Third Edition

The Circuits and Filters Handbook

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

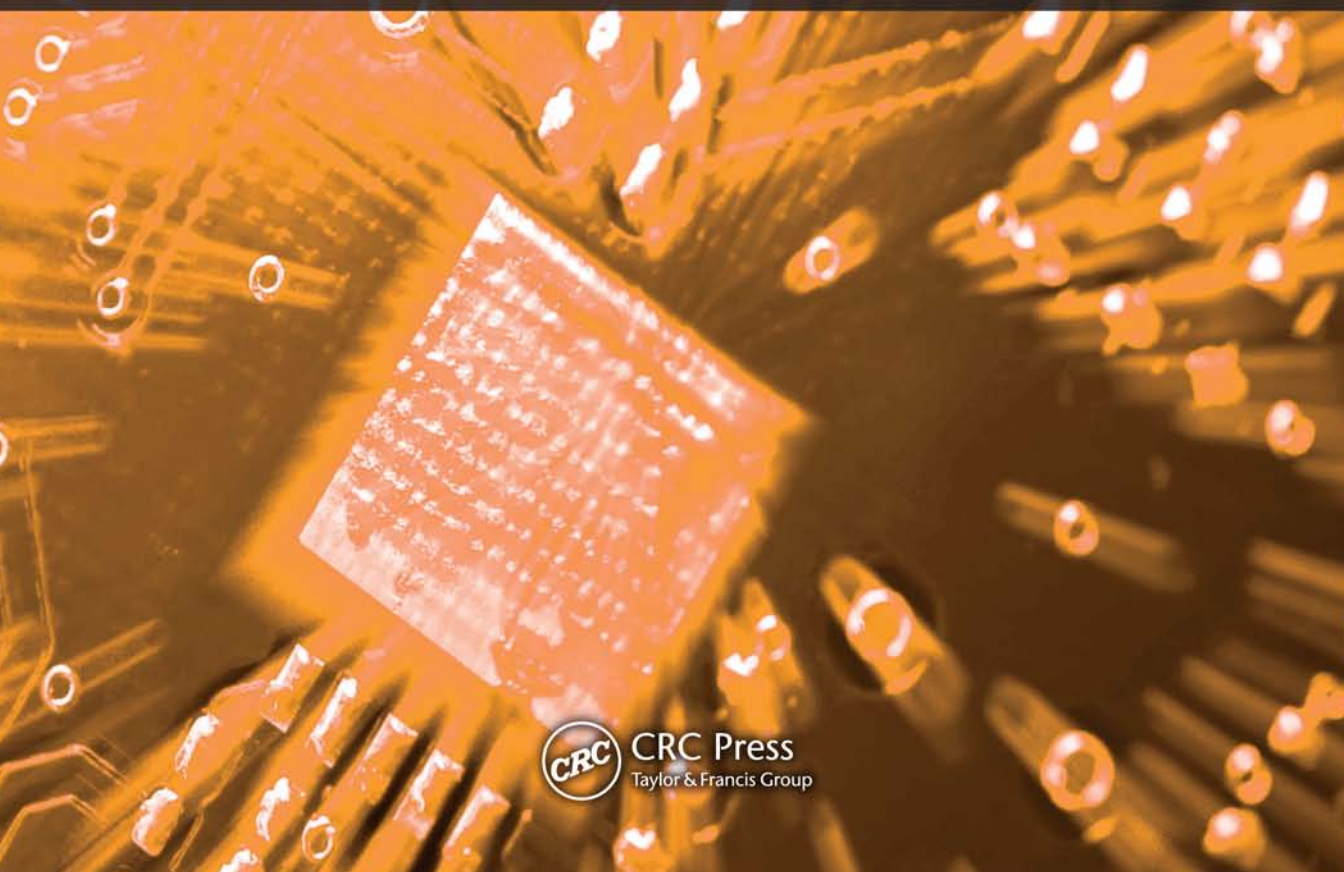
The Circuits and Filters Handbook

Third Edition

Feedback, Nonlinear, and Distributed Circuits

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

Feedback, Nonlinear, and Distributed Circuits

The Circuits and Filters Handbook

Third Edition

Edited by

Wai-Kai Chen

Fundamentals of Circuits and Filters

Feedback, Nonlinear, and Distributed Circuits

Analog and VLSI Circuits

Computer Aided Design and Design Automation

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Feedback, Nonlinear, and Distributed Circuits

Edited by

Wai-Kai Chen

University of Illinois

Chicago, U. S. A.

CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the

Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5881-9 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Feedback, nonlinear, and distributed circuits / edited by Wai-Kai Chen.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-4200-5881-9
ISBN-10: 1-4200-5881-9
1. Electronic circuits. 2. Electric circuits, Nonlinear. I. Chen, Wai-Kai, 1936- II. Title.

TK7867.N627 2009
621.3815--dc22 2008048115

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface vii

Editor-in-Chief ix

Contributors xi

SECTION I Feedback Circuits

1 Feedback Amplifier Theory **1**-1
John Choma, Jr.

2 Feedback Amplifier Configurations **2**-1
John Choma, Jr.

3 General Feedback Theory **3**-1
Wai-Kai Chen

4 Network Functions and Feedback **4**-1
Wai-Kai Chen

5 Measurement of Return Difference **5**-1
Wai-Kai Chen

6 Multiple-Loop Feedback Amplifiers **6**-1
Wai-Kai Chen

SECTION II Nonlinear Circuits

7 Qualitative Analysis **7**-1
Martin Hasler

8 Synthesis and Design of Nonlinear Circuits **8**-1
Angel Rodríguez-Vázquez, Manuel Delgado-Restituto, Jose L. Huertas, and F. Vidal

9 Representation, Approximation, and Identification **9**-1
Guanrong Chen

10 Transformation and Equivalence **10**-1
Wolfgang Mathis

11 Piecewise-Linear Circuits and Piecewise-Linear Analysis **11**-1
Joos Vandewalle and Lieven Vandenberghe

12 Simulation **12**-1
Erik Lindberg

13 Cellular Neural Networks and Cellular Wave Computers **13**-1
Tamás Roska, Ákos Zarándy, and Csaba Rekeczky

14 Bifurcation and Chaos **14**-1
Michael Peter Kennedy

SECTION III Distributed Circuits

15 Transmission Lines **15**-1
Thomas Koryu Ishii

16 Multiconductor Transmission Lines **16**-1
Daniël De Zutter and Luc Martens

17 Time and Frequency Domain Responses **17**-1
Luc Martens and Daniël De Zutter

18 Distributed RC Networks **18**-1
Vladimír Székely

19 Synthesis of Distributed Circuits **19**-1
Thomas Koryu Ishii

Index **IN**-1

Preface

The purpose of this book is to provide in a single volume a comprehensive reference work covering the broad spectrum of feedback amplifier design; analysis, synthesis, and design of nonlinear circuits; their representation, approximation, identification, and simulation; cellular neural networks; multiconductor transmission lines; and analysis and synthesis of distributed circuits. It also includes the design of multiple-loop feedback amplifiers. This book is written and developed for the practicing electrical engineers and computer scientists in industry, government, and academia. The goal is to provide the most up-to-date information in the field.

Over the years, the fundamentals of the field have evolved to include a wide range of topics and a broad range of practice. To encompass such a wide range of knowledge, this book focuses on the key concepts, models, and equations that enable the design engineer to analyze, design, and predict the behavior of feedback amplifiers, nonlinear and distributed systems. While design formulas and tables are listed, emphasis is placed on the key concepts and theories underlying the processes.

This book stresses fundamental theories behind professional applications and uses several examples to reinforce this point. Extensive development of theory and details of proofs have been omitted. The reader is assumed to have a certain degree of sophistication and experience. However, brief reviews of theories, principles, and mathematics of some subject areas are given. These reviews have been done concisely with perception.

The compilation of this book would not have been possible without the dedication and efforts of Professors Leon O. Chua and Thomas Koryu Ishii, and most of all the contributing authors. I wish to thank them all.

Wai-Kai Chen

Editor-in-Chief



Wai-Kai Chen is a professor and head emeritus of the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. He received his BS and MS in electrical engineering at Ohio University, where he was later recognized as a distinguished professor. He earned his PhD in electrical engineering at the University of Illinois at Urbana–Champaign.

Professor Chen has extensive experience in education and industry and is very active professionally in the fields of circuits and systems. He has served as a visiting professor at Purdue University, the University of Hawaii at Manoa, and Chuo University in Tokyo, Japan. He was the editor-in-chief of the *IEEE Transactions on Circuits and Systems*, *Series I and II*, the president of the IEEE Circuits and Systems Society, and is the founding editor and the editor-in-chief of the *Journal of Circuits, Systems and Computers*. He received the Lester R. Ford Award from the Mathematical Association of America; the Alexander von Humboldt Award from Germany; the JSPS Fellowship Award from the Japan Society for the Promotion of Science; the National Taipei University of Science and Technology Distinguished Alumnus Award; the Ohio University Alumni Medal of Merit for Distinguished Achievement in Engineering Education; the Senior University Scholar Award and the 2000 Faculty Research Award from the University of Illinois at Chicago; and the Distinguished Alumnus Award from the University of Illinois at Urbana–Champaign. He is the recipient of the Golden Jubilee Medal, the Education Award, and the Meritorious Service Award from the IEEE Circuits and Systems Society, and the Third Millennium Medal from the IEEE. He has also received more than a dozen honorary professorship awards from major institutions in Taiwan and China.

A fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the American Association for the Advancement of Science (AAAS), Professor Chen is widely known in the profession for the following works: *Applied Graph Theory* (North-Holland), *Theory and Design of Broadband Matching Networks* (Pergamon Press), *Active Network and Feedback Amplifier Theory* (McGraw-Hill), *Linear Networks and Systems* (Brooks/Cole), *Passive and Active Filters: Theory and Implements* (John Wiley), *Theory of Nets: Flows in Networks* (Wiley-Interscience), *The Electrical Engineering Handbook* (Academic Press), and *The VLSI Handbook* (CRC Press).

Contributors

Guanrong Chen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong, China

Wai-Kai Chen
Department of Electrical and Computer Engineering
University of Illinois
Chicago, Illinois

John Choma, Jr.
Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles
Los Angeles, California

Leon O. Chua
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, California

Manuel Delgado-Restituto
National Center of Microelectronics
Bureau of Science and Technology
Seville, Spain

Daniël De Zutter
Department of Information Technology
Ghent University
Ghent, Belgium

Martin Hasler
Laboratory of Nonlinear Systems
Swiss Federal Institute of Technology
Lausanne, Switzerland

Jose L. Huertas
National Center of Microelectronics
Bureau of Science and Technology
Seville, Spain

Thomas Koryu Ishii
Department Electrical and Computer Engineering
Marquette University
Milwaukee, Wisconsin

Michael Peter Kennedy
Department of Microelectronic Engineering
University College Cork
Cork, Ireland

Erik Lindberg
Department of Information Technology
Technical University of Denmark
Lyngby, Denmark

Luc Martens
Department of Information Technology
Ghent University
Ghent, Belgium

Wolfgang Mathis
Department of Electrical Engineering and Information Technology
University of Hanover
Hanover, Germany

Csaba Rekeczky
Eutecus Inc.
Berkeley, California
and
Pázmány Péter Catholic University
Budapest, Hungary

Angel Rodríguez-Vázquez
Electronics and Electromagnetism
University of Seville
Seville, Spain

Tamás Roska
Computer and Automation Research Institute
Hungarian Academy of Sciences
Budapest, Hungary
and
Department of Information Technology
Pázmány Péter Catholic University
Budapest, Hungary

Vladimír Székely
Department of Electron Devices
Budapest University of Technology and Economics
Budapest, Hungary

Lieven Vandenberghe
Electrical Engineering Department
University of California, Los Angeles
Los Angeles, California

Joos Vandewalle
Department of Electrical Engineering
Catholic University of Leuven
Leuven, Belgium

F. Vidal
Department of Electronics
University of Málaga
Málaga, Spain

Ákos Zarándy
Computer and Automation Research Institute
Hungarian Academy of Sciences
Budapest, Hungary

and

Pázmány Péter Catholic University
Budapest, Hungary

I

Feedback Circuits

Wai-Kai Chen
University of Illinois at Chicago

1 Feedback Amplifier Theory *John Choma, Jr.* **1**-1
Introduction • Methods of Analysis • Signal Flow Analysis • Global Single-Loop Feedback • Pole Splitting Open-Loop Compensation • Summary • References

2 Feedback Amplifier Configurations *John Choma, Jr.* **2**-1
Introduction • Series–Shunt Feedback Amplifier • Shunt–Series Feedback Amplifier • Shunt–Shunt Feedback Amplifier • Series–Series Feedback Amplifier • Dual-Loop Feedback • Summary • References

3 General Feedback Theory *Wai-Kai Chen* **3**-1
Introduction • Indefinite-Admittance Matrix • Return Difference • Null Return Difference • References

4 Network Functions and Feedback *Wai-Kai Chen* **4**-1
Blackman's Formula • Sensitivity Function

5 Measurement of Return Difference *Wai-Kai Chen* **5**-1
Blecher's Procedure • Impedance Measurements • References

6 Multiple-Loop Feedback Amplifiers *Wai-Kai Chen* **6**-1
Multiple-Loop Feedback Amplifier Theory • Return Different Matrix • Null Return Difference Matrix • Transfer-Function Matrix and Feedback • Sensitivity Matrix • Multiparameter Sensitivity • References

1

Feedback Amplifier Theory

1.1 Introduction **1**-1
1.2 Methods of Analysis **1**-2
1.3 Signal Flow Analysis **1**-4
1.4 Global Single-Loop Feedback **1**-6
Driving-Point I/O Resistances • Diminished Closed-Loop Damping Factor • Frequency Invariant Feedback Factor • Frequency Variant Feedback Factor (Compensation)
1.5 Pole Splitting Open-Loop Compensation **1**-11
Open-Loop Amplifier • Pole Splitting Analysis
1.6 Summary **1**-16
References **1**-17

John Choma, Jr.
University of Southern California

1.1 Introduction

Feedback, whether intentional or parasitic, is pervasive of all electronic circuits and systems. In general, feedback is comprised of a subcircuit that allows a fraction of the output signal of an overall network to modify the effective input signal in such a way as to produce a circuit response that can differ substantially from the response produced in the absence of such feedback. If the magnitude and relative phase angle of the fed back signal decreases the magnitude of the signal applied to the input port of an amplifier, the feedback is said to be *negative* or degenerative. On the other hand, *positive* (or regenerative) feedback, which gives rise to oscillatory circuit responses, is the upshot of a feedback signal that increases the magnitude of the effective input signal. Because negative feedback produces stable circuit responses, the majority of all intentional feedback architectures is degenerative [1,2]. However, parasitic feedback incurred by the energy storage elements associated with circuit layout, circuit packaging, and second-order high-frequency device phenomena often degrades an otherwise degenerative feedback circuit into either a potentially regenerative or severely underdamped network.

Intentional degenerative feedback applied around an analog network produces four circuit performance benefits. First, negative feedback desensitizes the gain of an open-loop amplifier (an amplifier implemented without feedback) with respect to variations in circuit element and active device model parameters. This desensitization property is crucial in view of parametric uncertainties caused by aging phenomena, temperature variations, biasing perturbations, and nonzero fabrication and manufacturing tolerances. Second, and principally because of the foregoing desensitization property, degenerative feedback reduces the dependence of circuit responses on the parameters of inherently nonlinear active devices, thereby reducing the total harmonic distortion evidenced in open loops. Third, negative feedback

broadbands the dominant pole of an open-loop amplifier, thereby affording at least the possibility of a closed-loop network with improved high-frequency performance. Finally, by modifying the driving-point input and output impedances of the open-loop circuit, negative feedback provides a convenient vehicle for implementing voltage buffers, current buffers, and matched interstage impedances.

The disadvantages of negative feedback include gain attenuation, a closed-loop configuration that is disposed to potential instability, and, in the absence of suitable frequency compensation, a reduction in the open-loop gain-bandwidth product (GBP). In uncompensated feedback networks, open-loop amplifier gains are reduced in almost direct proportion to the amount by which closed-loop amplifier gains are desensitized with respect to open-loop gains. Although the 3 dB bandwidth of the open-loop circuit is increased by a factor comparable to that by which the open-loop gain is decreased, the closed-loop GBP resulting from uncompensated degenerative feedback is never greater than that of the open-loop configuration [3]. Finally, if feedback is incorporated around an open-loop amplifier that does not have a dominant pole [4], complex conjugate closed-loop poles yielding nonmonotonic frequency responses are likely. Even positive feedback is possible if substantive negative feedback is applied around an open-loop amplifier for which more than two poles significantly influence its frequency response.

Although the foregoing detail is common knowledge deriving from Bode's pathfinding disclosures [5], most circuit designers remain uncomfortable with analytical procedures for estimating the frequency responses, I/O impedances, and other performance indices of practical feedback circuits. The purposes of this section are to formulate systematic feedback circuit analysis procedures and ultimately, to demonstrate their applicability to six specific types of commonly used feedback architectures. Four of these feedback types, the series–shunt, shunt–series, shunt–shunt, and series–series configurations, are single-loop architectures, while the remaining two types are the series–series/shunt–shunt and series–shunt/shunt–series dual-loop configurations.

1.2 Methods of Analysis

Several standard techniques are used for analyzing linear feedback circuits [6]. The most straightforward of these entails writing the Kirchhoff equilibrium equations for the small-signal model of the entire feedback system. This analytical tack presumably leads to the idealized feedback circuit block diagram abstracted in Figure 1.1. In this model, the circuit voltage or current response, X_R, is related to the source current or voltage excitation, X_S, by

$$G_{cl} \triangleq \frac{X_R}{X_S} = \frac{G_o}{1 + fG_o} \equiv \frac{G_o}{1 + T} \tag{1.1}$$

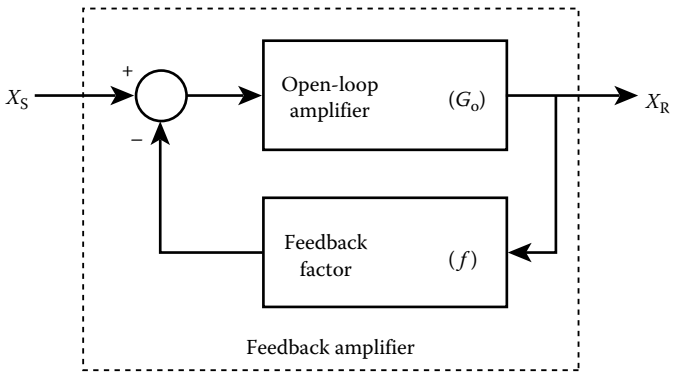


FIGURE 1.1 Block diagram model of a feedback network.

where

G_{cl} is the closed-loop gain of the feedback circuit

f, the feedback factor, is the proportion of circuit response fed back for antiphase superposition with the source signal

G_o represents the open-loop gain

The product fG_o is termed the loop gain T.

Equation 1.1 demonstrates that, for loop gains with magnitudes that are much larger than one, the closed-loop gain collapses to $1/f$, which is independent of the open-loop gain. To the extent that the open-loop amplifier, and not the feedback subcircuit, contains circuit elements and other parameters that are susceptible to modeling uncertainties, variations in the fabrication of active and passive elements, and nonzero manufacturing tolerances, large loop gain achieves a desirable parametric desensitization. Unfortunately, the determination of G_o and f directly from the Kirchhoff relationships is a nontrivial task, especially because G_o is rarely independent of f in practical electronics. Moreover, Equation 1.1 does not illuminate the manner in which the loop gain modifies the driving-point input and output impedances of the open-loop amplifier.

A second approach to feedback network analysis involves modeling the open loop, feedback, and overall closed-loop networks by a homogeneous set of two-port parameters [7]. When the two-port parameter model is selected judiciously, the two-port parameters for the closed-loop network derive from a superposition of the respective two-port parameters of the open loop and feedback subcircuits. Given the resultant parameters of the closed-loop circuit, standard formulas can then be exploited to evaluate closed-loop values of the circuit gain and the driving-point input and output impedances.

Unfortunately, several limitations plague the utility of feedback network analysis predicated on two-port parameters. First, the computation of closed-loop two-port parameters is tedious if the open-loop configuration is a multistage amplifier, or if multiloop feedback is utilized. Second, the two-loop method of feedback circuit analysis is straightforwardly applicable to only those circuits that implement global feedback (feedback applied from output port to input port). Many single-ended feedback amplifiers exploit only local feedback, wherein a fraction of the signal developed at the output port is fed back to a terminal pair other than that associated with the input port. Finally, the appropriate two-port parameters of the open-loop amplifier can be superimposed with the corresponding parameter set of the feedback subcircuit if and only if the Brune condition is satisfied [8]. This requirement mandates equality between the pre- and postconnection values of the two-port parameters of open loop and feedback cells, respectively. The subject condition is often not satisfied when the open-loop amplifier is not a simple three-terminal two-port configuration.

The third method of feedback circuit analysis exploits Mason's signal flow theory [9–11]. The circuit level application of this theory suffers few of the shortcomings indigenous to block diagram and two-port methods of feedback circuit analysis [12]. Signal flow analyses applied to feedback networks efficiently express I/O transfer functions, driving-point input impedances, and driving-point output impedances in terms of an arbitrarily selected critical or reference circuit parameters, say P.

An implicit drawback of signal flow methods is the fact that unless P is selected to be the feedback factor f, which is not always transparent in feedback architectures, expressions for the loop gain and the open-loop gain of feedback amplifiers are obscure. However, by applying signal flow theory to a feedback circuit model engineered from insights that derive from the results of two-port network analyses, the feedback factor can be isolated. The payoff of this hybrid analytical approach includes a conventional block diagram model of the I/O transfer function, as well as convenient mathematical models for evaluating the closed-loop driving-point input and output impedances. Yet, another attribute of hybrid methods of feedback circuit analysis is its ability to delineate the cause, nature, and magnitude of the feedforward transmittance produced by interconnecting a certain feedback subcircuit to a given open-loop amplifier. This information is crucial in feedback network design because feedforward invariably decreases gain and often causes undesirable phase shifts that can lead to significantly underdamped or unstable closed-loop responses.

1.3 Signal Flow Analysis

Guidelines for feedback circuit analysis by hybrid signal flow methods can be established with the aid of Figure 1.2 [13]. Figure 1.2a depicts a linear network whose output port is terminated in a resistance, R_L. The output signal variable is the voltage V_O, which is generated in response to an input port signal whose Thévenin voltage and resistance are respectively, V_S and R_S. Implicit to the linear network is a current-controlled voltage source (CCVS) Pi_b, with a value that is directly proportional to the indicated network branch current i_b. The problem at hand is the deduction of the voltage gain $G_v(R_S, R_L) = V_O/V_S$, the driving-point input resistance (or impedance) R_{in}, and the driving-point output resistance (or impedance) R_{out}, as explicit functions of the critical transimpedance parameter P. Although the following systematic procedure is developed in conjunction with the diagram in Figure 1.2, with obvious changes in notation, it is applicable to determine any type of transfer relationship for any linear network in terms of any type of reference parameter [14].

1. Set $P = 0$, as depicted in Figure 1.2b, and compute the resultant voltage gain $G_{vo}(R_S, R_L)$, where the indicated notation suggests an anticipated dependence of gain on source and load resistances. Also, compute the corresponding driving-point input and output resistances R_{in}, and R_{out}, respectively. In this case, the "critical" parameter P is associated with a controlled voltage source. Accordingly, $P = 0$ requires that the branch containing the controlled source be supplanted by a

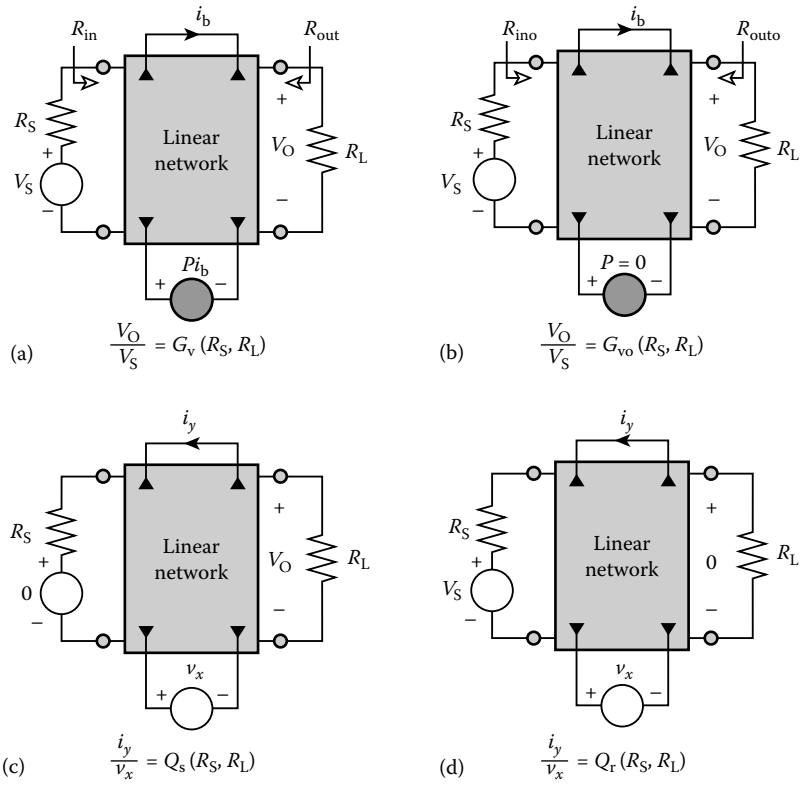


FIGURE 1.2 (a) Linear network with an identified critical parameter P. (b) Model for calculating the $P = 0$ value of voltage gain. (c) The return ratio with respect to P is $PQ_s(R_S, R_L)$. (d) The null return ratio with respect to P is $PQ_r(R_S, R_L)$.

short circuit. If, for example, P is associated with a controlled current source, $P = 0$ mandates the replacement of the controlled source by an open circuit.

2. Set the Thévenin source voltage V_S to zero, and replace the original controlled voltage source Pi_b by an independent voltage source of symbolic value, v_x. Then, calculate the ratio, i_y/v_x, where, as illustrated in Figure 1.2c, i_y flows in the branch that originally conducts the controlling current i_b. Note, however, that the reference polarity of i_y is opposite to that of i_b. The computed transfer function i_y/v_x is denoted by $Q_s(R_S, R_L)$. This transfer relationship, which is a function of the source and load resistances, is used to determine the *return ratio* $T_s(P, R_S, R_L)$ with respect to parameter P of the original network. In particular,

$$T_s(P, R_S, R_L) = PQ_s(R_S, R_L) \tag{1.2}$$

3. If P is associated with a controlled current source, the controlled generator Pi_b is replaced by a current source of value i_x. If the controlling variable is a voltage, instead of a current, the ratio v_y/v_x, is computed, where v_y is the voltage developed across the controlling branch and the polarity is opposite to that of the original controlling voltage.
4. The preceding computational step is repeated, but instead of setting V_S to zero, the output variable, which is the voltage V_O in the present case, is nulled, as indicated in Figure 1.2d. Let the computed ratio i_y/v_x, be symbolized as $Q_r(R_S, R_L)$. In turn, the *null return ratio* $T_r(P, R_S, R_L)$, with respect to parameter P is

$$T_r(P, R_S, R_L) = PQ_r(R_S, R_L) \tag{1.3}$$

5. Desired voltage gain $G_v(R_S, R_L)$, of the linear network undergoing study can be shown to be [5,12]

$$G_v(R_S, R_L) = \frac{V_O}{V_S} = G_{vo}(R_S, R_L)\left[\frac{1 + PQ_r(R_S, R_L)}{1 + PQ_s(R_S, R_L)}\right] \tag{1.4}$$

6. Given the function $Q_s(R_S, R_L)$, the driving-point input and output resistances follow straightforwardly from [12]

$$R_{in} = R_{ino}\left[\frac{1 + PQ_s(0, R_L)}{1 + PQ_s(\infty, R_L)}\right] \tag{1.5}$$

$$R_{out} = R_{outo}\left[\frac{1 + PQ_s(R_S, 0)}{1 + PQ_s(R_S, \infty)}\right] \tag{1.6}$$

An important special case entails a controlling electrical variable i_b associated with the selected parameter P that is coincidentally the voltage or current output of the circuit under investigation. In this situation, a factor P of the circuit response is fed back to the port (not necessarily the input port) defined by the terminal pair across which the controlled source is incident. When the controlling variable i_b is the output voltage or current of the subject circuit $Q_r(R_S, R_L)$, which is evaluated under the condition of a nulled network response, is necessarily zero. With $Q_r(R_S, R_L) = 0$, the algebraic form of Equation 1.4 is identical to that of Equation 1.1, where the loop gain T is the return ratio with respect to parameter P; that is,

$$PQ_s(R_S, R_L)|_{Q_r(R_S, R_L) = 0} = T \tag{1.7}$$

Moreover, a comparison of Equation 1.4 to Equation 1.1 suggests that $G_v(R_S, R_L)$ symbolizes the closed-loop gain of the circuit, $G_{vo}(R_S, R_L)$ represents the corresponding open-loop gain, and the circuit feedback factor f is

$$f = \frac{PQ_s(R_S, R_L)}{G_{vo}(R_S, R_L)} \tag{1.8}$$

1.4 Global Single-Loop Feedback

Consider the global feedback scenario illustrated in Figure 1.3a, in which a fraction P of the output voltage V_O is fed back to the voltage-driven input port. Figure 1.3b depicts the model used to calculate the return ratio $Q_s(R_S, R_L)$, where, in terms of the branch variables in the schematic diagram, $Q_s(R_S, R_L) = v_y/v_x$. An inspection of this diagram confirms that the transfer function v_y/v_x, is identical to the $P = 0$ value of the gain V_O/V_S, which derives from an analysis of the structure in Figure 1.3a. Thus, for global voltage feedback in which a fraction of the output voltage is fed back to a voltage-driven input port, $Q_s(R_S, R_L)$ is the open-loop voltage gain; that is, $Q_s(R_S, R_L) + G_{vo}(R_S, R_L)$. It follows from Equation 1.8 that the feedback factor f is identical to the selected critical parameter P. Similarly, for the global current feedback architecture of Figure 1.4a, a fraction P of the output current, I_O, is fed back to the current-driven input port $f = P$. As implied by the model of Figure 1.4b, $Q_s(R_S, R_L) \equiv G_{io}(R_S, R_L)$, the open-loop current gain.

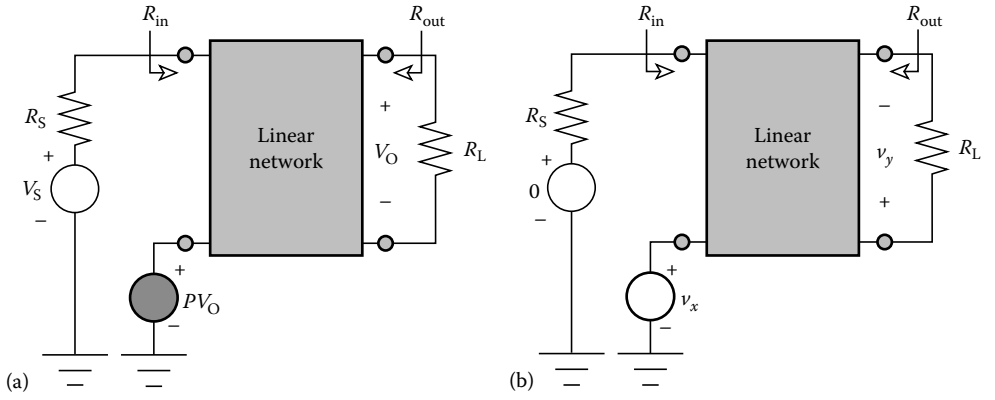


FIGURE 1.3 (a) Voltage-driven linear network with global voltage feedback. (b) Model for the calculation of loop gain.

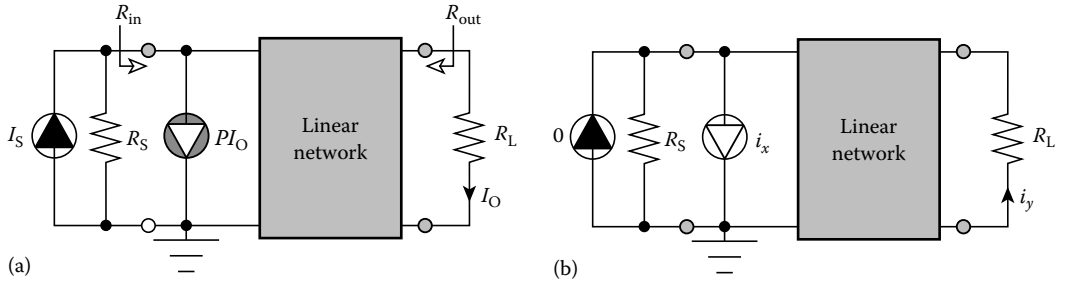


FIGURE 1.4 (a) Current-driven linear network with global current feedback. (b) Model for the calculation of loop gain.

1.4.1 Driving-Point I/O Resistances

Each of the two foregoing circuit architectures has a closed-loop gain where the algebraic form mirrors (Equation 1.1). It follows that for sufficiently large loop gain (equal to either $PG_{vo}(R_S, R_L)$ or $PG_{io}(R_S, R_L)$), the closed-loop gain approaches $(1/P)$ and is therefore desensitized with respect to open-loop gain parameters. However, such a desensitization with respect to the driving-point input and output resistances (or impedances) cannot be achieved. For the voltage feedback circuit in Figure 1.3a, $Q_s(\infty, R_L)$, is the $R_S = \infty$ value, $G_{vo}(R_S, R_L)$, of the open-loop voltage gain. This particular open-loop gain is zero, because $R_S = \infty$ decouples the source voltage from the input port of the amplifier. On the other hand, $Q_s(0, R_L)$ is the $R_S = 0$ value, $G_{vo}(0, R_L)$, of the open-loop voltage gain. This gain is at least as large as $G_{vo}(R_S, R_L)$, since a short-circuited Thévenin source resistance implies lossless coupling of the Thévenin signal to the amplifier input port. Recalling Equation 1.5, the resultant driving-point input resistance of the voltage feedback amplifier is

$$R_{in} = R_{ino}[1 + PG_{vo}(0, R_L)] \geq R_{ino}[1 + PG_{vo}(R_S, R_L)] \tag{1.9}$$

which shows that the closed-loop driving-point input resistance is larger than its open-loop counterpart and is dependent on open-loop voltage gain parameters.

Conversely, the corresponding driving-point output resistance in Figure 1.3a is smaller than the open-loop output resistance and approximately inversely proportional to the open-loop voltage gain. These assertions derive from the facts that $Q_s(R_S, 0)$ is the $R_L = 0$ value of the open-loop voltage gain $G_{vo}(R_S, R_L)$. Because $R_L = 0$ corresponds to the short-circuited load resistance, $G_{vo}(R_S, 0) = 0$. In contrast, $Q_s(R_S, \infty)$, is the $R_L = \infty$ value, $G_{vo}(R_S, \infty)$, of the open-loop gain, which is a least as large as $G_{vo}(R_S, R_L)$. By Equation 1.6,

$$R_{out} = \frac{R_{outo}}{1 + PG_{vo}(R_S, \infty)} \leq \frac{R_{outo}}{1 + PG_{vo}(R_S, R_L)} \tag{1.10}$$

Similarly, the driving-point input and output resistances of the global current feedback configuration of Figure 1.4a are sensitive to open-loop gain parameters. In contrast to the voltage amplifier of Figure 1.3a, the closed loop, driving-point input resistance of current amplifier is smaller than its open-loop value, while the driving-point output resistance is larger than its open-loop counterpart. Noting that the open-loop current gain $G_{io}(R_S, R_L)$ is zero for both $R_S = 0$ (which short circuits the input port), and $R_L = \infty$ (which open circuits the load port), Equations 1.5 and 1.6 give

$$R_{in} = \frac{R_{ino}}{1 + PG_{io}(\infty, R_L)} \tag{1.11}$$

$$R_{out} = R_{outo}[1 + PG_{io}(R_S, 0)] \tag{1.12}$$

1.4.2 Diminished Closed-Loop Damping Factor

In addition to illuminating the driving-point and forward transfer characteristics of single-loop feedback architectures, the special case of global single-loop feedback illustrates the potential instability problems pervasive of almost all feedback circuits. An examination of these problems begins by returning to Equation 1.1 and letting the open-loop gain, G_o, be replaced by the two-pole frequency-domain function,

$$G_o(s) = \frac{G_o(0)}{\left(1 + \frac{s}{p_1}\right)\left(1 + \frac{s}{p_2}\right)} \tag{1.13}$$

where $G_o(0)$ symbolizes the zero-frequency open-loop gain. The pole frequencies p_1 and p_2 in Equation 1.13 are either real numbers or complex conjugate pairs. Alternatively, Equation 1.13 is expressible as

$$G_s(s) = \frac{G_o(0)}{1 + \frac{2\zeta_{ol}}{\omega_{nol}}s + \frac{s^2}{\omega_{nol}^2}} \tag{1.14}$$

where

$$\omega_{nol} = \sqrt{p_1 p_2} \tag{1.15}$$

represents the *undamped natural frequency* of oscillation of the open-loop configuration, and

$$\zeta_{ol} = \frac{1}{2}\left[\sqrt{\frac{p_2}{p_1}} + \sqrt{\frac{p_1}{p_2}}\right] \tag{1.16}$$

is the *damping factor* of the open-loop circuit.

In Equation 1.1, let the feedback factor f be the single left-half-plane zero function,

$$f(s) = f_o\left(1 + \frac{s}{z}\right) \tag{1.17}$$

where

z is the frequency of the real zero introduced by feedback
f_o is the zero-frequency value of the feedback factor

The resultant loop gain is

$$T(s) = f_o\left(1 + \frac{s}{z}\right)G_o(s) \tag{1.18}$$

the zero-frequency value of the loop gain is

$$T(0) = f_o G_o(0) \tag{1.19}$$

and the zero-frequency closed-loop gain $G_{cl}(0)$, is

$$G_{cl}(0) = \frac{G_o(0)}{1 + f_o G_o(0)} = \frac{G_o(0)}{1 + T(0)} \tag{1.20}$$

Upon inserting Equations 1.14 and 1.17 into Equation 1.1, the closed-loop transfer function is determined to be

$$G_{cl}(s) = \frac{G_{cl}(0)}{1 + \frac{2\zeta_{cl}}{\omega_{ncl}}s + \frac{s^2}{\omega_{ncl}^2}} \tag{1.21}$$

where the closed-loop undamped natural frequency of oscillation ω_{ncl} relates to its open-loop counterpart ω_{nol}, in accordance with

$$\omega_{ncl} = \omega_{nol}\sqrt{1 + T(0)} \tag{1.22}$$

Moreover, the closed-loop damping factor ζ_{cl} is

$$\zeta_{cl} = \frac{\zeta_{ol}}{\sqrt{1+T(0)}} + \left[\frac{T(0)}{1+T(0)}\right]\frac{\omega_{ncl}}{2z} = \frac{\zeta_{ol}}{\sqrt{1+T(0)}} + \left[\frac{T(0)}{\sqrt{1+T(0)}}\right]\frac{\omega_{nol}}{2z} \tag{1.23}$$

A frequency invariant feedback factor $f(s)$ applied to the open-loop configuration whose transfer function is given by Equation 1.13 implies an infinitely large frequency, z, of the feedback zero. For this case, Equation 1.23 confirms a closed-loop damping factor that is always less than the open-loop damping factor. Indeed, for a smaller than unity open-loop damping factor (which corresponds to complex conjugate open-loop poles) and reasonable values of the zero-frequency loop gain $T(0)$, $\zeta_{cl} \ll 1$. Thus, constant feedback applied around an underdamped two-pole open-loop amplifier yields a severely underdamped closed-loop configuration. It follows that the closed-loop circuit has a transient step response plagued by overshoot and a frequency response that displays response peaking within the closed-loop passband. Observe that underdamping is likely to occur even in critically damped (identical real open-loop poles) or overdamped (distinct real poles) open-loop amplifiers, which correspond to $\zeta_{ol} = 1$ and $\zeta_{ol} > 1$, respectively, when a large zero-frequency loop gain is exploited.

Underdamped closed-loop amplifiers are not unstable systems, but they are nonetheless unacceptable. From a practical design perspective, closed-loop underdamping predicted by relatively simple mathematical models of the loop gain portend undesirable amplifier responses or even closed-loop instability. The problem is that simple transfer function models invoked in a manual circuit analysis are oblivious to presumably second-order parasitic circuit layout and device model energy storage elements with effects that include a deterioration of phase and gain margins.

1.4.3 Frequency Invariant Feedback Factor

Let the open-loop amplifier be overdamped, such that its real satisfy the relationship

$$p_2 = \kappa^2 p_1 \tag{1.24}$$

If the open-loop amplifier pole p_1 is dominant, κ^2 is a real number that is greater than the magnitude, $|G_o(0)|$, of the open-loop zero-frequency gain, which is presumed to be much larger than one. As a result, the open-loop damping factor in Equation 1.16 reduces to $\zeta_{o1} \approx \kappa/2$. With $\kappa^2 > |G_o(0)| \gg 1$, which formally reflects the *dominant pole approximation*, the 3 dB bandwidth B_{ol} of the open-loop amplifier is given approximately by [15]

$$B_{ol} \approx \frac{\omega_{nol}}{2\zeta_{ol}} = \frac{1}{\frac{1}{p_1}+\frac{1}{p_2}} = \left(\frac{\kappa^2}{\kappa^2+1}\right)p_1 \tag{1.25}$$

As expected, Equation 1.25 predicts an open-loop 3 dB bandwidth that is only slightly smaller than the frequency of the open-loop dominant pole.

The frequency, z, in Equation 1.23 is infinitely large if frequency invariant degenerative feedback is applied around an open-loop amplifier. For a critically damped or overdamped closed-loop amplifier, $\zeta_{cl} > 1$. Assuming open-loop pole dominance, this constraint imposes the open-loop pole requirement,

$$\frac{p_2}{p_1} \geq 4[1+T(0)] \tag{1.26}$$

Thus, for large zero-frequency loop gain, $T(0)$, an underdamped closed-loop response is avoided if and only if the frequency of the nondominant open-loop pole is substantially larger than that of the dominant open-loop pole. Unless frequency compensation measures are exploited in the open loop, Equation 1.26

is difficult to satisfy, especially if feedback is implemented expressly to realize a substantive desensitization of response with respect to open-loop parameters. On the chance that Equation 1.26 can be satisfied, and if the closed-loop amplifier emulates a dominant pole response, the closed-loop bandwidth is, using Equations 1.22, 1.23, and 1.25,

$$B_{\mathrm{cl}} \approx \frac{\omega_{\mathrm{ncl}}}{2\zeta_{\mathrm{cl}}} \approx [1 + T(0)]B_{\mathrm{ol}} \approx [1 + T(0)]p_1 \tag{1.27}$$

Observe from Equations 1.26 and 1.27 that the maximum possible closed-loop 3 dB bandwidth is 2 octaves below the minimum acceptable frequency of the nondominant open-loop pole.

Although Equation 1.27 theoretically confirms the broadbanding property of negative feedback amplifiers, the attainment of very large closed-loop 3 dB bandwidths is nevertheless a challenging undertaking. The problem is that Equation 1.26 is rarely satisfied. As a result, the open-loop configuration must be suitably compensated, usually by pole splitting methodology [16–18], to force the validity of Equation 1.26. However, the open-loop poles are not mutually independent, so any compensation that increases p_2 is accompanied by decreases in p_1. The pragmatic upshot of the matter is that the closed-loop 3 dB bandwidth is not directly proportional to the uncompensated value of p_1 but instead, it is proportional to the smaller, compensated value of p_1.

1.4.4 Frequency Variant Feedback Factor (Compensation)

Consider now the case where the frequency, z, of the compensating feedback zero is finite and positive. Equation 1.23 underscores the stabilizing property of a left-half-plane feedback zero in that a sufficiently small positive z renders a closed-loop damping factor ζ_{cl} that can be made acceptably large, regardless of the value of the open-loop damping factor ζ_{ol}. To this end, $\zeta_{\mathrm{cl}} > 1/\sqrt{2}$ is a desirable design objective in that it ensures a monotonically decreasing closed-loop frequency response. If, as is usually a design goal, the open-loop amplifier subscribes to pole dominance, Equation 1.23 translates the objective, $\zeta_{\mathrm{cl}} > 1/\sqrt{2}$, into the design constraint

$$z \leq \frac{\left[\frac{T(0)}{1+T(0)}\right]\omega_{\mathrm{ncl}}}{\sqrt{2} - \frac{\omega_{\mathrm{ncl}}}{[1+T(0)]B_{\mathrm{ol}}}} \tag{1.28}$$

where use is made of Equation 1.25 to cast ζ in terms of the open-loop bandwidth B_{ol}. When the closed-loop damping factor is precisely equal to $1/\sqrt{2}$ a maximally flat magnitude closed-loop response results for which the 3 dB bandwidth is ω_{ncl}. Equation 1.28 can then be cast into the more useful form

$$zG_{\mathrm{cl}}(0) = \frac{\mathrm{GBP_{ol}}}{\sqrt{2}\left(\frac{\mathrm{GBP_{ol}}}{\mathrm{GBP_{cl}}}\right) - 1} \tag{1.29}$$

where Equation 1.20 is exploited, $\mathrm{GBP_{ol}}$ is the gain-bandwidth product of the open-loop circuit, and $\mathrm{GBP_{cl}}$ is the gain-bandwidth product of the resultant closed-loop network.

For a given open-loop gain-bandwidth product $\mathrm{GBP_{ol}}$, a desired low-frequency closed-loop gain, $G_{\mathrm{cl}}(0)$, and a desired closed-loop gain-bandwidth product, $\mathrm{GBP_{cl}}$, Equation 1.29 provides a first-order estimate of the requisite feedback compensation zero. Additionally, note that Equation 1.29 imposes an upper limit on the achievable high-frequency performance of the closed-loop configuration. In particular, because z must be positive to ensure acceptable closed-loop damping, Equation 1.29 implies

$$\mathrm{GBP_{ol}} > \frac{\mathrm{GBP_{cl}}}{\sqrt{2}} \tag{1.30}$$

In effect, Equation 1.30 imposes a lower limit on the required open-loop GBP commensurate with feedback compensation implemented to achieve a maximally flat, closed-loop frequency response.

1.5 Pole Splitting Open-Loop Compensation

Equation 1.26 underscores the desirability of achieving an open-loop dominant pole frequency response in the design of a feedback network. In particular, Equation 1.26 shows that if the ultimate design goal is a closed-loop dominant pole frequency response, the frequency, p_2, of the nondominant open-loop amplifier pole must be substantially larger than its dominant pole counterpart, p_1. Even if closed-loop pole dominance is sacrificed as a trade-off for other performance merits, open-loop pole dominance is nonetheless a laudable design objective. This contention follows from Equations 1.16 and 1.23, which combine to suggest that the larger p_2 is in comparison to p_1, the larger is the open-loop damping factor. In turn, the unacceptably underdamped closed-loop responses that are indicative of small, closed-loop damping factors are thereby eliminated. Moreover, Equation 1.23 indicates that larger, open-loop damping factors impose progressively less demanding restrictions on the feedback compensation zero that may be required to achieve acceptable closed-loop damping. This observation is important because in an actual circuit design setting, small z in Equation 1.23 generally translates into a requirement of a correspondingly large RC time constant, where implementation may prove difficult in monolithic circuit applications.

Unfortunately, many amplifiers, and particularly broadbanded amplifiers, earmarked for use as open-loop cells in degenerative feedback networks, are not characterized by dominant pole frequency responses. The frequency response of these amplifiers is therefore optimized in accordance with a standard design practice known as pole splitting compensation. Such compensation entails the connection of a small capacitor between two high impedance, phase-inverting nodes of the open-loop topology [17,19–21]. Pole splitting techniques increase the frequency p_2 of the uncompensated nondominant open-loop pole to a compensated value, say p_{2c}. The frequency, p_1, of the uncompensated dominant open-loop pole is simultaneously reduced to a smaller frequency, say p_{1c}. Although these pole frequency translations complement the design requirement implicit to Equations 1.23 and 1.26, they do serve to limit the resultant closed-loop bandwidth, as discussed earlier. As highlighted next, they also impose other performance limitations on the open loop.

1.5.1 Open-Loop Amplifier

The engineering methods, associated mathematics, and engineering trade-offs underlying pole splitting compensation are best revealed in terms of the generalized, phase-inverting linear network abstracted in Figure 1.5. Although this amplifier may comprise the entire open-loop configuration, in the most general case, it is an interstage of the open loop. Accordingly, R_{st} in this diagram is viewed as the Thévenin equivalent resistance of either an input signal source or a preceding amplification stage. The response to the Thévenin driver, V_{st}, is the indicated output voltage, V_l, which is developed across the Thévenin load resistance, R_{lt}, seen by the stage under investigation. Note that the input current conducted by the amplifier is I_s, while the current flowing into the output port of the unit is denoted as I_l. The dashed branch containing the capacitor C_c, which is addressed later, is the pole splitting compensation element.

Because the amplifier under consideration is linear, any convenient set of two-port parameters can be used to model its terminal volt–ampere characteristics. Assuming the existence of the short-circuit admittance, or y parameters,

$$\begin{bmatrix} I_s \\ I_l \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_2 \end{bmatrix} \begin{bmatrix} V_i \\ V_l \end{bmatrix} \tag{1.31}$$

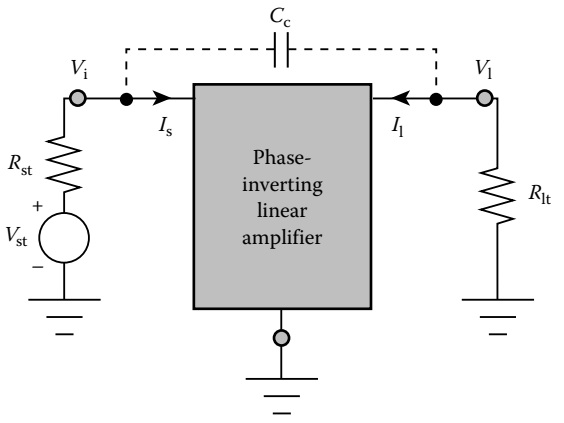


FIGURE 1.5 Linear amplifier for which a pole splitting compensation capacitance C_c is incorporated.

Defining

$$\begin{aligned} y_o &\triangleq y_{11} + y_{12} \\ y_o &\triangleq y_{22} + y_{12} \\ y_f &\triangleq y_{21} + y_{12} \\ y_r &\triangleq -y_{12} \end{aligned} \tag{1.32}$$

Equation 1.31 implies

$$I_s = y_i V_i + y_r(V_i - V_l) \tag{1.33}$$

$$I_l = y_f V_i + y_o V_1 + y_r(V_l - V_i) \tag{1.34}$$

The last two expressions produce the y-parameter model depicted in Figure 1.6a, in which y_i represents an effective shunt input admittance, y_o is a shunt output admittance, y_f is a forward transadmittance, and y_r reflects voltage feedback intrinsic to the amplifier.

Amplifiers amenable to pole splitting compensation have capacitive input and output admittances; that is, y_i and y_o are of the form

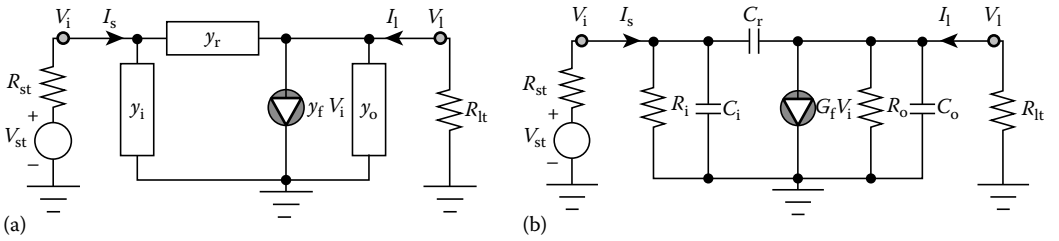


FIGURE 1.6 y-Parameter equivalent circuit of the phase-inverting linear amplifier in Figure 1.5. (b) Approximate form of the model in (a).

$$y_i = \frac{1}{R_i} + sC_i$$
$$y_o = \frac{1}{R_o} + sC_o \tag{1.35}$$

Similarly,

$$y_f = G_f - sC_f$$
$$y_r = \frac{1}{R_r} + sC_r \tag{1.36}$$

In Equation 1.36, the conductance component G_f of the forward transadmittance y_f positive in a phase-inverting amplifier. Moreover, the reactive component $-sC_f$ of y_f produces an *excess phase angle*, and hence, a *group delay*, in the forward gain function. This component, which deteriorates phase margin, can be ignored to first order if the signal frequencies of interest are not excessive in comparison to the upper-frequency limit of performance of the amplifier. Finally, the feedback internal to many practical amplifiers is predominantly capacitive so that the feedback resistance R_r can be ignored. These approximations allow the model in Figure 1.6a to be drawn in the form offered in Figure 1.6b.

It is worthwhile interjecting that the six parameters indigenous to the model in Figure 1.6b need not be deduced analytically from the small-signal models of the active elements embedded in the subject interstage. Instead, SPICE can be exploited to evaluate the y parameters in Equation 1.31 at the pertinent biasing level. Because these y parameters display dependencies on signal frequency, care should be exercised to evaluate their real and imaginary components in the neighborhood of the open loop, 3 dB bandwidth to ensure acceptable computational accuracy at high frequencies. Once the y parameters in Equation 1.31 are deduced by computer-aided analysis, the alternate admittance parameters in Equation 1.23, as well as numerical estimates for the parameters, R_i, C_i, R_o, C_o, C_r, and G_f, in Equations 1.35 and 1.36 follow straightforwardly.

1.5.2 Pole Splitting Analysis

An analysis of the circuit in Figure 1.6b produces a voltage transfer function $A_v(s)$ of the form

$$A_v(s) = \frac{V_l(s)}{V_{st}(s)} = A_v(0)\left[\frac{1 - \frac{s}{z_r}}{\left(1 + \frac{s}{p_1}\right)\left(1 + \frac{s}{p_2}\right)}\right] \tag{1.37}$$

Letting

$$R_{ll} = R_{lt} \| R_o \tag{1.38}$$

an inspection of the circuit in Figure 1.6b confirms that

$$A_v(0) = -G_f R_{ll}\left(\frac{R_i}{R_i + R_{st}}\right) \tag{1.39}$$

is the zero-frequency voltage gain. Moreover, the frequency, z_r, of the right-half-plane zero is

$$z_r = \frac{G_f}{C_r} \tag{1.40}$$

The lower pole frequency, p_1, and the higher pole frequency, p_2, derive implicitly from

$$\frac{1}{p_1}+\frac{1}{p_2}=R_{ll}(C_o+C_r)+R_{ss}[C_i+(1+G_fR_{ll})C_r] \tag{1.41}$$

and

$$\frac{1}{p_1p_2}=R_{ss}R_{ll}C_o\left[C_i+\left(\frac{C_o+C_i}{C_o}\right)C_r\right] \tag{1.42}$$

where

$$R_{ss}=R_{st}\stackrel{\Delta}{=}R_i \tag{1.43}$$

Most practical amplifiers, and particularly amplifiers realized in bipolar junction transistor technology, have very large forward transconductance, G_f, and small internal feedback capacitance, C_r. The combination of large G_f and small C_r renders the frequency in Equation 1.40 so large as to be inconsequential to the passband of interest. When utilized in a high-gain application, such as the open-loop signal path of a feedback amplifier, these amplifiers also operate with a large effective load resistance, R_{ll}. Accordingly, Equation 1.41 can be used to approximate the pole frequency p_1 as

$$p_1\approx\frac{1}{R_{ss}[C_i+(1+G_fR_{ll})C_r]} \tag{1.44}$$

Substituting this result into Equation 1.42, the approximate frequency p_2 of the high-frequency pole is

$$p_2\approx\frac{C_i+(1+G_fR_{ll})C_r}{R_{ll}C_o\left[C_i+\left(\frac{C_o+C_i}{C_o}\right)C_r\right]} \tag{1.45}$$

Figure 1.7 illustrates asymptotic frequency responses corresponding to pole dominance and to a two-pole response. Figure 1.7a depicts the frequency response of a dominant pole amplifier, which does not

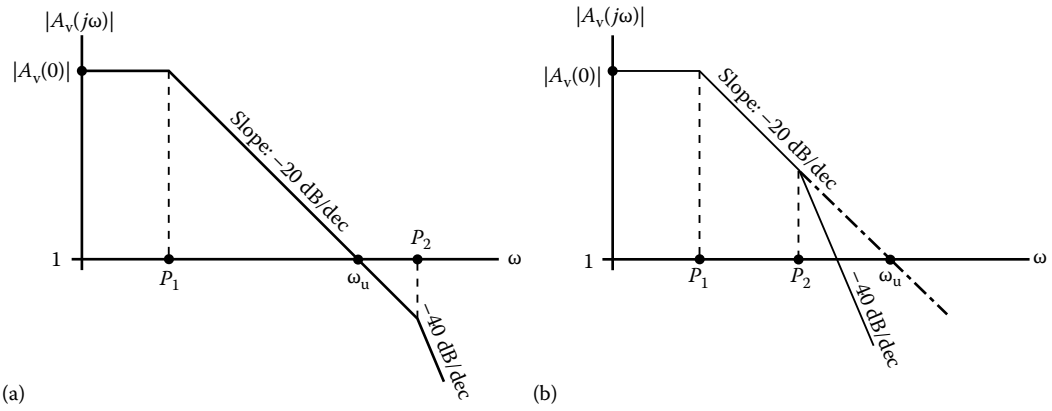


FIGURE 1.7 (a) Asymptotic frequency response for a dominant pole amplifier. Such an amplifier does not require pole splitting compensation because the two lowest frequency amplifier poles, p_1 and p_2, are already widely separated. (b) Frequency response of an amplifier with high-frequency response that is strongly influenced by both of its lowest frequency poles. The basic objective of pole splitting compensation is to transform the indicated frequency response to a form that emulates that depicted in (a).

require pole splitting compensation. Observe that its high-frequency response is determined by a single pole (p_1 in this case) through the signal frequency at which the gain ultimately degrades to unity. In this interpretation of a dominant pole amplifier, p_2 is not only much larger than p_1, but is in fact larger than the unity gain frequency, which is indicated as ω_u in the figure. This unity gain frequency, which can be viewed as an upper limit to the useful passband of the amplifier, is approximately, $|A_v(0)|p_1$. To the extent that p_1 is essentially the 3 dB bandwidth when $p_2 \gg p_1$, the unity gain frequency is also the GBP of the subject amplifier. In short, with $|A_v(j\omega_u)| \triangleq 1$, $p_2 \gg p_1$ in Equation 1.37 implies

$$\omega_u \approx |A_v(0)|p_1 \approx \text{GBP} \tag{1.46}$$

The contrasting situation of a response indigenous to the presence of two significant open-loop poles is illustrated in Figure 1.7b. In this case, the higher pole frequency p_2 is smaller than ω_u and hence, the amplifier does not emulate a single-pole response throughout its theoretically useful frequency range. The two critical frequencies, p_1 and p_2, remain real numbers, and as long as $p_2 \neq p_1$, the corresponding damping factor, is greater than 1. However, the damping factor of the two-pole amplifier (its response is plotted in Figure 1.7b) is nonetheless smaller than that of the dominant pole amplifier. It follows that, for reasonable loop gains, unacceptable underdamping is more likely when feedback is invoked around the two-pole amplifier, as opposed to the same amount of feedback applied around a dominant pole amplifier. Pole splitting attempts to circumvent this problem by transforming the pole conglomeration of the two-pole amplifier into one that emulates the dominant pole situation inferred by Figure 1.7a.

To the foregoing end, append the compensation capacitance C_c between the input and the output ports of the phase-inverting linear amplifier, as suggested in Figure 1.5. With reference to the equivalent circuit in Figure 1.6b, the electrical impact of this additional element is the effective replacement of the internal feedback capacitance C_r by the capacitance sum $(C_r + C_c)$. Letting

$$C_p \triangleq C_r + C_c \tag{1.47}$$

it is apparent that Equations 1.40 through 1.42 remain applicable, provided that C_r in these relationships is supplanted by C_p. Because C_p is conceivably significantly larger than C_c, however, the approximate expressions for the resultant pole locations differ from those of Equations 1.44 and 1.45. In particular, a reasonable approximation for the compensated value, say P_{1c}, of the lower pole frequency is now

$$p_{1c} \approx \frac{1}{[R_{ll} + (1 + G_f R_{ll})R_{ss}]C_p} \tag{1.48}$$

while the higher pole frequency, p_{2c}, becomes

$$p_{2c} \approx \frac{1}{\left(R_{ss} \| R_{ll} \| \frac{1}{G_f}\right)(C_o + C_i)} \tag{1.49}$$

Clearly, $p_{1c} < p_1$ and $p_{2c} > p_2$. Moreover, for large G_f, p_{2c} is potentially much larger than p_{1c}. It should also be noted that the compensated value, say, z_{rc}, of the right-half-plane zero is smaller than its uncompensated value, z_r, because Equation 1.40 demonstrates that

$$z_{rc} = \frac{G_f}{C_p} = z_r\left(\frac{C_r}{C_r + C_c}\right) \tag{1.50}$$

Although z_{rc} can conceivably exert a significant influence on the high-frequency response of the compensated amplifier, the following discussion presumes tacitly that $z_{rc} > p_{2c}$ [2].

Assuming a dominant pole frequency response, the compensated unity gain frequency, ω_{uc}, is, using Equations 1.39, 1.46, and 1.48,

$$\omega_{uc} \approx |A_v(0)|p_{1c} \approx \left(\frac{1}{R_{st}C_p}\right)\left[G_f\left(R_{ss}\|R_{ll}\|\frac{1}{G_f}\right)\right] \tag{1.51}$$

It is interesting to note that

$$\omega_{uc} < \left(\frac{1}{R_{st}C_p}\right) \tag{1.52}$$

that is, the unity gain frequency is limited by the inverse of the RC time constant formed by the Thévenin source resistance R_{st} and the net capacitance C_p appearing between the input port and the phase inverted output port. The subject inequality comprises a significant performance limitation, for if p_{2c} is indeed much larger than p_{ic}, ω_{uc} is approximately the GBP of the compensated cell. Accordingly, for a given source resistance, a required open-loop gain, and a desired open-loop bandwidth, Equation 1.52 imposes an upper limit on the compensation capacitance that can be exploited for pole splitting purposes.

In order for the compensated amplifier to behave as a dominant pole configuration, p_{2c} must exceed ω_{uc}, as defined by Equation 1.51. Recalling Equation 1.49, the requisite constraint is found to be

$$R_{st}C_p > G_f\left(R_{ss}\|R_{ll}\|\frac{1}{G_f}\right)^2(C_o + C_i) \tag{1.53}$$

Assuming $G_f\,(R_{ss}/R_{ll}) \ll 1$, Equation 1.53 reduces to the useful simple form

$$C_fR_{st} > \frac{C_o + C_i}{C_p} \tag{1.54}$$

which confirms the need for large forward transconductance G_f if pole splitting is to be an effective compensation technique.

1.6 Summary

The use of negative feedback is fundamental to the design of reliable and reproducible analog electronic networks. Accordingly, this chapter documents the salient features of the theory that underlies the efficient analysis and design of commonly used feedback networks. Four especially significant points are postulated in this section.

1. By judiciously exploiting signal flow theory, the classical expression, Equation 1.1, for the I/O transfer relationship of a linear feedback system is rendered applicable to a broad range of electronic feedback circuits. This expression is convenient for design-oriented analysis because it clearly identifies the open-loop gain, G_o, and the loop gain, T. The successful application of signal flow theory is predicated on the requirement that the feedback factor, to which T is proportional and that appears in the signal flow literature as a ''critical'' or ''reference'' parameter, can be identified in a given feedback circuit.
2. Signal flow theory, as applied to electronic feedback architectures, proves to be an especially expedient analytical tool because once the loop gain T is identified, the driving-point input and output impedances follow with minimal additional calculations. Moreover, the functional dependence of T on the Thévenin source and terminating load impedances unambiguously brackets the

magnitudes of the driving point I/O impedances attainable in particular types of feedback arrangements.

3. Damping factor concept is advanced herewith as a simple way of assessing the relative stability of both the open and closed loops of a feedback circuit. The open-loop damping factor derives directly from the critical frequencies of the open-loop gain, while these frequencies and any zeros appearing in the loop gain unambiguously define the corresponding closed-loop damping factor. Signal flow theory is once again used to confirm the propensity of closed loops toward instability unless the open-loop subcircuit functions as a dominant pole network. Also confirmed is the propriety of the common practice of implementing a feedback zero as a means of stabilizing an otherwise potentially unstable closed loop.
4. Pole splitting as a means to achieve dominant pole open-loop responses is definitively discussed. Generalized design criteria are formulated for this compensation scheme, and limits of performance are established. Of particular interest is the fact that pole splitting limits the GBP of the compensated amplifier to a value that is determined by a source resistance–compensation capacitance time constant.

References

1. J. A. Mataya, G. W. Haines, and S. B. Marshall, IF amplifier using C_c-compensated transistors, *IEEE J. Solid-State Circuits*, SC-3, 401–407, Dec. 1968.
2. W. G. Beall and J. Choma Jr., Charge-neutralized differential amplifiers, *J. Analog Integr. Circuits Signal Process.*, 1, 33–44, Sept. 1991.
3. J. Choma Jr., A generalized bandwidth estimation theory for feedback amplifiers, *IEEE Trans. Circuits Syst.*, CAS-31, 861–865, Oct. 1984.
4. R. D. Thornton, C. L. Searle, D. O. Pederson, R. B. Adler, and E. J. Angelo Jr., *Multistage Transistor Circuits*, New York: John Wiley & Sons, 1965, Chaps. 1 and 8.
5. H. W. Bode, *Network Analysis and Feedback Amplifier Design*, New York: Van Nostrand, 1945.
6. P. J. Hurst, A comparison of two approaches to feedback circuit analysis, *IEEE Trans. Educ.*, 35, 253–261, Aug. 1992.
7. M. S. Ghausi, *Principles and Design of Linear Active Networks*, New York: McGraw-Hill, 1965, pp. 40–56.
8. A. J. Cote Jr. and J. B. Oakes, *Linear Vacuum-Tube and Transistor Circuits*, New York: McGraw-Hill, 1961, pp. 40–46.
9. S. J. Mason, Feedback theory—Some properties of signal flow graphs, *Proc. IRE*, 41, 1144–1156, Sept. 1953.
10. S. J. Mason, Feedback theory—Further properties of signal flow graphs, *Proc. IRE*, 44, 920–926, July 1956.
11. N. Balabanian and T. A. Bickart, *Electrical Network Theory*, New York: John Wiley & Sons, 1969, pp. 639–669.
12. J. Choma Jr., Signal flow analysis of feedback networks, *IEEE Trans. Circuits Syst.*, 37, 455–463, Apr. 1990.
13. J. Choma Jr., *Electrical Networks: Theory and Analysis*, New York: Wiley Interscience, 1985, pp. 589–605.
14. P. J. Hurst, Exact simulation of feedback circuit parameters, *IEEE Trans. Circuits Syst.*, 38, 1382–1389, Nov. 1991.
15. J. Choma Jr. and S. A. Witherspoon, Computationally efficient estimation of frequency response and driving point impedance in wideband analog amplifiers, *IEEE Trans. Circuits Syst.*, 37, 720–728, June 1990.
16. R. G. Meyer and R. A. Blauschild, A wide-band low-noise monolithic transimpedance amplifier, *IEEE J. Solid-State Circuits*, SC-21, 530–533, Aug. 1986.

17. Y. P. Tsividis, Design considerations in single-channel MOS analog integrated circuits, *IEEE J. Solid-State Circuits*, SC-13, 383–391, June 1978.
18. J. J. D'Azzo and C. H. Houpis, *Feedback Control System Analysis and Synthesis*, New York: McGraw-Hill, 1960, pp. 230–234.
19. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, New York: John Wiley & Sons, 1977, pp. 512–521.
20. P. R. Gray, Basic MOS operational amplifier design—An overview, in *Analog MOS Integrated Circuits*, P. R. Gray, D. A. Hodges, and R.W. Brodersen (Eds.), New York: IEEE, 1980, pp. 28–49.
21. J. E. Solomon, The monolithic op-amp: A tutorial study, *IEEE J. Solid-State Circuits*, SC-9, 314–332, Dec. 1974.

2

Feedback Amplifier Configurations

2.1 Introduction **2**-1
2.2 Series–Shunt Feedback Amplifier **2**-2
Circuit Modeling and Analysis • Feed-Forward Compensation
2.3 Shunt–Series Feedback Amplifier **2**-10
2.4 Shunt–Shunt Feedback Amplifier **2**-12
Circuit Modeling and Analysis • Design Considerations
2.5 Series–Series Feedback Amplifier **2**-16
2.6 Dual-Loop Feedback **2**-20
Series–Series/Shunt–Shunt Feedback Amplifier • Series–Shunt/Shunt–Series Feedback Amplifier
2.7 Summary **2**-28
References **2**-29

John Choma, Jr.
University of Southern California

2.1 Introduction

Four basic types of single-loop feedback amplifiers are available: the series–shunt, shunt–series, shunt–shunt, and series–series architectures [1]. Each of these cells is capable of a significant reduction of the dependence of forward transfer characteristics on the ill-defined or ill-controlled parameters implicit to the open-loop gain; but none of these architectures can simultaneously offer controlled driving-point input and output impedances. Such additional control is afforded only by dual global loops comprised of series and/or shunt feedback signal paths appended to an open-loop amplifier [2,3]. Only two types of global dual-loop feedback architectures are used: the series–series/shunt–shunt feedback amplifier and the series–shunt/shunt–series feedback amplifier.

Although only bipolar technology is exploited in the analysis of the aforementioned four single-loop and two dual-loop feedback cells, all disclosures are generally applicable to metal-oxide-silicon (MOS), heterostructure bipolar transistor (HBT), and III–V compound metal-semiconductor field-effect transistor (MESFET) technologies. All analytical results derive from an application of a hybrid, signal flow/two-port parameter analytical tack. Because the thought processes underlying this technical approach apply to all feedback circuits, the subject analytical procedure is developed in detail for only the series–shunt feedback amplifier.

2.2 Series–Shunt Feedback Amplifier

2.2.1 Circuit Modeling and Analysis

Figure 2.1a depicts the ac schematic diagram (a circuit diagram divorced of biasing details) of a series–shunt feedback amplifier. In this circuit, the output voltage V_O, which is established in response to a single source represented by the Thévenin voltage V_{ST}, and the Thévenin resistance, R_{ST}, is sampled by the feedback network composed of the resistances, R_{EE} and R_F. The sampled voltage is fed back in such a way that the closed-loop input voltage, V_I, is the sum of the voltage, V_{1A}, across the input port of the amplifier and the voltage V_{1F}, developed across R_{EE} in the feedback subcircuit. Because $V_I = V_{1A} + V_{1F}$, the output port of the feedback configuration can be viewed as connected in series with the amplifier input port. On the other hand, output voltage sampling constrains the net load current, I_O, to be the algebraic sum of the amplifier output port current, I_{2A}, and the feedback network input current, I_{2F}. Accordingly, the output topology is indicative of a shunt connection between the feedback subcircuit and the amplifier output port. The fact that voltage is fed back to a voltage-driven input port renders the driving-point input resistance, R_{in}, of the closed-loop amplifier large, whereas the driving-point output resistance, R_{out}, seen by the terminating load resistance, R_{LT}, is small. The resultant closed-loop amplifier

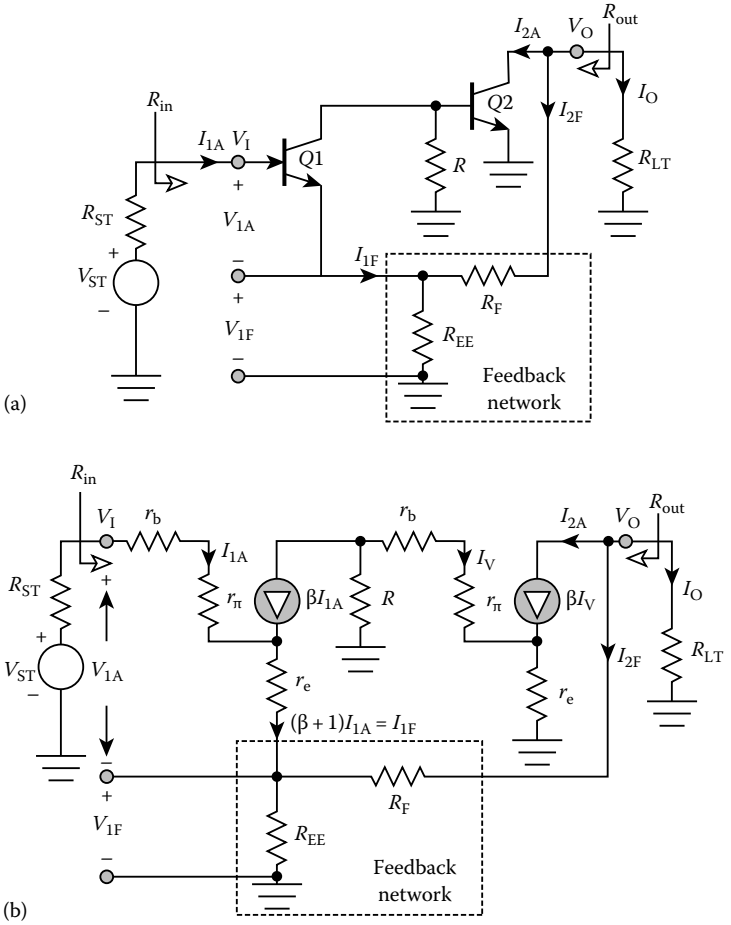


FIGURE 2.1 (a) AC schematic diagram of a bipolar series–shunt feedback amplifier. (b) Low-frequency small-signal equivalent circuit of the feedback amplifier.

is therefore best suited for voltage amplification, in the sense that the closed-loop voltage gain, V_O/V_{ST}, can be made approximately independent of source and load resistances. For large loop gain, this voltage transfer function is also nominally independent of transistor parameters.

Assuming that transistors $Q1$ and $Q2$ are identical devices that are biased identically, Figure 2.1b is the applicable low-frequency equivalent circuit. This equivalent circuit exploits the hybrid-π model [4] of a bipolar junction transistor, subject to the proviso that the forward Early resistance [5] used to emulate base conductivity modulation is sufficiently large to warrant its neglect. Because an infinitely large forward Early resistance places the internal collector resistance (not shown in the figure) of a bipolar junction transistor in series with the current-controlled current source, this collector resistance can be ignored as well.

The equivalent circuit of Figure 2.1b can be reduced to a manageable topology by noting that the ratio of the signal current, I_V, flowing into the base of transistor $Q2$ to the signal current, I_{1A}, flowing into the base of transistor $Q1$ is

$$\frac{I_V}{I_{1A}} \triangleq -K_\beta = -\frac{\beta R}{R + r_b + r_\pi + (\beta + 1)r_e} = -\frac{\alpha R}{r_{ib} + (1 - \alpha)R} \tag{2.1}$$

where

$$\alpha = \frac{\beta}{\beta + 1} \tag{2.2}$$

is the small-signal, short-circuit common base current gain, and

$$r_{ib} = r_e + \frac{r_\pi + r_b}{\beta + 1} \tag{2.3}$$

symbolizes the short-circuit input resistance of a common base amplifier. It follows that the current source βI_v in Figure 2.1b can be replaced by the equivalent current $(-\beta K_\beta I_{1A})$.

A second reduction of the equivalent circuit in Figure 2.1b results when the feedback subcircuit is replaced by a model that reflects the h-parameter relationships

$$\begin{bmatrix} V_{1F} \\ I_{2F} \end{bmatrix} = \begin{bmatrix} h_{if} & h_{rf} \\ h_{ff} & h_{of} \end{bmatrix} \begin{bmatrix} I_{1F} \\ V_O \end{bmatrix} \tag{2.4}$$

where

$V_{1F}(V_O)$ represents the signal voltage developed across the output (input) port of the feedback subcircuit

$I_{1F}(I_{2F})$ symbolizes the corresponding current flowing into the feedback output (input) port

Although any homogeneous set of two-port parameters can be used to model the feedback subcircuit, h-parameters are the most convenient selection herewith. In particular, the feedback amplifier undergoing study is a series–shunt configuration. The h-parameter equivalent circuit represents its input port as a Thévenin circuit and its input port as a Norton configuration, therefore, the h-parameter equivalent circuit is likewise a series–shunt structure.

For the feedback network at hand, which is redrawn for convenience in Figure 2.2a, the h-parameter equivalent circuit is as depicted in Figure 2.2b. The latter diagram exploits the facts that the short-circuit input resistance h_{if} is a parallel combination of the resistance R_{EE} and R_F, and the open-circuit output conductance h_{of} is $1/(R_{EE} + R_F)$. The open-circuit reverse voltage gain h_{rf} is

$$h_{rf} = \frac{R_{EE}}{R_{EE} + R_F} \tag{2.5}$$

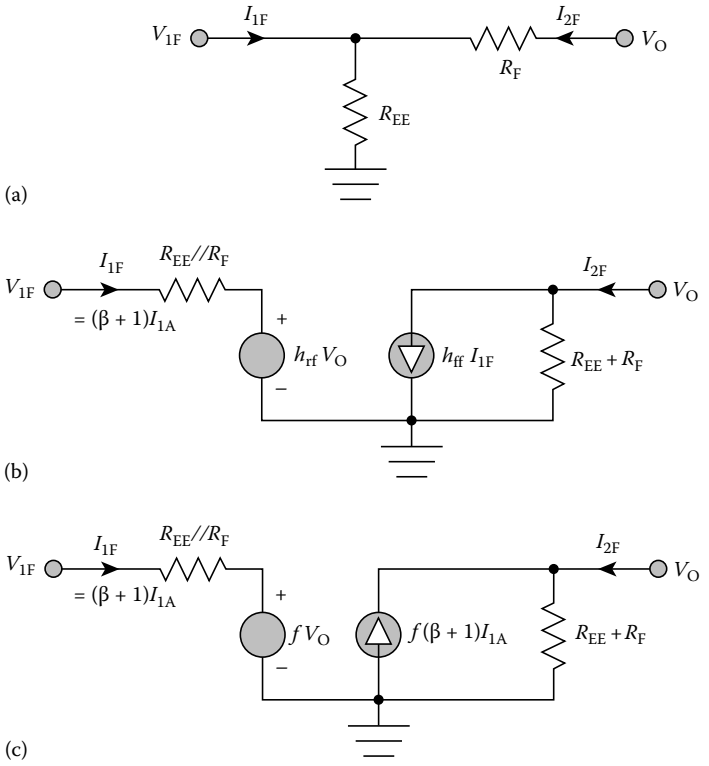


FIGURE 2.2 (a) Feedback subcircuit in the series–shunt feedback amplifier of Figure 2.1a. (b) *h*-Parameter equivalent circuit of the feedback subcircuit. (c) Alternative form of the *h*-parameter equivalent circuit.

while the short-circuit forward current gain h_{ff} is

$$h_{ff} = \frac{R_{EE}}{R_{EE} + R_F} = -h_{rf} \tag{2.6}$$

Figure 2.2c modifies the equivalent circuit in Figure 2.2b in accordance with the following two arguments. First, h_{rf} in Equation 2.5 is recognized as the fraction of the feedback subcircuit input signal that is fed back as a component of the feedback subcircuit output voltage, V_{1F}. But this subcircuit input voltage is identical to the closed-loop amplifier output signal V_O. Moreover, V_{1F} superimposes with the Thévenin input signal applied to the feedback amplifier to establish the amplifier input port voltage, V_{1A}. It follows that h_{rf} is logically referenced as a feedback factor, say *f*, of the amplifier under consideration; that is,

$$h_{rf} = \frac{R_{EE}}{R_{EE} + R_F} \triangleq f \tag{2.7}$$

and by Equation 2.6,

$$h_{ff} = -\frac{R_{EE}}{R_{EE} + R_F} = -f \tag{2.8}$$

Second, the feedback subcircuit output current, I_{1F}, is, as indicated in Figure 2.1b, the signal current, $(\beta + 1)I_{1A}$. Thus, in the model of Figure 2.2b,

$$h_{ff}I_{1F} = -f(\beta + 1)I_{1A} \tag{2.9}$$

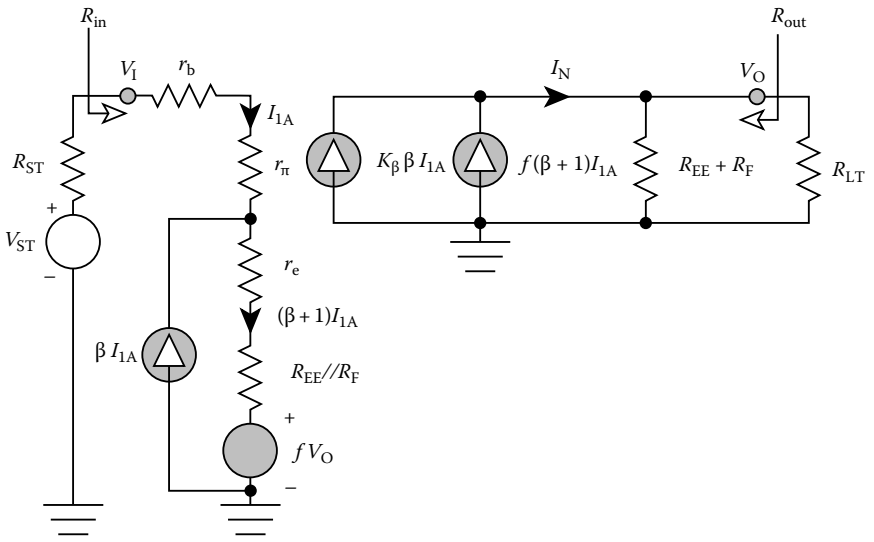


FIGURE 2.3 Modified small-signal model of the series–shunt feedback amplifier.

If the model in Figure 2.2c is used to replace the feedback network in Figure 2.1b the equivalent circuit of the series–shunt feedback amplifier becomes the alternative structure offered in Figure 2.3. In arriving at this model, care has been exercised to ensure that the current flowing through the emitter of transistor $Q1$ is $(\beta + 1)I_{1A}$. It is important to note that the modified equivalent circuit delivers transfer and driving-point impedance characteristics that are identical to those implicit to the equivalent circuit of Figure 2.1b. In particular, the traditional analytical approach to analyzing a series–shunt feedback amplifier tacitly presumes the satisfaction of the Brune condition [6] to formulate a composite structure where the h-parameter matrix is the sum of the respective h-parameter matrices for the open loop and feedback circuits. In contrast, the model of Figure 2.3 derives from Figure 2.1b without invoking the Brune requirement, which is often not satisfied. It merely exploits the substitution theorem; that is, the feedback network in Figure 2.1b is substituted by its h-parameter representation.

In addition to modeling accuracy, the equivalent circuit in Figure 2.3 boasts at least three other advantages. The first is an illumination of the vehicle by which feedback is implemented in the series–shunt configuration. This vehicle is the voltage-controlled voltage source, fV_O, which feeds back a fraction of the output signal to produce a branch voltage that algebraically superimposes with, and thus modifies, the applied source voltage effectively seen by the input port of the open-loop amplifier. Thus, with $f = 0$, no feedback is evidenced, and the model at hand emulates an open-loop configuration. But even with $f = 0$, the transfer and driving-point impedance characteristics of the resultant open-loop circuit are functionally dependent on the feedback elements, R_{EE} and R_F, because appending the feedback network to the open-loop amplifier incurs additional impedance loads at both the input and the output ports of the amplifier.

The second advantage of the subject model is its revelation of the magnitude and nature of feed-forward through the closed loop. In particular, note that the signal current, I_N, driven into the effective load resistance comprised of the parallel combination of $(R_{EE} + R_F)$ and R_{LT}, is the sum of two current components. One of these currents, $\beta K_\beta I_{1A}$, materializes from the transfer properties of the two transistors utilized in the amplifier. The other current, $f(\beta + 1)I_{1A}$, is the feed-forward current resulting from the bilateral nature of the passive feedback network. In general, negligible feed-forward through the feedback subcircuit is advantageous, particularly in high-frequency signal-processing applications. To this end, the model in Figure 2.3 suggests the design requirement,

$$f \ll \alpha K_\beta \tag{2.10}$$

When the resistance, R, in Figure 2.1a is the resistance associated with the output port of a PNP current source used to supply biasing current to the collector of transistor $Q1$ and the base of transistor $Q2$, K_β approaches β, and Equation 2.10 is easily satisfied; however, PNP current sources are undesirable in broadband low-noise amplifiers. In these applications, the requisite biasing current must be supplied by a passive resistance, R, connected between the positive supply voltage and the junction of the $Q1$ collector and the $Q2$ base. Unfortunately, the corresponding value of K_β can be considerably smaller than β, with the result that Equation 2.10 may be difficult to satisfy. Circumvention schemes for this situation are addressed later.

A third attribute of the model in Figure 2.3 is its disposition to an application of signal flow theory. For example, with the feedback factor f selected as the reference parameter for signal flow analysis, the open-loop voltage gain $G_{vo}(R_{ST}, R_{LT})$, of the series–shunt feedback amplifier is computed by setting f to zero. Assuming that Equation 2.10 is satisfied, circuit analysis reveals this gain as

$$G_{vo}(R_{ST}, R_{LT}) = \alpha K_\beta \left[\frac{(R_{EE} + R_F) \| R_{LT}}{r_{ib} + (1 - \alpha) R_{ST} + (R_{EE} \| R_F)} \right] \tag{2.11}$$

The corresponding input and output driving-point resistances, R_{ino} and R_{outo}, respectively, are

$$R_{ino} = r_B + r_\pi + (\beta + 1)(r_E + R_{EE} \| R_F) \tag{2.12}$$

and

$$R_{outo} = R_{EE} + R_F \tag{2.13}$$

It follows that the closed-loop gain $G_v(R_{ST}, R_{LT})$ of the series–shunt feedback amplifier is

$$G_v(R_{ST}, R_{LT}) = \frac{G_{vo}(R_{ST}, R_{LT})}{1 + T} \tag{2.14}$$

where the loop gain T is

$$\begin{aligned} T = fG_{vo}(R_{ST}, R_{LT}) &= \left(\frac{R_{EE}}{R_{EE} + R_F} \right) G_{vo}(R_{ST}, R_{LT}) \\ &= \alpha K_\beta \left(\frac{R_{EE}}{R_{EE} + R_F + R_{LT}} \right) \left[\frac{R_{LT}}{r_{ib} + (1 - \alpha) R_{ST} + (R_{EE} \| R_F)} \right] \end{aligned} \tag{2.15}$$

For $T \gg 1$, which mandates a sufficiently large K_β in Equation 2.11, the closed-loop gain collapses to

$$G_v(R_{ST}, R_{LT}) \approx \frac{1}{f} = 1 + \frac{R_F}{R_{EE}} \tag{2.16}$$

which is independent of active element parameters. Moreover, to the extent that $T \gg 1$ the series–shunt feedback amplifier behaves as an ideal voltage-controlled voltage source in the sense that its closed-loop voltage gain is independent of source and load terminations. The fact that the series–shunt feedback network behaves approximately as an ideal voltage amplifier implies that its closed-loop driving-point input resistance is very large and its closed-loop driving-point output resistance is very small. These facts are confirmed analytically by noting that

$$\begin{aligned} R_{in} = R_{ino}[1 + fG_{vo}(0, R_L)] &\approx fR_{ino}G_{vo}(0, R_L) \\ &= \beta K_\beta \left(\frac{R_{EE}}{R_{EE} + R_F + R_{LT}} \right) R_{LT} \end{aligned} \tag{2.17}$$

and

$$R_{\text{out}} = \frac{R_{\text{outo}}}{1 + fG_{\text{vo}}(R_{\text{S}}, \infty)} \approx \frac{R_{\text{outo}}}{fG_{\text{vo}}(R_{\text{S}}, \infty)} = \left(1 + \frac{R_{\text{F}}}{R_{\text{EE}}}\right)\left[\frac{r_{\text{ib}} + (1-\alpha)R_{\text{ST}} + R_{\text{EE}} \| R_{\text{F}}}{\alpha K_{\beta}}\right] \tag{2.18}$$

To the extent that the interstage biasing resistance, R, is sufficiently large to allow K_{β} to approach β, observe that R_{in} in Equation 2.17 is nominally proportional to β^2, while R_{out} in Equation 2.18 is inversely proportional to β.

2.2.2 Feed-Forward Compensation

When practical design restrictions render the satisfaction of Equation 2.10 difficult, feed-forward problems can be circumvented by inserting an emitter follower between the output port of transistor $Q2$ in the circuit diagram of Figure 2.1a and the node to which the load termination and the input terminal of the feedback subcircuit are incident [2]. The resultant circuit diagram, inclusive now of simple biasing subcircuits, is shown in Figure 2.4. The buffer transistor $Q3$ increases the original short-circuit forward current gain, $K_{\beta}\beta$, of the open-loop amplifier by a factor approaching $(\beta + 1)$, while not altering the feed-forward factor implied by the feedback network in Figure 2.1a. In effect, K_{β} is increased by a factor of almost $(\beta + 1)$, thereby making Equation 2.10 easy to satisfy. Because of the inherently low output resistance of an emitter follower, the buffer also reduces the driving-point output resistance achievable by the original configuration.

The foregoing contentions can be confirmed through an analysis of the small-signal model for the modified amplifier in Figure 2.4. Such an analysis is expedited by noting that the circuit to the left of the current-controlled current source, $K_{\beta}\beta I_{1\text{A}}$, in Figure 2.3 remains applicable. For zero feedback, it follows that the small-signal current $I_{1\text{A}}$ flowing into the base of transistor $Q1$ derives from

$$\left.\frac{I_{1\text{A}}}{V_{\text{ST}}}\right|_{f=0} = \frac{1-\alpha}{r_{\text{ib}} + (1-\alpha)R_{\text{ST}} + (R_{\text{EE}} \| R_{\text{F}})} \tag{2.19}$$

The pertinent small-signal model for the buffered series–shunt feedback amplifier is resultantly the configuration offered in Figure 2.5.

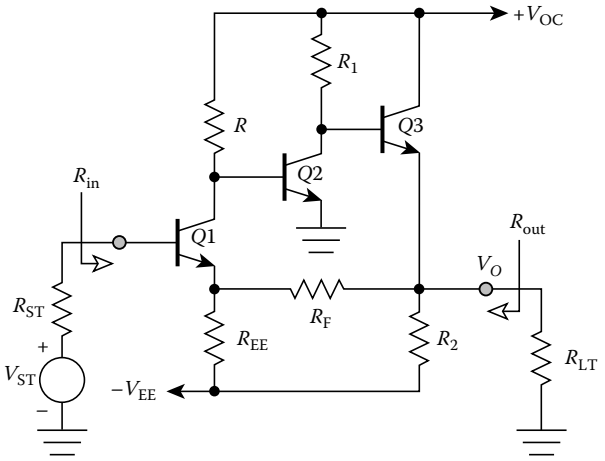


FIGURE 2.4 Series–shunt feedback amplifier that incorporates an emitter follower output stage to reduce the effects of feed-forward through the feedback network.

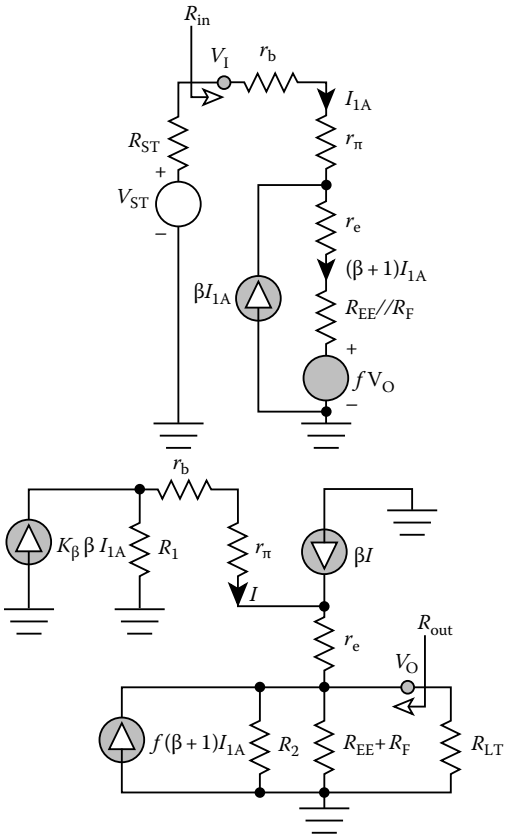


FIGURE 2.5 Small-signal model of the buffered series–shunt feedback amplifier.

Letting

$$R' = R_2 \| (R_{EE} + R_F) \| R_{LT} \tag{2.20}$$

an analysis of the structure in Figure 2.5 reveals

$$\frac{V_O}{I_{1A}} = (\beta + 1)\left[\frac{R'}{R' + r_{ib} + (1-\alpha)R_1}\right]\left\{\alpha K_\beta R_1 + f[r_{ib} + (1-\alpha)R_1]\right\} \tag{2.21}$$

which suggests negligible feed-forward for

$$f \ll \frac{\alpha K_\beta R_1}{r_{ib} + (1-\alpha)R_1} \tag{2.22}$$

Note that for large R_1, Equation 2.22 implies the requirement $f \ll \beta K_\beta$, which is easier to satisfy than is Equation 2.10. Assuming the validity of Equations 2.19, 2.21, and 2.22 deliver an open-loop voltage gain, $G_{vo}(R_{ST}, R_{LT})$, of

$$G_{vo}(R_{ST}, R_{LT}) = \alpha K_\beta \left[\frac{R'}{r_{ib} + (1-\alpha)R_{ST} + R_{EE} \| R_F}\right]\left[\frac{R_1}{R' + r_{ib} + (1-\alpha)R_1}\right] \tag{2.23}$$

Recalling Equation 2.1, which demonstrates that K_β approaches β for large R, Equation 2.23 suggests an open-loop gain that is nominally proportional to β^2 if R_1 is also large.

Using the concepts evoked by Equations 2.17 and 2.18, the driving-point input and output impedances can now be determined. In a typical realization of the buffered series–shunt feedback amplifier, the resistance, R_2, in Figure 2.4 is very large because it is manifested as the output resistance of a common base current sink that is employed to stabilize the operating point of transistor $Q3$. For this situation, and assuming the resistance R_1 is large, the resultant driving-point input resistance is larger than its predecessor input resistance by a factor of approximately $(\beta + 1)$. Similarly, it is easy to show that for large R_1 and large R_2, the driving-point output resistance is smaller than that predicted by Equation 2.18 by a factor approaching $(\beta + 1)$.

Although the emitter follower output stage in Figure 2.4 all but eliminates feed-forward signal transmission through the feedback network and increases both the driving-point input resistance and output conductance, a potential bandwidth penalty is paid by its incorporation into the basic series–shunt feedback cell. The fundamental problem is that if R_1 is too large, potentially significant Miller multiplication of the base–collector transition capacitance of transistor $Q2$ materializes. The resultant capacitive loading at the collector of transistor $Q1$ is exacerbated by large R, which may produce a dominant pole at a frequency that is too low to satisfy closed-loop bandwidth requirements. The bandwidth problem may be mitigated by coupling resistance R_1 to the collector of $Q2$ through a common base cascode. This stage appears as transistor $Q4$ in Figure 2.6.

Unfortunately, the use of the common base cascode indicated in Figure 2.6 may produce an open-loop amplifier with transfer characteristics that do not emulate a dominant pole response. In other words, the frequency of the compensated pole established by capacitive loading at the collector of transistor $Q1$ may be comparable to the frequencies of poles established elsewhere in the circuit, and particularly at the base node of transistor $Q1$. In this event, frequency compensation aimed toward achieving acceptable closed-loop damping can be implemented by replacing the feedback resistor R_F with the parallel combination of R_F and a feedback capacitance, say C_F, as indicated by the dashed branch in Figure 2.6. The resultant frequency-domain feedback factor $f(s)$ is

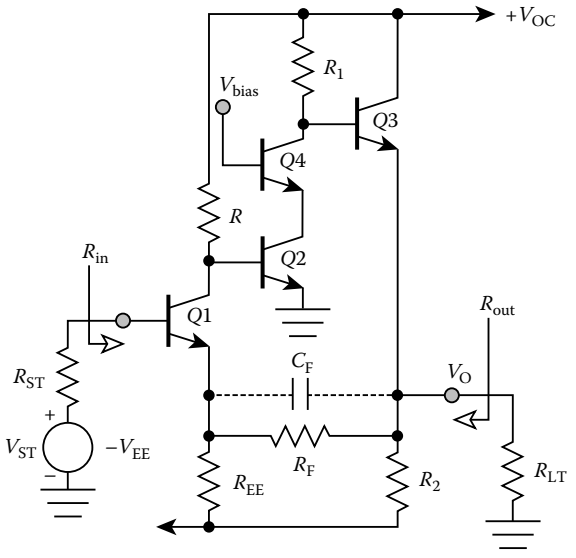


FIGURE 2.6 Buffered series–shunt feedback amplifier with common base cascode compensation of the common emitter amplifier formed by transistor $Q2$. A feedback zero is introduced by the capacitance C_F to achieve acceptable closed-loop damping.

$$f(s) = f\left[\frac{1+\frac{s}{z}}{1+\frac{fs}{z}}\right] \tag{2.24}$$

where

f is the feedback factor given by Equation 2.7

z is the frequency of the introduced compensating zero

$$z = \frac{1}{R_F C_F} \tag{2.25}$$

The pole in Equation 2.24 is inconsequential if the closed-loop amplifier bandwidth B_{cl} satisfies the restriction, $fB_{cl}R_FC_F = B_{cl}(R_{EE}||R_F)C_F \ll 1$.

2.3 Shunt–Series Feedback Amplifier

Although the series–shunt circuit functions as a voltage amplifier, the shunt–series configuration (see the ac schematic diagram depicted in Figure 2.7a) is best suited as a current amplifier. In the subject circuit,

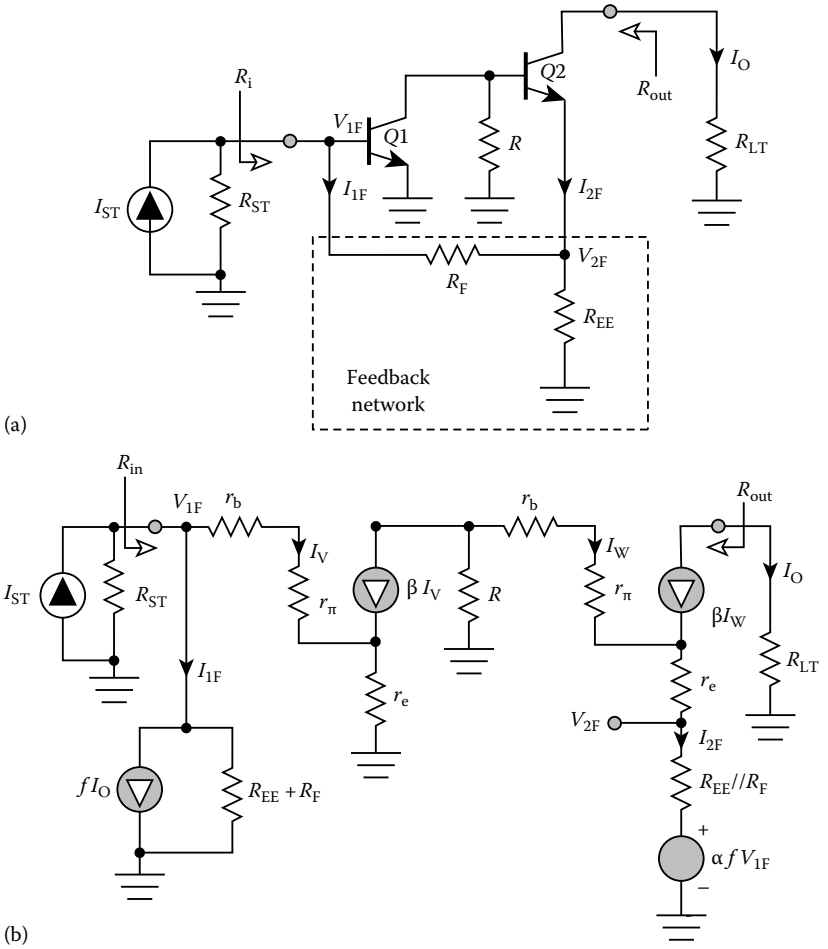


FIGURE 2.7 (a) AC schematic diagram of a bipolar shunt–series feedback amplifier. (b) Low-frequency small-signal equivalent circuit of the feedback amplifier.

the $Q2$ emitter current, which is a factor of $(1/\alpha)$ of the output signal current, I_O, is sampled by the feedback network formed of the resistances, R_{EE} and R_F. The sampled current is fed back as a current in shunt with the amplifier input port. Because output current is fed back as a current to a current-driven input port, the resultant driving-point output resistance is large, and the driving-point input resistance is small. These characteristics allow for a closed-loop current gain, $G_1(R_{ST}, R_{LT}) = I_O/I_{ST}$, that is relatively independent of source and load resistances and insensitive to transistor parameters.

In the series–shunt amplifier, h-parameters were selected to model the feedback network because the topology of an h-parameter equivalent circuit is, similar to the amplifier in which the feedback network is embedded, a series shunt, or Thévenin–Norton, topology. In analogous train of thought compels the use of g-parameters to represent the feedback network in Figure 2.7a. With reference to the branch variables defined in the schematic diagram,

$$\begin{bmatrix} I_{1F} \\ V_{2F} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_{EE}+R_F} & -\frac{R_{EE}}{R_{EE}+R_F} \\ \frac{R_{EE}}{R_{EE}+R_F} & R_{EF} \| R_F \end{bmatrix} \begin{bmatrix} V_{1F} \\ I_{2F} \end{bmatrix} \tag{2.26}$$

Noting that the feedback network current, I_{2F}, relates to the amplifier output current, I_O, in accordance with

$$I_{2F} = -\frac{I_O}{\alpha} \tag{2.27}$$

and letting the feedback factor, f, be

$$f = \frac{1}{\alpha}\left(\frac{R_{EE}}{R_{EE}+R_F}\right) \tag{2.28}$$

the small-signal equivalent circuit of shunt–series feedback amplifier becomes the network diagrammed in Figure 2.7b. Note that the voltage-controlled voltage source, $\alpha f V_{1F}$, models the feed-forward transfer mechanism of the feedback network, where the controlling voltage, V_{1F}, is

$$V_{1F} = [r_b + r_\pi + (\beta+1)r_c]I_V = (\beta+1)r_{ib}I_V \tag{2.29}$$

An analysis of the model in Figure 2.7b confirms that the second-stage, signal-base current I_W relates to the first-stage, signal-base current I_v as

$$\frac{I_W}{I_V} = -\frac{\alpha(R + f r_{ib})}{r_{ib} + R_{EE} \| R_F + (1-\alpha)R} \tag{2.30}$$

For

$$f \ll \frac{R}{r_{ib}} \tag{2.31}$$

which offsets feed-forward effects,

$$\frac{I_W}{I_V} \approx -\frac{\alpha R}{r_{ib} + R_{EE} \| R_F + (1-\alpha)R} \triangleq -K_r \tag{2.32}$$

Observe that the constant K_r tends toward β for large R, as can be verified by an inspection of Figure 2.7b.

Using Equation 2.32, the open-loop current gain, found by setting f to zero, is

$$G_{IO}(R_{ST}, R_{LT}) = \frac{I_O}{I_{ST}}\bigg\|_{f=0} = \alpha K_r \left\{ \frac{R_{ST} \| (R_{EE} + R_F)}{r_{ib} + (1-\alpha)[R_{ST} \| (R_{EE} + R_F)]} \right\} \tag{2.33}$$

and recalling Equation 2.28, the loop gain T is

$$\begin{aligned} T = fG_{IO}(R_{ST}, R_{LT}) &= \frac{1}{\alpha}\left(\frac{R_{EE}}{R_{EE} + R_F}\right) G_{IO}(R_{ST}, R_{LT}) \\ &= K_r \left(\frac{R_{EE}}{R_{EE} + R_F + R_{ST}}\right) \left\{ \frac{R_{ST}}{r_{ib} + (1-\alpha)[R_{ST} \| (R_{EE} + R_F)]} \right\} \end{aligned} \tag{2.34}$$

By inspection of the model in Figure 2.7b, the open-loop input resistance, R_{ino}, is

$$R_{ino} = (R_{EE} + R_F) \| [(\beta + 1) r_{ib}] \tag{2.35}$$

and, within the context of an infinitely large Early resistance, the open-loop output resistance, R_{outo}, is infinitely large.

The closed-loop current gain of the shunt–series feedback amplifier is now found to be

$$G_1(R_{ST}, R_{LT}) = \frac{G_{IO}(R_{ST}, R_{LT})}{1 + T} \approx \alpha \left(1 + \frac{R_F}{R_{EE}}\right) \tag{2.36}$$

where the indicated approximation exploits the presumption that the loop gain T is much larger than one. As a result of the large loop-gain assumption, note that the closed-loop gain is independent of the source and load resistances and is invulnerable to uncertainties and perturbations in transistor parameters. The closed-loop output resistance, which exceeds its open-loop counterpart, remains infinitely large. Finally, the closed-loop driving-point input resistance of the shunt–series amplifier is

$$R_{in} = \frac{R_{ino}}{1 + fG_{IO}(\infty, R_{LT})} \approx \left(1 + \frac{R_F}{R_{EE}}\right) \frac{r_{ib}}{K_r} \tag{2.37}$$

2.4 Shunt–Shunt Feedback Amplifier

2.4.1 Circuit Modeling and Analysis

The ac schematic diagram of the third type of single-loop feedback amplifier, the shunt–shunt triple, is drawn in Figure 2.8a. A cascade interconnection of three transistors $Q1$, $Q2$, and $Q3$, forms the open loop, while the feedback subcircuit is the single resistance, R_F. This resistance samples the output voltage, V_O, as a current fed back to the input port. Output voltage is fed back as a current to a current-driven input port, so both the driving-point input and output resistances are very small. Accordingly, the circuit operates best as a transresistance amplifier in that its closed-loop transresistance, $R_M(R_{ST}, R_{LT}) = V_O/I_{ST}$, is nominally invariant with source resistance, load resistance, and transistor parameters.

The shunt–shunt nature of the subject amplifier suggests the propriety of y-parameter modeling of the feedback network. For the electrical variables indicated in Figure 2.8a,

$$\begin{bmatrix} I_{1F} \\ I_{2F} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_F} & -\frac{1}{R_F} \\ -\frac{1}{R_F} & \frac{1}{R_F} \end{bmatrix} \begin{bmatrix} V_{1F} \\ V_O \end{bmatrix} \tag{2.38}$$

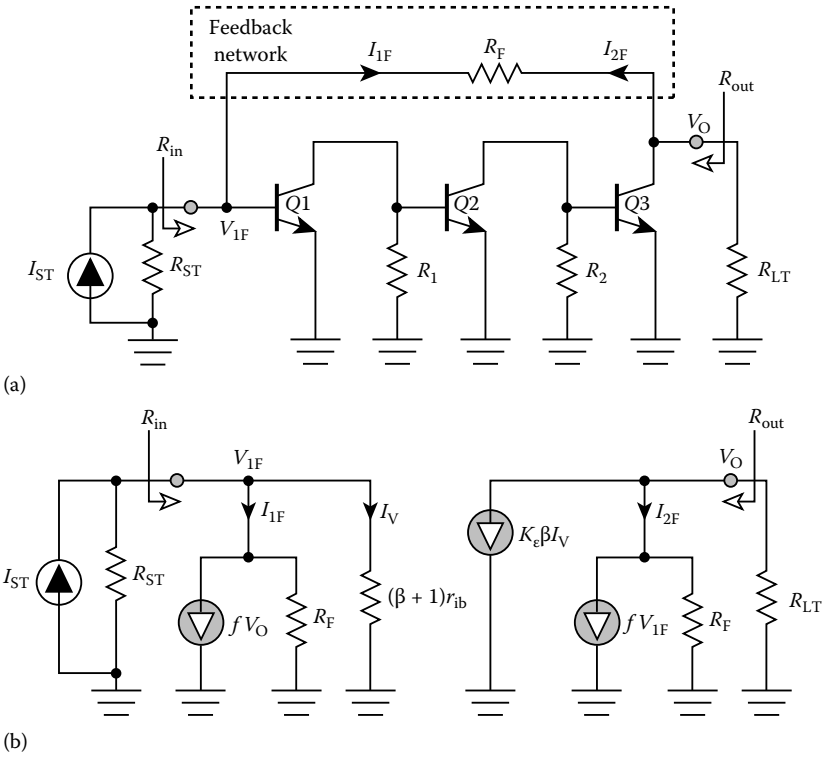


FIGURE 2.8 (a) AC schematic diagram of a bipolar shunt–shunt feedback amplifier. (b) Low-frequency small-signal equivalent circuit of the feedback amplifier.

which implies that a resistance, R_F, loads both the input and the output ports of the open-loop three-stage cascade. The short-circuit admittance relationship in Equation 2.38 also suggests a feedback factor, f, given by

$$f = \frac{1}{R_F} \tag{2.39}$$

The foregoing observations and the small-signal modeling experience gained with the preceding two feedback amplifiers lead to the equivalent circuit submitted in Figure 2.8b. For analytical simplicity, the model reflects the assumption that all three transistors in the open loop have identical small-signal parameters. Moreover, the constant, K_ε, which symbolizes the ratio of the signal-base current flowing into transistor $Q3$ to the signal-base current conducted by transistor $Q1$, is given by

$$K_\varepsilon = \left[\frac{\alpha R_1}{r_{ib} + (1-\alpha)R_1}\right]\left[\frac{\alpha R_2}{r_{ib} + (1-\alpha)R_2}\right] \tag{2.40}$$

Finally, the voltage-controlled current source, fV_{1F}, accounts for feed-forward signal transmission through the feedback network. If such feed-forward is to be negligible, the magnitude of this controlled current must be significantly smaller than $K_\varepsilon\beta I_v$, a current that emulates feed-forward through the open-loop amplifier. Noting that the input port voltage, V_{1F}, in the present case remains the same as that specified by Equation 2.29, negligible feed-forward through the feedback network mandates

$$R_F \gg \frac{r_{ib}}{\alpha K_\varepsilon} \tag{2.41}$$

Because the constant K_ε in Equation 2.40 tends toward β^2 if R_1 and R_2 are large resistances, Equation 2.41 is relatively easy to satisfy.

With feed-forward through the feedback network ignored, an analysis of the model in Figure 2.8b provides an open-loop transresistance, $R_{MO}(R_{ST}, R_{LT})$, of

$$R_{MO}(R_{ST}, R_{LT}) = -\alpha K_\varepsilon \left[\frac{R_F \| R_{ST}}{r_{ib}(1-\alpha)(R_F \| R_{ST})}\right](R_F \| R_{LT}) \tag{2.42}$$

while the loop gain is

$$\begin{aligned} T = fR_{MO}(R_{ST}, R_{LT}) &= -\frac{R_{MO}(R_{ST}, R_{LT})}{R_F} \\ &= \alpha K_\varepsilon \left[\frac{R_{ST}}{R_{ST}+R_F}\right]\left[\frac{R_F \| R_{ST}}{r_{ib}(1-\alpha)(R_F \| R_{ST})}\right] \end{aligned} \tag{2.43}$$

For $T \gg 1$, the corresponding closed-loop transresistance $R_M(R_{ST}, R_{LT})$ is

$$R_M(R_{ST}, R_{LT}) = \frac{R_{MO}(R_{ST}, R_{LT})}{1+T} \approx -R_F \tag{2.44}$$

Finally, the approximate driving-point input and output resistances are, respectively,

$$R_{in} \approx \left(\frac{r_{ib}}{\alpha K_\varepsilon}\right)\left(1 + \frac{R_F}{R_{LT}}\right) \tag{2.45}$$

$$R_{out} \approx \left[\frac{r_{ib} + (1-\alpha)(R_F \| R_{ST})}{\alpha K_\varepsilon}\right]\left(1 + \frac{R_F}{R_{ST}}\right) \tag{2.46}$$

2.4.2 Design Considerations

Because the shunt–shunt triple uses three gain stages in the open-loop amplifier, its loop gain is significantly larger than the loop gains provided by either of the previously considered feedback cells. Accordingly, the feedback triple affords superior desensitization of the closed-loop gain with respect to transistor parameters and source and load resistances; but the presence of a cascade of three common emitter gain stages in the open loop of the amplifier complicates frequency compensation and limits the 3 dB bandwidth. The problem is that, although each common emitter stage approximates a dominant pole amplifier, none of the critical frequencies in the cluster of poles established by the cascade interconnection of these units is likely to be dominant. The uncompensated closed loop is therefore predisposed to unacceptable underdamping, thereby making compensation via an introduced feedback zero difficult.

At least three compensation techniques can be exploited to optimize the performance of the shunt–shunt feedback amplifier [3,7–9]. The first of these techniques entail pole splitting of the open-loop interstage through the introduction of a capacitance, C_c, between the base and the collector terminals of transistor $Q2$, as depicted in the ac schematic diagram of Figure 2.9. In principle, pole splitting can be invoked on any one of the three stages of the open loop; but pole splitting of the interstage is most desirable because such compensation of the first stage proves effective only for large source resistance. Moreover, the resultant dominant pole becomes dependent on the source termination. On the other hand, pole splitting of the third stage produces a dominant pole that is sensitive to load termination. In conjunction with pole splitting, a feedback zero can be introduced, if necessary, to increase closed-loop damping by replacing the feedback resistance, R_F, by the parallel combination of R_F and a feedback

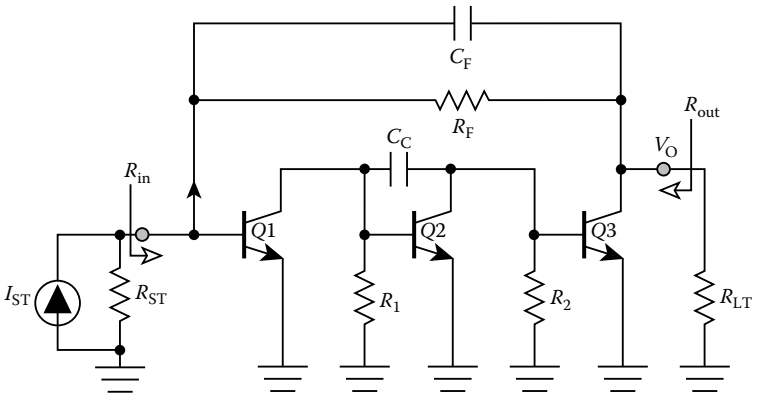


FIGURE 2.9 AC schematic diagram of a frequency-compensated shunt–shunt triple. The capacitance, C_c, achieves open-loop pole splitting, while the capacitance, C_F, implements a compensating feedback network zero.

capacitance, C_F, as illustrated in Figure 2.9. This compensation produces left-half-plane zero in the feedback factor at $s = -(1/R_F)$.

A second compensation method broadbands the interstage of the open-loop amplifier through local current feedback introduced by the resistance, R_X, in Figure 2.10. Simultaneously, the third stage is broadbanded by way of a common base cascode transistor $Q4$. Because emitter degeneration of the interstage reduces the open-loop gain, an emitter follower (transistor $Q5$) is embedded between the feedback network and the output port of the open-loop third stage. As in the case of the series–shunt feedback amplifier, the first-order effect of this emitter follower is to increase feed-forward signal transmission through the open-loop amplifier by a factor that approaches $(\beta + 1)$.

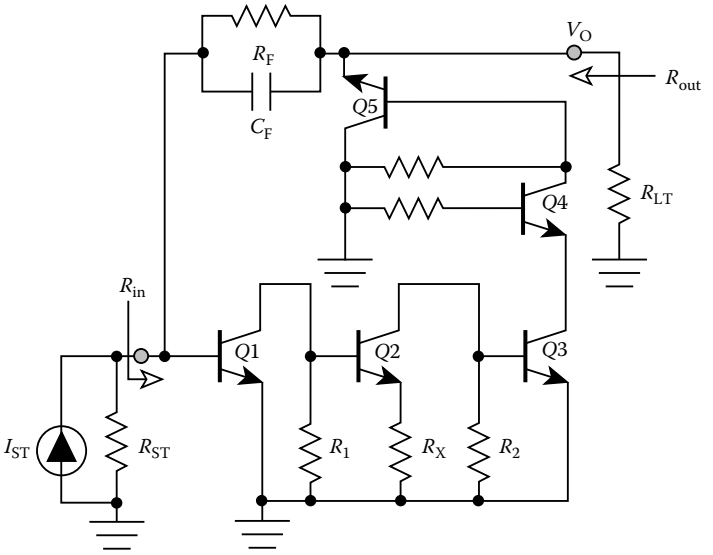


FIGURE 2.10 AC schematic diagram of an alternative compensation scheme for the shunt–shunt triple. Transistor $Q2$ is broadbanded by the emitter degeneration resistance R_X and transistor $Q3$ is broadbanded by the common base cascode transistor $Q4$. The emitter follower transistor, $Q5$, minimizes feed-forward signal transmission through the feedback network.

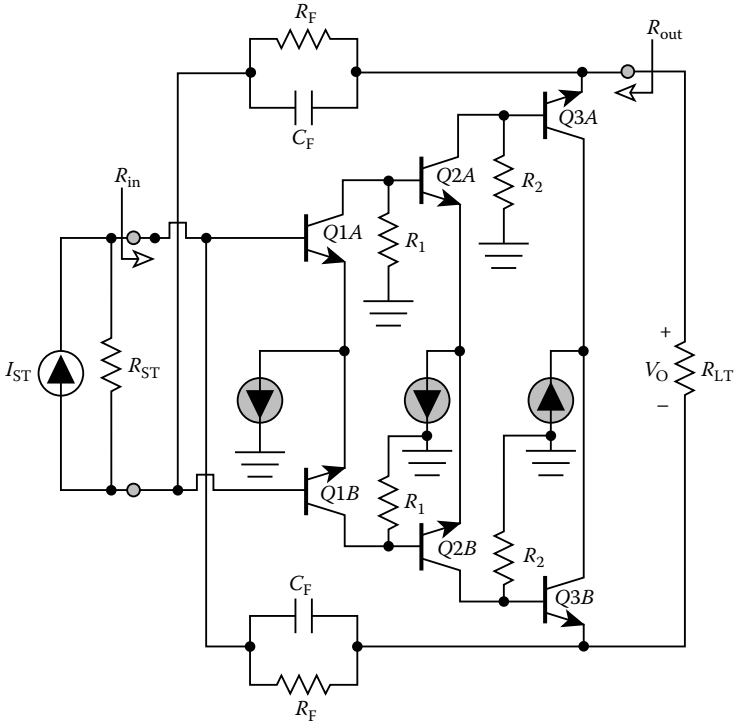


FIGURE 2.11 AC schematic diagram of a differential realization of the compensated shunt–shunt feedback amplifier. The balanced stage boasts improved bandwidth over its single-ended counterpart because of its use of only two high-gain stages in the open loop. The emitter follower pair *Q*3*A* and *Q*3*B* diminishes feed-forward transmission through the feedback network composed of the shunt interconnection of resistor R_F with capacitor C_F.

A final compensation method is available if shunt–shunt feedback is implemented as the balanced differential architecture (see the ac schematic diagram offered in Figure 2.11). By exploiting the antiphase nature of opposite collectors in a balanced common emitter topology, a shunt–shunt feedback amplifier can be realized with only two gain stages in the open loop. The resultant closed-loop 3 dB bandwidth is invariably larger than that of its three-stage single-ended counterpart, because the open loop is now characterized by only two, as opposed to three, fundamental critical frequencies. Because the forward gain implicit to two amplifier stages is smaller than the gain afforded by three stages of amplification, a balanced emitter follower (transistors *Q*3*A* and *Q*3*B*) is incorporated to circumvent the deleterious relative effects of feed-forward signal transmission through the feedback network.

2.5 Series–Series Feedback Amplifier

Figure 2.12a is the ac schematic diagram of the series–series feedback amplifier. Three transistors, *Q*1, *Q*2, and *Q*3, are embedded in the open-loop amplifier, while the feedback subcircuit is the wye configuration formed of the resistances R_X, R_Y, and R_Z. Although it is possible to realize series–series feedback via emitter degeneration of a single-stage amplifier, the series–series triple offers substantially more loop gain and thus better desensitization of the forward gain with respect to both transistor parameters and source and load terminations.

In Figure 2.12a, the feedback wye senses the *Q*3 emitter current, which is a factor of $(1/\alpha)$ of the output signal current I_O. This sampled current is fed back as a voltage in series with the emitter of *Q*1. Because output current is fed back as a voltage to a voltage-driven input port, both the driving-point input and

output resistances are large. The circuit is therefore best suited as a transconductance amplifier in the sense that for large loop gain, its closed-loop transconductance, $G_M(R_{ST}, R_{LT}) = I_O/V_{ST}$, is almost independent of the source and load resistances.

The series–series topology of the subject amplifier conduces *z*-parameter modeling of the feedback network. Noting the electrical variables delineated in the diagram of Figure 2.12a,

$$\begin{bmatrix} V_{1F} \\ V_{2F} \end{bmatrix} = \begin{bmatrix} R_X + R_Z & R_Z \\ R_Z & R_Y + R_Z \end{bmatrix} \begin{bmatrix} I_{1F} \\ I_{2F} \end{bmatrix} \tag{2.47}$$

Equation 2.47 suggests that the open-circuit feedback network resistances loading the emitters of transistors $Q1$ and $Q3$ are $(R_X + R_Z)$ and $(R_Y + R_Z)$, respectively, and the voltage fed back to the emitter of transistor $Q1$ is $R_Z I_{2F}$. Because the indicated feedback network current I_{2F} is $(-I_O/\alpha)$, this fed back voltage is equivalent to $(-R_Z I_O/\alpha)$, which suggests a feedback factor, f, of

$$f = \frac{R_Z}{\alpha} \tag{2.48}$$

Finally, the feed-forward through the feedback network if $R_Z I_{1F}$. Because I_{1F} relates to the signal-base current I_V flowing into transistor $Q1$ by $I_{1F} = (\beta + 1)I_V$, this feed-forward voltage is also expressible

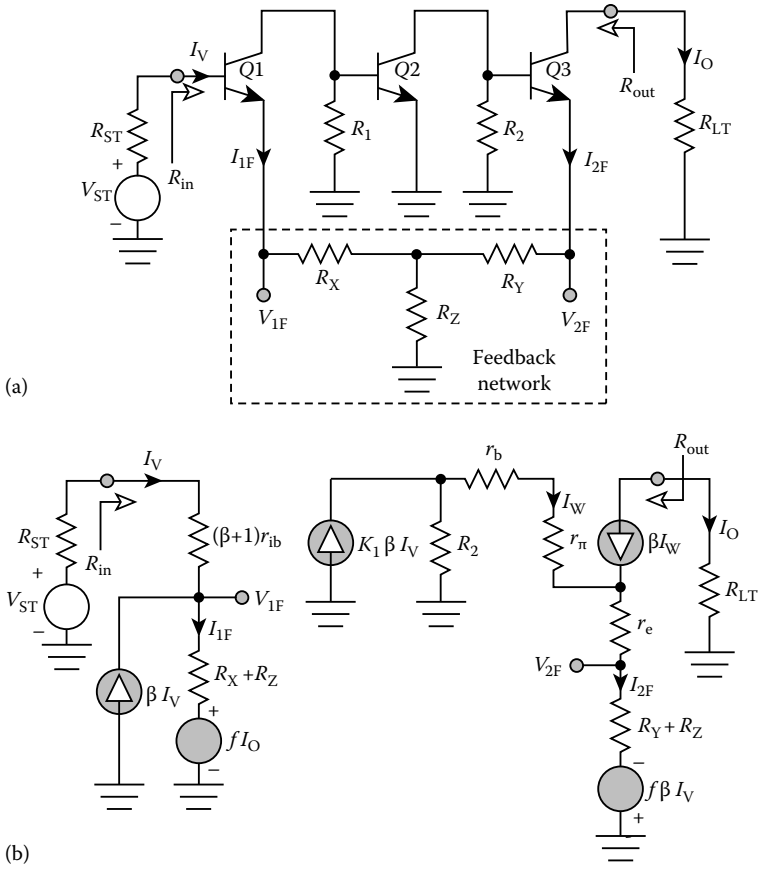


FIGURE 2.12 (a) AC schematic diagram of a bipolar series–series feedback amplifier. (b) Low-frequency, small-signal equivalent circuit of the feedback amplifier.

as $(-f\beta I_V)$. The foregoing observations and the hybrid-π method of a bipolar junction transistor produce the small-signal model depicted in Figure 2.12b. In this model, all transistors are presumed to have identical corresponding small-signal parameters, and the constant, K_1, is

$$K_1 = \frac{\alpha R_1}{r_{ib} + (1-\alpha)R_1} \tag{2.49}$$

An analysis of the model of Figure 2.12b confirms that the ratio of the signal current, I_W, flowing into the base of transistor $Q3$ to the signal-base current, I_V, of transistor $Q1$ is

$$\frac{I_W}{I_V} = \frac{\alpha K_1 R_2 \left(1 + \frac{f}{K_1 R_2}\right)}{r_{ib} + R_Y + R_Z + (1-\alpha)R_2} \tag{2.50}$$

This result suggests that feed-forward effects through the feedback network are negligible if $|f| \ll K_1 R_2$, which requires

$$R_Z \ll \alpha K_1 R_2 \tag{2.51}$$

In view of the fact that the constant, K_1, approaches β for large values of the resistance, R_1, Equation 2.51 is not a troublesome inequality. Introducing a second constant, K_2, such that

$$K_2 \triangleq \frac{\alpha R_2}{r_{ib} + R_Y + R_Z + (1-\alpha)R_2} \tag{2.52}$$

the ratio I_W/I_V in Equation 2.50 becomes

$$\frac{I_W}{I_V} \approx K_1 K_2 \tag{2.53}$$

assuming Equation 2.51 is satisfied.

Given the propriety of Equation 2.50 and using Equation 2.53 the open-loop transconductance, $G_{MO}(R_{ST}, R_{LT})$ is found to be

$$G_{MO}(R_{ST}, R_{LT}) = -\left\{\frac{\alpha K_1 K_2}{r_{ib} + R_X + R_Z + (1-\alpha)R_{ST}}\right\} \tag{2.54}$$

and recalling Equation 2.48, the loop gain T is

$$T = -\left(\frac{R_Z}{\alpha}\right) G_{MO}(R_{ST}, R_{LT}) = \frac{K_1 K_2 R_Z}{r_{ib} + R_X + R_Z + (1-\alpha)R_{ST}} \tag{2.55}$$

It follows that for $T \gg 1$, the closed-loop transconductance is

$$G_M(R_{ST}, R_{LT}) = \frac{G_{MO}(R_{ST}, R_{LT})}{1+T} \approx -\frac{\alpha}{R_Z} \tag{2.56}$$

The Early resistance is large enough to justify its neglect, so the open loop, and thus the closed-loop, driving-point output resistances are infinitely large. On the other hand, the closed-loop driving-point input resistance R_{in} can be shown to be

$$R_{in} = R_{ino}[1 + fG_{MO}(0, R_{LT})] \approx (\beta + 1)K_1 K_2 R_Z \tag{2.57}$$

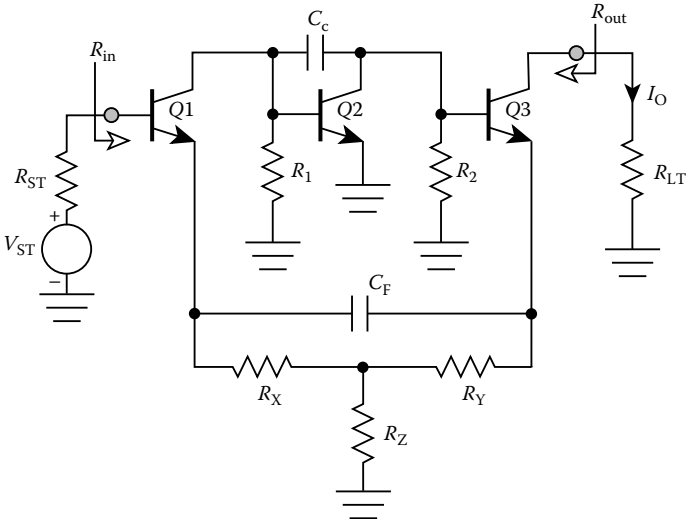


FIGURE 2.13 AC schematic diagram of a frequency-compensated series–series feedback triple. The capacitance, C_c, achieves pole splitting in the open-loop configuration, while the capacitance, C_F, introduces a zero in the feedback factor of the closed-loop amplifier.

Similar to its shunt–shunt counterpart, the series–series feedback amplifier uses three open-loop gain stages to produce large loop gain. However, also similar to the shunt–shunt triple, frequency compensation via an introduced feedback zero is difficult unless design care is exercised to realize a dominant pole open-loop response. To this end, the most commonly used compensation is pole splitting in the open loop, combined, if required, with the introduction of a zero in the feedback factor. The relevant ac schematic diagram appears in Figure 2.13 where the indicated capacitance, C_c, inserted across the base–collector terminals of transistor $Q3$ achieves the aforementioned pole splitting compensation. The capacitance, C_F, in Figure 2.13 delivers a frequency-dependent feedback factor, $f(s)$ of

$$f(s) = f\left[\frac{1+\frac{s}{z}}{1+\frac{s}{z}\left(\frac{R_Z}{R_Z+R_X\|R_Y}\right)}\right] \tag{2.58}$$

where the frequency z of the introduced zero derives from

$$\frac{1}{z} = (R_X + R_Y)\left(1+\frac{R_X\|R_Y}{R_Z}\right)C_F \tag{2.59}$$

The corresponding pole in Equation 2.58 is insignificant if the closed-loop amplifier is designed for a bandwidth, B_{cl} that satisfies the inequality, $B_{c1}(R_X + R_Y)C_F \ll 1$.

As is the case with shunt–shunt feedback, an alternative frequency compensation scheme is available if series–series feedback is implemented as a balanced differential architecture. The pertinent ac schematic diagram, inclusive of feedback compensation, appears in Figure 2.14. This diagram exploits the fact that the feedback wye consisting of the resistances, R_X, R_Y, and R_Z as utilized in the single-ended configurations of Figures 2.12a and 2.13 can be transformed into the feedback delta of Figure 2.15. The terminal volt–ampere characteristics of the two networks in Figure 2.15 are identical, provided that the delta subcircuit elements, R_F, R_U, and R_V, are chosen in accordance with

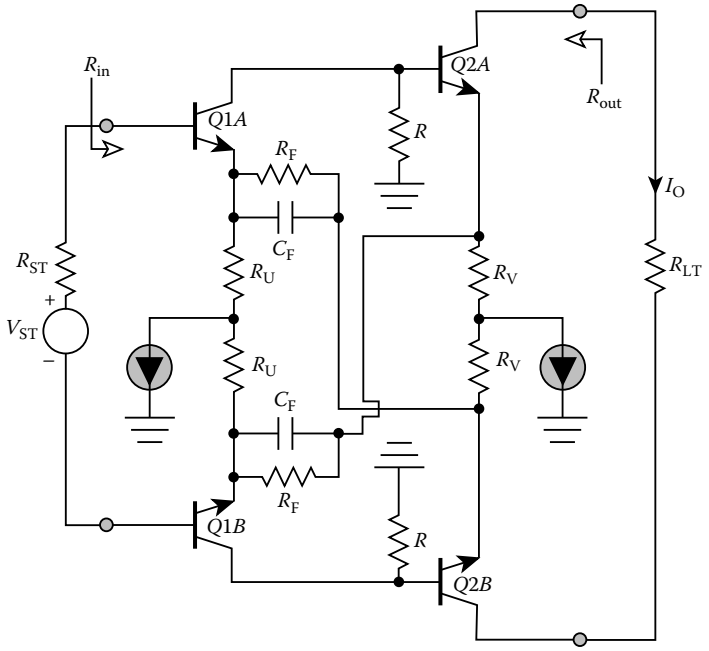


FIGURE 2.14 AC schematic diagram of a balanced differential version of the series–series feedback amplifier. The circuit utilizes only two, as opposed to three, gain stages in the open loop.

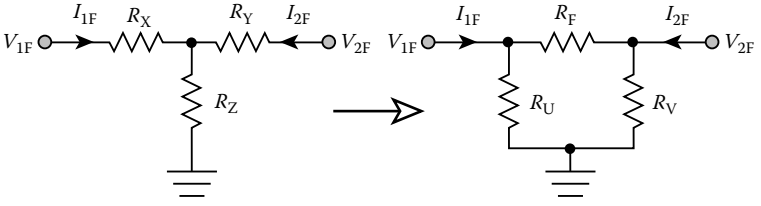


FIGURE 2.15 Transformation of the wye feedback subcircuit used in the amplifier of Figure 2.13 to the delta subcircuit exploited in Figure 2.14. The resistance transformation equations are given by Equations 2.60 through 2.62.

$$R_F = (R_X + R_Y)\left(1 + \frac{R_X \| R_Y}{R_Z}\right) \tag{2.60}$$

$$\frac{R_U}{R_F} = \frac{R_Z}{R_Y} \tag{2.61}$$

$$\frac{R_V}{R_F} = \frac{R_Z}{R_X} \tag{2.62}$$

2.6 Dual-Loop Feedback

As mentioned previously, a simultaneous control of the driving-point I/O resistances, as well as the closed-loop gain, mandates the use of dual global loops comprised of series and shunt feedback signal paths. The two global dual-loop feedback architectures are the series–series/shunt–shunt feedback amplifier and the series–shunt/shunt–series feedback amplifier. In Sections 2.6.1 and 2.6.2, both of

these units are studied by judiciously applying the relevant analytical results established earlier for pertinent single-loop feedback architectures. The ac schematic diagrams of these respective circuit realizations are provided, and engineering design considerations are offered.

2.6.1 Series–Series/Shunt–Shunt Feedback Amplifier

Figure 2.16 is a behavioral abstraction of the series–series/shunt–shunt feedback amplifier. Two port z parameters are used to model the series–series feedback subcircuit, for which feed-forward is tacitly ignored and the feedback factor associated with its current-controlled voltage source is f_{ss}. On the other hand, y parameters model the shunt–shunt feedback network, where the feedback factor relative to its voltage-controlled current source is f_{pp}. As in the series–series network, feed-forward in the shunt–shunt subcircuit is presumed negligible. The four-terminal amplifier around which the two feedback units are connected has an open loop (meaning $f_{ss} = 0$ and $f_{pp} = 0$, but with the loading effects of both feedback circuits considered) transconductance of $G_{MO}(R_{ST}, R_{LT})$.

With f_{pp} set to zero to deactivate shunt–shunt feedback, the resultant series–series feedback network is a transconductance amplifier with a closed-loop transconductance, $G_{MS}(R_{ST}, R_{LT})$, is

$$G_{MS}(R_{ST}, R_{LT}) = \frac{I_O}{V_{ST}} = \frac{G_{MO}(R_{ST}, R_{LT})}{1 + f_{ss}G_{MO}(R_{ST}, R_{LT})} \approx \frac{1}{f_{ss}} \tag{2.63}$$

where the loop gain, $f_{ss}G_{MO}(R_{ST}, R_{LT})$, is presumed much larger than one, and the loading effects of both the series–series feedback subcircuit and the deactivated shunt–shunt feedback network are incorporated into $G_{MO}(R_{ST}, R_{LT})$. The transresistance, $R_{MS}(R_{ST}, R_{LT})$, implied by Equation 2.63, which expedites the study of the shunt–shunt component of the feedback configuration, is

$$R_{MS}(R_{ST}, R_{LT}) = \frac{V_O}{I_{ST}} = R_{ST}R_{LT}\frac{I_O}{V_{ST}} \approx \frac{R_{ST}R_{LT}}{f_{ss}} \tag{2.64}$$

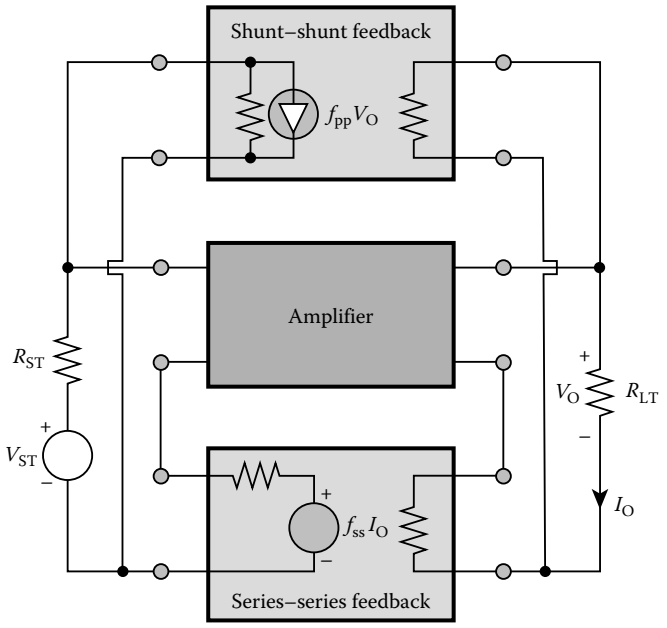


FIGURE 2.16 System-level diagram of a series–series/shunt–shunt dual-loop feedback amplifier. Note that feed-forward signal transmission through either feedback network is ignored.

The series–series feedback input and output resistances R_{ins} and R_{outs}, respectively, are large and given by

$$R_{ins} = R_{ino}[1 + f_{ss}G_{MO}(0, R_{LT})] \tag{2.65}$$

and

$$R_{outs} = R_{outo}[1 + f_{ss}G_{MO}(R_{ST}, 0)] \tag{2.66}$$

where the zero feedback ($f_{ss} = 0$ and $f_{pp} = 0$) values, R_{ino} and R_{outo}, of these driving-point quantities are computed with due consideration given to the loading effects imposed on the amplifier by both feedback subcircuits.

When shunt–shunt feedback is applied around the series–series feedback cell, the configuration becomes a transresistance amplifier. The effective open-loop transresistance is $R_{MS}(R_{ST}, R_{LT})$, as defined by Equation 2.64. Noting a feedback of f_{pp}, the corresponding closed-loop transresistance is

$$R_M(R_{ST}, R_{LT}) \approx \frac{\frac{R_{ST}R_{LT}}{f_{ss}}}{1 + f_{pp}\left(\frac{R_{ST}R_{LT}}{f_{ss}}\right)} \tag{2.67}$$

which is independent of amplifier model parameters, despite the unlikely condition of an effective loop gain $f_{pp}R_{ST}R_{LT}/f_{ss}$ that is much larger than 1. It should be interjected, however, that Equation 2.67 presumes negligible feed-forward through the shunt–shunt feedback network. This presumption may be inappropriate owing to the relatively low closed-loop gain afforded by the series–series feedback subcircuit. Ignoring this potential problem temporarily, Equation 2.67 suggests a closed-loop voltage gain $A_V(R_{ST}, R_{LT})$ of

$$A_V(R_{ST}, R_{LT}) = \frac{V_O}{V_S} = \frac{R_M(R_{ST}, R_{LT})}{R_{ST}} \approx \frac{R_{LT}}{f_{ss} + f_{pp}R_{ST}R_{LT}} \tag{2.68}$$

The closed-loop, driving-point output resistance R_{out}, can be straightforwardly calculated by noting that the open circuit ($R_{LT} \to \infty$) voltage gain, A_{VO}, predicted by Equation 2.68 is $A_{VO} = 1/f_{pp}\ R_{ST}$. Accordingly, Equation 2.68 is alternatively expressible as

$$A_V(R_{ST}, R_{LT}) \approx A_{VO}\left(\frac{R_{LT}}{R_{LT} + \frac{f_{ss}}{f_{pp}R_{ST}}}\right) \tag{2.69}$$

Because Equation 2.69 is a voltage divider relationship stemming from a Thévenin model of the output port of the dual-loop feedback amplifier, as delineated in Figure 2.17, it follows that the driving-point output resistance is

$$R_{out} \approx \frac{f_{ss}}{f_{pp}R_{ST}} \tag{2.70}$$

Observe that, similar to the forward gain characteristics, the driving-point output resistance is nominally insensitive to changes and other uncertainties in open-loop amplifier parameters. Moreover, this output resistance is directly proportional to the ratio f_{ss}/f_{pp} of feedback factors. As illustrated in preceding sections, the individual feedback factors, and thus the ratio of feedback factors, is likely to be proportional to a ratio of resistances. In view of the fact that resistance ratios can be tightly controlled in a monolithic fabrication process, R_{out} in Equation 2.70 is accurately prescribed for a given source termination.

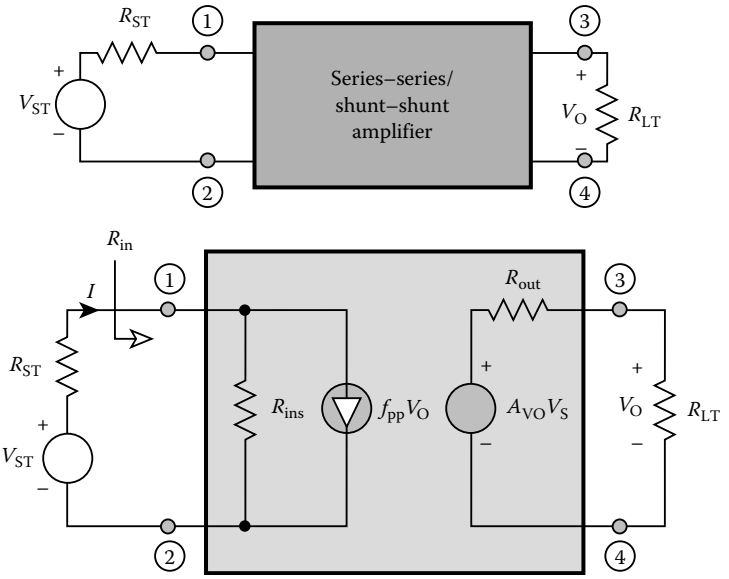


FIGURE 2.17 Norton equivalent input and Thévenin equivalent output circuits for the series–series/shunt–shunt dual-loop feedback amplifier.

The driving-point input resistance R_{in} can be determined from a consideration of the input port component of the system-level equivalent circuit depicted in Figure 2.17. This resistance is the ratio of V_{ST} to I, under the condition of $R_S = 0$. With $R_S = 0$, Equation 2.68 yields $V_O = R_{LT}V_{ST}/f_{ss}$ and thus, Kirchhoff's voltage law (KVL) applied around the input port of the model at hand yields

$$R_{in} = \frac{R_{ins}}{1 + \frac{f_{pp}R_{LT}R_{ins}}{f_{ss}}} \approx \frac{f_{ss}}{f_{pp}R_{LT}} \tag{2.71}$$

where the "open-loop" input resistance R_{ins}, defined by Equation 2.65, is presumed large. Similar to the driving-point output resistance of the series–series/shunt–shunt feedback amplifier, the driving-point input resistance is nominally independent of open-loop amplifier parameters.

It is interesting to observe that the input resistance in Equation 2.71 is inversely proportional to the load resistance by the same factor (f_{ss}/f_{pp}) that the driving-point output resistance in Equation 2.70 is inversely proportional to the source resistance. As a result,

$$\frac{f_{ss}}{f_{pp}} \approx R_{in}R_{LT} \equiv R_{out}R_{ST} \tag{2.72}$$

Thus, in addition to being stable performance indices for well-defined source and load terminations, the driving-point input and output resistances track one another, despite manufacturing uncertainties and changes in operating temperature that might perturb the individual values of the two feedback factors f_{ss} and f_{pp}.

The circuit property stipulated by Equation 2.72 has immediate utility in the design of wideband communication transceivers and other high-speed signal-processing systems [10–14]. In these and related applications, a cascade of several stages is generally required to satisfy frequency response, distortion, and noise specifications. A convenient way of implementing a cascade interconnection is to force each member of the cascade to operate under the match terminated case of

$R_{ST} = R_{in} = R_{LT} = R_{out} \triangleq R$. From Equation 2.72 match terminated operation demands feedback factors selected so that

$$R = \sqrt{\frac{f_{ss}}{f_{pp}}} \tag{2.73}$$

which forces a match terminated closed-loop voltage gain A_V^* of

$$A_V^* \approx \frac{1}{2f_{pp}R} = \frac{1}{2\sqrt{f_{pp}f_{ss}}} \tag{2.74}$$

The ac schematic diagram of a practical, single-ended series–series/shunt–shunt amplifier is submitted in Figure 2.18. An inspection of this diagram reveals a topology that coalesces the series–series and shunt–shunt triples studied earlier. In particular, the wye network formed of the three resistances, R_{Xx}, R_Y, and R_Z, comprises the series–series component of the dual-loop feedback amplifier. The capacitor, C_c, narrowbands the open-loop amplifier to facilitate frequency compensation of the series–series loop through the capacitance, C_{F1}. Compensated shunt feedback of the network is achieved by the parallel combination of the resistance, R_F and the capacitance, C_{F2}. If C_{F1} and C_c combine to deliver a dominant pole series–series feedback amplifier, C_{F2} is not necessary. Conversely, C_{F1} is superfluous if C_{F2} and C_c interact to provide a dominant pole shunt–shunt feedback amplifier. As in the single-ended series–series configuration, transistor $Q3$ can be broadbanded via a common base cascode. Moreover, if feedback through the feedback networks poses a problem, an emitter follower can be inserted at the port to which the shunt feedback path and the load termination are incident.

A low-frequency analysis of the circuit in Figure 2.18 is expedited by assuming high-β transistors having identical corresponding small-signal model parameters. This analysis, which in contrast to the

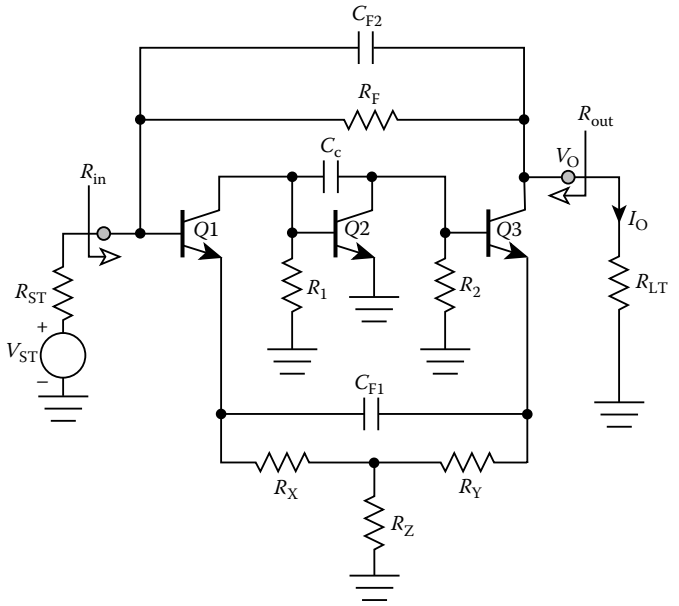


FIGURE 2.18 AC schematic diagram of a frequency-compensated, series–series/shunt–shunt, dual-loop feedback amplifier. The compensation is affected by the capacitances C_{F1} and C_{F2}, while C_c achieves pole splitting in the open-loop amplifier.

simplified behavioral analysis, does not ignore the electrical effects of the aforementioned feed-forward through the shunt–shunt feedback network, yields a voltage gain $A_V(R_{ST}, R_{LT})$, of

$$A_V(R_{ST}, R_{LT}) \approx -\left(\frac{R_{in}}{R_{in} + R_{ST}}\right)\left(\frac{R_{LT}}{R_{LT} + R_F}\right)\left(\frac{\alpha R_F}{R_Z} - 1\right) \tag{2.75}$$

where the driving-point input resistance of the amplifier R_{in} is

$$R_{in} \approx \frac{R_F + R_{LT}}{1 + \frac{\alpha R_{LT}}{R_Z}} \tag{2.76}$$

The driving-point output resistance R_{out} is

$$R_{out} \approx \frac{R_F + R_{ST}}{1 + \frac{\alpha R_{ST}}{R_z}} \tag{2.77}$$

As predicted by the behavioral analysis R_{in}, R_{out}, and $A_V(R_{ST}, R_{LT})$, are nominally independent of transistor parameters. Observe that the functional dependence of R_{in} on the load resistance, R_{LT}, is identical to the manner in which R_{out} is related to the source resistance R_{ST}. In particular, $R_{in} \equiv R_{out}$ if $R_{ST} \equiv R_{LT}$. For the match terminated case in which $R_{ST} = R_{in} = R_{LT} = R_{out} \triangleq R$,

$$R \approx \sqrt{\frac{R_F R_Z}{\alpha}} \tag{2.78}$$

The corresponding match terminated voltage gain in Equation 2.75 collapses to

$$A_V^* \approx -\left(\frac{R_F - R}{2R}\right) \tag{2.79}$$

Similar to the series–series and shunt–shunt triples, many of the frequency compensation problems implicit to the presence of three open-loop stages can be circumvented by realizing the series–series/shunt–shunt amplifier as a two-stage differential configuration. Figure 2.19 is the ac schematic diagram of a compensated differential series–series/shunt–shunt feedback dual.

2.6.2 Series–Shunt/Shunt–Series Feedback Amplifier

The only other type of global dual-loop architecture is the series–shunt/shunt–series feedback amplifier; the behavioral diagram appears in Figure 2.20. The series–shunt component of this system, which is modeled by *h*-parameters, has a negligibly small feed-forward factor and a feedback factor of f_{sp}. Hybrid *g*-parameters model the shunt–series feedback structure, which has a feedback factor of f_{ps} and a presumably negligible feed-forward factor. The four-terminal amplifier around which the two feedback units are connected has an open loop (meaning $f_{sp} = 0$ and $f_{ps} = 0$, but with the loading effects of both feedback circuits considered) voltage gain of $A_{VO}(R_{ST}, R_{LT})$.

For $f_{ps} = 0$, the series–shunt feedback circuit voltage gain $A_{VS}(R_{ST}, R_{LT})$, is

$$A_{VS}(R_{ST}, R_{LT}) = \frac{V_O}{V_{ST}} = \frac{A_{VO}(R_{ST}, R_{LT})}{1 + f_{sp}A_{VO}(R_{ST}, R_{LT})} \approx \frac{1}{f_{sp}} \tag{2.80}$$

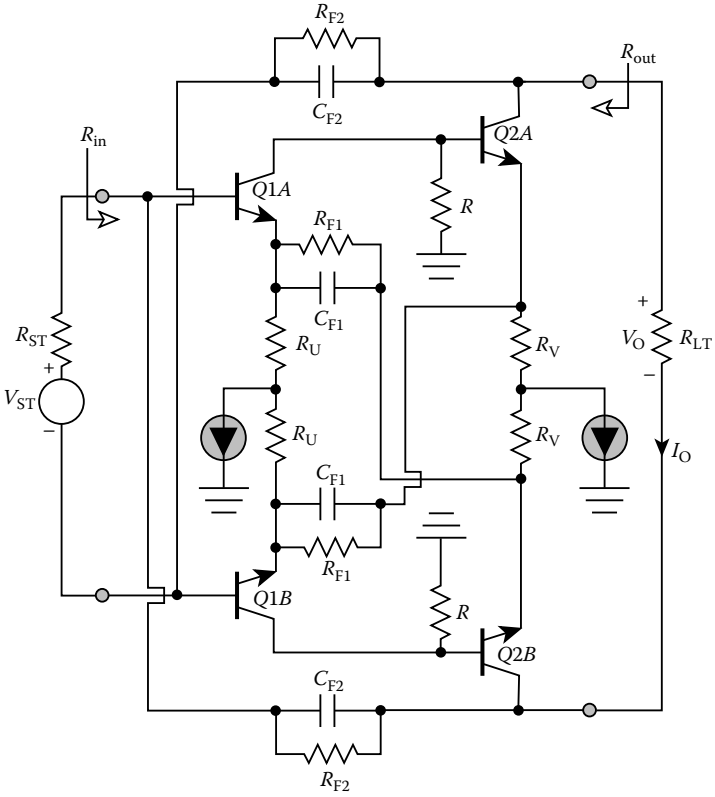


FIGURE 2.19 AC schematic diagram of the differential realization of a compensated series–series/shunt–shunt feedback amplifier.

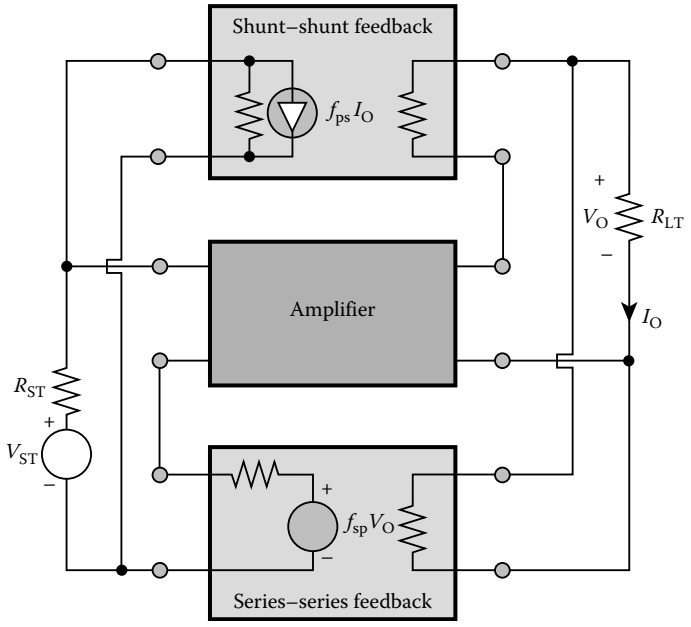


FIGURE 2.20 System-level diagram of a series–shunt/shunt–series, dual-loop feedback amplifier. Note that feedforward signal transmission through either feedback network is ignored.

where the approximation reflects an assumption of a large loop gain. When the shunt–series component of the feedback amplifier is activated, the dual-loop configuration functions as a current amplifier. Its effective open-loop transfer function is the current gain, $A_{IS}(R_{ST}, R_{LT})$, established by the series–shunt amplifier; namely,

$$A_{IS}(R_{ST}, R_{LT}) = \frac{I_O}{I_{ST}} = \left(\frac{R_{ST}}{R_{LT}}\right)\frac{V_O}{V_{ST}} \approx \frac{R_{ST}}{f_{sp}R_{LT}} \tag{2.81}$$

It follows that the current gain, $A_I(R_{ST}, R_{LT})$, of the closed loop is

$$A_I(R_{ST}, R_{LT}) \approx \frac{\frac{R_{ST}}{f_{sp}R_{LT}}}{1 + f_{ps}\left(\frac{R_{ST}}{f_{sp}R_{LT}}\right)} = \frac{R_{ST}}{f_{sp}R_{LT} + f_{ps}R_{ST}} \tag{2.82}$$

while the corresponding voltage gain, $A_V(R_{ST}, R_{LT})$, assuming negligible feed-forward through the shunt–series feedback network, is

$$A_V(R_{ST}, R_{LT}) = \frac{R_{LT}}{R_{ST}}A_I(R_{ST}, R_{LT}) \approx \frac{R_{LT}}{f_{sp}R_{LT} + f_{ps}R_{ST}} \tag{2.83}$$

Repeating the analytical strategy employed to determine the input and output resistances of the series–series/shunt–shunt configuration, Equation 2.83 delivers a driving-point input resistance of

$$R_{in} \approx \frac{f_{sp}R_{LT}}{f_{ps}} \tag{2.84}$$

and a driving-point output resistance of

$$R_{out} \approx \frac{f_{ps}R_{ST}}{f_{sp}} \tag{2.85}$$

Similar to the forward voltage gain, the driving-point input and output resistances of the series–shunt/shunt–series feedback amplifier are nominally independent of active element parameters. Note, however, that the input resistance is directly proportional to the load resistance by a factor (f_{sp}/f_{ps}), which is the inverse of the proportionality constant that links the output resistance to the source resistance. Specifically,

$$\frac{f_{sp}}{f_{ps}} = \frac{R_{in}}{R_{LT}} = \frac{R_{ST}}{R_{out}} \tag{2.86}$$

Thus, although R_{in} and R_{out} are reliably determined for well-defined load and source terminations, they do not track one another as well as they do in the series–series/shunt–shunt amplifier. Using Equation 2.86, the voltage gain in Equation 2.83 is expressible as

$$A_V(R_{ST}, R_{LT}) \approx \frac{1}{f_{sp}\left(1 + \sqrt{\frac{R_{out}R_{ST}}{R_{in}R_{LT}}}\right)} \tag{2.87}$$

The simplified ac schematic diagram of a practical series–shunt/shunt–series feedback amplifier appears in Figure 2.21. In this circuit, series–shunt feedback derives from the resistances, R_{EE1} and R_{F1}, and

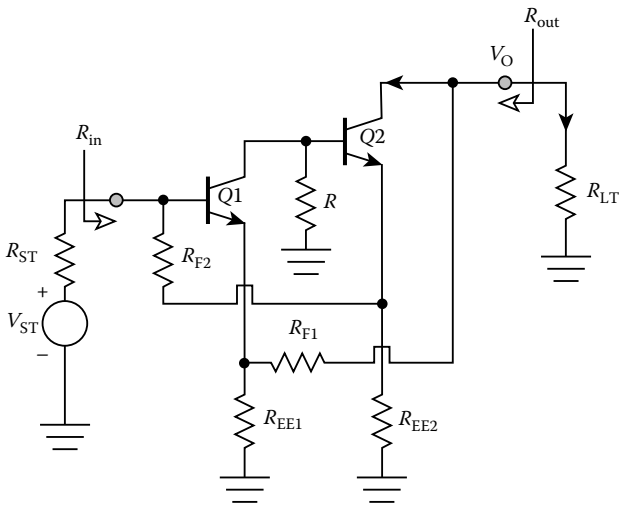


FIGURE 2.21 AC schematic diagram of a series–shunt/shunt–series, dual-loop feedback amplifier.

shunt–series feedback is determined by the resistances, R_{EE2} and R_{F2}. Because this circuit topology merges the series–shunt and shunt–series pairs, requisite frequency compensation, which is not shown in the subject figure, mirrors the relevant compensation schemes studied earlier. Note, however, that a cascade of only two open-loop gain stages renders compensation easier to implement and larger 3 dB bandwidths easier to achieve in the series–series/shunt–shunt circuit, which requires three open-loop gain stages for a single-ended application.

For high-β transistors having identical corresponding small-signal model parameters, a low-frequency analysis of the circuit in Figure 2.21 gives a voltage gain of

$$A_V(R_{ST}, R_{LT}) \approx \left(\frac{\alpha R_{in}}{R_{in} + \alpha R_S}\right)\left(1 + \frac{R_{F1}}{R_{EE1}}\right) \tag{2.88}$$

where the driving-point input resistance, R_{in}, of the subject amplifier is

$$R_{in} \approx \alpha R_{LT}\left(\frac{1 + \frac{R_{F2}}{R_{EE2}}}{1 + \frac{R_{F1}}{R_{EE1}} + \frac{R_{LT}}{R_{EE1} \| R_{EE2}}}\right) \tag{2.89}$$

The driving-point output resistance, R_{out}, is

$$R_{out} \approx R_{ST}\left(\frac{1 + \frac{R_{F1}}{R_{EE1}}}{1 + \frac{R_{F2}}{R_{EE2}} + \frac{R_{ST}}{R_{EE1} \| R_{EE2}}}\right) \tag{2.90}$$

2.7 Summary

This section documents small-signal performance equations, general operating characteristics, and engineering design guidelines for the six most commonly used global feedback circuits. These observations derive from analyses based on the judicious application of signal flow theory to the small-signal model that results when the subject feedback network is supplanted by an appropriate two-port parameter equivalent circuit.

Four of the six fundamental feedback circuits are single-loop architectures.

1. Series–shunt feedback amplifier functions best as a voltage amplifier in that its input resistance is large, and its output resistance is small. Because only two gain stages are required in the open loop, the amplifier is relatively easy to compensate for acceptable closed-loop damping and features potentially large 3 dB bandwidth. A computationally efficient analysis aimed toward determining loop gain, closed-loop gain, I/O resistances, and the condition that renders feed-forward through the feedback network inconsequential is predicated on replacing the feedback subcircuit with its *h*-parameter model.
2. Shunt–series feedback amplifier is a current amplifier in that its input resistance is small, and its output resistance is large. Similar to its series–shunt dual, only two gain stages are required in the open loop. Computationally efficient analyses are conducted by replacing the feedback subcircuit with its *g*-parameter model.
3. Shunt–shunt feedback amplifier is a transresistance signal processor in that both its input and output resistances are small. Although this amplifier can be realized theoretically with only a single open-loop stage, a sufficiently large loop gain generally requires a cascade of three open-loop stages. As a result, pole splitting is invariably required to ensure an open-loop dominant pole response, thereby limiting the achievable closed-loop bandwidth. In addition compensation of the feedback loop may be required for acceptable closed-loop damping. The bandwidth and stability problems implicit to the use of three open-loop gain stages can be circumvented by a balanced differential realization, which requires a cascade of only two open-loop gain stages. Computationally efficient analyses are conducted by replacing the feedback subcircuit with its *y*-parameter model.
4. Series–series feedback amplifier is a transconductance signal processor in that both its input and output resistances are large. Similar to its shunt–shunt counterpart, its implementation generally requires a cascade of three open-loop gain stages. Computationally efficient analyses are conducted by replacing the feedback subcircuit with its *z*-parameter model.

The two remaining feedback circuits are dual-loop topologies that can stabilize the driving-point input and output resistances, as well as the forward gain characteristics, with respect to shifts in active element parameters. One of these latter architectures, the series–series/shunt–shunt feedback amplifier, is particularly well suited to electronic applications that require a multistage cascade.

1. Series–series/shunt–shunt feedback amplifier coalesces the series–series architecture with its shunt–shunt dual. It is particularly well suited to applications, such as wideband communication networks, which require match terminated source and load resistances. Requisite frequency compensation and broadbanding criteria mirror those incorporated in the series–series and shunt–shunt single-loop feedback topologies.
2. Series–shunt/shunt–series feedback amplifier coalesces the series–shunt architecture with its shunt–series dual. Although its input resistance can be designed to match the source resistance seen by the input port of the amplifier, and its output resistance can be matched to the load resistance driven by the amplifier, match terminated operating ($R_{in} = R_{ST} = R_{LT} = R_{out}$) is not feasible. Requisite frequency compensation and broadbanding criteria mirror those incorporated in the series–shunt and shunt–series single-loop feedback topologies.

References

1. J. Millman and A. Grabel, *Microelectronics,* 2nd edn., New York: McGraw-Hill, 1987, Chap. 12.
2. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley-Interscience, 1984, pp. 424–432.
3. R. G. Meyer, R. Eschenbach, and R. Chin, A wideband ultralinear amplifier from DC to 300 MHz, *IEEE J. Solid-State Circuits*, SC-9, 167–175, Aug. 1974.

4. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, New York: Holt, Rinehart Winston, 1987, pp. 428–441.
5. J. M. Early, Effects of space-charge layer widening in junction transistors, *Proc. IRE*, 46, 1141–1152, Nov. 1952.
6. A. J. Cote Jr. and J. B. Oakes, *Linear Vacuum-Tube and Transistor Circuits*, New York: McGraw-Hill, 1961, pp. 40–46.
7. R. G. Meyer and R. A. Blauschild, A four-terminal wideband monolithic amplifier, *IEEE J. Solid-State Circuits*, SC-17, 634–638, Dec. 1981.
8. M. Ohara, Y. Akazawa, N. Ishihara, and S. Konaka, Bipolar monolithic amplifiers for a gigabit optical repeater, *IEEE J. Solid-State Circuits*, SC-19, 491–497, Aug. 1985.
9. M. J. N. Sibley, R. T. Univin, D. R. Smith, B. A. Boxall, and R. J. Hawkins, A monolithic transimpedance preamplifier for high speed optical receivers, *Br. Telecommun. Tech. J.*, 2, 64–66, July 1984.
10. J. F. Kukielka and C. P. Snapp, Wideband monolithic cascadable feedback amplifiers using silicon bipolar technology, *IEEE Microw. Millimeter-Wave Circuits Symp. Dig.*, 2, 330–331, June 1982.
11. R. G. Meyer, M. J. Shensa, and R. Eschenbach, Cross modulation and intermodulation in amplifiers at high frequencies, *IEEE J. Solid-State Circuits*, SC-7, 16–23, Feb. 1972.
12. K. H. Chan and R. G. Meyer, A low distortion monolithic wide-band amplifier, *IEEE J. Solid-State Circuits*, SC-12, 685–690, Dec. 1977.
13. A. Arbel, Multistage transistorized current modules, *IEEE Trans. Circuits Syst.*, CT-13, 302–310, Sept. 1966.
14. A. Arbel, *Analog Signal Processing and Instrumentation*, London: Cambridge University, 1980, Chap. 3.
15. W. G. Beall, New feedback techniques for high performance monolithic wideband amplifiers, Electron. Res. Group, University of Southern California, Tech. Memo., Jan. 1990.

3

General Feedback Theory

3.1 Introduction **3**-1
3.2 Indefinite-Admittance Matrix **3**-1
3.3 Return Difference **3**-7
3.4 Null Return Difference **3**-11
References **3**-13

Wai-Kai Chen
University of Illinois at Chicago

3.1 Introduction

In Section 1.2, we used the ideal feedback model to study the properties of feedback amplifiers. The model is useful only if we can separate a feedback amplifier into the basic amplifier $\mu(s)$ and the feedback network $\beta(s)$. The procedure is difficult and sometimes virtually impossible, because the forward path may not be strictly unilateral, the feedback path is usually bilateral, and the input and output coupling networks are often complicated. Thus, the ideal feedback model is not an adequate representation of a practical amplifier. In the remainder of this section, we shall develop Bode's feedback theory, which is applicable to the general network configuration and avoids the necessity of identifying the transfer functions $\mu(s)$ and $\beta(s)$.

Bode's feedback theory [1] is based on the concept of return difference, which is defined in terms of network determinants. We show that the return difference is a generalization of the concept of the feedback factor of the ideal feedback model, and can be measured physically from the amplifier itself. We then introduce the notion of null return difference and discuss its physical significance. Because the feedback theory will be formulated in terms of the first- and second-order cofactors of the elements of the indefinite-admittance matrix of a feedback circuit, we first review briefly the formulation of the indefinite-admittance matrix.

3.2 Indefinite-Admittance Matrix

Figure 3.1 is an n-terminal network N composed of an arbitrary number of active and passive network elements connected in any way whatsoever. Let $V_1, V_2, \ldots, V_n$ be the Laplace-transformed potentials measured between terminals $1, 2, \ldots, n$ and some arbitrary but unspecified reference point, and let $I_1, I_2, \ldots, I_n$ be the Laplace-transformed currents entering the terminals $1, 2, \ldots, n$ from outside the network. The network N together with its load is linear, so the terminal current and voltages are related by the equation

$$\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} + \begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{bmatrix} \tag{3.1}$$

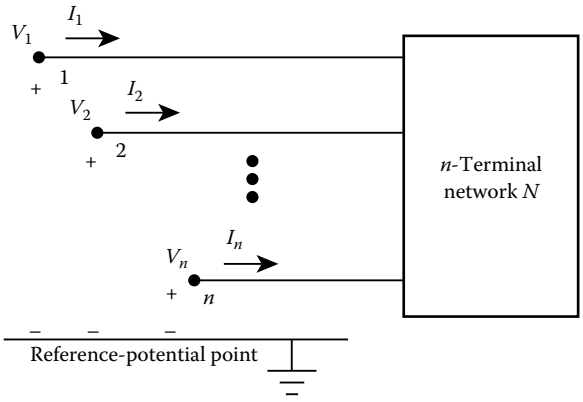


FIGURE 3.1 General symbolic representation of an n-terminal network.

or more succinctly as

$$\mathbf{I}(s) = \mathbf{Y}(s)\mathbf{V}(s) + \mathbf{J}(s) \tag{3.2}$$

where J_k ($k = 1, 2, \ldots, n$) denotes the current flowing into the kth terminal when all terminals of N are grounded to the reference point. The coefficient matrix $\mathbf{Y}(s)$ is called the indefinite-admittance matrix because the reference point for the potentials is some arbitrary but unspecified point outside the network. Notice that the symbol $\mathbf{Y}(s)$ is used to denote either the admittance matrix or the indefinite-admittance matrix. This should not create any confusion because the context will tell. In the remainder of this section, we shall deal exclusively with the indefinite-admittance matrix.

We remark that the short-circuit currents J_k result from the independent sources and/or initial conditions in the interior of N. For our purposes, we shall consider all independent sources outside the network and set all initial conditions to zero. Hence, $\mathbf{J}(s)$ is considered to be zero, and Equation 3.2 becomes

$$\mathbf{I}(s) = \mathbf{Y}(s)\mathbf{V}(s) \tag{3.3}$$

where the elements y_{ij} of $\mathbf{Y}(s)$ can be obtained as

$$y_{ij} = \left.\frac{I_i}{V_j}\right|_{v_x=0, x\neq j} \tag{3.4}$$

As an illustration, consider a small-signal equivalent model of a transistor in Figure 3.2. Its indefinite-admittance matrix is found to be

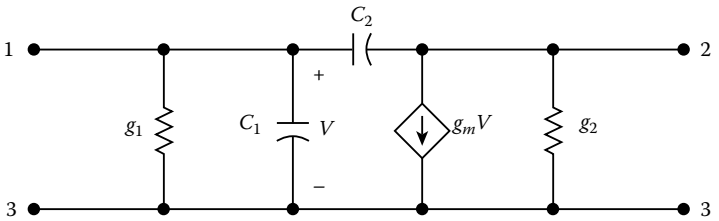


FIGURE 3.2 Small-signal equivalent network of a transistor.

$$\mathbf{Y}(s) = \begin{bmatrix} g_1 + sC_1 + sC_2 & -sC_2 & -g_1 - sC_1 \\ g_m - sC_2 & g_2 + sC_2 & -g_2 - g_m \\ -g_1 - sC_1 - g_m & -g_2 & g_1 + g_2 + g_m + sC_1 \end{bmatrix} \tag{3.5}$$

Observe that the sum of elements of each row or column is equal to zero. The fact that these properties are valid in general for the indefinite-admittance matrix will now be demonstrated.

To see that the sum of the elements in each column of $\mathbf{Y}(s)$ equals zero, we add all n equations of Equation 3.1 to yield

$$\sum_{i=1}^{n} \sum_{j=1}^{n} y_{ji} V_i = \sum_{m=1}^{n} I_m - \sum_{m=1}^{n} J_m = 0 \tag{3.6}$$

The last equation is obtained by appealing to Kirchhoff's current law for the node corresponding to the reference point. Setting all the terminal voltages to zero except the kth one, which is nonzero, gives

$$V_k \sum_{j=1}^{n} y_{jk} = 0 \tag{3.7}$$

Because $V_k \neq 0$, it follows that the sum of the elements of each column of $\mathbf{Y}(s)$ equals zero. Thus, the indefinite-admittance matrix is always singular.

To demonstrate that each row sum of $\mathbf{Y}(s)$ is also zero, we recognize that because the point of zero potential may be chosen arbitrarily, the currents J_k and I_k remain invariant when all the terminal voltages V_k are changed by the same but arbitrary constant amount. Thus, if V_0 is an n-vector, each element of which is $v_0 \neq 0$, then

$$\mathbf{I}(s) - \mathbf{J}(s) = \mathbf{Y}(s)[\mathbf{V}(s) + \mathbf{V}_0] = \mathbf{Y}(s)\mathbf{V}(s) + \mathbf{Y}(s)\mathbf{V}_0 \tag{3.8}$$

which after invoking Equation 3.2 yields that

$$\mathbf{Y}(s)\mathbf{V}_0 = 0 \tag{3.9}$$

or

$$\sum_{j=1}^{n} y_{ij} = 0, \quad i = 1, 2, \ldots, n \tag{3.10}$$

showing that each row sum of $\mathbf{Y}(s)$ equals zero.

Thus, if $\mathbf{Y}_{uv}$ denotes the submatrix obtained from an indefinite-admittance matrix $\mathbf{Y}(s)$ by deleting the uth row and vth column, then the (first order) cofactor, denoted by the symbol Y_{uv}, of the element y_{uv} of $\mathbf{Y}(s)$, is defined by

$$Y_{uv} = (-1)^{u+v} \det \mathbf{Y}_{uv} \tag{3.11}$$

As a consequence of the zero-row-sum and zero-column-sum properties, all the cofactors of the elements of the indefinite-admittance matrix are equal. Such a matrix is also referred to as the equicofactor matrix. If Y_{uv} and Y_{ij} are any two cofactors of the elements of $\mathbf{Y}(s)$, then

$$Y_{uv} = Y_{ij} \tag{3.12}$$

for all u, v, i, and j. For the indefinite-admittance matrix $\mathbf{Y}(s)$ of Equation 3.5 it is straightforward to verify that all of its nine cofactors are equal to

$$Y_{uv} = s^2C_1C_2 + s(C_1g_2 + C_2g_1 + C_2g_2 + g_mC_2) + g_1g_2 \tag{3.13}$$

for $u, v = 1, 2, 3$.

Denote by $\mathbf{Y}_{rp,sq}$ the submatrix obtained from $\mathbf{Y}(s)$ by striking out rows r and s and columns p and q. Then the second-order cofactor, denoted by the symbol $Y_{rp,sq}$ of the elements y_{rp}, and y_{sq} of $\mathbf{Y}$(s) is a scalar quantity defined by the relation

$$Y_{rp,sq} = \text{sgn}(r-s)\text{sgn}(p-q)(-1)^{r+p+s+q}\ \det\mathbf{Y}_{rp,sq} \tag{3.14}$$

where $r \neq s$ and $p \neq q$, and

$$\text{sgn}\ u = +1 \quad \text{if}\ u > 0 \tag{3.15a}$$

$$\text{sgn}\ u = -1 \quad \text{if}\ u < 0 \tag{3.15b}$$

The symbols $\mathbf{Y}_{uv}$ and Y_{uv} or $\mathbf{Y}_{rp,sq}$ and $Y_{rp,sq}$ should not create any confusion because one is in boldface whereas the other is italic. Also, for our purposes, it is convenient to define

$$Y_{rp,sq} = 0, \quad r = s \text{ or } p = q \tag{3.16a}$$

or

$$\text{sgn}\ 0 = 0 \tag{3.16b}$$

This convention will be followed throughout the remainder of this section.

As an example, consider the hybrid-pi equivalent network of a transistor in Figure 3.3. Assume that each node is an accessible terminal of a four-terminal network. Its indefinite-admittance matrix is

$$\mathbf{Y}(s) = \begin{bmatrix} 0.02 & 0 & -0.02 & 0 \\ 0 & 5\times10^{-12}s & 0.2 - 5\times10^{-12}s & -0.2 \\ -0.02 & -5\times10^{-12}s & 0.024 + 105\times10^{-12}s & -0.004 - 10^{-10}s \\ 0 & 0 & -0.204 - 10^{-10}s & 0.204 + 10^{-10}s \end{bmatrix} \tag{3.17}$$

The second-order cofactor $Y_{31,42}$ and $Y_{11,34}$ of the elements of $\mathbf{Y}(s)$ of Equation 3.17 are computed as follows:

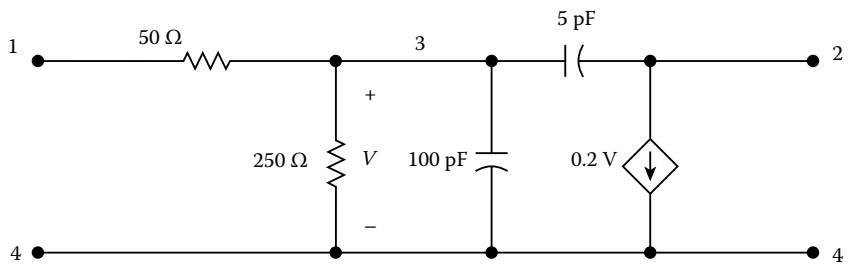


FIGURE 3.3 Hybrid-pi equivalent network of a transistor.

$$Y_{31,42} = \text{sgn}(3-4)\text{sgn}(1-2)(-1)^{3+1+4+2}\ \det\begin{bmatrix} -0.02 & 0 \\ 0.2 - 5\times 10^{-12}s & -0.2 \end{bmatrix}$$
$$= 0.004 \tag{3.18a}$$

$$Y_{11,34} = \text{sgn}(1-3)\text{sgn}(1-4)(-1)^{1+1+3+4}\ \det\begin{bmatrix} 5\times 10^{-12}s & 0.2 - 5\times 10^{-12}s \\ 0 & -0.204 - 10^{-10}s \end{bmatrix}$$
$$= 5\times 10^{-12}s\left(0.204 + 10^{-10}s\right) \tag{3.18b}$$

The usefulness of the indefinite-admittance matrix lies in the fact that it facilitates the computation of the driving-point or transfer functions between any pair of nodes or from any pair of nodes to any other pair. In the following, we present elegant, compact, and explicit formulas that express the network functions in terms of the ratios of the first- and/or second-order cofactors of the elements of the indefinite-admittance matrix.

Assume that a current source is connected between any two nodes r and s so that a current I_{sr} is injected into the rth node and at the same time is extracted from the sth node. Suppose that an ideal voltmeter is connected from node p to node q so that it indicates the potential rise from q to p, as depicted symbolically in Figure 3.4. Then the transfer impedance, denoted by the symbol $z_{rp,sq}$, between the node pairs rs and pq of the network of Figure 3.4 is defined by the relation

$$z_{rp,sq} = \frac{V_{pq}}{I_{sr}} \tag{3.19}$$

with all initial conditions and independent sources inside N set to zero. The representation is, of course, quite general. When $r = p$ and $s = q$, the transfer impedance $z_{rp,sq}$, becomes the *driving-point impedance* $z_{rr,ss}$ between the terminal pair rs.

In Figure 3.4, set all initial conditions and independent sources in N to zero and choose terminal q to be the reference-potential point for all other terminals. In terms of Equation 3.1, these operations are equivalent to setting $\mathbf{J} = 0$, $V_q = 0$, $I_x = 0$ for $x \neq r, s$ and $I_r = -I_s = I_{sr}$. Because $\mathbf{Y}(s)$ is an equicofactor matrix, the equations of Equation 3.1 are not linearly independent and one of them is superfluous. Let us suppress the sth equation from Equation 3.1, which then reduces to

$$\mathbf{I}_{-s} = \mathbf{Y}_{sq}\mathbf{V}_{-q} \tag{3.20}$$

where $\mathbf{I}_{-s}$ and $\mathbf{V}_{-q}$ denote the subvectors obtained from $\mathbf{I}$ and $\mathbf{V}$ of Equation 3.3 by deleting the sth row and qth row, respectively. Applying Cramer's rule to solve for V_p yields

$$V_p = \frac{\det\tilde{\mathbf{Y}}_{sq}}{\det\mathbf{Y}_{sq}} \tag{3.21}$$

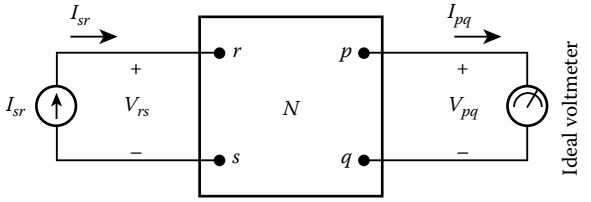


FIGURE 3.4 Symbolic representation for the measurement of the transfer impedance.

where $\tilde{\mathbf{Y}}_{sq}$ is the matrix derived from $\mathbf{Y}_{sq}$ by replacing the column corresponding to V_p by $\mathbf{I}_{-s}$. We recognize that $\mathbf{I}_{-s}$ is in the pth column if $p < q$ but in the $(p-1)$th column if $p > q$. Furthermore, the row in which I_{sr} appears is the rth row if $r < s$, but is the $(r-1)$th row if $r > s$. Thus, we obtain

$$(-1)^{s+q} \det\tilde{\mathbf{Y}}_{sq} = I_{sr} Y_{rp,sq} \tag{3.22}$$

In addition, we have

$$\det\mathbf{Y}_{sq} = (-1)^{s+q} Y_{sq} \tag{3.23}$$

Substituting these in Equation 3.21 in conjunction with Equation 3.19, we obtain

$$z_{rp,sq} = \frac{Y_{rp,sq}}{Y_{uv}} \tag{3.24}$$

$$z_{rr,ss} = \frac{Y_{rr,ss}}{Y_{uv}} \tag{3.25}$$

in which we have invoked the fact that $Y_{sq} = Y_{uv}$.

The *voltage gain*, denoted by $g_{rp,sq}$, between the node pairs rs and pq of the network of Figure 3.4 is defined by

$$g_{rp,sq} = \frac{V_{pq}}{V_{rs}} \tag{3.26}$$

again with all initial conditions and independent sources in N being set to zero. Thus, from Equations 3.24 and 3.25 we obtain

$$g_{rp,sq} = \frac{z_{rp,sq}}{z_{rr,ss}} = \frac{Y_{rp,sq}}{Y_{rr,ss}} \tag{3.27}$$

The symbols have been chosen to help us remember. In the numerators of Equations 3.24, 3.25, and 3.27, the order of the subscripts is as follows: r, the current injecting node; p, the voltage measurement node; s, the current extracting node; and q, the voltage reference node. Nodes r and p designate the input and output transfer measurement, and nodes s and q form a sort of double datum.

As an illustration, we consider the hybrid-pi transistor equivalent network of Figure 3.3. For this transistor, suppose that we connect a 100 Ω load resistor between nodes 2 and 4, and excite the resulting circuit by a voltage source V_{14}, as depicted in Figure 3.5. To simplify our notation, let $p = 10^{-9}s$. The indefinite-admittance matrix of the amplifier is

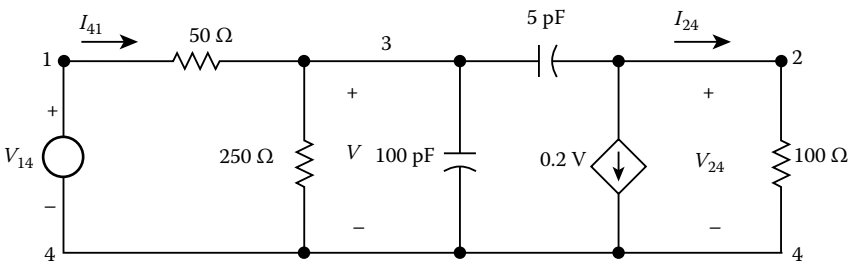


FIGURE 3.5 Transistor amplifier used to illustrate the computation of $g_{rp,sq}$.

$$\mathbf{Y}(s) = \begin{bmatrix} 0.02 & 0 & -0.02 & 0 \\ 0 & 0.01 + 0.005p & 0.2 - 0.005p & -0.21 \\ -0.02 & -0.005p & 0.024 + 0.105p & -0.004 - 0.1p \\ 0 & -0.01 & -0.204 - 0.1p & 0.214 + 0.1p \end{bmatrix} \tag{3.28}$$

To compute the voltage gain $g_{12,44}$, we appeal to Equation 3.27 and obtain

$$g_{12,44} = \frac{V_{24}}{V_{14}} = \frac{Y_{12,44}}{Y_{11,44}} = \frac{p - 40}{5p^2 + 21.7p + 2.4} \tag{3.29}$$

The input impedance facing the voltage source V_{14} is determined by

$$z_{11,44} = \frac{V_{14}}{I_{41}} = \frac{Y_{11,44}}{Y_{uv}} = \frac{Y_{11,44}}{Y_{44}} = \frac{50p^2 + 217p + 24}{p^2 + 4.14p + 0.08} \tag{3.30}$$

To compute the current gain defined as the ratio of the current I_{24} in the 100 Ω resistor to the input current I_{41}, we apply Equation 3.24 and obtain

$$\frac{I_{24}}{I_{41}} = 0.01\frac{V_{24}}{I_{41}} = 0.01 z_{12,44} = 0.01\frac{Y_{12,44}}{Y_{44}} = \frac{0.1p - 4}{p^2 + 4.14p + 0.08} \tag{3.31}$$

Finally, to compute the transfer admittance defined as the ratio of the load current I_{24} to the input voltage V_{14}, we appeal to Equation 3.27 and obtain

$$\frac{I_{24}}{V_{14}} = 0.01\frac{V_{24}}{V_{14}} = 0.01 g_{12,44} = 0.01\frac{Y_{12,44}}{Y_{11,44}} = \frac{p - 40}{500p^2 + 2170p + 240} \tag{3.32}$$

3.3 Return Difference

In the study of feedback amplifier response, we are usually interested in how a particular element of the amplifier affects that response. This element is either crucial in terms of its effect on the entire system or of primary concern to the designer. It may be the transfer function of an active device, the gain of an amplifier, or the immittance of a one-port network. For our purposes, we assume that this element x is the controlling parameter of a voltage-controlled current source defined by the equation

$$I = xV \tag{3.33}$$

To focus our attention on the element x, Figure 3.6 is the general configuration of a feedback amplifier in which the controlled source is brought out as a two-port network connected to a general four-port network, along with the input source combination of I_s and admittance Y_1 and the load admittance Y_2.

We remark that the two-port representation of a controlled source Equation 3.33 is quite general. It includes the special situation where a one-port element is characterized by its immittance. In this case, the controlling voltage V is the terminal voltage of the controlled current source I, and x become the one-port admittance.

The *return difference* $F(x)$ of a feedback amplifier with respect to an element x is defined as the ratio of the two functional values assumed by the first-order cofactor of an element of its indefinite-admittance matrix under the condition that the element x assumes its nominal value and the condition that the element x assumes the value zero. To emphasize the importance of the feedback element x, we express the

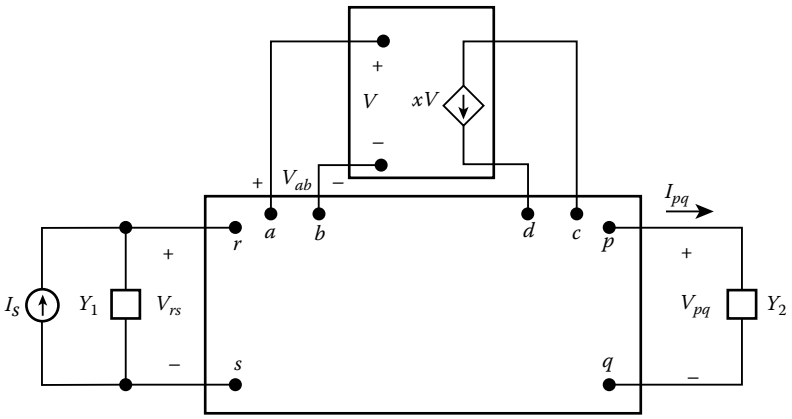


FIGURE 3.6 General configuration of a feedback amplifier.

indefinite-admittance matrix $\mathbf{Y}$ of the amplifier as a function of x, even though it is also a function of the complex-frequency variable s, and write $\mathbf{Y} = \mathbf{Y}(x)$. Then, we have [2]

$$F(x) \equiv \frac{Y_{uv}(x)}{Y_{uv}(0)} \tag{3.34}$$

where

$$Y_{uv}(0) = Y_{uv}(x)|_{x=0} \tag{3.35}$$

The physical significance of the return difference will now be considered. In the network of Figure 3.6, the input, the output, the controlling branch, and the controlled source are labeled as indicated. Then, the element x enters the indefinite-admittance matrix $\mathbf{Y}(x)$ in a rectangular pattern as shown next:

$$\mathbf{Y}(x) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} & & & \\ & & & \\ x & -x & & \\ -x & x & & \end{bmatrix} \end{array} \tag{3.36}$$

If in Figure 3.6 we replace the controlled current source xV by an independent current source of xA and set the excitation current source I_s to zero, the indefinite-admittance matrix of the resulting network is simply $\mathbf{Y}(0)$. By appealing to Equation 3.24, the new voltage V'_{ab} appearing at terminals a and b of the controlling branch is

$$V'_{ab} = x\frac{Y_{da,cb}(0)}{Y_{uv}(0)} = -x\frac{Y_{ca,db}(0)}{Y_{uv}(0)} \tag{3.37}$$

Notice that the current injecting point is terminal d, not c.

The preceding operation of replacing the controlled current source by an independent current source and setting the excitation I_s to zero can be represented symbolically as in Figure 3.7. Observe that the controlling branch is broken off as marked and a 1 V voltage source is applied to the right of the breaking mark. This 1 V sinusoidal voltage of a fixed angular frequency produces a current of x A at the controlled

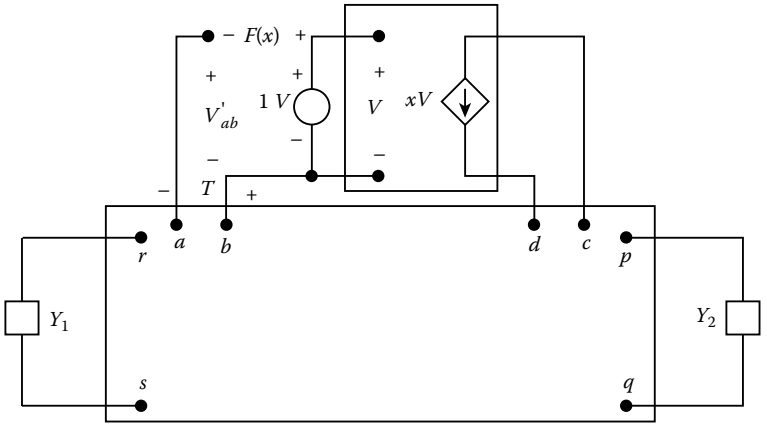


FIGURE 3.7 Physical interpretation of the return difference with respect to the controlling parameter of a voltage-controlled current source.

current source. The voltage appearing at the left of the breaking mark caused by this 1 V excitation is then V'_{ab} as indicated. This returned voltage V'_{ab} has the same physical significance as the loop transmission $\mu\beta$ defined for the ideal feedback model in Chapter 1. To see this, we set the input excitation to the ideal feedback model to zero, break the forward path, and apply a unit input to the right of the break, as depicted in Figure 3.8. The signal appearing at the left of the break is precisely the loop transmission.

For this reason, we introduce the concept of return ratio T, which is defined as the negative of the voltage appearing at the controlling branch when the controlled current source is replaced by an independent current source of x A and the input excitation is set to zero. Thus, the return ratio T is simply the negative of the returned voltage V'_{ab}, or $T = -V'_{ab}$. With this in mind, we next compute the difference between the 1 V excitation and the returned voltage V'_{ab} obtaining

$$\begin{aligned} 1 - V'_{ab} &= 1 + x\frac{Y_{ca,db}}{Y_{uv}(0)} = \frac{Y_{uv}(0) + xY_{ca,db}}{Y_{uv}(0)} = \frac{Y_{db}(0) + xY_{ca,db}}{Y_{db}(0)} \\ &= \frac{Y_{db}(x)}{Y_{db}(0)} = \frac{Y_{uv}(x)}{Y_{uv}(0)} = F(x) \end{aligned} \tag{3.38}$$

in which we have invoked the identities $Y_{uv} = Y_{ij}$ and

$$Y_{db}(x) = Y_{db}(0) + xY_{ca,db} \tag{3.39}$$

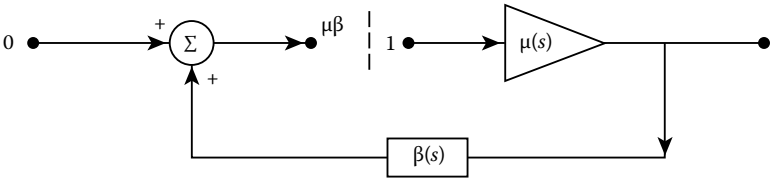


FIGURE 3.8 Physical interpretation of the loop transmission.

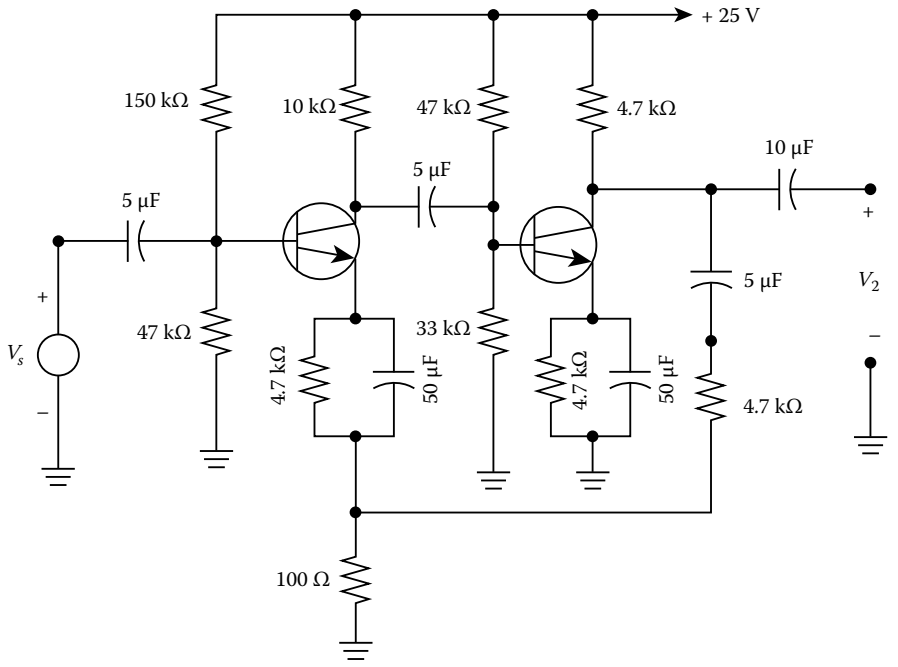


FIGURE 3.9 Voltage-series feedback amplifier together with its biasing and coupling circuitry.

We remark that we write $Y_{ca,db}(x)$ as $Y_{ca,db}$ because it is independent of x. In other words, the return difference $F(x)$ is simply the difference of the 1 V excitation and the returned voltage V'_{ab} as illustrated in Figure 3.7, and hence its name. Because

$$F(x) = 1 + T = 1 - \mu\beta \tag{3.40}$$

we conclude that the return difference has the same physical significance as the feedback factor of the ideal feedback model. The significance of the previous physical interpretations is that it permits us to determine the return ratio T or $-\mu\beta$ by measurement. Once the return ratio is measured, the other quantities such as return difference and loop transmission are known.

To illustrate, consider the voltage-series or the series-parallel feedback amplifier of Figure 3.9. Assume that the two transistors are identical with the following hybrid parameters:

$$h_{ie} = 1.1\ \text{k}\Omega, \quad h_{fe} = 50, \quad h_{re} = h_{oe} = 0 \tag{3.41}$$

After the biasing and coupling circuitry have been removed, the equivalent network is presented in Figure 3.10. The effective load of the first transistor is composed of the parallel combination of the 10, 33, 47, and 1.1 kΩ resistors. The effect of the 150 and 47 kΩ resistors can be ignored; they are included in the equivalent network to show their insignificance in the computation.

To simplify our notation, let

$$\tilde{\alpha}_k = \alpha_k \times 10^{-4} = \frac{h_{fe}}{h_{ie}} = 455 \times 10^{-4}, \quad k = 1, 2 \tag{3.42}$$

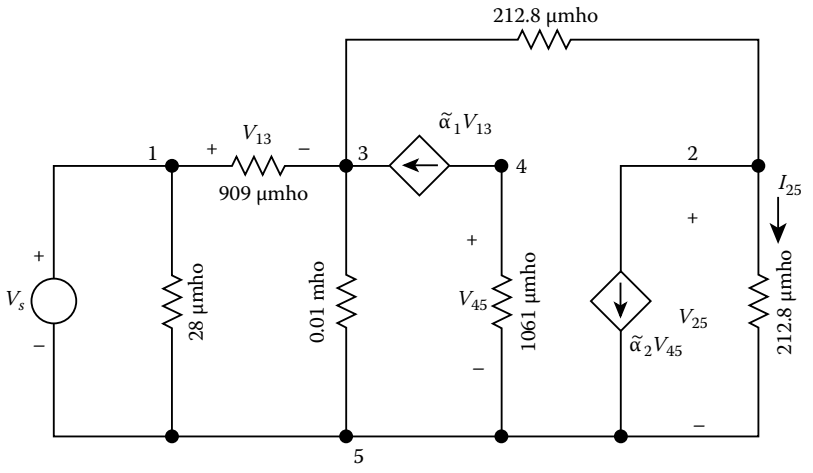


FIGURE 3.10 Equivalent network of the feedback amplifier of Figure 3.9.

The subscript k is used to distinguish the transconductances of the first and the second transistors. The indefinite-admittance matrix of the feedback amplifier of Figure 3.9 is

$$\mathbf{Y} = 10^{-4}\begin{bmatrix} 9.37 & 0 & -9.09 & 0 & -0.28 \\ 0 & 4.256 & -2.128 & \alpha_2 & -2.128-\alpha_2 \\ -9.09-\alpha_1 & -2.128 & 111.218+\alpha_1 & 0 & -100 \\ \alpha_1 & 0 & -\alpha_1 & 10.61 & -10.61 \\ -0.28 & -2.128 & -100 & -10.61-\alpha_1 & 113.018+\alpha_1 \end{bmatrix} \tag{3.43}$$

By applying Equation 3.27, the amplifier voltage gain is computed as

$$g_{12,25} = \frac{V_{25}}{V_s} = \frac{V_{12,25}}{V_{11,25}} = \frac{211.54 \times 10^{-7}}{4.66 \times 10^{-7}} = 45.39 \tag{3.44}$$

To calculate the return differences with respect to the transconductances $\tilde{\alpha}_k$ of the transistors, we short circuit the voltage source V_s. The resulting indefinite-admittance matrix is obtained from Equation 3.43 by adding the first row to the fifth row and the first column to the fifth column and then deleting the first row and column. Its first-order cofactor is simply $Y_{11,55}$. Thus, the return differences with respect to $\tilde{\alpha}_k$ are

$$F(\tilde{\alpha}_1) = \frac{Y_{11,55}(\tilde{\alpha}_1)}{Y_{11,55}(0)} = \frac{466.1 \times 10^{-9}}{4.97 \times 10^{-9}} = 93.70 \tag{3.45a}$$

$$F(\tilde{\alpha}_2) = \frac{Y_{11,55}(\tilde{\alpha}_2)}{Y_{11,55}(0)} = \frac{466.1 \times 10^{-9}}{25.52 \times 10^{-9}} = 18.26 \tag{3.45b}$$

3.4 Null Return Difference

In this section, we introduce the notion of null return difference, which is found to be very useful in measurement situations and in the computation of the sensitivity for the feedback amplifiers.

The null return difference $\hat{F}(x)$ of a feedback amplifier with respect to an element x is defined to be the ratio of the two functional values assumed by the second-order cofactor $Y_{rp,sq}$ of the elements of its

indefinite-admittance matrix **Y** under the condition that the element x assumes its nominal value and the condition that the element x assumes the value zero where r and s are input terminals, and p and q are the output terminals of the amplifier, or

$$\hat{F}(x) = \frac{Y_{rp,sq}(x)}{Y_{rp,sq}(0)} \tag{3.46}$$

Likewise, the null return ratio $\hat{T}$, with respect to a voltage-controlled current source $I = xV$, is the negative of the voltage appearing at the controlling branch when the controlled current source is replaced by an independent current source of xA and when the input excitation is adjusted so that the output of the amplifier is identically zero.

Now, we demonstrate that the null return difference is simply the return difference in the network under the situation that the input excitation I_s has been adjusted so that the output is identically zero. In the network of Figure 3.6, suppose that we replace the controlled current source by an independent current source of xA. Then by applying Equation 3.24 and the superposition principle, the output current I_{pq} at the load is

$$I_{pq} = Y_2\left[I_s\frac{Y_{rp,sq}(0)}{Y_{uv}(0)} + x\frac{Y_{dp,cq}(0)}{Y_{uv}(0)}\right] \tag{3.47}$$

Setting $I_{pq} = 0$ or $V_{pq} = 0$ yields

$$I_s \equiv I_0 = -x\left[\frac{Y_{dp,cq}(0)}{Y_{rp,sq}(0)}\right] \tag{3.48}$$

in which $Y_{dp,cq}$ is independent of x. This adjustment is possible only if a direct transmission occurs from the input to the output when x is set to zero. Thus, in the network of Figure 3.7, if we connect an independent current source of strength I_0 at its input port, the voltage V'_{ab} is the negative of the null return ratio $\hat{T}$. Using Equation 3.24, we obtain [3]

$$\begin{aligned}\hat{T} = -V'_{ab} &= -x\frac{Y_{da,cb}(0)}{Y_{uv}(0)} - I_0\frac{Y_{ra,sb}(0)}{Y_{uv}(0)}\\ &= -\frac{x\left[Y_{da,cb}(0)Y_{rp,sq}(0) - Y_{ra,sb}(0)Y_{dp,cq}(0)\right]}{Y_{uv}(0)Y_{rp,sq}(0)}\\ &= \frac{x\dot{Y}_{rp,sq}}{Y_{rp,sq}(0)} = \frac{Y_{rp,sq}(x)}{Y_{rp,sq}(0)} - 1\end{aligned} \tag{3.49}$$

where

$$\dot{Y}_{rp,sq} \equiv \frac{\mathrm{d}Y_{rp,sq}(x)}{\mathrm{d}x} \tag{3.50}$$

This leads to

$$\hat{F}(x) = 1 + \hat{T} = 1 - V'_{ab} \tag{3.51}$$

which demonstrates that the null return difference $\hat{F}(x)$ is simply the difference of the 1 V excitation applied to the right of the breaking mark of the broken controlling branch of the controlled source and

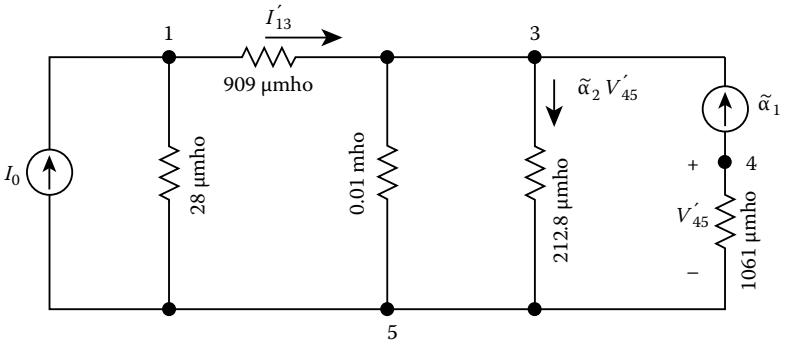


FIGURE 3.11 Network used to compute the null return difference $\hat{F}(\tilde{\alpha}_1)$ by its physical interpretation.

the returned voltage V'_{ab} appearing at the left of the breaking mark under the situation that the input signal I_s is adjusted so that the output is identically zero.

As an illustration, consider the voltage-series feedback amplifier of Figure 3.9, an equivalent network of which is presented in Figure 3.10. Using the indefinite-admittance matrix of Equation 3.43 in conjunction with Equation 3.42, the null return differences with respect to $\tilde{\alpha}_k$ are

$$\hat{F}(\tilde{\alpha}_1) = \frac{Y_{12,55}(\tilde{\alpha}_1)}{Y_{12,55}(0)} = \frac{211.54 \times 10^{-7}}{205.24 \times 10^{-12}} = 103.07 \times 10^3 \tag{3.52a}$$

$$\hat{F}(\tilde{\alpha}_2) = \frac{Y_{12,55}(\tilde{\alpha}_2)}{Y_{12,55}(0)} = \frac{211.54 \times 10^{-7}}{104.79 \times 10^{-10}} = 2018.70 \tag{3.52b}$$

Alternatively, $\hat{F}(\tilde{\alpha}_1)$ can be computed by using its physical interpretation as follows. Replace the controlled source $\tilde{\alpha}_1 V_{13}$ in Figure 3.10 by an independent current source of $\tilde{\alpha}_1$ A. We then adjust the voltage source V_s so that the output current I_{25} is identically zero. Let I_0 be the input current resulting from this source. The corresponding network is presented in Figure 3.11. From this network, we obtain

$$\hat{F}(\tilde{\alpha}_1) = 1 + \hat{T} = 1 - V'_{13} = 1 - \frac{100V'_{35} + \alpha_2 V'_{45} - \alpha_1}{9.09} = 103.07 \times 10^3 \tag{3.53}$$

Likewise, we can use the same procedure to compute the return difference $\hat{F}(\tilde{\alpha}_2)$.

References

1. H. W. Bode, *Network Analysis and Feedback Amplifier Design*, Princeton, NJ: Van Nostrand, 1945.
2. W.-K. Chen, Indefinite-admittance matrix formulation of feedback amplifier theory, *IEEE Trans. Circuits Syst.*, CAS-23, 498–505, 1976.
3. W.-K. Chen, On second-order cofactors and null return difference in feedback amplifier theory, *Int. J. Circuit Theory Appl.*, 6, 305–312, 1978.

4

Network Functions and Feedback

Wai-Kai Chen
University of Illinois at Chicago

4.1 Blackman's Formula **4**-1
4.2 Sensitivity Function **4**-6

We now study the effects of feedback on amplifier impedance and gain and obtain some useful relations among the return difference, the null return difference, and impedance functions in general.

Refer to the general feedback configuration of Figure 3.6. Let w be a transfer function. As before, to emphasize the importance of the feedback element x, we write $w = w(x)$. To be definite, let $w(x)$ for the time being be the current gain between the output and input ports. Then, from Equation 3.24 we obtain

$$w(x) = \frac{I_{pq}}{I_s} = \frac{Y_2 V_{pq}}{I_s} = \frac{Y_{rp,sq}(x)}{Y_{uv}(x)} Y_2 \tag{4.1}$$

yielding

$$\frac{w(x)}{w(0)} = \frac{Y_{rp,sq}(x)}{Y_{uv}(x)} \frac{Y_{uv}(0)}{Y_{rp,sq}(0)} = \frac{\hat{F}(x)}{F(x)} \tag{4.2}$$

provided that $w(0) \neq 0$. This gives a very useful formula for computing the current gain:

$$w(x) = w(0) \frac{\hat{F}(x)}{F(x)} \tag{4.3}$$

Equation 4.3 remains valid if $w(x)$ represents the transfer impedance $z_{rp,sq} = V_{pq}/I_s$ instead of the current gain.

4.1 Blackman's Formula

In particular, when $r = p$ and $s = q$, $w(x)$ represents the driving-point impedance $z_{rr,ss}(x)$ looking into the terminals r and s, and we have a somewhat different interpretation. In this case, $F(x)$ is the return difference with respect to the element x under the condition $I_s = 0$. Thus, $F(x)$ is the return difference for the situation when the port where the input impedance is defined is left open without a source and we write $F(x) = F$(input open circuited). Likewise, from Figure 3.6, $\hat{F}(x)$ is the return difference with respect to x for the input excitation I_s and output response V_{rs} under the condition I_s is adjusted so that V_{rs} is identically zero. Thus, $\hat{F}(x)$ is the return difference for the situation when the port where the input

impedance is defined is short circuited, and we write $\hat{F}(x) = F$ (input short circuited). Consequently, the input impedance $Z(x)$ looking into a terminal pair can be conveniently expressed as

$$Z(x) = Z(0)\frac{F(\text{input short circuited})}{F(\text{input open circuited})} \tag{4.4}$$

This is the well-known Blackman's formula for computing an active impedance. The formula is extremely useful because the right-hand side can usually be determined rather easily. If x represents the controlling parameter of a controlled source in a single-loop feedback amplifier, then setting $x = 0$ opens the feedback loop and $Z(0)$ is simply a passive impedance. The return difference for x when the input port is short circuited or open circuited is relatively simple to compute because shorting out or opening a terminal pair frequently breaks the feedback loop. In addition, Blackman's formula can be used to determine the return difference by measurements. Because it involves two return differences, only one of them can be identified and the other must be known in advance. In the case of a single-loop feedback amplifier, it is usually possible to choose a terminal pair so that either the numerator or the denominator on the right-hand side of Equation 4.4 is unity. If F(input short circuited) = 1, F(input open circuited) becomes the return difference under normal operating condition and we have

$$F(x) = \frac{Z(0)}{Z(x)} \tag{4.5}$$

On the other hand, if F(input open circuited) = 1, F(input short circuited) becomes the return difference under normal operating condition and we obtain

$$F(x) = \frac{Z(x)}{Z(0)} \tag{4.6}$$

Example 4.1

The network of Figure 4.1 is a general active RC one-port realization of a rational impedance. We use Blackman's formula to verify that its input admittance is given by

$$Y = 1 + \frac{Z_3 - Z_4}{Z_1 - Z_2} \tag{4.7}$$

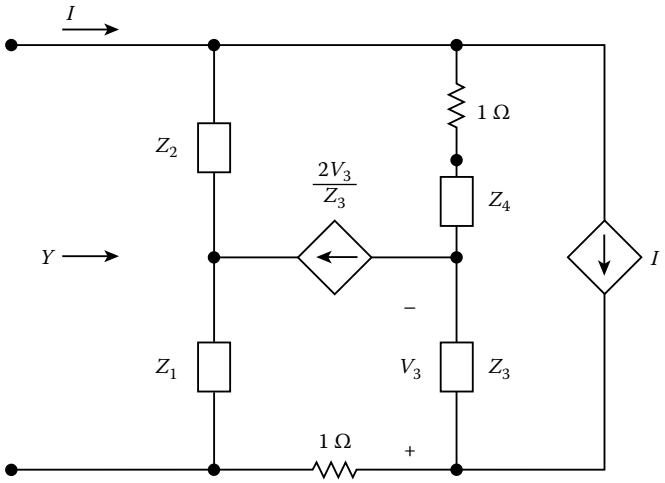


FIGURE 4.1 General active RC one-port realization of a rational function.

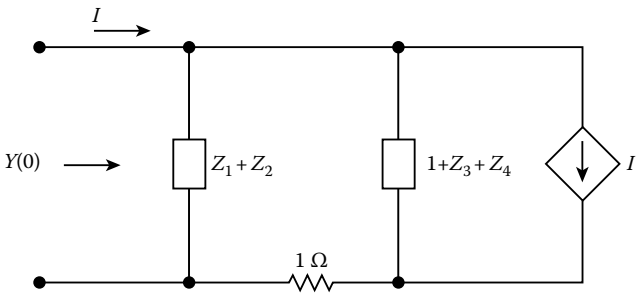


FIGURE 4.2 Network used to compute $Y(0)$.

Appealing to Equation 4.4, the input admittance written as $Y = Y(x)$ can be written as

$$Y(x) = Y(0)\frac{F(\text{input open circuited})}{F(\text{input short circuited})} \tag{4.8}$$

where $x = 2/Z_3$. By setting x to zero, the network used to compute $Y(0)$ is shown in Figure 4.2. Its input admittance is

$$Y(0) = \frac{Z_1 + Z_2 + Z_3 + Z_4 + 2}{Z_1 + Z_2} \tag{4.9}$$

When the input port is open circuited, the network of Figure 4.1 degenerates to that depicted in Figure 4.3. The return difference with respect to x is

$$F(\text{input open circuited}) = 1 - V_3' = \frac{Z_1 + Z_3 - Z_2 - Z_4}{2 + Z_1 + Z_2 + Z_3 + Z_4} \tag{4.10}$$

where the returned voltage V_3' at the controlling branch is given by

$$V_3' = \frac{2(1 + Z_2 + Z_4)}{2 + Z_1 + Z_2 + Z_3 + Z_4} \tag{4.11}$$

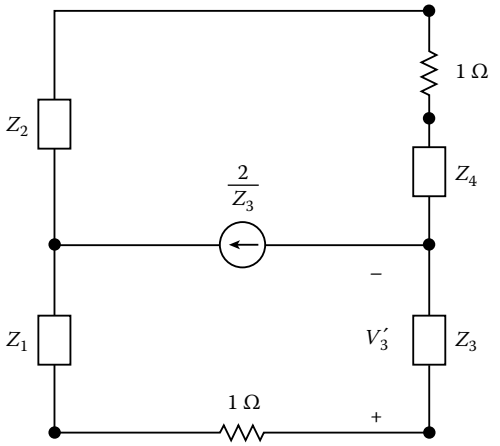


FIGURE 4.3 Network used to compute F(input open circuited).

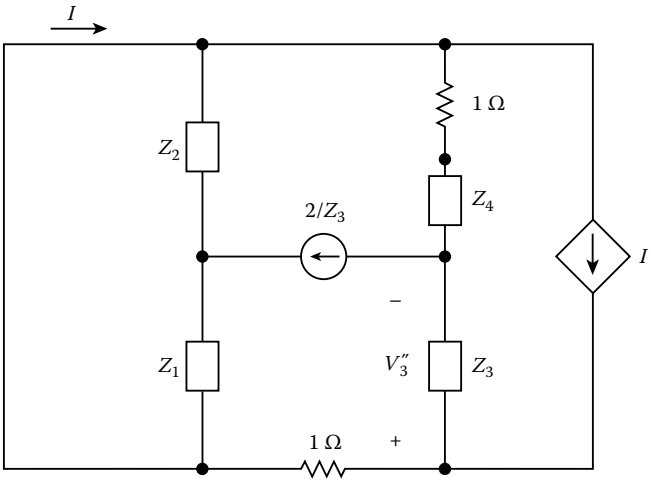


FIGURE 4.4 Network used to compute F(input short circuited).

To compute the return difference when the input port is short circuited, we use the network of Figure 4.4 and obtain

$$F(\text{input short circuited}) = 1 - V_3'' = \frac{Z_1 - Z_2}{Z_1 + Z_2} \tag{4.12}$$

where the return voltage V_3'' at the controlling branch is found to be

$$V_3'' = \frac{2Z_2}{Z_1 + Z_2} \tag{4.13}$$

Substituting Equations 4.9, 4.10, and 4.12 in Equation 4.8 yields the desired result.

$$Y = 1 + \frac{Z_3 - Z_4}{Z_1 - Z_2} \tag{4.14}$$

To determine the effect of feedback on the input and output impedances, we choose the series-parallel feedback configuration of Figure 4.5. By shorting the terminals of Y_2, we interrupt the feedback loop,

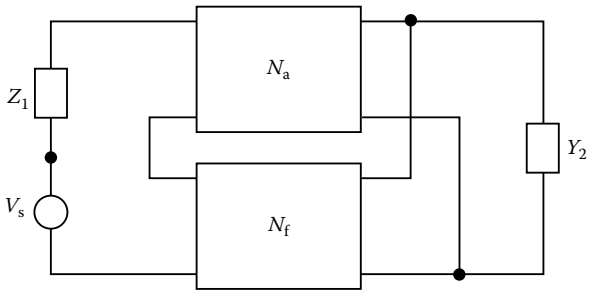


FIGURE 4.5 Series-parallel feedback configuration.

therefore, formula (Equation 4.5) applies and the output impedance across the load admittance Y_2 becomes

$$Z_{\text{out}}(x) = \frac{Z_{\text{out}}(0)}{F(x)} \tag{4.15}$$

demonstrating that the impedance measured across the path of the feedback is reduced by the factor that is the normal value of the return difference with respect to the element x, where x is an arbitrary element of interest. For the input impedance of the amplifier looking into the voltage source V_s of Figure 4.5, by open circuiting or removing the voltage source V_s, we break the feedback loop. Thus, formula (Equation 4.6) applies and the input impedance becomes

$$Z_{\text{in}}(x) = F(x)Z_{\text{in}}(0) \tag{4.16}$$

meaning that the impedance measured in series lines is increased by the same factor $F(x)$. Similar conclusions can be reached for other types of configurations discussed in Chapter 2 by applying Blackman's formula.

Again, refer to the general feedback configuration of Figure 3.6. If $w(x)$ represents the voltage gain V_{pq}/V_{rs} or the transfer admittance I_{pq}/V_{rs}. Then, from Equation 4.27 we can write

$$\frac{w(x)}{w(0)} = \frac{Y_{rp,sq}(x)}{Y_{rp,sq}(0)} \frac{Y_{rr,ss}(0)}{Y_{rr,ss}(x)} \tag{4.17}$$

The first term in the product on the right-hand side is the null return difference $\hat{F}(x)$ with respect to x for the input terminals r and s and output terminals p and q. The second term is the reciprocal of the null return difference with respect to x for the same input and output port at terminals r and s. This reciprocal can then be interpreted as the return difference with respect to x when the input port of the amplifier is short circuited. Thus, the voltage gain or the transfer admittance can be expressed as

$$w(x) = w(0)\frac{\hat{F}(x)}{F(\text{input short circuited})} \tag{4.18}$$

Finally, if $w(x)$ denotes the short circuit current gain I_{pq}/I_s as Y_2 approaches infinity, we obtain

$$\frac{w(x)}{w(0)} = \frac{Y_{rp,sq}(x)}{Y_{rp,sq}(0)} \frac{Y_{pp,qq}(0)}{Y_{pp,qq}(x)} \tag{4.19}$$

The second term in the product on the right-hand side is the reciprocal of the return difference with respect to x when the output port of the amplifier is short circuited, giving a formula for the short circuit current gain as

$$w(x) = w(0)\frac{\hat{F}(x)}{F(\text{output short circuited})} \tag{4.20}$$

Again, consider the voltage-series or series-parallel feedback amplifier of Figure 3.9 an equivalent network of which is given in Figure 3.10. The return differences $F(\tilde{\alpha}_k)$, the null return differences $\hat{F}(\tilde{\alpha}_k)$ and the voltage gain w were computed earlier in Equations 3.45, 3.52, and 3.44, and are repeated next:

$$F(\tilde{\alpha}_1) = 93.70, \quad F(\tilde{\alpha}_2) = 18.26 \tag{4.21a}$$

$$\hat{F}(\tilde{\alpha}_1) = 103.07 \times 10^3, \quad \hat{F}(\tilde{\alpha}_2) = 2018.70 \tag{4.21b}$$

$$w = \frac{V_{25}}{V_s} = w(\tilde{\alpha}_1) = w(\tilde{\alpha}_2) = 45.39 \tag{4.21c}$$

We apply Equation 4.18 to calculate the voltage gain w, as follows:

$$w(\tilde{\alpha}_1) = w(0)\frac{\hat{F}(\tilde{\alpha}_1)}{F(\text{input short circuited})} = 0.04126\frac{103.07 \times 10^3}{93.699} = 45.39 \tag{4.22}$$

where

$$w(0) = \left.\frac{Y_{12,55}(\tilde{\alpha}_1)}{Y_{11,55}(\tilde{\alpha}_1)}\right|_{\tilde{\alpha}_1=0} = \frac{205.24 \times 10^{-12}}{497.41 \times 10^{-11}} = 0.04126 \tag{4.23a}$$

$$F(\text{input short circuited}) = \frac{Y_{11,55}(\tilde{\alpha}_1)}{Y_{11,55}(0)} = \frac{466.07 \times 10^{-9}}{4.9741 \times 10^{-9}} = 93.699 \tag{4.23b}$$

and

$$w(\tilde{\alpha}_2) = w(0)\frac{\hat{F}(\tilde{\alpha}_2)}{F(\text{input short circuited})} = 0.41058\frac{2018.70}{18.26} = 45.39 \tag{4.24}$$

where

$$w(0) = \frac{Y_{12,55}(\tilde{\alpha}_2)}{Y_{11,55}(\tilde{\alpha}_2)}|_{\tilde{\alpha}_2=0} = \frac{104.79 \times 10^{-10}}{255.22 \times 10^{-10}} = 0.41058 \tag{4.25a}$$

$$F(\text{input short circuited}) = \frac{Y_{11,55}(\tilde{\alpha}_2)}{Y_{11,55}(0)} = \frac{466.07 \times 10^{-9}}{25.52 \times 10^{-9}} = 18.26 \tag{4.25b}$$

4.2 Sensitivity Function

One of the most important effects of negative feedback is its ability to make an amplifier less sensitive to the variations of its parameters because of aging, temperature variations, or other environment changes. A useful quantitative measure for the degree of dependence of an amplifier on a particular parameter is known as the sensitivity. The sensitivity function, written as $\mathcal{S}(x)$, for a given transfer function with respect to an element x is defined as the ratio of the fractional change in a transfer function to the fractional change in x for the situation when all changes concerned are differentially small. Thus, if $w(x)$ is the transfer function, the sensitivity function can be written as

$$\mathcal{S}(x) = \lim_{\Delta x \to 0} \frac{\Delta w/w}{\Delta x/x} = \frac{x}{w}\frac{\partial w}{\partial x} = x\frac{\partial \ln w}{\partial x} \tag{4.26}$$

Refer to the general feedback configuration of Figure 3.6, and let $w(x)$ represent either the current gain I_{pq}/I_s or the transfer impedance V_{pq}/I_s for the time being. Then, we obtain from Equation 3.24

$$w(x) = Y_2\frac{Y_{rp,sq}(x)}{Y_{uv}(x)} \text{ or } \frac{Y_{rp,sq}(x)}{Y_{uv}(x)} \tag{4.27}$$

As before, we write

$$\dot{Y}_{uv}(x) = \frac{\partial Y_{uv}(x)}{\partial x} \tag{4.28a}$$

$$\dot{Y}_{rp,sq}(x) = \frac{\partial Y_{rp,sq}(x)}{\partial x} \tag{4.28b}$$

obtaining

$$Y_{uv}(x) = Y_{uv}(0) + x\dot{Y}_{uv}(x) \tag{4.29a}$$

$$Y_{rp,sq}(x) = Y_{rp,sq}(0) + x\dot{Y}_{rp,sq}(x) \tag{4.29b}$$

Substituting Equation 4.27 in Equation 4.26, in conjunction with Equation 4.29, yields

$$\begin{aligned}\mathcal{S}(x) &= x\frac{\dot{Y}_{rp,sq}(x)}{Y_{rp,sq}(x)} - x\frac{\dot{Y}_{uv}(x)}{Y_{uv}(x)} = \frac{Y_{rp,sq}(x) - Y_{rp,sq}(0)}{Y_{rp,sq}(x)} - \frac{Y_{uv}(x) - Y_{uv}(0)}{Y_{uv}(x)} \\ &= \frac{Y_{uv}(0)}{Y_{uv}(x)} - \frac{Y_{rp,sq}(0)}{Y_{rp,sq}(x)} = \frac{1}{F(x)} - \frac{1}{\hat{F}(x)}\end{aligned} \tag{4.30}$$

Combining this with Equation 4.3, we obtain

$$\mathcal{S}(x) = \frac{1}{F(x)}\left[1 - \frac{w(0)}{w(x)}\right] \tag{4.31}$$

Observe that if $w(0) = 0$, Equation 4.31 becomes

$$\mathcal{S}(x) = \frac{1}{F(x)} \tag{4.32}$$

meaning that sensitivity is equal to the reciprocal of the return difference. For the ideal feedback model, the feedback path is unilateral. Hence, $w(0) = 0$ and

$$\mathcal{S} = \frac{1}{F} = \frac{1}{1+T} = \frac{1}{1-\mu\beta} \tag{4.33}$$

For a practical amplifier, $w(0)$ is usually very much smaller than $w(x)$ in the passband, and $F \approx 1/\mathcal{S}$ may be used as a good estimate of the reciprocal of the sensitivity in the same frequency band. A single-loop feedback amplifier composed of a cascade of common-emitter stages with a passive network providing the desired feedback fulfills this requirements. If in such a structure any one of the transistors fails, the forward transmission is nearly zero and $w(0)$ is practically zero. Our conclusion is that if the failure of any element will interrupt the transmission through the amplifier as a whole to nearly zero, the sensitivity is approximately equal to the reciprocal of the return difference with respect to that element. In the case of driving-point impedance, $w(0)$ is not usually smaller than $w(x)$, and the reciprocity relation is not generally valid.

Now assume that $w(x)$ represents the voltage gain. Substituting Equation 4.27 in Equation 4.26 results in

$$\begin{aligned}\mathcal{S}(x) &= x\frac{\dot{Y}_{rp,sq}(x)}{Y_{rp,sq}(x)} - x\frac{\dot{Y}_{rr,ss}(x)}{Y_{rr,ss}(x)} = \frac{Y_{rp,sq}(x) - Y_{rp,sq}(0)}{Y_{rp,sq}(x)} - \frac{Y_{rr,ss}(x) - Y_{rr,ss}(0)}{Y_{rr,ss}(x)} \\ &= \frac{Y_{rr,ss}(0)}{Y_{rr,ss}(x)} - \frac{Y_{rp,sq}(0)}{Y_{rp,sq}(x)} = \frac{1}{F(\text{input short circuited})} - \frac{1}{\hat{F}(x)}\end{aligned} \tag{4.34}$$

Combining this with Equation 4.18 gives

$$\mathcal{S}(x) = \frac{1}{F(\text{input short circuited})}\left[1 - \frac{w(0)}{w(x)}\right] \tag{4.35}$$

Finally, if $w(x)$ denotes the short circuit current gain I_{pq}/I_s as Y_2 approaches infinity, the sensitivity function can be written as

$$\mathcal{S}(x) = \frac{Y_{pp,qq}(0)}{Y_{pp,qq}(x)} - \frac{Y_{rp,sq}(0)}{Y_{rp,sq}(x)} = \frac{1}{F(\text{output short circuited})} - \frac{1}{\hat{F}(x)} \tag{4.36}$$

which when combined with Equation 4.20 yields

$$\mathcal{S}(x) = \frac{1}{F(\text{output short circuited})}\left[1 - \frac{w(0)}{w(x)}\right] \tag{4.37}$$

We remark that Equations 4.31, 4.35, and 4.39 are quite similar. If the return difference $F(x)$ is interpreted properly, they can all be represented by the single relation Equation 4.31. As before, if $w(0) = 0$, the sensitivity for the voltage gain function is equal to the reciprocal of the return difference under the situation that the input port of the amplifier is short circuited, whereas the sensitivity for the short circuit current gain is the reciprocal of the return difference when the output port is short circuited.

Example 4.2

The network of Figure 4.6 is a common-emitter transistor amplifier. After removing the biasing circuit and using the common-emitter hybrid model for the transistor at low frequencies, an equivalent network of the amplifier is presented in Figure 4.7 with

$$I_s' = \frac{V_s}{R_1 + r_x} \tag{4.38a}$$

$$G_1' = \frac{1}{R_1'} = \frac{1}{R_1 + r_x} + \frac{1}{r_\pi} \tag{4.38b}$$

$$G_2' = \frac{1}{R_2'} = \frac{1}{R_2} + \frac{1}{R_c} \tag{4.38c}$$

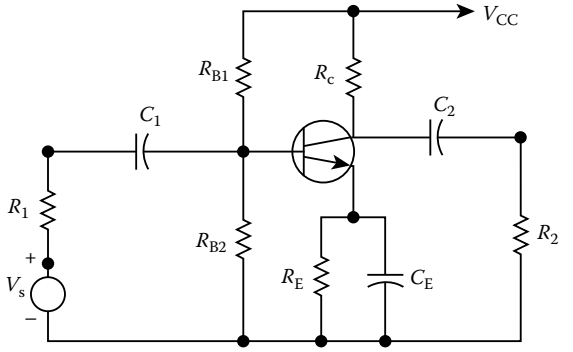


FIGURE 4.6 Common-emitter transistor feedback amplifier.

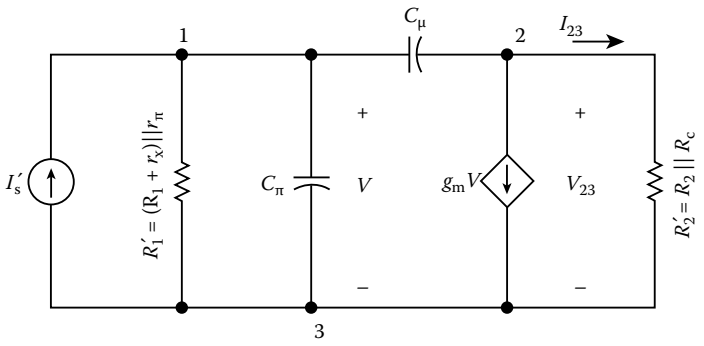


FIGURE 4.7 Equivalent network of the feedback amplifier of Figure 4.6.

The indefinite admittance matrix of the amplifier is

$$\mathbf{Y} = \begin{bmatrix} G'_1 + sC_\pi + sC_\mu & -sC_\mu & -G'_1 - sC_\pi \\ g_m - sC_\mu & G'_2 + sC_\mu & -G'_2 - g_m \\ -G'_1 - sC_\pi - g_m & -G'_2 & G'_1 + G'_2 + sC_\pi + g_m \end{bmatrix} \tag{4.39}$$

Assume that the controlling parameter g_m is the element of interest. The return difference and the null return difference with respect to g_m in Figure 4.7 with I'_s as the input port and R'_2, as the output port, are

$$F(g_m) = \frac{Y_{33}(g_m)}{Y_{33}(0)} = \frac{(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu(G'_2 + g_m)}{(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu G'_2} \tag{4.40}$$

$$\hat{F}(g_m) = \frac{Y_{12,33}(g_m)}{Y_{12,33}(0)} = \frac{sC_\mu - g_m}{sC_\mu} = 1 - \frac{g_m}{sC_\mu} \tag{4.41}$$

The current gain I_{23}/I'_s as defined in Figure 4.7, is computed as

$$w(g_m) = \frac{Y_{12,33}(g_m)}{R'_2 Y_{33}(g_m)} = \frac{sC_\mu - g_m}{R'_2[(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu(G'_2 + g_m)]} \tag{4.42}$$

Substituting these in Equations 4.30 or 4.31 gives

$$\mathcal{S}(g_m) = -\frac{g_m(G'_1 + sC_\pi + sC_\mu)(G'_2 + sC_\mu)}{(sC_\mu - g_m)[(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu(G'_2 + g_m)]} \tag{4.43}$$

Finally, we compute the sensitivity for the driving-point impedance facing the current source I'_s. From Equation 4.31, we obtain

$$\mathcal{S}(g_m) = \frac{1}{F(g_m)}\left[1 - \frac{Z(0)}{Z(g_m)}\right] = -\frac{sC_\mu g_m}{(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu(G'_2 + g_m)} \tag{4.44}$$

where

$$Z(g_m) = \frac{Y_{11,33}(g_m)}{Y_{33}(g_m)} = \frac{G'_2 + sC_\mu}{(G'_1 + sC_\pi)(G'_2 + sC_\mu) + sC_\mu(G'_2 + g_m)} \tag{4.45}$$

5

Measurement of Return Difference

5.1 Blecher's Procedure **5**-2
5.2 Impedance Measurements **5**-4
References **5**-7

Wai-Kai Chen
University of Illinois at Chicago

The zeros of the network determinant are called the natural frequencies. Their locations in the complex-frequency plane are extremely important in that they determine the stability of the network. A network is said to be stable if all of its natural frequencies are restricted to the open left-half side of the complex-frequency plane. If a network determinant is known, its roots can readily be computed explicitly with the aid of a computer if necessary, and the stability problem can then be settled directly. However, for a physical network there remains the difficulty of getting an accurate formulation of the network determinant itself, because every equivalent network is, to a greater or lesser extent, an idealization of the physical reality. As frequency is increased, parasitic effects of the physical elements must be taken into account. What is really needed is some kind of experimental verification that the network is stable and will remain so under certain prescribed conditions. The measurement of the return difference provides an elegant solution to this problem.

The return difference with respect to an element x in a feedback amplifier is defined by

$$F(x) = \frac{Y_{uv}(x)}{Y_{uv}(0)} \tag{5.1}$$

Because $Y_{uv}(x)$ denotes the nodal determinant, the zeros of the return difference are exactly the same as the zeros of the nodal determinant provided that there is no cancellation of common factors between $Y_{uv}(x)$ and $Y_{uv}(0)$. Therefore, if $Y_{uv}(0)$ is known to have no zeros in the closed right-half side of the complex-frequency plane, which is usually the case in a single-loop feedback amplifier when x is set to zero, $F(x)$ gives precisely the same information about the stability of a feedback amplifier as does the nodal determinant itself. The difficulty inherent in the measurement of the return difference with respect to the controlling parameter of a controlled source is that, in a physical system, the controlling branch and the controlled source both form part of a single device such as a transistor, and cannot be physically separated. In the following, we present a scheme that does not require the physical decomposition of a device.

Let a device of interest be brought out as a two-port network connected to a general four-port network as shown in Figure 5.1. For our purposes, assume that this device is characterized by its y parameters, and represented by its y-parameter equivalent two-port network as indicated in Figure 5.2, in which the parameter y_{21} controls signal transmission in the forward direction through the device, whereas y_{12} gives

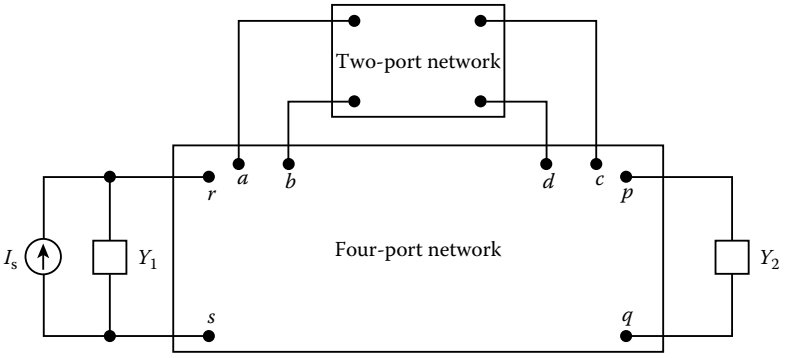


FIGURE 5.1 The general configuration of a feedback amplifier with a two-port device.

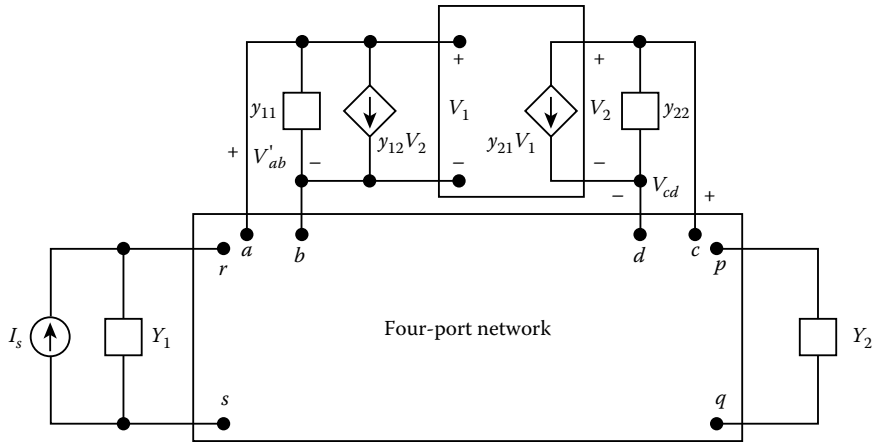


FIGURE 5.2 The representation of a two-port device in Figure 5.1 by its *y* parameters.

the reverse transmission, accounting for the internal feedback within the device. Our objective is to measure the return difference with respect to the forward short circuit transfer admittance y_{21}.

5.1 Blecher's Procedure [1]

Let the two-port device be a transistor operated in the common-emitter configuration with terminals a, $b = d$, and c representing, respectively, the base, emitter, and collector terminals. To simplify our notation, let $a = 1$, $b = d = 3$, and $c = 2$, as exhibited explicitly in Figure 5.3.

To measure $F(y_{21})$, we break the base terminal of the transistor and apply a 1 V excitation at its input as exhibited in Figure 5.3. To ensure that the controlled current source $y_{21}V_{13}$ drives a replica of what it sees during normal operation, we connect an active one-port network composed of a parallel combination of the admittance y_{11} and a controlled current source $y_{12}V_{23}$ at terminals 1 and 3. The returned voltage V_{13} is precisely the negative of the return ratio with respect to the element y_{21}. If, in the frequency band of interest, the externally applied feedback is large compared with the internal feedback of the transistor, the controlled source $y_{12}V_{23}$ can be ignored. If, however, we find that this internal feedback cannot be ignored, we can simulate it by using an additional transistor, connected as shown in Figure 5.4. This additional transistor must be matched as closely as possible to the one in question. The one-port

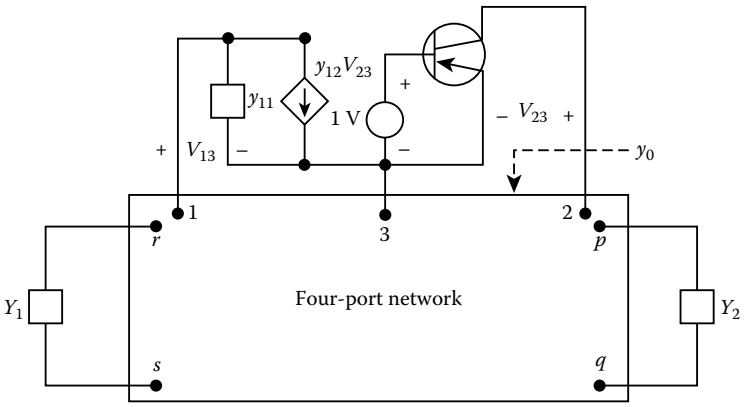


FIGURE 5.3 A physical interpretation of the return difference $F(y_{21})$ for a transistor operated in the common-emitter configuration and represented by its y parameters y_{ij}.

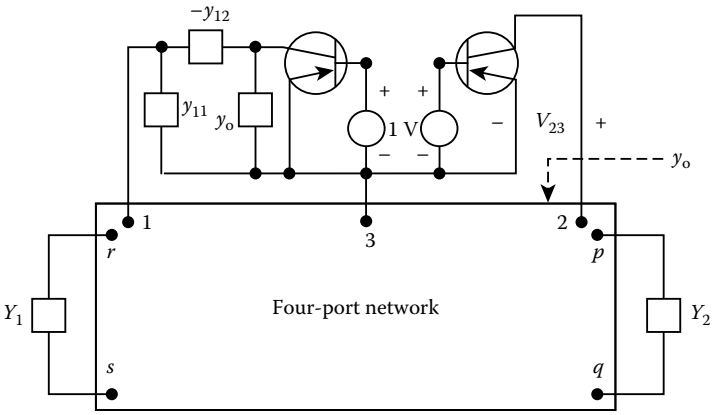


FIGURE 5.4 The measurement of return difference $F(y_{21})$ for a transistor operated in the common-emitter configuration and represented by its y parameters y_{ij}.

admittance y_o denotes the admittance presented to the output port of the transistor under consideration as indicated in Figures 5.3 and 5.4. For a common-emitter state, it is perfectly reasonable to assume that $|y_0| \gg |y_{12}|$ and $|y_{11}| \gg |y_{12}|$. Under these assumptions, it is straightforward to show that the Norton equivalent network looking into the two-port network at terminals 1 and 3 of Figure 5.4 can be approximated by the parallel combination of y_{11} and $y_{12}V_{23}$, as indicated in Figure 5.3. In Figure 5.4, if the voltage sources have very low internal impedances, we can join together the two base terminals of the transistors and feed them both from a single voltage source of very low internal impedance. In this way, we avoid the need of using two separate sources. For the procedure to be feasible, we must demonstrate the admittances y_{11} and $-y_{12}$ can be realized as the input admittances of one-port RC networks.

Consider the hybrid-pi equivalent network of a common-emitter transistor of Figure 5.5, the short circuit admittance matrix of which is found to be

$$\mathbf{Y}_{sc} = \frac{1}{g_x + g_\pi + sC_\pi + sC_\mu} \begin{bmatrix} g_x(g_\pi + sC_\pi + sC_\mu) & -g_x sC_\mu \\ g_x(g_m - sC_\mu) & sC_\mu(g_x + g_\pi + sC_\pi + g_m) \end{bmatrix} \tag{5.2}$$

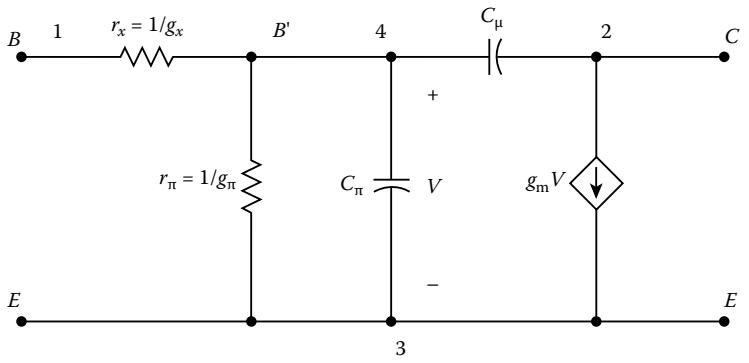


FIGURE 5.5 The hybrid-pi equivalent network of a common-emitter transistor.

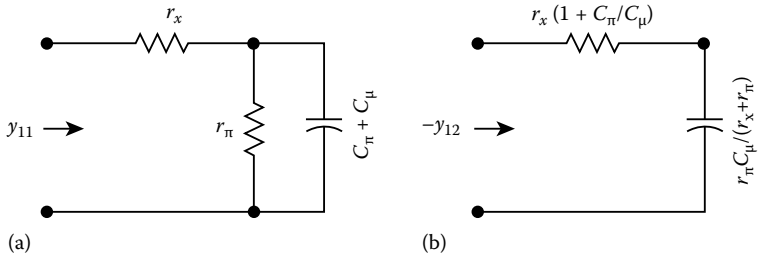


FIGURE 5.6 (a) The realization of y_{11} and (b) the realization of $-y_{12}$.

It is easy to confirm that the admittance y_{11} and $-y_{12}$ can be realized by the one-port networks of Figure 5.6.

5.2 Impedance Measurements

In this section, we show that the return difference can be evaluated by measuring two driving-point impedances at a convenient port in the feedback amplifier [2].

Refer again to the general feedback configuration of Figure 5.2. Suppose that we wish to evaluate the return difference with respect to the forward short circuit transfer admittance y_{21}. The controlling parameters y_{12} and y_{21} enter the indefinite-admittance matrix $\mathbf{Y}$ in the rectangular patterns as shown next:

$$\mathbf{Y}(x) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} & & y_{12} & -y_{12} \\ & & -y_{12} & y_{12} \\ y_{21} & -y_{21} & & \\ -y_{21} & y_{21} & & \end{bmatrix} \end{array} \tag{5.3}$$

To emphasize the importance of y_{12} and y_{21}, we again write $Y_{uv}(x)$ as $Y_{uv}(y_{12}, y_{21})$ and $z_{aa,bb}(x)$ as $z_{aa,bb}(y_{12}, y_{21})$. By appealing to Equation 3.25, the impedance looking into terminals a and b of Figure 5.2 is

$$z_{aa,bb}(y_{12}, y_{21}) = \frac{Y_{aa,bb}(y_{12}, y_{21})}{Y_{dd}(y_{12}, y_{21})} \tag{5.4}$$

The return difference with respect to y_{21} is given by

$$F(y_{21}) = \frac{Y_{dd}(y_{12}, y_{21})}{Y_{dd}(y_{12}, 0)} \tag{5.5}$$

Combining these yields

$$\begin{aligned} F(y_{21})z_{aa,bb}(y_{12}, y_{21}) &= \frac{Y_{aa,bb}(y_{12}, y_{21})}{Y_{dd}(y_{12}, 0)} = \frac{Y_{aa,bb}(0, 0)}{Y_{dd}(y_{12}, 0)} \\ &= \frac{Y_{aa,bb}(0, 0)}{Y_{dd}(0, 0)} \frac{Y_{dd}(0, 0)}{Y_{dd}(y_{12}, 0)} = \frac{z_{aa,bb}(0, 0)}{F(y_{12})|_{y_{21}=0}} \end{aligned} \tag{5.6}$$

obtaining a relation

$$F(y_{12})|_{y_{21}=0} F(y_{21}) = \frac{z_{aa,bb}(0, 0)}{z_{aa,bb}(y_{12}, y_{21})} \tag{5.7}$$

among the return differences and the driving-point impedances. $F(y_{12})|_{y21\,=\,0}$ is the return difference with respect to y_{12} when y_{21} is set to zero. This quantity can be measured by the arrangement of Figure 5.7. $z_{aa,bb}(y_{12}, y_{21})$ is the driving-point impedance looking into terminals a and b of the network of Figure 5.2. Finally, $z_{aa,bb}(0, 0)$ is the impedance to which $z_{aa,bb}(y_{12}, y_{21})$ reduces when the controlling parameters y_{12} and y_{21} are both set to zero. This impedance can be measured by the arrangement of Figure 5.8. Note that, in all three measurements, the independent current source I_s is removed.

Suppose that we wish to measure the return difference $F(y_{21})$ with respect to the forward transfer admittance y_{21} of a common-emitter transistor shown in Figure 5.2. Then, the return difference $F(y_{12})$ when y_{21} is set to zero, for all practical purposes, is indistinguishable from unity. Therefore, Equation 5.7 reduces to the following simpler form:

$$F(y_{21}) \approx \frac{z_{11,\,33}(0, 0)}{z_{11,\,33}(y_{12}, y_{21})} \tag{5.8}$$

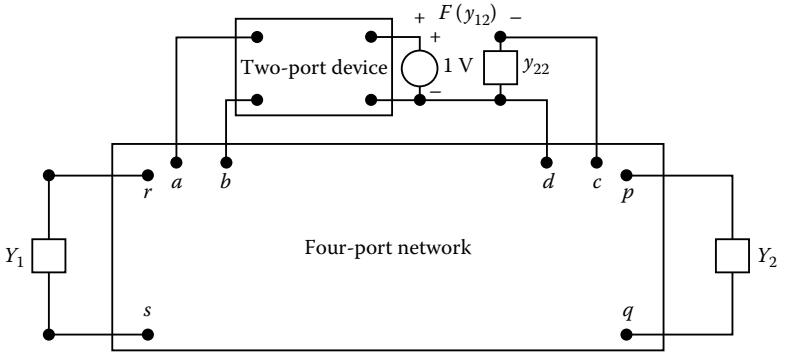


FIGURE 5.7 The measurement of the return difference $F(y_{12})$ with y_{21} set to zero.

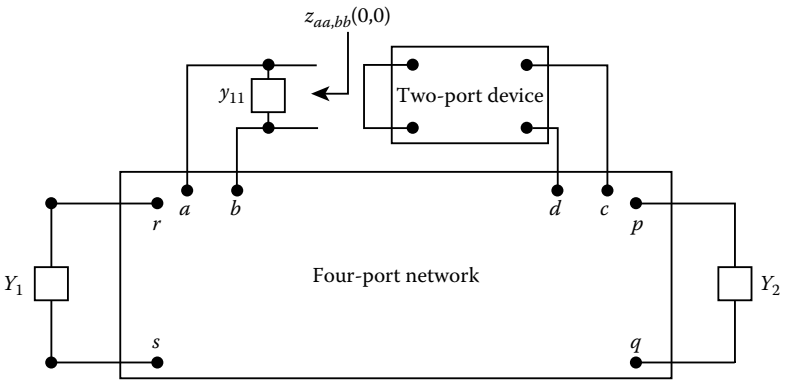


FIGURE 5.8 The measurement of the driving-point impedance $z_{aa,bb}(0, 0)$.

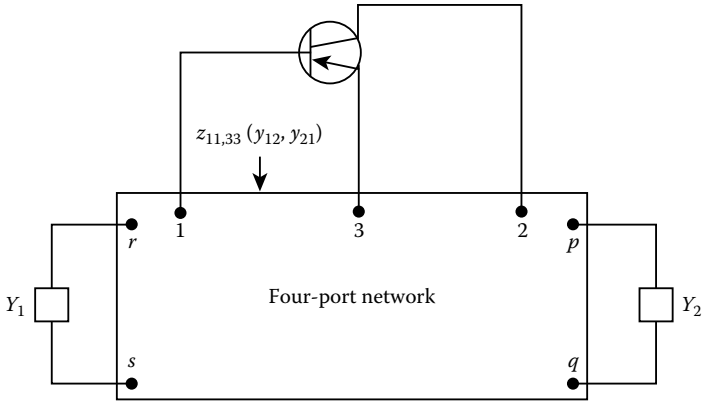


FIGURE 5.9 The measurement of the driving-point impedance $z_{11,33}(y_{12}, y_{21})$.

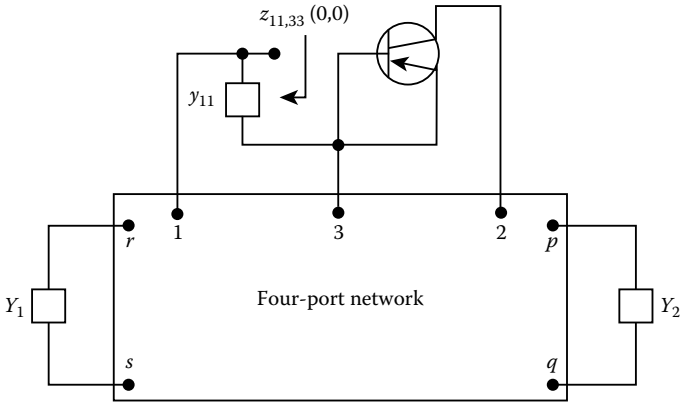


FIGURE 5.10 The measurement of the driving-point impedance $z_{11,33}(0, 0)$.

showing that the return difference $F(y_{21})$ effectively equals the ratio of two functional values assumed by the driving-point impedance looking into terminals 1 and 3 of Figure 5.2 under the condition that the controlling parameters y_{12} and y_{21} are both set to zero and the condition that they assume their nominal values. These two impedances can be measured by the network arrangements of Figures 5.9 and 5.10.

References

1. F. H. Blecher, Design principles for single loop transistor feedback amplifiers, *IRE Trans. Circuit Theory*, CT-4, 145–156, 1957.
2. S. S. Haykin, *Active Network Theory*, Reading, MA: Addison-Wesley, 1970.

6

Multiple-Loop Feedback Amplifiers

6.1 Multiple-Loop Feedback Amplifier Theory **6**-1
6.2 Return Different Matrix **6**-5
6.3 Null Return Difference Matrix **6**-6
6.4 Transfer-Function Matrix and Feedback **6**-7
6.5 Sensitivity Matrix **6**-10
6.6 Multiparameter Sensitivity **6**-14
References **6**-17

Wai-Kai Chen
University of Illinois at Chicago

So far, we have studied the single-loop feedback amplifiers. The concept of feedback was introduced in terms of return difference. We found that return difference is the difference between the unit applied signal and the returned signal. The returned signal has the same physical meaning as the loop transmission in the ideal feedback mode. It plays an important role in the study of amplifier stability, its sensitivity to the variations of the parameters, and the determination of its transfer and driving point impedances. The fact that return difference can be measured experimentally for many practical amplifiers indicates that we can include all the parasitic effects in the stability study, and that stability problem can be reduced to a Nyquist plot.

In this section, we study amplifiers that contain a multiplicity of inputs, outputs, and feedback loops. They are referred to as the multiple-loop feedback amplifiers. As might be expected, the notion of return difference with respect to an element is no longer applicable, because we are dealing with a group of elements. For this, we generalize the concept of return difference for a controlled source to the notion of return difference matrix for a multiplicity of controlled sources. For measurement situations, we introduce the null return difference matrix and discuss its physical significance. We demonstrate that the determinant of the overall transfer function matrix can be expressed explicitly in terms of the determinants of the return difference and the null return difference matrices, thereby allowing us to generalize Blackman's formula for the input impedance.

6.1 Multiple-Loop Feedback Amplifier Theory

The general configuration of a multiple-input, multiple-output, and multiple-loop feedback amplifier is presented in Figure 6.1, in which the input, output, and feedback variables may be either currents or voltages. For the specific arrangement of Figure 6.1, the input and output variables are represented by an n-dimensional vector $\mathbf{u}$ and an m-dimensional vector $\mathbf{y}$ as

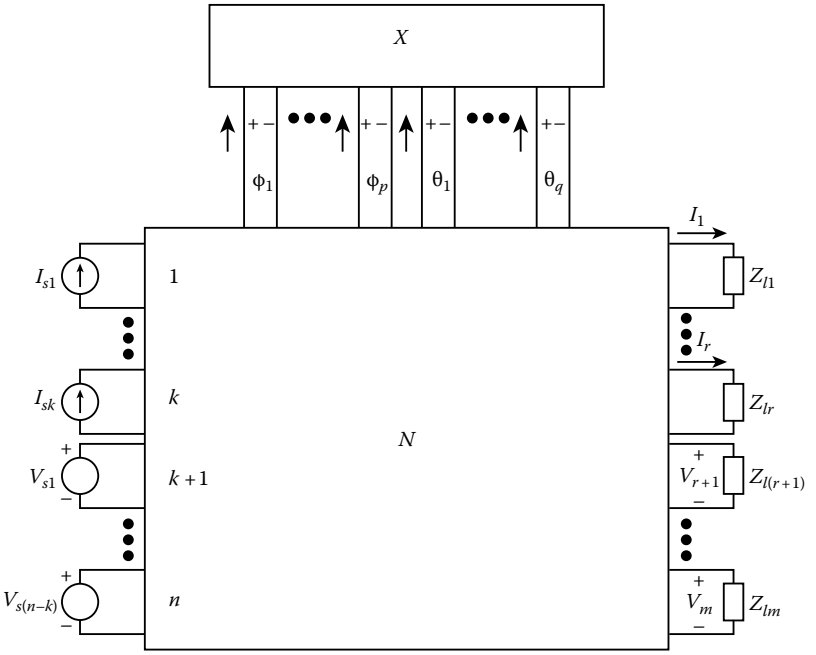


FIGURE 6.1 The general configuration of a multiple-input, multiple-output, and multiple-loop feedback amplifier.

$$\mathbf{u}(s) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} I_{s1} \\ I_{s2} \\ \vdots \\ I_{sk} \\ V_{s1} \\ V_{s2} \\ \vdots \\ V_{s(n-k)} \end{bmatrix}, \quad \mathbf{y}(s) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \\ y_{r+1} \\ y_{r+2} \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_r \\ V_{r+1} \\ V_{r+2} \\ \vdots \\ V_m \end{bmatrix} \tag{6.1}$$

respectively. The elements of interest can be represented by a rectangular matrix $\mathbf{X}$ of order $q \times p$ relating the controlled and controlling variables by the matrix equation

$$\mathbf{\Theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_q \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{q1} & x_{q2} & \cdots & x_{qp} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \mathbf{X}\mathbf{\Phi} \tag{6.2}$$

where the p-dimensional vector $\mathbf{\Phi}$ is called the controlling vector, and the q-dimensional vector $\mathbf{\Theta}$ is the controlled vector. The controlled variables θ_k and the controlling variables Φ_k can either be currents or voltages. The matrix $\mathbf{X}$ can represent either a transfer-function matrix or a driving-point function matrix. If $\mathbf{X}$ represents a driving-point function matrix, the vectors $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are of the same dimension ($q = p$) and their components are the currents and voltages of a p-port network.

The general configuration of Figure 6.1 can be represented equivalently by the block diagram of Figure 6.2 in which N is a $(p + q + m + n)$-port network and the elements of interest are exhibited

explicitly by the block **X**. For the $(p+q+m+n)$-port network N, the vectors **u** and are **Θ** are its inputs, and the vectors **Φ** and **y** its outputs. Since N is linear, the input and output vectors are related by the matrix equations

$$\mathbf{\Phi} = \mathbf{A\Theta} + \mathbf{Bu} \tag{6.3a}$$

$$\mathbf{y} = \mathbf{C\Theta} + \mathbf{Du} \tag{6.3b}$$

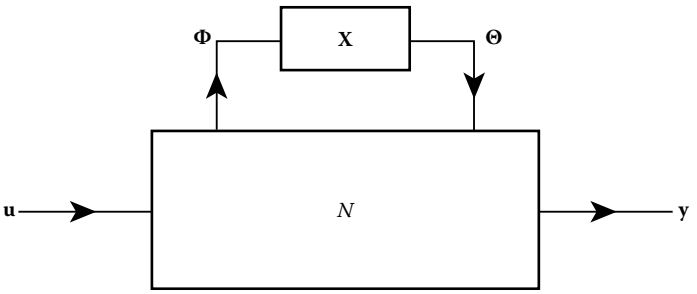


FIGURE 6.2 The block diagram of the general feedback configuration of Figure 6.1.

where **A**, **B**, **C**, and **D** are transfer-function matrices of orders $p \times q$, $p \times n$, $m \times q$, and $m \times n$, respectively. The vectors **Θ** and **Φ** are not independent and are related by

$$\mathbf{\Theta} = \mathbf{X\Phi} \tag{6.3c}$$

The relationships among the above three linear matrix equations can also be represented by a matrix signal-flow graph as shown in Figure 6.3 known as the fundamental matrix feedback-flow graph. The overall closed-loop transfer-function matrix of the multiple-loop feedback amplifier is defined by the equation

$$\mathbf{y} = \mathbf{W}(\mathbf{X})\mathbf{u} \tag{6.4}$$

where $\mathbf{W}(\mathbf{X})$ is of order $m \times n$. As before, to emphasize the importance of **X**, the matrix **W** is written as $\mathbf{W}(\mathbf{X})$ for the present discussion, even though it is also a function of the complex-frequency variable s. Combining the previous matrix equations, the transfer-function matrix is

$$\mathbf{W}(\mathbf{X}) = \mathbf{D} + \mathbf{CX}(\mathbf{1}_p - \mathbf{AX})^{-1}\mathbf{B} \tag{6.5a}$$

or

$$\mathbf{W}(\mathbf{X}) = \mathbf{D} + \mathbf{C}(\mathbf{1}_q - \mathbf{XA})^{-1}\mathbf{XB} \tag{6.5b}$$

where $\mathbf{1}_p$ denotes the identity matrix of order p. Clearly, we have

$$\mathbf{W}(\mathbf{0}) = \mathbf{D} \tag{6.6}$$

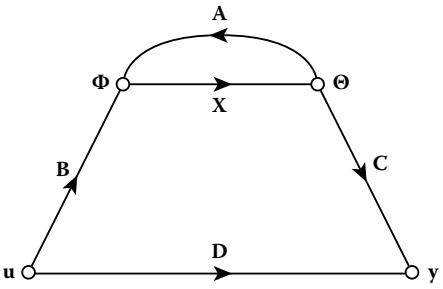


FIGURE 6.3 The fundamental matrix feedback-flow graph.

In particular, when **X** is square and nonsingular, Equation 6.5 can be written as

$$\mathbf{W}(\mathbf{X}) = \mathbf{D} + \mathbf{C}(\mathbf{X}^{-1} - \mathbf{A})^{-1}\mathbf{B} \tag{6.7}$$

Example 6.1

Consider the voltage-series feedback amplifier of Figure 3.9. An equivalent network is shown in Figure 6.4 in which we have assumed that the two transistors

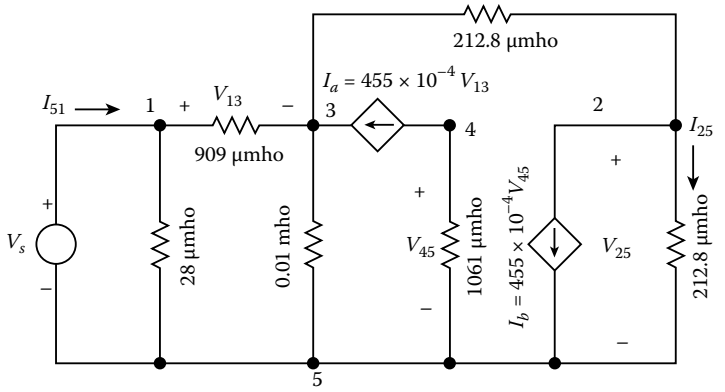


FIGURE 6.4 An equivalent network of the voltage-series feedback amplifier of Figure 3.9.

are identical with $h_{ie} = 1.1$ kΩ, $h_{fe} = 50$, $h_{re} = h_{oe} = 0$. Let the controlling parameters of the two controlled sources be the elements of interest. Then we have

$$\mathbf{\Theta} = \begin{bmatrix} I_a \\ I_b \end{bmatrix} = 10^{-4} \begin{bmatrix} 455 & 0 \\ 0 & 455 \end{bmatrix} \begin{bmatrix} V_{13} \\ V_{45} \end{bmatrix} = \mathbf{X\Phi} \tag{6.8}$$

Assume that the output voltage V_{25} and input current I_{51} are the output variables. Then the seven-port network N defined by the variables V_{13}, V_{45}, V_{25}, I_{51}, I_a, I_b, and V_s can be characterized by the matrix equations

$$\begin{aligned} \mathbf{\Phi} = \begin{bmatrix} V_{13} \\ V_{45} \end{bmatrix} &= \begin{bmatrix} -90.782 & 45.391 \\ -942.507 & 0 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \end{bmatrix} + \begin{bmatrix} 0.91748 \\ 0 \end{bmatrix} [V_s] \\ &= \mathbf{A\Theta} + \mathbf{Bu} \end{aligned} \tag{6.9a}$$

$$\begin{aligned} \mathbf{y} = \begin{bmatrix} V_{25} \\ I_{51} \end{bmatrix} &= \begin{bmatrix} 45.391 & -2372.32 \\ -0.08252 & 0.04126 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \end{bmatrix} + \begin{bmatrix} 0.041260 \\ 0.000862 \end{bmatrix} [V_s] \\ &= \mathbf{C\Theta} + \mathbf{Du} \end{aligned} \tag{6.9b}$$

According to Equation 6.4, the transfer-function matrix of the amplifier is defined by the matrix equation

$$\mathbf{y} = \begin{bmatrix} V_{25} \\ I_{51} \end{bmatrix} = \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} [V_s] = \mathbf{W}(\mathbf{X})\mathbf{u} \tag{6.10}$$

Because **X** is square and nonsingular, we can use Equation 6.7 to calculate **W**(**X**):

$$\mathbf{W}(\mathbf{X}) = \mathbf{D} + \mathbf{C}(\mathbf{X}^{-1} - \mathbf{A})^{-1}\,\mathbf{B} = \begin{bmatrix} 45.387 \\ 0.369 \times 10^{-4} \end{bmatrix} = \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} \tag{6.11}$$

where

$$(\mathbf{X}^{-1} - \mathbf{A})^{-1} = 10^{-4} \begin{bmatrix} 4.856 & 10.029 \\ -208.245 & 24.914 \end{bmatrix} \tag{6.12}$$

obtaining the closed-loop voltage gain w_{11} and input impedance Z_{in} facing the voltage source V_s as

$$w_{11} = \frac{V_{25}}{V_s} = 45.387, \quad Z_{in} = \frac{V_s}{I_{51}} = \frac{1}{w_{21}} = 27.1\ \text{k}\Omega \tag{6.13}$$

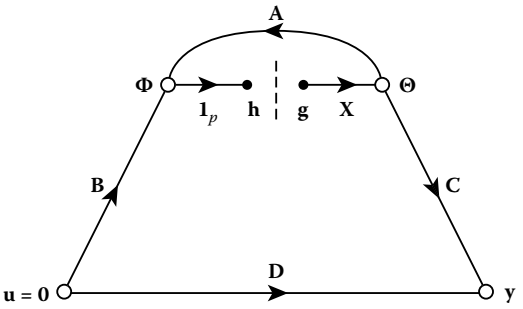


FIGURE 6.5 The physical interpretation of the loop-transmission matrix.

6.2 Return Different Matrix

In this section, we extend the concept of return difference with respect to an element to the notion of return difference matrix with respect to a group of elements.

In the fundamental matrix feedback-flow graph of Figure 6.3, suppose that we break the input of the branch with transmittance **X**, set the input excitation vector **u** to zero, and apply a signal p-vector **g** to the right of the breaking mark, as depicted in Figure 6.5. Then the returned signal p-vector **h** to the left of the breaking mark is found to be

$$\mathbf{h} = \mathbf{AXg} \tag{6.14}$$

The square matrix **AX** is called the loop-transmission matrix and its negative is referred to as the return ratio matrix denoted by

$$\mathbf{T(X)} = -\mathbf{AX} \tag{6.15}$$

The difference between the applied signal vector **g** and the returned signal vector **h** is given by

$$\mathbf{g} - \mathbf{h} = (\mathbf{1}_p - \mathbf{AX})\mathbf{g} \tag{6.16}$$

The square matrix $\mathbf{1}_p - \mathbf{AX}$ relating the applied signal vector **g** to the difference of the applied signal vector **g** and the returned signal vector **h** is called the return difference matrix with respect to **X** and is denoted by

$$\mathbf{F(X)} = \mathbf{1}_p - \mathbf{AX} \tag{6.17}$$

Combining this with Equation 6.15 gives

$$\mathbf{F(X)} = \mathbf{1}_p + \mathbf{T(X)} \tag{6.18}$$

For the voltage-series feedback amplifier of Figure 6.4, let the controlling parameters of the two controlled current sources be the elements of interest. Then the return ratio matrix is found from Equations 6.8 and 6.9a

$$\begin{aligned}\mathbf{T(X)} = -\mathbf{AX} &= -\begin{bmatrix} -90.782 & 45.391 \\ -942.507 & 0 \end{bmatrix}\begin{bmatrix} 455 \times 10^{-4} & 0 \\ 0 & 455 \times 10^{-4} \end{bmatrix} \\ &= \begin{bmatrix} 4.131 & -2.065 \\ 42.884 & 0 \end{bmatrix}\end{aligned} \tag{6.19}$$

obtaining the return difference matrix as

$$\mathbf{F}(\mathbf{X}) = \mathbf{1}_2 + \mathbf{T}(\mathbf{X}) = \begin{bmatrix} 5.131 & -2.065 \\ 42.884 & 1 \end{bmatrix} \tag{6.20}$$

6.3 Null Return Difference Matrix

A direct extension of the null return difference for the single-loop feedback amplifier is the null return difference matrix for the multiple-loop feedback networks.

Refer again to the fundamental matrix feedback-flow graph of Figure 6.3. As before, we break the branch with transmittance **X** and apply a signal p-vector **g** to the right of the breaking mark, as illustrated in Figure 6.6. We then adjust the input excitation n-vector **u** so that the total output m-vector **y** resulting from the inputs **g** and **u** is zero. From Figure 6.6, the desired input excitation **u** is found:

$$\mathbf{Du} + \mathbf{CXg} = \mathbf{0} \tag{6.21}$$

or

$$\mathbf{u} = -\mathbf{D}^{-1}\mathbf{CXg} \tag{6.22}$$

provided that the matrix **D** is square and nonsingular. This requires that the output **y** be of the same dimension as the input **u** or $m = n$. Physically, this requirement is reasonable because the effects at the output caused by **g** can be neutralized by a unique input excitation **u** only when **u** and **y** are of the same dimension. With these inputs **u** and **g**, the returned signal **h** to the left of the breaking mark in Figure 6.6 is computed as

$$\mathbf{h} = \mathbf{Bu} + \mathbf{AXg} = (-\mathbf{BD}^{-1}\mathbf{CX} + \mathbf{AX})\mathbf{g} \tag{6.23}$$

obtaining

$$\mathbf{g} - \mathbf{h} = (\mathbf{1}_p - \mathbf{AX} + \mathbf{BD}^{-1}\mathbf{CX})\mathbf{g} \tag{6.24}$$

The square matrix

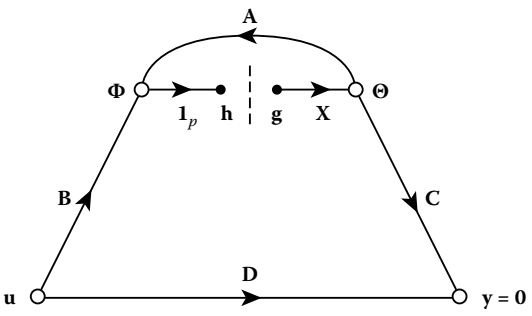


FIGURE 6.6 The physical interpretation of the null return difference matrix.

$$\begin{aligned} \hat{\mathbf{F}}(\mathbf{X}) &= \mathbf{1}_p + \hat{\mathbf{T}}(\mathbf{X}) \\ &= \mathbf{1}_p - \mathbf{AX} + \mathbf{BD}^{-1}\mathbf{CX} = \mathbf{1}_p - \hat{\mathbf{A}}\mathbf{X} \end{aligned} \tag{6.25}$$

relating the input signal vector **g** to the difference of the input signal vector **g**, and the returned signal vector **h** is called the null return difference matrix with respect to **X**, where

$$\hat{\mathbf{T}}(\mathbf{X}) = -\mathbf{AX} + \mathbf{BD}^{-1}\mathbf{CX} = -\hat{\mathbf{A}}\mathbf{X} \tag{6.26a}$$

$$\hat{\mathbf{A}} = \mathbf{A} - \mathbf{BD}^{-1}\mathbf{C} \tag{6.26b}$$

The square matrix $\hat{\mathbf{T}}(\mathbf{X})$ is known as the *null return ratio matrix.*

Example 6.2

Consider again the voltage-series feedback amplifier of Figure 3.9, an equivalent network of which is illustrated in Figure 6.4. Assume that the voltage V_{25} is the output variable. Then from Equation 6.9

$$\begin{aligned}\mathbf{\Phi} = \begin{bmatrix} V_{13} \\ V_{45} \end{bmatrix} &= \begin{bmatrix} -90.782 & 45.391 \\ -942.507 & 0 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \end{bmatrix} + \begin{bmatrix} 0.91748 \\ 0 \end{bmatrix} [V_s] \\ &= \mathbf{A\Theta} + \mathbf{B}u \end{aligned} \tag{6.27a}$$

$$\begin{aligned} y = [V_{25}] &= [45.391 \quad -2372.32] \begin{bmatrix} I_a \\ I_b \end{bmatrix} + [0.04126]\,[V_s] \\ &= \mathbf{C\Theta} + \mathbf{D}u \end{aligned} \tag{6.27b}$$

Substituting the coefficient matrices in Equation 6.26b, we obtain

$$\hat{\mathbf{A}} = \mathbf{A} - \mathbf{BD}^{-1}\mathbf{C} = \begin{bmatrix} -1,\ 100.12 & 52,\ 797.6 \\ -942.507 & 0 \end{bmatrix} \tag{6.28}$$

giving the null return difference matrix with respect to $\mathbf{X}$ as

$$\hat{\mathbf{F}}(\mathbf{X}) = \mathbf{1}_2 - \hat{\mathbf{A}}\mathbf{X} = \begin{bmatrix} 51.055 & -2402.29 \\ 42.884 & 1 \end{bmatrix} \tag{6.29}$$

Suppose that the input current I_{51} is chosen as the output variable. Then, from Equation 6.9b we have

$$y = [I_{51}] = [-0.08252\ \ 0.04126] \begin{bmatrix} I_a \\ I_b \end{bmatrix} + [0.000862]\,[V_s] = \mathbf{C\Theta} + Du \tag{6.30}$$

The corresponding null return difference matrix becomes

$$\hat{\mathbf{F}}(\mathbf{X}) = \mathbf{1}_2 - \hat{\mathbf{A}}\mathbf{X} = \begin{bmatrix} 1.13426 & -0.06713 \\ 42.8841 & 1 \end{bmatrix} \tag{6.31}$$

where

$$\hat{\mathbf{A}} = \begin{bmatrix} -2.95085 & 1.47543 \\ -942.507 & 0 \end{bmatrix} \tag{6.32}$$

6.4 Transfer-Function Matrix and Feedback

In this section, we show the effect of feedback on the transfer-function matrix $\mathbf{W}(\mathbf{X})$. Specifically, we express det $\mathbf{W}(\mathbf{X})$ in terms of the det $\mathbf{X}(\mathbf{0})$ and the determinants of the return difference and null return difference matrices, thereby generalizing Blackman's impedance formula for a single input to a multiplicity of inputs.

Before we proceed to develop the desired relation, we state the following determinant identity for two arbitrary matrices $\mathbf{M}$ and $\mathbf{N}$ of order $m \times n$ and $n \times m$:

$$\det(\mathbf{1}_m + \mathbf{MN}) = \det(\mathbf{1}_n + \mathbf{NM}) \tag{6.33}$$

a proof of which may be found in [1,2]. Using this, we next establish the following generalization of Blackman's formula for input impedance.

THEOREM 6.1

In a multiple-loop feedback amplifier, if $\mathbf{W}(\mathbf{0}) = \mathbf{D}$ *is nonsingular, then the determinant of the transfer-function matrix* $\mathbf{W}(\mathbf{X})$ *is related to the determinants of the return difference matrix* $\mathbf{F}(\mathbf{X})$ *and the null return difference matrix* $\hat{\mathbf{F}}(\mathbf{X})$ *by*

$$\det \mathbf{W}(\mathbf{X}) = \det \mathbf{W}(\mathbf{0}) \frac{\det \hat{\mathbf{F}}(\mathbf{X})}{\det \mathbf{F}(\mathbf{X})} \tag{6.34}$$

PROOF: From Equation 6.5a, we obtain

$$\mathbf{W}(\mathbf{X}) = \mathbf{D}\left[\mathbf{1}_n + \mathbf{D}^{-1}\mathbf{C}\mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\mathbf{B}\right] \tag{6.35}$$

yielding

$$\begin{aligned} \det \mathbf{W}(\mathbf{X}) &= [\det \mathbf{W}(\mathbf{0})]\det\left[\mathbf{1}_n + \mathbf{D}^{-1}\mathbf{C}\mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\mathbf{B}\right] \\ &= [\det \mathbf{W}(\mathbf{0})]\det\left[\mathbf{1}_p + \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\right] \\ &= [\det \mathbf{W}(\mathbf{0})]\det\left[\mathbf{1}_p - \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\mathbf{X}\right](\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1} \\ &= \frac{\det \mathbf{W}(\mathbf{0})\det \hat{\mathbf{F}}(\mathbf{X})}{\det \mathbf{F}(\mathbf{X})} \end{aligned} \tag{6.36}$$

The second line follows directly from Equation 6.33. This completes the proof of the theorem.

As indicated in Equation 4.4, the input impedance $Z(x)$ looking into a terminal pair can be conveniently expressed as

$$Z(x) = Z(0)\frac{F(\text{input short ciruited})}{F(\text{input open circuited})} \tag{6.37}$$

A similar expression can be derived from Equation 6.34 if $\mathbf{W}(\mathbf{X})$ denotes the impedance matrix of an n-port network of Figure 6.1. In this case, $\mathbf{F}(\mathbf{X})$ is the return difference matrix with respect to $\mathbf{X}$ for the situation when the n ports where the impedance matrix are defined are left open without any sources, and we write $\mathbf{F}(\mathbf{X}) = \mathbf{F}(\text{input open-circuited})$. Likewise, $\hat{\mathbf{F}}(\mathbf{X})$ is the return difference matrix with respect to $\mathbf{X}$ for the input port-current vector $\mathbf{I}_s$ and the output port-voltage vector $\mathbf{V}$ under the condition that $\mathbf{I}_s$ is adjusted so that the port-voltage vector $\mathbf{V}$ is identically zero. In other words, $\hat{\mathbf{F}}(\mathbf{X})$ is the return difference matrix for the situation when the n ports, where the impedance matrix is defined, are short-circuited, and we write $\hat{\mathbf{F}}(\mathbf{X}) = \mathbf{F}$ (input short-circuited). Consequently, the determinant of the impedance matrix $\mathbf{Z}(\mathbf{X})$ of an n-port network can be expressed from Equation 6.34 as

$$\det \mathbf{Z}(\mathbf{X}) = \det \mathbf{Z}(\mathbf{0})\frac{\det \mathbf{F}\,(\text{input short circuited})}{\det \mathbf{F}\,(\text{input open circuited})} \tag{6.38}$$ ■

Example 6.3

Refer again to the voltage-series feedback amplifier of Figure 3.9, an equivalent network of which is illustrated in Figure 6.4. As computed in Equation 6.20, the return difference matrix with respect to the two controlling parameters is given by

$$\mathbf{F}(\mathbf{X}) = \mathbf{1}_2 + \mathbf{T}(\mathbf{X}) = \begin{bmatrix} 5.131 & -2.065 \\ 42.884 & 1 \end{bmatrix} \tag{6.39}$$

the determinant of which is

$$\det \mathbf{F}(\mathbf{X}) = 93.68646 \tag{6.40}$$

If V_{25} of Figure 6.4 is chosen as the output and V_s as the input, the null return difference matrix is, from Equation 6.29,

$$\hat{\mathbf{F}}(\mathbf{X}) = \mathbf{1}_2 - \hat{\mathbf{A}}\mathbf{X} = \begin{bmatrix} 51.055 & -2402.29 \\ 42.884 & 1 \end{bmatrix} \tag{6.41}$$

the determinant of which is

$$\det \hat{\mathbf{F}}(\mathbf{X}) = 103,\ 071 \tag{6.42}$$

By appealing to Equation 6.34, the feedback amplifier voltage gain V_{25}/V_s can be written as

$$w(\mathbf{X}) = \frac{V_{25}}{V_s} = w(\mathbf{0})\frac{\det \hat{\mathbf{F}}(\mathbf{X})}{\det \mathbf{F}(\mathbf{X})} = 0.04126\ \frac{103,\ 071}{93.68646} = 45.39 \tag{6.43}$$

confirming Equation 3.44, where $w(\mathbf{0}) = 0.04126$, as given in Equation 6.27b.

Suppose, instead, that the input current I_{51} is chosen as the output and V_s as the input. Then, from Equation 6.31, the null return difference matrix becomes

$$\hat{\mathbf{F}}(\mathbf{X}) = \mathbf{1}_2 - \hat{\mathbf{A}}(\mathbf{X}) = \begin{bmatrix} 1.13426 & -0.06713 \\ 42.8841 & 1 \end{bmatrix} \tag{6.44}$$

the determinant of which is

$$\det \hat{\mathbf{F}}(\mathbf{X}) = 4.01307 \tag{6.45}$$

By applying Equation 6.34, the amplifier input admittance is obtained as

$$\begin{aligned} w(\mathbf{X}) = \frac{I_{51}}{V_s} &= w(\mathbf{0})\frac{\det \hat{\mathbf{F}}(\mathbf{X})}{\det \mathbf{F}(\mathbf{X})} \\ &= 8.62 \times 10^{-4}\ \frac{4.01307}{93.68646} = 36.92\ \mu\text{mho} \end{aligned} \tag{6.46}$$

or 27.1 kΩ, confirming Equation 6.13, where $w(\mathbf{0}) = 862\ \mu$mho is found from Equation 6.30.

Another useful application of the generalized Blackman's formula (Equation 6.38) is that it provides the basis of a procedure for the indirect measurement of return difference. Refer to the general feedback network of Figure 6.2. Suppose that we wish to measure the return difference $F(y_{21})$ with respect to the forward short circuit transfer admittance y_{21} of a two-port device characterized by its y parameters y_{ij}.

Choose the two controlling parameters y_{21} and y_{12} to be the elements of interest. Then, from Figure 5.2 we obtain

$$\mathbf{\Theta} = \begin{bmatrix} I_a \\ I_b \end{bmatrix} = \begin{bmatrix} y_{21} & 0 \\ 0 & y_{12} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \mathbf{X\Phi} \tag{6.47}$$

where I_a and I_b are the currents of the voltage-controlled current sources. By appealing to Equation 6.38, the impedance looking into terminals a and b of Figure 5.2 can be written as

$$z_{aa,bb}(y_{12}, y_{21}) = z_{aa,bb}(0, 0)\frac{\det \mathbf{F} \text{ (input short circuited)}}{\det \mathbf{F} \text{ (input open circuited)}} \tag{6.48}$$

When the input terminals a and b are open-circuited, the resulting return difference matrix is exactly the same as that found under normal operating conditions, and we have

$$\mathbf{F} \text{ (input open circuited)} = \mathbf{F}(\mathbf{X}) = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{6.49}$$

Because

$$\mathbf{F}(\mathbf{X}) = \mathbf{1}_2 - \mathbf{AX} \tag{6.50}$$

the elements F_{11} and F_{21} are calculated with $y_{12} = 0$, whereas F_{12} and F_{22} are evaluated with $y_{21} = 0$. When the input terminals a and b are short circuited, the feedback loop is interrupted and only the second row and first column element of the matrix $\mathbf{A}$ is nonzero, and we obtain

$$\det \mathbf{F} \text{ (input short circuited)} = 1 \tag{6.51}$$

Because $\mathbf{X}$ is diagonal, the return difference function $F(y_{21})$ can be expressed in terms of $\det \mathbf{F}(\mathbf{X})$ and the cofactor of the first row and first column element of $\mathbf{F}(\mathbf{X})$:

$$F(y_{21}) = \frac{\det \mathbf{F}(\mathbf{X})}{F_{22}} \tag{6.52}$$

Substituting these in Equation 6.48 yields

$$F(y_{12})|_{y_{21}=0}\ F(y_{21}) = \frac{z_{aa,bb}(0, 0)}{z_{aa,bb}(y_{12}, y_{21})} \tag{6.53}$$

where

$$F_{22} = 1 - a_{22}y_{12}|_{y_{21}=0} = F(y_{12})|_{y_{21}=0} \tag{6.54}$$

and a_{22} is the second row and second column element of $\mathbf{A}$. Formula (Equation 6.53) was derived earlier in Equation 5.7 using the network arrangements of Figures 5.7 and 5.8 to measure the elements $F(y_{12})|_{y_{21}=0}$ and $z_{aa,bb}(0,0)$, respectively.

6.5 Sensitivity Matrix

We have studied the sensitivity of a transfer function with respect to the change of a particular element in the network. In a multiple-loop feedback network, we are usually interested in the sensitivity of a transfer function with respect to the variation of a set of elements in the network. This set may include either

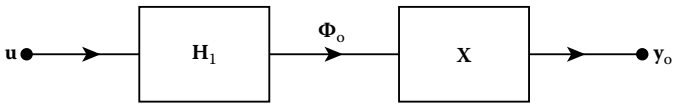


FIGURE 6.7 The block diagram of a multivariable open-loop control system.

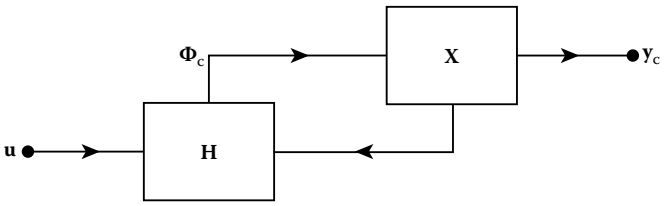


FIGURE 6.8 The general feedback structure.

elements that are inherently sensitive to variation or elements where the effect on the overall amplifier performance is of paramount importance to the designers. For this, we introduce a sensitivity matrix and develop formulas for computing multiparameter sensitivity function for a multiple-loop feedback amplifier [3].

Figure 6.7 is the block diagram of a multivariable open-loop control system with n inputs and m outputs, whereas Figure 6.8 is the general feedback structure. If all feedback signals are obtainable from the output and if the controllers are linear, no loss of generality occurs by assuming the controller to be of the form given in Figure 6.9.

Denote the set of Laplace-transformed input signals by the n-vector $\mathbf{u}$, the set of inputs to the network $\mathbf{X}$ in the open-loop configuration of Figure 6.7 by the p-vector $\mathbf{\Phi}_o$, and the set of outputs of the network $\mathbf{X}$ of Figure 6.7 by the m-vector $\mathbf{y}_o$. Let the corresponding signals for the closed-loop configuration of Figure 6.9 be denoted by the n-vector $\mathbf{u}$, the p-vector $\mathbf{\Phi}_c$, and the m-vector $\mathbf{y}_c$, respectively. Then, from Figures 6.7 and 6.9, we obtain the following relations:

$$\mathbf{y}_o = \mathbf{X}\mathbf{\Phi}_o \tag{6.55a}$$

$$\mathbf{\Phi}_o = \mathbf{H}_1\mathbf{u} \tag{6.55b}$$

$$\mathbf{y}_c = \mathbf{X}\mathbf{\Phi}_c \tag{6.55c}$$

$$\mathbf{\Phi}_c = \mathbf{H}_2(\mathbf{u} + \mathbf{H}_3\mathbf{y}_c) \tag{6.55d}$$

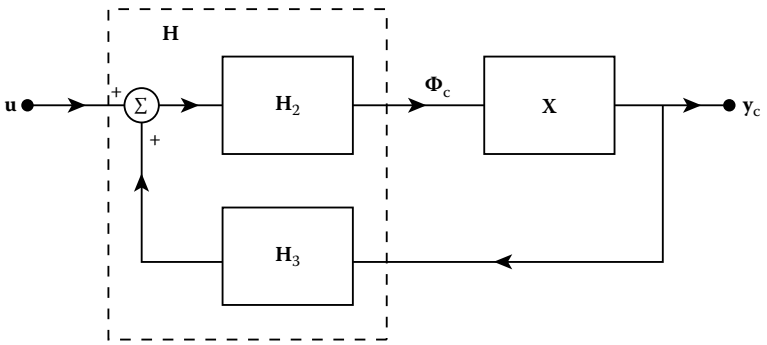


FIGURE 6.9 The general feedback configuration.

where the transfer-function matrices $\mathbf{X}$, $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_3$ are of order $m \times p$, $p \times n$, $p \times n$, and $n \times m$, respectively. Combining Equation 6.55c and d yields

$$(\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)\mathbf{y}_c = \mathbf{X}\mathbf{H}_2\mathbf{u} \tag{6.56}$$

or

$$\mathbf{y}_c = (\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}\mathbf{H}_2\mathbf{u} \tag{6.57}$$

The closed-loop transfer function matrix $\mathbf{W}(\mathbf{X})$ that relates the input vector $\mathbf{u}$ to the output vector $\mathbf{y}_c$ is defined by the equation

$$\mathbf{y}_c = \mathbf{W}(\mathbf{X})\mathbf{u} \tag{6.58}$$

identifying from Equation 6.57 the $m \times n$ matrix

$$\mathbf{W}(\mathbf{X}) = (\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}\mathbf{H}_2 \tag{6.59}$$

Now, suppose that $\mathbf{X}$ is perturbed from $\mathbf{X}$ to $\mathbf{X} + \boldsymbol{\delta}\mathbf{X}$. The outputs of the open-loop and closed-loop systems of Figures 6.7 and 6.9 will no longer be the same as before. Distinguishing the new from the old variables by the superscript $+$, we have

$$\mathbf{y}_o^+ = \mathbf{X}^+\Phi_o \tag{6.60a}$$

$$\mathbf{y}_c^+ = \mathbf{X}^+\boldsymbol{\Phi}_c^+ \tag{6.60b}$$

$$\boldsymbol{\Phi}_c^+ = \mathbf{H}_2\left(\mathbf{u} + \mathbf{H}_3\mathbf{y}_c^+\right) \tag{6.60c}$$

where $\boldsymbol{\Phi}_o$ remains the same.

We next proceed to compare the relative effects of the variations of $\mathbf{X}$ on the performance of the open-loop and the closed-loop systems. For a meaningful comparison, we assume that $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_3$ are such that when there is no variation of $\mathbf{X}$, $\mathbf{y}_o = \mathbf{y}_c$. Define the error vectors resulting from perturbation of $\mathbf{X}$ as

$$\mathbf{E}_o = \mathbf{y}_o - \mathbf{y}_o^+ \tag{6.61a}$$

$$\mathbf{E}_c = \mathbf{y}_c - \mathbf{y}_c^+ \tag{6.61b}$$

A square matrix relating $\mathbf{E}_o$ to $\mathbf{E}_c$ is called the **sensitivity matrix** $\mathcal{S}(\mathbf{X})$ for the transfer function matrix $\mathbf{W}(\mathbf{X})$ with respect to the variations of $\mathbf{X}$:

$$\mathbf{E}_c = \mathcal{S}(\mathbf{X})\mathbf{E}_o \tag{6.62}$$

In the following, we express the **sensitivity matrix** $\mathcal{S}(\mathbf{X})$ in terms of the system matrices $\mathbf{X}$, $\mathbf{H}_2$, and $\mathbf{H}_3$.

The input and output relation similar to that given in Equation 6.57 for the perturbed system can be written as

$$\mathbf{y}_c^+ = (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}^+\mathbf{H}_2\mathbf{u} \tag{6.63}$$

Substituting Equations 6.57 and 6.63 in Equation 6.61b gives

$$\begin{aligned}\mathbf{E}_c &= \mathbf{y}_c - \mathbf{y}_c^+ = \left[(\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}\mathbf{H}_2 - (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}^+\mathbf{H}_2\right]\mathbf{u} \\ &= (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\left\{[\mathbf{1}_m - (\mathbf{X} + \boldsymbol{\delta}\mathbf{X})\mathbf{H}_2\mathbf{H}_3](\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}\mathbf{H}_2 - (\mathbf{X} + \boldsymbol{\delta}\mathbf{X})\mathbf{H}_2\right\}\mathbf{u} \\ &= (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\left[\mathbf{X}\mathbf{H}_2 - \boldsymbol{\delta}\mathbf{X}\mathbf{H}_2\mathbf{H}_3(\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{X}\mathbf{H}_2 - \mathbf{X}\mathbf{H}_2 - \boldsymbol{\delta}\mathbf{X}\mathbf{H}_2\right]\mathbf{u} \\ &= -(\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\boldsymbol{\delta}\mathbf{X}\mathbf{H}_2[\mathbf{1}_n + \mathbf{H}_3\mathbf{W}(\mathbf{X})]\mathbf{u}\end{aligned} \tag{6.64}$$

From Equations 6.55d and 6.58, we obtain

$$\boldsymbol{\Phi}_c = \mathbf{H}_2[\mathbf{1}_n + \mathbf{H}_3\mathbf{W}(\mathbf{X})]\mathbf{u} \tag{6.65}$$

Because by assuming that $\mathbf{y}_o = \mathbf{y}_c$, we have

$$\boldsymbol{\Phi}_o = \boldsymbol{\Phi}_c = \mathbf{H}_2[\mathbf{1}_n + \mathbf{H}_3\mathbf{W}(\mathbf{X})]\mathbf{u} \tag{6.66}$$

yielding

$$\mathbf{E}_o = \mathbf{y}_o - \mathbf{y}_o^+ = (\mathbf{X} - \mathbf{X}^+)\boldsymbol{\Phi}_o = -\boldsymbol{\delta}\mathbf{X}\mathbf{H}_2[\mathbf{1}_n + \mathbf{H}_3\mathbf{W}(\mathbf{X})]\mathbf{u} \tag{6.67}$$

Combining Equations 6.64 and 6.67 yields an expression relating the error vectors $\mathbf{E}_c$ and $\mathbf{E}_o$ of the closed-loop and open-loop systems by

$$\mathbf{E}_c = (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1}\mathbf{E}_o \tag{6.68}$$

obtaining the sensitivity matrix as

$$\mathcal{S}(\mathbf{X}) = (\mathbf{1}_m - \mathbf{X}^+\mathbf{H}_2\mathbf{H}_3)^{-1} \tag{6.69}$$

For small variations of $\mathbf{X}$, $\mathbf{X}^+$ is approximately equal to $\mathbf{X}$. Thus, in Figure 6.9, if the matrix triple product $\mathbf{X}\mathbf{H}_2\mathbf{H}_3$ is regarded as the *loop-transmission matrix* and $-\mathbf{X}\mathbf{H}_2\mathbf{H}_3$ as the *return ratio matrix,* then the difference between the unit matrix and the loop-transmission matrix,

$$\mathbf{1}_m - \mathbf{X}\mathbf{H}_2\mathbf{H}_3 \tag{6.70}$$

can be defined as the *return difference matrix*. Therefore, Equation 6.69 is a direct extension of the sensitivity function defined for a single-input, single-output system and for a single parameter. Recall that in Equation 4.33 we demonstrated that, using the ideal feedback model, the sensitivity function of the closed-loop transfer function with respect to the forward amplifier gain is equal to the reciprocal of its return difference with respect to the same parameter.

In particular, when $\mathbf{W}(\mathbf{X})$, $\boldsymbol{\delta}\mathbf{X}$, and $\mathbf{X}$ are square and nonsingular, from Equations 6.55a, 6.55b, and 6.58, 6.61 can be rewritten as

$$\mathbf{E}_c = \mathbf{y}_c - \mathbf{y}_c^+ = [\mathbf{W}(\mathbf{X}) - \mathbf{W}^+(\mathbf{X})]\mathbf{u} = -\boldsymbol{\delta}\mathbf{W}(\mathbf{X})\mathbf{u} \tag{6.71a}$$

$$\mathbf{E}_o = \mathbf{y}_o - \mathbf{y}_o^+ = [\mathbf{X}\mathbf{H}_1 - \mathbf{X}^+\mathbf{H}_1]\mathbf{u} = -\boldsymbol{\delta}\mathbf{X}\mathbf{H}_1\mathbf{u} \tag{6.71b}$$

If $\mathbf{H}_1$ is nonsingular, $\mathbf{u}$ in Equation 6.71b can be solved for and substituted in Equation 6.71a to give

$$\mathbf{E}_c = \boldsymbol{\delta}\mathbf{W}(\mathbf{X})\mathbf{H}_1^{-1}(\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{E}_o \tag{6.72}$$

As before, for meaningful comparison, we require that $\mathbf{y}_o = \mathbf{y}_c$ or

$$\mathbf{X}\mathbf{H}_1 = \mathbf{W}(\mathbf{X}) \tag{6.73}$$

From Equation 6.72, we obtain

$$\mathbf{E}_c = \boldsymbol{\delta}\mathbf{W}(\mathbf{X})\mathbf{W}^{-1}(\mathbf{X})\mathbf{X}(\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{E}_o \tag{6.74}$$

identifying that

$$\mathcal{S}(\mathbf{X}) = \mathrm{d}\mathbf{W}(\mathbf{X})\mathbf{W}^{-1}(\mathbf{X})\mathbf{X}(\mathrm{d}\mathbf{X})^{-1} \tag{6.75}$$

This result is to be compared with the scalar sensitivity function defined in Equation 4.26, which can be put in the form

$$\mathcal{S}(x) = (\delta w)w^{-1}x(\delta x)^{-1} \tag{6.76}$$

6.6 Multiparameter Sensitivity

In this section, we derive formulas for the effect of change of $\mathbf{X}$ on a scalar transfer function $w(\mathbf{X})$.

Let x_k, $k = 1, 2, \ldots, pq$, be the elements of $\mathbf{X}$. The multivariable Taylor series expansion of $w(\mathbf{X})$ with respect to x_k is given by

$$\delta w = \sum_{k=1}^{pq} \frac{\partial w}{\partial x_k}\delta x_k + \sum_{j=1}^{pq}\sum_{k=1}^{pq} \frac{\partial^2 w}{\partial x_j \partial x_k}\frac{\delta x_j \delta x_k}{2!} + \cdots \tag{6.77}$$

The first-order perturbation can then be written as

$$\delta w \approx \sum_{k=1}^{pq} \frac{\partial w}{\partial x_k}\delta x_k \tag{6.78}$$

Using Equation 4.26, we obtain

$$\frac{\delta w}{w} \approx \sum_{k=1}^{pq} \mathcal{S}(x_k)\frac{\delta x_k}{x_k} \tag{6.79}$$

This expression gives the fractional change of the transfer function w in terms of the scalar sensitivity functions $\mathcal{S}(x_k)$.

Refer to the fundamental matrix feedback-flow graph of Figure 6.3. If the amplifier has a single input and a single output from Equation 6.35, the overall transfer function $w(\mathbf{X})$ of the multiple-loop feedback amplifier becomes

$$w(\mathbf{X}) = D + \mathbf{C}\mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\mathbf{B} \tag{6.80}$$

When $\mathbf{X}$ is perturbed to $\mathbf{X}^+ = \mathbf{X} + \boldsymbol{\delta}\mathbf{X}$, the corresponding expression of Equation 6.80 is given by

$$w(\mathbf{X}) + \delta w(\mathbf{X}) = D + \mathbf{C}(\mathbf{X} + \boldsymbol{\delta}\mathbf{X})(\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{B} \tag{6.81}$$

or

$$\delta w(\mathbf{X}) = \mathbf{C}\big[(\mathbf{X} + \boldsymbol{\delta}\mathbf{X})(\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})^{-1} - \mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\big]\mathbf{B} \tag{6.82}$$

As $\boldsymbol{\delta}\mathbf{X}$ approaches zero, we obtain

$$\begin{aligned}
\delta w(\mathbf{X}) &= \mathbf{C}\big[(\mathbf{X} + \boldsymbol{\delta}\mathbf{X}) - \mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}(\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})\big](\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{B} \\
&= \mathbf{C}\big[\boldsymbol{\delta}\mathbf{X} + \mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\mathbf{A}\boldsymbol{\delta}\mathbf{X}\big](\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{B} \\
&= \mathbf{C}(\mathbf{1}_q - \mathbf{X}\mathbf{A})^{-1}(\boldsymbol{\delta}\mathbf{X})(\mathbf{1}_p - \mathbf{A}\mathbf{X} - \mathbf{A}\boldsymbol{\delta}\mathbf{X})^{-1}\mathbf{B} \\
&\approx \mathbf{C}(\mathbf{1}_q - \mathbf{X}\mathbf{A})^{-1}(\boldsymbol{\delta}\mathbf{X})(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1}\mathbf{B}
\end{aligned} \tag{6.83}$$

where $\mathbf{C}$ is a row q vector and $\mathbf{B}$ is a column p vector. Write

$$\mathbf{C} = [c_1 \;\; c_2 \cdots c_q] \tag{6.84a}$$

$$\mathbf{B}' = [b_1 \;\; b_2 \cdots b_p] \tag{6.84b}$$

$$\tilde{\mathbf{W}} = \mathbf{X}(\mathbf{1}_p - \mathbf{A}\mathbf{X})^{-1} = (\mathbf{1}_q - \mathbf{X}\mathbf{A})^{-1}\mathbf{X} = [\tilde{w}_{ij}] \tag{6.84c}$$

The increment $\delta w(\mathbf{X})$ can be expressed in terms of the elements of Equation 6.84 and those of $\mathbf{X}$. In the case where $\mathbf{X}$ is diagonal with

$$\mathbf{X} = \text{diag}[x_1 \;\; x_2 \cdots x_p] \tag{6.85}$$

where $p = q$, the expression for $\delta w(\mathbf{X})$ can be succinctly written as

$$\begin{aligned}
\delta w(\mathbf{X}) &= \sum_{i=1}^{p}\sum_{k=1}^{p}\sum_{j=1}^{p} c_i\left(\frac{\tilde{w}_{ik}}{x_k}\right)(\delta x_k)\left(\frac{\tilde{w}_{kj}}{x_k}\right)b_j \\
&= \sum_{i=1}^{p}\sum_{k=1}^{p}\sum_{j=1}^{p} \frac{c_i\tilde{w}_{ik}\tilde{w}_{kj}b_j}{x_k}\,\frac{\delta x_k}{x_k}
\end{aligned} \tag{6.86}$$

Comparing this with Equation 6.79, we obtain an explicit form for the single-parameter sensitivity function as

$$\mathcal{S}(x_k) = \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{c_i\tilde{w}_{ik}\tilde{w}_{kj}b_j}{x_k w(\mathbf{X})} \tag{6.87}$$

Thus, knowing Equations 6.84 and 6.85, we can calculate the multiparameter sensitivity function for the scalar transfer function $w(\mathbf{X})$ immediately.

Example 6.4

Consider again the voltage-series feedback amplifier of Figure 3.9, an equivalent network of which is shown in Figure 6.4. Assume that V_s is the input and V_{25} the output. The transfer function of interest is the amplifier voltage gain V_{25}/V_s. The elements of main concern are the two controlling parameters of the controlled sources. Thus, we let

$$\mathbf{X} = \begin{bmatrix} \tilde{\alpha}_1 & 0 \\ 0 & \tilde{\alpha}_2 \end{bmatrix} = \begin{bmatrix} 0.0455 & 0 \\ 0 & 0.0455 \end{bmatrix} \tag{6.88}$$

From Equation 6.27 we have

$$\mathbf{A} = \begin{bmatrix} -90.782 & 45.391 \\ -942.507 & 0 \end{bmatrix} \tag{6.89a}$$

$$\mathbf{B}' = [0.91748 \;\; 0] \tag{6.89b}$$

$$\mathbf{C} = [45.391 \;\; -2372.32] \tag{6.89c}$$

yielding

$$\tilde{\mathbf{W}} = \mathbf{X}(\mathbf{1}_2 - \mathbf{AX})^{-1} = 10^{-4}\begin{bmatrix} 4.85600 & 10.02904 \\ -208.245 & 24.91407 \end{bmatrix} \tag{6.90}$$

Also, from Equation 6.13 we have

$$w(\mathbf{X}) = \frac{V_{25}}{V_s} = 45.387 \tag{6.91}$$

To compute the sensitivity functions with respect to $\tilde{\alpha}_1$ and $\tilde{\alpha}_2$, we apply Equation 6.87 and obtain

$$\mathcal{S}(\tilde{\alpha}_1) = \sum_{i=1}^{2}\sum_{j=1}^{2}\frac{c_i\tilde{w}_{i1}\tilde{w}_{1j}b_j}{\tilde{\alpha}_1 w(\mathbf{X})} = \frac{c_1\tilde{w}_{11}\tilde{w}_{11}b_1 + c_1\tilde{w}_{11}\tilde{w}_{12}b_2 + c_2\tilde{w}_{21}\tilde{w}_{11}b_1 + c_2\tilde{w}_{21}\tilde{w}_{12}b_2}{\tilde{\alpha}_1 w}$$
$$= 0.01066 \tag{6.92a}$$

$$\mathcal{S}(\tilde{\alpha}_2) = \frac{c_1\tilde{w}_{12}\tilde{w}_{21}b_1 + c_1\tilde{w}_{12}\tilde{w}_{22}b_2 + c_2\tilde{w}_{22}\tilde{w}_{21}b_1 + c_2\tilde{w}_{22}\tilde{w}_{22}b_2}{\tilde{\alpha}_2 w} = 0.05426 \tag{6.92b}$$

As a check, we use Equation 4.30 to compute these sensitivities. From Equations 3.45 and 3.52, we have

$$F(\tilde{\alpha}_1) = 93.70 \tag{6.93a}$$

$$F(\tilde{\alpha}_2) = 18.26 \tag{6.93b}$$

$$\hat{F}(\tilde{\alpha}_1) = 103.07 \times 10^3 \tag{6.93c}$$

$$\hat{F}(\tilde{\alpha}_2) = 2018.70 \tag{6.93d}$$

Substituting these in Equation 4.30 the sensitivity functions are

$$\mathcal{S}(\tilde{\alpha}_1) = \frac{1}{F(\tilde{\alpha}_1)} - \frac{1}{\hat{F}(\tilde{\alpha}_1)} = 0.01066 \tag{6.94a}$$

$$\mathcal{S}(\tilde{\alpha}_2) = \frac{1}{F(\tilde{\alpha}_2)} - \frac{1}{\hat{F}(\tilde{\alpha}_2)} = 0.05427 \tag{6.94b}$$

confirming Equation 6.92.

Suppose that $\tilde{\alpha}_1$ is changed by 4% and $\tilde{\alpha}_2$ by 6%. The fractional change of the voltage gain $w(\mathbf{X})$ is found from Equation 6.79 as

$$\frac{\delta w}{w} \approx \mathcal{S}(\tilde{\alpha}_1)\frac{\delta\tilde{\alpha}_1}{\tilde{\alpha}_1} + \mathcal{S}(\tilde{\alpha}_2)\frac{\delta\tilde{\alpha}_2}{\tilde{\alpha}_2} = 0.003683 \tag{6.95}$$

or 0.37%.

References

1. W.-K. Chen, *Active Network and Feedback Amplifier Theory*, New York: McGraw-Hill, 1980, Chaps. 2, 4, 5, 7.
2. W.-K. Chen, *Active Network and Analysis*, Singapore: World Scientific, 1991, Chaps. 2, 4, 5, 7.
3. J. B. Cruz Jr. and W. R. Perkins, A new approach to the sensitivity problem in multivariable feedback system design, *IEEE Trans. Autom. Control*, AC-9, 216–223, 1964.
4. E. S. Kuh and R. A. Rohrer, *Theory of Linear Active Networks*, San Francisco: Holden-Day, 1967.
5. I. W. Sandberg, On the theory of linear multi-loop feedback systems, *Bell Syst. Tech. J.*, 42, 355–382, 1963.

II

Nonlinear Circuits

Leon O. Chua
University of California, Berkeley

7 **Qualitative Analysis** *Martin Hasler* **7**-1
Introduction • Resistive Circuits • Autonomous Dynamic Circuits • Nonautonomous Dynamic Circuits • References

8 **Synthesis and Design of Nonlinear Circuits** *Angel Rodríguez-Vázquez, Manuel Delgado-Restituto, Jose L. Huertas, and F. Vidal* **8**-1
Introduction • Approximation Issues • Aggregation, Scaling, and Transformation Circuits • Piecewise-Linear Circuitry • Polynomials, Rational, and Piecewise-Polynomial Functions • Sigmoids, Bells, and Collective Computation Circuits • Extension to Dynamic Systems • Appendix A: Catalog of Primitives • Appendix B: Value and Slope Hermite Basis Functions • References

9 **Representation, Approximation, and Identification** *Guanrong Chen* **9**-1
Introduction • Representation • Approximation • Identification • References

10 **Transformation and Equivalence** *Wolfgang Mathis* **10**-1
Transformation of Nonlinear Dynamical Circuit Equations • Normal Forms of Nonlinear Dynamical Circuit Equations • Dimensionless Forms of Circuit Equations • Equivalence of Nonlinear Resistive *n*-Ports • Equivalence of Nonlinear Dynamical *n*-Ports • Equivalence of Reciprocal Nonlinear Dynamic Circuits • References

11 **Piecewise-Linear Circuits and Piecewise-Linear Analysis** *Joos Vandewalle and Lieven Vandenberghe* **11**-1
Introduction and Motivation • Hierarchy of Piecewise-Linear Models and Their Representations • Piecewise-Linear Models for Electronic Components • Structural Properties of Piecewise-Linear Resistive Circuits • Analysis of Piecewise-Linear Resistive Circuits • Piecewise-Linear Dynamic Circuits • Efficient Computer-Aided Analysis of PWL Circuits • Acknowledgment • References

12 **Simulation** *Erik Lindberg* **12**-1
Numerical Solution of Nonlinear Algebraic Equations • Numerical Integration of Nonlinear Differential Equations • Use of Simulation Programs • References

13 Cellular Neural Networks and Cellular Wave Computers *Tamás Roska, Ákos Zarándy, and Csaba Rekeczky* **13**-1
Introduction: Definition and Classification • Simple CNN Circuit Structure • Stored Program CNN Universal Machine and the Analogic Supercomputer Chip • Applications • Template Library: Analogical CNN Algorithms • Recent Advances • Recent Developments and Outlook • References

14 Bifurcation and Chaos *Michael Peter Kennedy* **14**-1
Introduction to Chaos • Chua's Circuit: A Paradigm for Chaos • Chua's Oscillator • van der Pol Neon Bulb Oscillator • Synchronization of Chaotic Circuits • Applications of Chaos • References • Further Information

7

Qualitative Analysis

7.1 Introduction 7-1
7.2 Resistive Circuits 7-1
Number of Solutions of a Resistive Circuit • Bounds on Voltages and Currents • Monotonic Dependence
7.3 Autonomous Dynamic Circuits 7-13
Introduction • Convergence to DC-Operating Points
7.4 Nonautonomous Dynamic Circuits 7-19
Introduction • Boundedness of the Solutions • Unique Asymptotic Behavior
References 7-21

Martin Hasler
Swiss Federal Institute of Technology

7.1 Introduction

The main goal of circuit analysis is to determine the solution of the circuit, i.e., the voltages and the currents in the circuit, usually as functions of time. The advent of powerful computers and circuit analysis software has greatly simplified this task. Basically, the circuit to be analyzed is fed to the computer through some circuit description language, or it is analyzed graphically, and the software will produce the desired voltage or current waveforms. Progress has rendered the traditional paper-and-pencil methods obsolete, in which the engineer's skill and intuition led the way through series of clever approximations, until the circuits equations can be solved analytically.

A closer comparison of the numerical and the approximate analytical solution reveals, however, that the two are not quite equivalent. Although the former is precise, it only provides the solution of the circuit with given parameters, whereas the latter is an approximation, but the approximate solutions most often is given explicitly as a function of some circuit parameters. Therefore, it allows us to assess the influence of these parameters on the solution.

If we rely entirely on the numerical solution of a circuit, we never get a global picture of its behavior, unless we carry out a huge number of analyses. Thus, the numerical analysis should be complemented by a qualitative analysis, one that concentrates on general properties of the circuit, properties that do not depend on the particular set of circuit parameters.

7.2 Resistive Circuits

The term ''resistive circuits'' is not used, as one would imagine, for circuits that are composed solely of resistors. It admits all circuit elements that are not dynamic, i.e., whose constitutive relations do not involve time derivatives, integrals over time, or time delays, etc. Expressed positively, resistive circuit elements are described by constitutive relations that involve only currents and voltages at the same time instants.

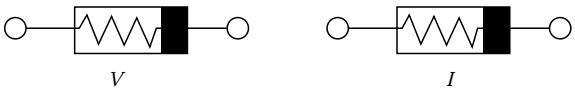


FIGURE 7.1 Symbols of the *V*- and the *I*-resistor.

Physical circuits can never be modeled in a satisfactory way by resistive circuits, but resistive circuits appear in many contexts as auxiliary constructs. The most important problem that leads to a resistive circuit is the determination of the equilibrium points, or, as is current use in electronics, the DC-operating points, of a dynamic circuit. The DC-operating points of a circuit correspond in a one-to-one fashion to the solutions of the resistive circuit obtained by removing the capacitors and by short circuiting the inductors. The resistive circuit associated with the state equations of a dynamic circuit in discussed in Ref. [1].

Among the resistive circuit elements we find, of course, the resistors. For the purposes of this introduction, we distinguish between, linear resistors, *V*-resistors, and *I*-resistors. *V*-resistors are voltage controlled, i.e., defined by constitutive relations of the form

$$i = g(v) \tag{7.1}$$

In addition, we require that g is a continuous, increasing function of v, defined for all real v. Dually, an *I*-resistor is current controlled, i.e., defined by a constitutive relation of the form

$$v = h(i) \tag{7.2}$$

In addition, we require that h is a continuous, increasing function of i, defined for all real i. We use the symbols of Figure 7.1 for *V*- and *I*-resistor. Linear resistors are examples of both *I*- and *V*-resistors. An example of a *V*-resistor that is not an *I*-resistor is the junction diode, modeled by its usual exponential constitutive relation

$$i = I_s(e^{v/nV_T} - 1) \tag{7.3}$$

Although Equation 7.3 could be solved for v and thus the constitutive relation could be written in the form of Equation 7.2, the resulting function h would be defined only for currents between $-I_s$ and $+\infty$, which is not enough to qualify for an *I*-resistor. For the same reason, the static model for a Zener diode would be an *I*-resistor, but not a *V*-resistor. Indeed, the very nature of the Zener diode limits its voltages on the negative side.

A somewhat strange by-product of our definition of *V*- and *I*-resistors is that independent voltage sources are *I*-resistors and independent current sources are *V*-resistors. Indeed, a voltage source of value E has the constitutive relation

$$v = E \tag{7.4}$$

which clearly is of the form (Equation 7.2), with a constant function h, and a current source of value I has the form

$$i = I \tag{7.5}$$

which is of the form (Equation 7.1) with a constant function g. Despite this, we shall treat the independent sources as a different type of element.

Another class of resistive elements is the controlled sources. We consider them to be two-ports, e.g., a voltage-controlled voltage source (VCVS). A VCVS is the two-port of Figure 7.2, where the constitutive relations are

$$v_1 = \alpha v_2 \tag{7.6}$$

$$i_1 = 0 \tag{7.7}$$

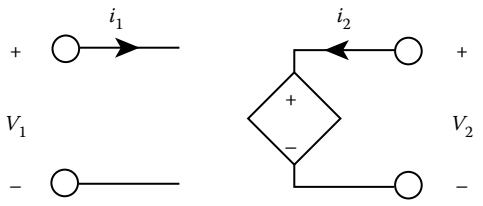


FIGURE 7.2 VCVS as a two-port.

The other controlled sources have similar forms. Another useful resistive circuit element is the ideal operational amplifier. It is a two-port defined by the two constitutive relations

$$v_1 = 0 \tag{7.8}$$

$$i_1 = 0 \tag{7.9}$$

This two-port can be decomposed into the juxtaposition of two singular one-ports, the nullator and the norator, as shown in Figure 7.3. The nullator has two constitutive relations:

$$v = 0, \quad i = 0 \tag{7.10}$$

whereas the norator has no constitutive relation.

For all practical purposes, the resistive circuit elements mentioned thus far are sufficient. By this we mean that all nonlinear resistive circuits encountered in practice possess an equivalent circuit composed of nonlinear resistors, independent and controlled sources, and nullator–norator pairs. Figure 7.4 illustrates this fact. Here, the equivalent circuit of the bipolar transistor is modeled by the Ebers–Moll equations:

$$\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = \begin{pmatrix} 1 + \frac{1}{\beta_F} & -1 \\ -1 & 1 + \frac{1}{\beta_R} \end{pmatrix} \begin{pmatrix} g(v_1) \\ g(v_2) \end{pmatrix} \tag{7.11}$$

The function g is given by the right-hand side of Equation 7.3.

Actually, the list of basic resistive circuit elements given so far is redundant, and the nullator–norator pairs render the controlled sources superfluous. An example of a substitution of controlled sources by nullator–norator pairs is given in Figure 7.4. Equivalent circuits exist for all four types of controlled sources with nullator–norator pairs. Figure 7.5 gives an equivalent circuit for a voltage-controlled current source (VCCS), where the input port is floating with respect to the output port.

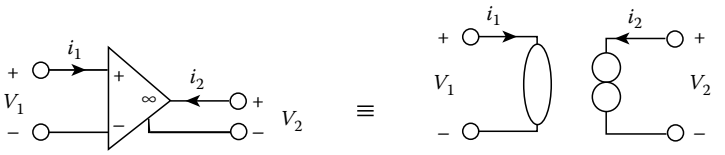


FIGURE 7.3 Operational amplifier as a juxtaposition of a nullator and a norator.

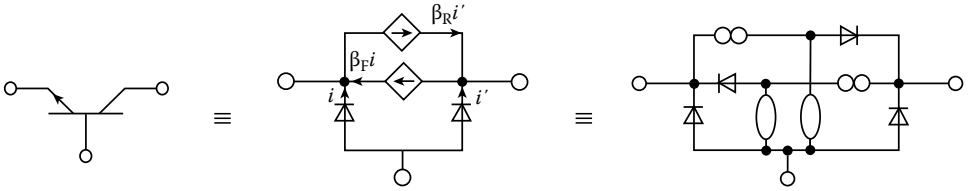


FIGURE 7.4 Equivalent circuit of a bipolar npn transistor.

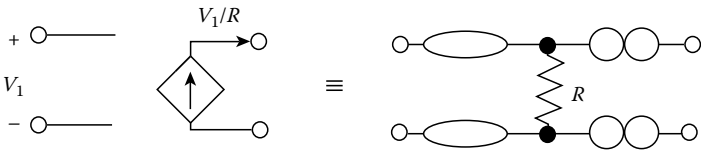


FIGURE 7.5 Equivalent circuit for a floating VCCS.

The system of equations that describes a resistive circuit is the collection of Kirchhoff equations and the constitutive relations of the circuit elements. It has the following form (if we limit ourselves to resistors, independent sources, nullators, and norators):

$$\mathbf{Ai} = \mathbf{0} \text{ (Kirchhoff's voltage law)} \tag{7.12}$$

$$\mathbf{Bv} = \mathbf{0} \text{ (Kirchhoff's voltage law)} \tag{7.13}$$

$$i_k = g(v_k) \text{ } (V\text{-resistor}) \tag{7.14}$$

$$v_k = h(i_k) \text{ } (I\text{-resistor}) \tag{7.15}$$

$$v_k = E_k \text{ (independent voltage source)} \tag{7.16}$$

$$i_k = I_k \text{ (independent current source)} \tag{7.17}$$

$$\begin{matrix} v_k = 0 \\ i_k = 0 \end{matrix} \text{ (nullators)} \tag{7.18}$$

In this system of equations, the unknowns are the branch voltages and the branch currents

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_b \end{pmatrix}, \quad \mathbf{i} = \begin{pmatrix} i_1 \\ i_2 \\ \vdots \\ i_b \end{pmatrix} \tag{7.19}$$

where the b is the number of branches. Because we have b linearly independent Kirchhoff equations [2], the system contains $2b$ equations and $2b$ unknowns. A solution $\xi = \begin{pmatrix} v \\ i \end{pmatrix}$ of the system is called a solution of the circuit. It is a collection of branch voltages and currents that satisfy Equations 7.12 through 7.19.

7.2.1 Number of Solutions of a Resistive Circuit

As we found earlier, the number of equations of a resistive circuit equals the number of unknowns. One may therefore expect a unique solution. This may be the norm, but it is far from being generally true. It is not even true for linear resistive circuits. In fact, the equations for a linear resistive circuit are of the form

$$\mathbf{H}\xi = \mathbf{e} \tag{7.20}$$

where the $2b \times 2b$ matrix $\mathbf{H}$ contains the resistances and elements of value 0, ± 1, whereas the vector $\mathbf{e}$ contains the source values and zeroes. The solution of Equation 7.20 is unique if the determinant of $\mathbf{H}$ differs from zero. If it is zero, then the circuit has either infinitely many solutions or no solution at all. Is such a case realistic? The answer is yes and no. Consider two voltages sources connected as shown in Figure 7.6.

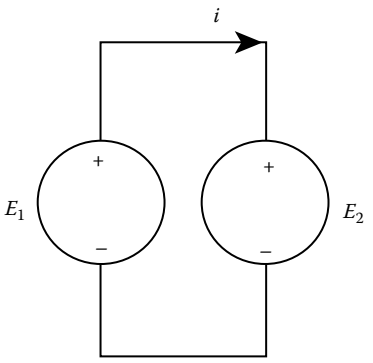


FIGURE 7.6 Circuit with zero or infinite solutions.

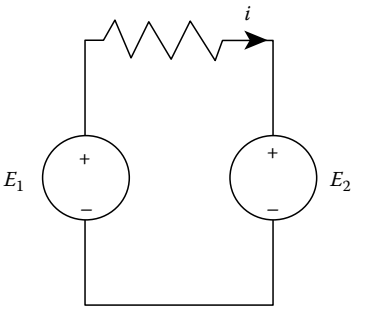


FIGURE 7.7 Circuit with exactly one solution.

If $E_1 \neq E_2$, the constitutive relations of the sources are in contradiction with Kirchhoff's voltage law (KVL), and thus the circuit has no solution, whereas when $E_1 = E_2$, the current i in Figure 7.6 is not determined by the circuit equations, and thus the circuit has infinitely many solutions. One may object that the problem is purely academic, because in practice wires as connections have a small, but positive, resistance, and therefore one should instead consider the circuit of Figure 7.7, which has exactly one solution.

Examples of singular linear resistive circuits exist that are much more complicated. However, the introduction of parasitic elements always permits us to obtain a circuit with a single solution, and thus the special case in which the matrix **H** in Equation 7.9 is singular can be disregarded. Within the framework of linear circuits, this attitude is perfectly justified. When a nonlinear circuit model is chosen, however, the situation changes. An example clarifies this point.

Consider the linear circuit of Figure 7.8. It is not difficult to see that it has exactly one solution, except when

$$R_1 R_3 = R_2 R_4 \tag{7.21}$$

In this case, the matrix **H** in Equation 7.29 is singular and the circuit of Figure 7.8 has zero or infinitely many solutions, depending on whether E differs from zero. From the point of view of linear circuits, we can disregard this singular case because it arises only when Equation 7.21 is exactly satisfied with infinite precision.

Now, replace resistor R_4 by a nonlinear resistor, where the characteristic is represented by the bold line in Figure 7.9. The resulting circuit is equivalent to the connection of a voltage source, a linear resistor, and the nonlinear resistor, as shown in Figure 7.10. Its solutions correspond to the intersections of the nonlinear resistor characteristic and the load line (Figure 7.9). Depending on the value of E, either one, two, or three solutions are available. Although we still need infinite precision to obtain two solutions, this is not the case for one or three solutions. Thus, more than one DC-operating point may be observed in electronic circuits. Indeed, for static memories, and multivibrators in general, multiple DC-operating points are an essential feature.

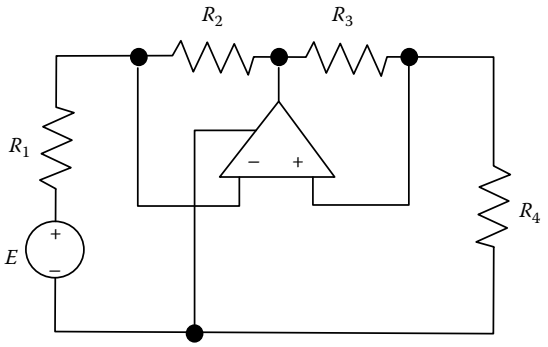


FIGURE 7.8 Circuit with one, zero, or infinite solutions.

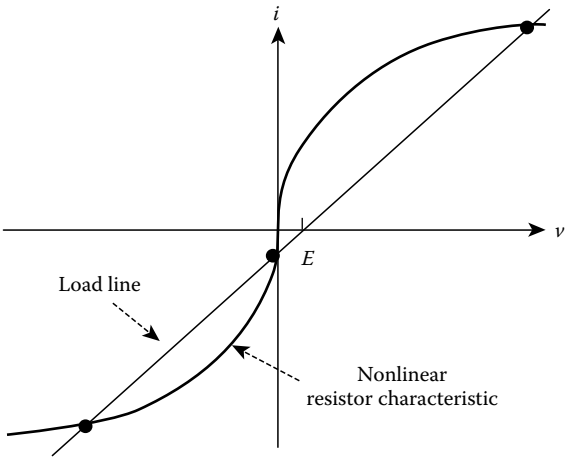


FIGURE 7.9 Characteristic of the nonlinear resistor and solutions of the circuit of Figure 7.10.

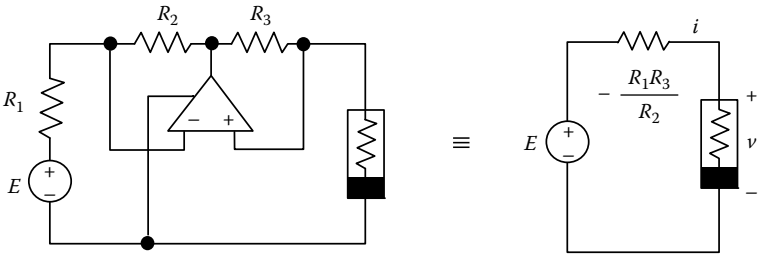


FIGURE 7.10 Circuit with one, two, or three solutions.

The example of Figure 7.10 shows an important aspect of the problem. The number of solutions depends on the parameter values of the circuit. In the example the value of E determines whether one, two, or three solutions are available. This is not always the case. An important class of nonlinear resistive circuits always has exactly one solutions, irrespective of circuit parameters. In fact, for many applications, e.g., amplification, signal shaping, logic operations, etc., it is necessary that a circuit has exactly one DC-operating point. Circuits that are designed for these functionalities should thus have a unique DC-operating point for any choice of element values.

If a resistive circuit contains only two-terminal resistors with increasing characteristics and sources, but no nonreciprocal element such as controlled sources, operational amplifiers, or transistors, the solution is usually unique. The following theorem gives a precise statement.

THEOREM 7.1

A circuit composed of independent voltage and current sources and strictly increasing resistors without loop of voltage sources and without cutset of current sources has at most one solution.

The interconnection condition concerning the sources is necessary. The circuit of Figure 7.6 is an illustration of this statement. Its solution is not unique because of the loop of voltage sources. The loop is

no longer present in the circuit of Figure 7.7, which satisfies the conditions of Theorem 7.1, and which indeed has a unique solution.

If the resistor characteristics are not *strictly* increasing but only increasing (i.e., if the v–i curves have horizontal or vertical portions), the theorem still holds, if we exclude loops of voltage sources and I-resistors, and cutsets of current sources and V-resistors.

Theorem 7.1 guarantees the uniqueness of the solution, but it cannot assure its existence. On the other hand, we do not need increasing resistor characteristics for the existence.

THEOREM 7.2

Let a circuit be composed of independent voltage and current sources and resistors whose characteristics are continuous and satisfy the following passivity condition at infinity:

$$v \to +\infty \Leftrightarrow i \to +\infty \quad \text{and} \quad v \to -\infty \Leftrightarrow i \to -\infty \tag{7.22}$$

If no loop of voltage sources and no cutset of current sources exist, then we have at least one solution of the circuit.

For refinements of this theorem, refer to Refs. [1,3].

If we admit nonreciprocal elements, neither Theorem 7.1 nor Theorem 7.2 remain valid. Indeed, the solution of the circuit of Figure 7.10 may be nonunique, even though the nonlinear resistor has a strictly increasing characteristic. In order to ensure the existence and uniqueness of a nonreciprocal nonlinear resistive circuit, nontrivial constraints on the interconnection of the elements must be observed. The theorems below give different, but basically equivalent, ways to formulate these constraints.

The first result is the culminating point of a series of papers by Sandberg and Wilson [3]. It is based on the following notion.

Definition 7.1:

- The connection of the two bipolar transistors shown in Figure 7.11 is called a feedback structure. The type of the transistors and the location of the collectors and emitters are arbitrary.
- A circuit composed of bipolar transistors, resistors, and independent sources contains a feedback structure, if it can be reduced to the circuit of Figure 7.11 by replacing each voltage source by a short circuit, each current source by an open circuit, each resistor and diode by an open or a short circuit, and each transistor by one of the five short–open circuit combinations represented in Figure 7.12.

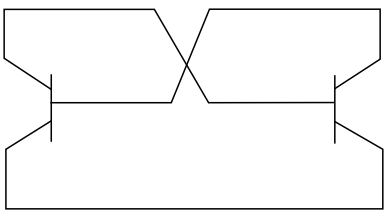


FIGURE 7.11 Feedback structure.

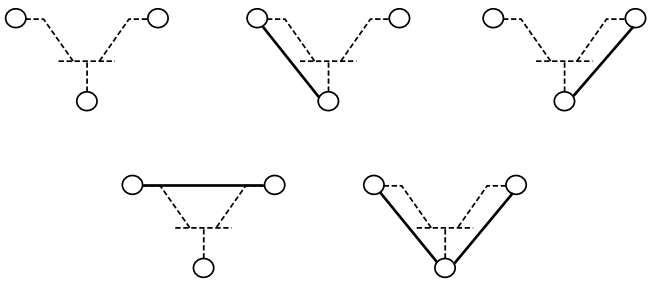


FIGURE 7.12 Short-open-circuit combinations for replacing the transistors.

THEOREM 7.3

Let a circuit be composed of bipolar transistors, described by the Ebers–Moll model, positive linear resistors, and independent sources. Suppose we have no loop of voltage sources and no cutset of current sources. If the circuit contains no feedback structure, it has exactly one solution.

This theorem [4] is extended in Ref. [5] to MOS transistors.

The second approach was developed by Nishi and Chua [6]. Instead of transistors, it admits controlled sources. In order to formulate the theorem, two notions must be introduced.

Definition 7.2: A circuit composed of controlled sources, resistors, and independent sources satisfies the interconnection condition, if the following conditions are satisfied:

- No loop is composed of voltage sources, output ports of (voltage or current) controlled voltage sources, and input ports of current-controlled (voltage or current) sources.
- No cutset is composed of current sources, outputs ports of (voltage or current) controlled current sources, and input ports of voltage-controlled (voltage or current) sources.

Definition 7.3: A circuit composed exclusively of controlled sources has a complementary tree structure if both the input and output ports each form a tree. The fundamental loop matrix of the input port tree has the form

$$B = [B_T|1] \tag{7.23}$$

The circuit is said to have a positive (negative) complementary tree structure, if the determinant of B_T is positive (negative).

THEOREM 7.4

Suppose a circuit composed of controlled sources, strictly increasing resistors satisfying (*Equation 7.22*), *and independent sources satisfies the interconnection condition. If, by replacing each resistor either by a short circuit or an open circuit, all independent and some dependent voltage sources by short circuits, and all independent and some dependent current sources by open circuits, one never obtains a negative complementary tree structure, the circuit has exactly one solution* [6].

A similar theorem for circuits with operational amplifiers instead of controlled sources is proved in Ref. [7].

The third approach is that of Hasler [1,8]. The nonreciprocal elements here are nullator–norator pairs. Instead of reducing the circuit by some operations in order to obtain a certain structure, we must orient the resistors in certain way. Again, we must first introduce a new concept.

Definition 7.4: Let a circuit be composed of nullator–norator pairs, resistors, and independent voltage and current sources. A partial orientation of the resistors is uniform, if the following two conditions are satisfied:

- Every oriented resistor is part of an evenly directed loop composed only of oriented resistors and voltages sources.
- Every oriented resistor is part of an evenly directed cutset composed only of norators, oriented resistors, and voltage sources.

THEOREM 7.5

Let a circuit be composed of nullator–norator pairs, V- *and* I-*resistors, and independent voltage and current sources. If the following conditions are satisfied, the circuit has exactly one solutions*:

- *Norators*, I-*resistors, and voltage sources together form a tree.*
- *Nullators*, I-*resistors, and voltage sources together form a tree.*
- *Resistors have no uniform partial orientation, except for the trivial case, in which no resistor is oriented.*

We illustrate the conditions of this theorem with the example of Figure 7.10. In Figure 7.13 the resistors are specified as *V*- and *I*-resistors and a uniform orientation of the resistors is indicated. Note that the nonlinear resistor is a *V*-resistor, but not an *I*-resistor, because its current saturates. The linear resistors, however, are both *V*- and *I*-resistors. The choice in Figure 7.13 is made in order to satisfy the first two conditions of Theorem 7.5. Correspondingly, in Figures 7.14 and 7.15 the norator–*I*-resistor–voltage source tree and the nullator–*I*-resistor–voltage source tree are represented. Because the third condition is not satisfied, Theorem 7.5 cannot guarantee a unique solution. Indeed, as explained earlier, this circuit may have three solutions.

Theorem 7.5 has been generalized to controlled sources, to resistors that are increasing but neither voltage nor current controlled (e.g., the ideal diode), and to resistors that are decreasing instead of increasing [9].

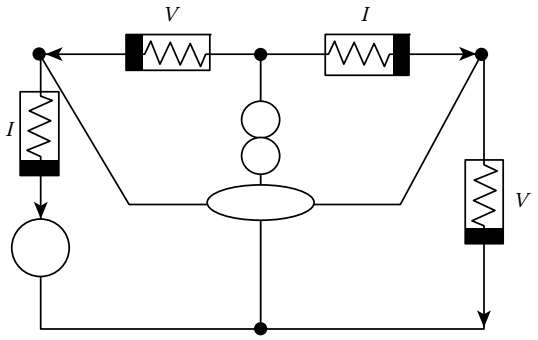


FIGURE 7.13 Circuit of Figure 7.10 with nullator and norator.

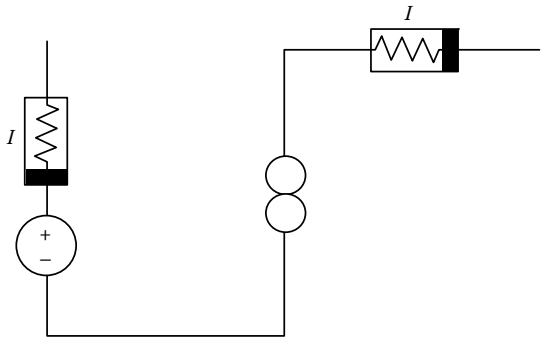


FIGURE 7.14 Norator–*I*-resistor–voltage source tree.

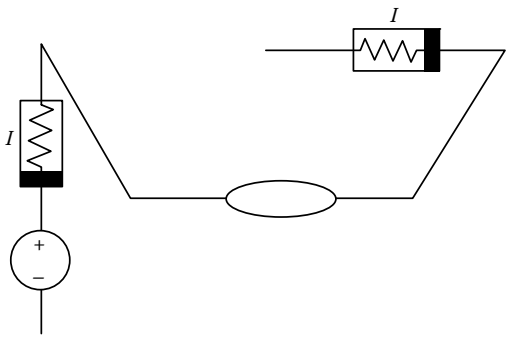


FIGURE 7.15 Nullator–*I*-resistor–voltage source tree.

Theorems 7.3, 7.4, and 7.5 have common features. Their conditions concern the circuit structure—the circuit graph that expresses the interconnection of the elements and the type of elements that occupy the branches of the graph, but not the element values. Therefore, the theorems guarantee the existence and uniqueness of the solution for whole classes of circuits, in which the individual circuits differ by their element values and parameters. In this sense the conditions are not only sufficient, but also necessary. This means, for example, in the case of Theorem 7.5 if all circuits with the same structure have exactly one solution, then the three conditions must be satisfied. However, by logical contraposition, if one of the three conditions is not satisfied for a given circuit structure, a circuit with this structure exists which has either no solution or more than one solutions.

On the other hand, if we consider a specific circuit, the conditions are only sufficient. They permit us to prove that the solution exists and is unique, but some circuits do not satisfy the conditions and still have exactly one solution. However, if the parameters of such a circuit are varied, one eventually falls onto a circuit with no solution or more than one solution.

The main conditions of Theorems 7.3 and 7.4 have an evident intuitive meaning. The orientations to look for in Theorem 7.5 are linked to the sign of the currents and the voltages of the difference of two solutions. Because the resistors are increasing, these signs are the same for the voltage and current differences. If we extend the analysis of the signs of solutions or solution differences to other elements, we must differentiate between voltages and currents. This approach, in which two orientations for all branches are considered, one corresponding to the currents and one corresponding to the voltages, is pursued in Ref. [10].

The conditions of Theorems 7.3 through 7.5 can be verified by inspection for small circuits. For larger circuits, one must resort to combinatorial algorithms. Such algorithms are proposed in Refs. [11,12]. As can be expected from the nature of conditions, the algorithms grow exponentially with the number of resistors. It is not known whether algorithms of polynomial complexity exist.

Some circuits always have either no solution or an infinite number of solutions, irrespective of the element and parameter values. Figure 7.6 gives the simplest example. Such circuits clearly are not very useful in practice. The remaining circuits are those that may have a finite number $n > 1$ of solutions if the circuit parameters are chosen suitably. These are the circuits that are useful for static memories and for multivibrators in general. This class is characterized by the following theorem.

THEOREM 7.6

Let circuit be composed of nullator–norator pairs, V- and I-resistors, and independent voltage and current sources. If the following three conditions are satisfied, the circuit has more than one, but a finite number of solutions for a suitable choice of circuit parameters:

- *Norators, I-resistors, and voltage sources together form a tree.*
- *Nullators, I-resistors, and voltage sources together form a tree.*
- *A nontrivial, uniform partial orientation of the resistors occurs.*

Can we be more precise and formulate conditions on the circuit structure that guarantee four solutions, for example? This is not possible because changing the parameters of the circuit will lead to another number of solutions. Particularly with a circuit structure that satisfies the conditions of Theorem 7.6, there is a linear circuit that always has an infinite number of solutions. If we are more restrictive on the resistor characteristics, e.g., imposing convex or concave characteristics for certain resistors, it is possible to determine the maximum number of solutions. A method to determine an upper bound is given in Ref. [14], whereas the results of Ref. [15] allow us to determine the actual maximum number under certain conditions. Despite these results, however, the maximum number of solutions is still an open problem.

7.2.2 Bounds on Voltages and Currents

It is common sense for electrical engineers that in an electronic circuit all node voltages lie between zero and the power supply voltage, or between the positive and the negative power supply voltages, if both are present. Actually, this is only true for the DC-operating point, but can we prove it in this case? The following theorems give the answer. They are based on the notion of passivity.

Definition 7.5: A resistor is passive if it can only absorb, but never produce power. This means that for any point (v, i) on its characteristic we have

$$v \cdot i \geq 0 \tag{7.24}$$

A resistor is strictly passive, if in addition to Equation 7.24 it satisfies the condition

$$v \cdot i = 0 \rightarrow v = i = 0 \tag{7.25}$$

THEOREM 7.7

Let a circuit be composed of strictly passive resistors and independent voltage and current sources. Then, for every branch k of the circuit the following bounds can be given:

$$|v_k| \leq \sum_{\text{source branches } j} |v_j| \tag{7.26}$$

$$|i_k| \leq \sum_{\text{source branches } j} |i_j| \tag{7.27}$$

If, in addition, the circuit is connected and all sources have a common node, the ground node, then the maximum and the minimum node voltage are at a source terminal.

The theorem implies in particular that in a circuit with a single voltage source, all branch voltages are bounded by the source voltage in magnitude, and all node voltages lie between zero and the source voltage. Similarly, if a circuit has a single current source, all branch currents are bounded by the source current in magnitude. Finally, if several voltage sources are present that are all connected to ground and have positive value, then the node voltages lie between zero and the maximum source voltage. If some sources have positive values and others have negative values, then all node voltages lie between the maximum and the minimum source values.

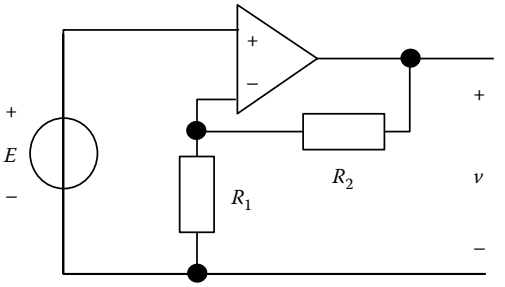


FIGURE 7.16 Voltage amplifier.

This theorem and various generalizations can be found in Ref. [1]. The main drawback is that it does not admit nonreciprocal elements. A simple counterexample is the voltage amplifier of Figure 7.16. The voltage of the output node of the operational amplifier is

$$v = \frac{R_1 + R_2}{R_1} E \tag{7.28}$$

Thus, the output node voltage is higher than the source voltage. Of course, the reason is that the operational amplifier is an active element. It is realized by transistors and needs a positive and a negative voltage source as the power supply. The output voltage of the operational amplifier cannot exceed these supply voltages. This fact is not contained in the model of the ideal operational amplifier, but follows from the extension of Theorem 7.7 to bipolar transistors [1,16].

THEOREM 7.8

Let a circuit be composed of bipolar transistors modeled by the Ebers–Moll equations, of strictly passive resistors, and of independent voltage and current sources. Then, the conclusion of Theorem 7.7 hold.

At first glance, Theorem 7.8 appears to imply that it is impossible to build an amplifier with bipolar transistors. Indeed, it is impossible to build such an amplifier with a single source, the input signal. We need at least one power supply source that sets the limits of dynamic range of the voltages according to Theorem 7.8. The signal source necessarily has a smaller amplitude and the signal can be amplified roughly up to the limit set by the power supply source.

Theorem 7.8 can be extended to MOS transistors. The difficulty is that the nonlinear characteristics of the simplest model is not strictly increasing, and therefore some interconnection condition must be added to avoid parts with undetermined node voltages.

7.2.3 Monotonic Dependence

Instead of looking at single solutions of resistive circuits, as done earlier in the chapter, we consider here a solution as a function of a parameter. The simplest and at the same time the most important case is the dependence of a solution on the value of a voltage or current source. To have a well-defined situation, we suppose that the circuit satisfies the hypotheses of Theorem 7.5. In this case [1,8], the solution is a continuous function of the source values.

As an example, let us consider the circuit of Figure 7.17. We are interested in the dependence of the various currents on the source voltage E. Because the circuit contains only strictly increasing resistors, we

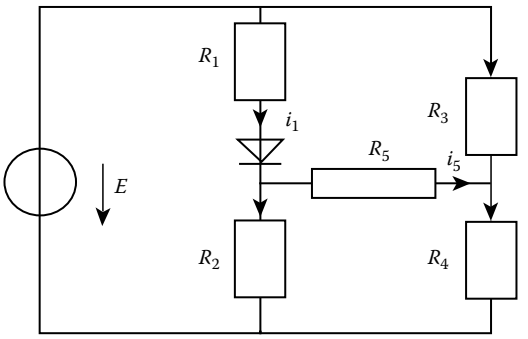


FIGURE 7.17 Circuit example for source dependence.

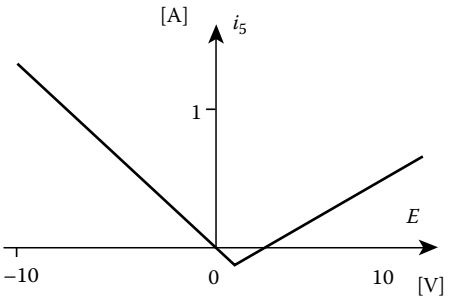


FIGURE 7.18 Nonmonotonic dependence.

expect all currents to be strictly monotonic functions of E. This is not true. In Figure 7.18, the current $i_5(E)$ is represented for $R_1 = R_2 = R_3 = 2R_4 = R_5 = 1\ \Omega$ and for standard diode model parameters. Clearly, it is nonmonotonic.

7.3 Autonomous Dynamic Circuits

7.3.1 Introduction

This section adds to the resistive elements of Section 7.2—the capacitors and the inductors. A nonlinear capacitor is defined by the constitutive relation

$$v = h(q) \tag{7.29}$$

where the auxiliary variable q is the charge of the capacitor, which is linked to the current by

$$i = \frac{\mathrm{d}q}{\mathrm{d}t} \tag{7.30}$$

The dual element, the nonlinear inductor, is defined by

$$i = g(\varphi) \tag{7.31}$$

where the auxiliary variable φ, the flux, is linked to the voltage by

$$v = \frac{\mathrm{d}\varphi}{\mathrm{d}t} \tag{7.32}$$

The symbols of these two elements are represented in Figure 7.19.

The system of equations that describes an autonomous dynamic circuit is composed of Equations 7.12 through 7.17, completed with Equations 7.29 and 7.30 for capacitor branches and Equations 7.31 and 7.32 for inductor branches. Hence, it becomes a mixed differential–nondifferential system of equations. Its solutions are the voltages, currents, charges, and fluxes as functions of time. Because it contains differential equations, we have infinitely many solutions, each one determined by some set of initial conditions.

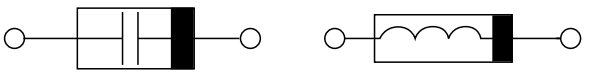


FIGURE 7.19 Symbols of the nonlinear capacitor and the nonlinear inductor.

If all variables except the charges and fluxes are eliminated from the system of equations, one obtains a reduced, purely differential system of equations

$$\frac{d\mathbf{q}}{dt} = \mathbf{f}(\mathbf{q},\varphi) \tag{7.33}$$

$$\frac{d\varphi}{dt} = \mathbf{g}(\mathbf{q},\varphi) \tag{7.34}$$

where $\mathbf{q}$ and φ are the vectors composed of, respectively, the capacitor charges and the inductor fluxes. These are the state equations of the circuit. Under mild assumptions on the characteristics of the nonlinear elements (local Lipschitz continuity and eventual passivity), it can be shown that the solutions are uniquely determined by the initial values of the charges and fluxes at some time t_0, $\mathbf{q}(t_0)$, and $\varphi(t_0)$, and that they exist for all times $t_0 \leq t < \infty$ [1,17].

It cannot be taken for granted, however, that the circuit equations actually can be reduced to that state Equations 7.33 and 7.34. On the one hand, the charges and fluxes may be dependent and thus their initial values cannot be chosen freely. However, the state equations may still exist, in terms of a subset of charges and fluxes. This means that only these charges and fluxes can be chosen independently as initial conditions. On the other hand, the reduction, even to some alternative set of state variables, may be simply impossible. This situation is likely to lead to impasse points, i.e., nonexistence of the solution at a finite time. We refer the reader to the discussion in Ref. [1]. In the sequel we suppose that the solutions exist from the initial time t_0 to $+\infty$ and that they are determined by the charges and fluxes at t_0.

We are interested in the asymptotic behavior, i.e., the behavior of the solutions when the time t goes to infinity. If the dynamic circuit is linear and strictly stable, i.e., if all its natural frequencies are in the open left half of the complex plane, then all solutions converge to one and the same DC-operating (equilibrium) point. This property still holds for many nonlinear circuits, but not for all by far. In particular, the solutions may converge to different DC-operating points, depending on the initial conditions (static memories), they may converge to periodic solutions (free-running oscillators), or they may even show chaotic behavior (e.g., Chua's circuit). Here, we give conditions that guarantee the solutions converge to a unique solution or one among several DC-operating points.

7.3.2 Convergence to DC-Operating Points

The methods to prove convergence to one or more DC-operating points is based on Lyapunov functions. A Lyapunov function is a continuously differentiable function $W(\xi)$, where ξ is the vector composed of the circuit variables (the voltages, currents, charges, and fluxes). In the case of autonomous circuits, a Lyapunov function must have the following properties:

1. W is bounded below, i.e., there exists a constant W_0 such that

$$W(\xi) \geq W_0 \quad \text{for all } \xi \tag{7.35}$$

2. The set of voltages, currents, charges, and fluxes of the circuit such that $W(\xi) \leq E$ is bounded for any real E.
3. For any solution $\xi(t)$ of the circuit

$$\frac{d}{dt} W(\xi(t)) \leq 0 \tag{7.36}$$

4. If

$$\frac{d}{dt} W(\xi(t)) = 0 \tag{7.37}$$

then $\xi(t)$ is a DC-operating point.

If an autonomous circuit has a Lyapunov function and if it has at least one, but a finite number of DC-operating points, then every solution converges to a DC-operating point. The reason is that the Lyapunov function must decrease along each solution, and thus must result in a local minimum, a stable DC-operating point. If more than one DC-operating point exists, it may, as a mathematical exception that cannot occur in practice, end up in a saddle point, i.e., an unstable DC-operating point.

The problem with the Lyapunov function method is that it gives no indication as to how to find such a function. Basically, three methods are available to deal with this problem:

1. Some standard candidates for Lyapunov functions, e.g., the stored energy.
2. Use a certain kind of function and adjust the parameters in order to satisfy points 2 and 3 in the previous list. Often, quadratic functions are used.
3. Use an algorithm to generate Lyapunov functions [18–20].

The following theorems were obtained via approach 1, and we indicate which Lyapunov function was used to prove them. At first glance, this may seem irrelevant from an engineering point of view. However, if we are interested in designing circuits to solve optimization problems, we are likely to be interested in Lyapunov functions. Indeed, as mentioned previously, along any solution of the circuit, the Lyapunov function decreases and approaches a minimum of the function. Thus, the dynamics of the circuit solve a minimization problem. In this case, we look for a circuit with a given Lyapunov function, however, usually we look for a Lyapunov function for a given circuit.

THEOREM 7.9

Let a circuit be composed of capacitors and inductors with a strictly increasing characteristic, resistors with a strictly increasing characteristic, and independent voltage and current sources. Suppose the circuit has a DC-operating point $\bar{\xi}$. *By Theorem 7.1, this DC-operating point is unique. Finally, suppose the circuit has no loop composed of capacitors, inductors, and voltage sources and no cutset composed of capacitors, inductors, and current sources. Then, all solutions of the circuit converge to* $\bar{\xi}$.

The Lyapunov function of this circuit is given by a variant of the stored energy in the capacitors and the resistors, the stored energy with respect to $\bar{\xi}$ [1,17]. If the constitutive relations of the capacitors and the inductors are given by $v_k = h_k(q_k)$ and $i_k = g_k(v_k)$, respectively, then this Lyapunov function becomes

$$W(\xi) = \sum_{\substack{\text{capacitor}\\ \text{branches } k}} \int_{\bar{q}k}^{qk} (h_k(q) - h_k(\bar{q}_k))\mathrm{d}q + \sum_{\substack{\text{inductor}\\ \text{branches } k}} \int_{\bar{\varphi}_k}^{\varphi_k} (g_k(\varphi) - g_k(\bar{\varphi}_k))\mathrm{d}\varphi \tag{7.38}$$

The main condition (Equation 7.36) for a Lyapunov function follows from the fact that the derivative of the stored energy is the absorbed power, here in incremental form:

$$\frac{\mathrm{d}}{\mathrm{d}t} W(\xi) = \sum_{\substack{\text{capacitor}\\ \text{and inductor}\\ \text{branches } k}} \Delta v_k \Delta i_k = -\sum_{\substack{\text{resistor}\\ \text{branches } k}} \Delta v_k \Delta i_k \leq 0 \tag{7.39}$$

Various generalizations of Theorem 7.9 have been given. The condition ''strictly increasing resistor characteristic'' has been relaxed to a condition that depends on $\bar{\xi}$ in Refs. [1,17] and mutual inductances and capacitances have been admitted in Ref. [17].

Theorem 7.10 admits resistors with nonmonotonic characteristics. However, it does not allow for both inductors and capacitors.

THEOREM 7.10

Let a circuit be composed of capacitors with a strictly increasing characteristic, voltage-controlled resistors such that

$$v \to +\infty \Rightarrow i > I_+ > 0 \quad \text{and} \quad v \to -\infty \Rightarrow i < I_- < 0 \tag{7.40}$$

and independent voltage sources. Furthermore, suppose that the circuit has a finite number of DC-operating points. Then every solution of the circuit converges toward a DC-operating point.

This theorem is based on the then following Lyapunov function, called cocontent:

$$W(\xi(t)) = \sum_{\substack{\text{resistor} \\ \text{branches } k}} \int_o^{v_k} g_k(v)\mathrm{d}v \tag{7.41}$$

where $i_k = g_k(v_k)$ is the constitutive relation of the resistor on branch k. The function W is decreasing along a solution of the circuit because

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} W(\xi(t)) &= \sum_{\substack{\text{resistor} \\ \text{branches } k}} \frac{\mathrm{d}v_k}{\mathrm{d}t} i_k = -\sum_{\substack{\text{capacitor} \\ \text{branches } k}} \frac{\mathrm{d}v_k}{\mathrm{d}t} i_k \\ &= -\sum_{\substack{\text{capacitor} \\ \text{branches } k}} \frac{\mathrm{d}h_k}{\mathrm{d}q} i_k^2 \le 0 \end{aligned} \tag{7.42}$$

where $h_k(q_k)$ is the constitutive relation of the capacitor on branch k.

Theorem 7.10 has a dual version. It admits inductors instead of capacitors, current-controlled resistors, and current sources. The corresponding Lyapunov function is the content:

$$W(\xi) = \sum_{\substack{\text{resistor} \\ \text{branches } k}} \int_o^{i_k} h_k(i)\mathrm{d}i \tag{7.43}$$

where $v_k = h_k(i_k)$ is the constitutive relation of the resistor on branch k.

The main drawback of the two preceding theorems is that they do not admit nonreciprocal elements such as controlled sources, operational amplifiers, etc. In other words, no statement about the analog neural network of Figure 7.20 can be made. In this network the nonreciprocal element is the VCVS with the nonlinear characteristics $v_2 = \sigma(v_1)$. However, Theorem 7.10 can be generalized to a reciprocal voltage-controlled N-port resistor closed on capacitors and voltage sources. Such an N-port (Figure 7.21) is described by a constitutive relation of the form

$$i_k = g_k(v_1, \ldots, v_N) \tag{7.44}$$

and it is reciprocal, if for all $\mathbf{v}$, and all k, j we have

$$\frac{\partial g_k}{\partial v_j}(\mathbf{v}) = \frac{\partial g_j}{\partial v_k}(\mathbf{v}) \tag{7.45}$$

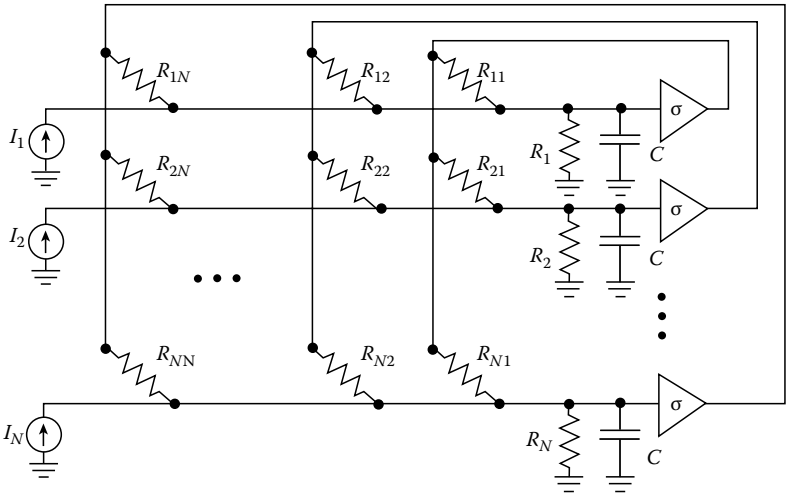


FIGURE 7.20 Analog neural network.

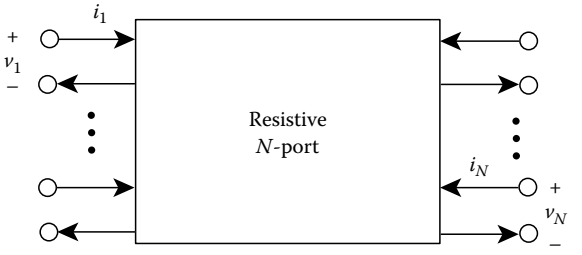


FIGURE 7.21 Resistive N-port.

THEOREM 7.11

Let a circuit be composed of charge-controlled capacitors with a strictly increasing characteristic and independent voltage sources that terminate a reciprocal voltage-controlled N-port with constitutive relation (*Equation 7.42*) *so that we find constants V and* $P > 0$ *such that*

$$\|\mathbf{v}\| \geq V \Rightarrow \mathbf{g} \cdot \mathbf{v} = \sum_{k=1}^{N} g_k(\mathbf{v}) v_k \geq P \tag{7.46}$$

If the number of DC-operating points is finite, then all solutions converge toward a DC-operating point.

The proof of this theorem is based on the Lyapunov function $W(\mathbf{v})$ that satisfies

$$\frac{\partial W}{\partial v_k}(\mathbf{v}) = g_k(\mathbf{v}) \tag{7.47}$$

Thanks to Equation 7.45, function W exists. The first two conditions for a Lyapunov function are a consequence of Equation 7.46. Finally

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}W(\xi(t)) &= \sum_{\substack{\text{resistor}\\ \text{branches}k}} g_k(\mathbf{v})\frac{\mathrm{d}v_k}{\mathrm{d}t}\\ &= \sum_{\substack{\text{resistor}\\ \text{branches } k}} i_k\frac{\mathrm{d}v_k}{\mathrm{d}t}\\ &= -\sum_{\substack{\text{capacitor}\\ \text{branches } k}} \frac{\mathrm{d}h_k}{\mathrm{d}q}i_k^2 \le 0\end{aligned} \tag{7.48}$$

where $h_k(q_k)$ is the constitutive relation of the capacitor on branch k.

To illustrate how Theorem 7.11 can be applied when Theorem 7.10 fails, consider the analog neural network of Figure 7.20. If the capacitor voltages are denoted by u_i and the voltages at the output of the voltage sources by v_i, the state equations for the network of Figure 7.1 become

$$-C_i\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{u_i}{R_i} + \sum_{j=1}^{N}\frac{u_i - v_j}{R_{ij}} + I_i \tag{7.49}$$

Suppose that the nonlinear characteristic $\sigma(u)$ is invertible. The state equations can be written in terms of the voltages v_i:

$$-C\frac{\mathrm{d}\sigma^{-1}}{\mathrm{d}v}(v_i)\frac{\mathrm{d}v_i}{\mathrm{d}t} = G_i\sigma^{-1}(v_i) - \sum_{j=1}^{N}\frac{v_j}{R_{ij}} + I_i \tag{7.50}$$

where

$$G_i = \frac{1}{R_i} + \sum_{j=1}^{N}\frac{1}{R_{ij}} \tag{7.51}$$

Equations 7.40 can be reinterpreted as the equations of a resistive N-port with the constitutive relations

$$g_i(\mathbf{v}) = G_i\sigma^{-1}(v_i) - \sum_{j=1}^{N}\frac{v_j}{R_{ij}} + I_i \tag{7.52}$$

closed on nonlinear capacitors with the constitutive relation

$$v = \sigma\left(\frac{q}{C}\right) \tag{7.53}$$

If σ is a sigmoidal function, as is most often supposed in this context (i.e., a strictly increasing function with $s(u) \to \pm 1$ for $u \to \pm\infty$), then the capacitors have a strictly increasing characteristic, as required by Theorem 7.11. Furthermore, the resistive N-port is reciprocal if for $i \neq j$

$$\frac{\partial g_i}{\partial v_j} = -\frac{1}{R_{ij}} = \frac{\partial g_j}{\partial v_i} = -\frac{1}{R_{ji}} \tag{7.54}$$

In other words, if for all i, j

$$R_{ij} = R_{ji} \tag{7.55}$$

On the other hand, inequality Equation 7.46 must be modified because the sigmoids have values only in the interval $[-1, +1]$ and thus Equation 7.50 are defined only on the invariant bounded set $S = \{\mathbf{v} | -1 < v_i < +1\}$. Therefore, inequality Equation 7.50 must be satisfied for vectors $\mathbf{v}$ sufficiently close to the boundary of S. This is indeed the case, because $\sigma^{-1}(v) \to \pm\infty$ as $v \to \pm 1$, whereas the other terms of the right-hand side of Equation 7.52 remain bounded.

It follows that all solutions of the analog neural network of Figure 7.20 converge to a DC-operating point as $t \to \infty$, provided σ is a sigmoid function and the connection matrix R_{ij} (synaptic matrix) is symmetrical. The Lyapunov function can be given explicitly:

$$W(\mathbf{v}) = \sum_{i=1}^{N} G_i \int_0^{v_i} \sigma^{-1}(v)\mathrm{d}v - \frac{1}{2}\sum_{i,j=1}^{N} \frac{v_i v_j}{R_{ij}} + \sum_{i=1}^{N} v_i I_i \tag{7.56}$$

7.4 Nonautonomous Dynamic Circuits

7.4.1 Introduction

This section is a consideration of circuits that contain elements where constitutive relations depend explicitly on time. However, we limit time dependence to the independent sources. For most practical purposes, this is sufficient. A time-dependent voltage source has a constitutive relation

$$v = e(t) \tag{7.57}$$

and a time-dependent current source

$$i = e(t) \tag{7.58}$$

where $e(t)$ is a given function of time which we suppose here to be continuous. In information processing circuits, $e(t)$ represents a signal that is injected into the circuit, whereas in energy transmission circuits $e(t)$ usually is a sinusoidal or nearly sinusoidal function related to a generator.

The time-dependent sources may drive the voltages and the currents to infinity, even if they only inject bounded signals into the circuit. Therefore, the discussion begins with the conditions that guarantee the boundedness of the solutions.

7.4.2 Boundedness of the Solutions

In electronic circuits, even active elements become passive when the voltages and currents grow large. This is the reason that solutions remain bounded.

Definition 7.6: A resistor is eventually passive if, for sufficiently large voltages and/or currents, it can only absorb power. More precisely, eventual passivity means that constants V and I exist such that, for all points (v, i) on the resistor characteristic with $|v| > V$ or $|i| > I$, we have

$$v \cdot i \geq 0 \tag{7.59}$$

Note that sources are not eventually passive, but as soon as an internal resistance of a source is taken into account, the source becomes eventually passive. The notion of eventual passivity can be extended to time-varying resistors.

Definition 7.7: A time-varying resistor is eventually passive if constants V and I are independent of time and are such that all points (v, i), with $|v| > V$ or $|i| > I$ that at some time lie on the characteristic of the resistor, satisfy the passivity condition (Equation 7.59). According to this definition, time-dependent sources with internal resistance are eventually passive if the source signal remains bounded.

Eventual passivity allows us to deduce bounds for the solutions. These bounds are uniform in the sense that they do not depend on the particular solution. To be precise, this is true only asymptotically, as $t \to \infty$.

Definition 7.8: The solutions of a circuit are eventually uniformly bounded if there exist constants V, I, Q, and $\mathbf{\Phi}$ such that, for any solution there exists a time T such that for any $t > T$, the voltages $v_k(t)$ are bounded by V, the currents $i_k(t)$ are bounded by I, the charges $q_k(t)$ are bounded by Q, and the fluxes $\varphi_k(t)$ are bounded by $\mathbf{\Phi}$.

Another manner of expressing the same property is to say that an attracting domain exists in state space [1].

THEOREM 7.12

A circuit composed of eventually passive resistors with $v \cdot i \to +\infty$ as $|v| \to \infty$ or $|i| \to \infty$, capacitors with $v \to \pm\infty$ as $q \to \neq\infty$, and inductors with $i \to \pm\infty$ as $\varphi \to \infty$ has eventually uniformly bounded solutions if no loop or cutset exists without a resistor [1,17].

Again, this theorem is proved by using a Lyapunov function, namely the stored energy

$$W(\xi) = \sum_{\substack{\text{capacitor} \\ \text{branches } k}} \int_0^{q_k} h_k(q)\mathrm{d}q + \sum_{\substack{\text{capacitor} \\ \text{branches } k}} \int_0^{\varphi_k} g_k(\varphi)\mathrm{d}\varphi \tag{7.60}$$

Inequality Equation 7.36 holds only outside of a bounded domain.

7.4.3 Unique Asymptotic Behavior

In the presence of signals with complicated waveforms that are injected into a circuit, we cannot expect simple waveforms for the voltages and the currents, not even asymptotically, as $t \to \infty$. However, we can hope that two solutions, starting from different initial conditions, but subject to the same source, have the same steady-state behavior. The latter term needs a more formal definition.

Definition 7.9: A circuit has unique asymptotic behavior if the following two conditions are satisfied:

1. All solutions are bounded.
2. For any two solutions $\xi_1(t)$ and $\xi_2(t)$

$$\|\xi_1(t) - \xi_2(t)\| \to_{t\to\infty} 0 \tag{7.61}$$

In order to prove unique asymptotic behavior, it is necessary to extend the notion of the Lyapunov function [1]. This does not lead very far, but at least it permits us to prove the following theorem.

THEOREM 7.13

Suppose a circuit is composed of resistors with a strictly increasing characteristic such that $v \cdot i \to \infty$ *as* $|v| \to \infty$ *or* $|i| \to \infty$, *positive linear capacitors*, positive linear inductors, *time-depending voltage* (*current*) *sources with bounded voltage* (*current*) *and a positive resistor in series* (*parallel*). *If no loop or cutset is composed exclusively of capacitors and inductors, the circuit has unique asymptotic behavior* [1,17].

This theorem is unsatisfactory because linear reactances are required and real devices are never exactly linear. It has been shown that slight nonlinearities can be tolerated without losing the unique asymptotic behavior [21]. On the other hand, we cannot expect to get much stronger general results because nonautonomous nonlinear circuits may easily have multiple steady-state regimes and even more complicated dynamics, such as chaos, even if the characteristics of the nonlinear elements are all strictly increasing.

Another variant of Theorem 7.13 considers linear resistors and nonlinear reactances [17].

References

1. M. Hasler and J. Neirynck, *Nonlinear Circuits*, Boston, MA: Artech House, 1986.
2. L.O. Chua, C.A. Desoer, and E.S. Kuh, *Linear and Nonlinear Circuits*, Electrical and Electronic Engineering Series, Singapore: McGraw-Hill International Editors, 1987.
3. A.N. Willson, Ed., *Nonlinear Networks: Theory and Analysis*, New York: IEEE Press, 1974.
4. R.O. Nielsen and A.N. Willson, A fundamental result concerning the topology of transistor circuits with multiple equilibria, *Proc. IEEE*, 68, 196–208, 1980.
5. A.N. Willson, On the topology of FET circuits and the uniqueness of their dc operating points, *IEEE Trans. Circuits Syst*., 27, 1045–1051, 1980.
6. T. Nishi and L.O. Chua, Topological criteria for nonlinear resistive circuits containing controlled sources to have a unique solution, *IEEE Trans. Circuits Syst*., 31, 722–741, Aug. 1984.
7. T. Nishi and L.O. Chua, Nonlinear op-amp circuits: Existence and uniqueness of solution by inspection, *Int. J. Circuit Theory Appl*., 12, 145–173, 1984.
8. M. Hasler, Nonlinear nonreciprocal resistive circuits with a unique solution, *Int. J. Circuit Theory Appl*., 14, 237–262, 1986.
9. M. Fosséprez, *Topologie et Comportement des Circuits non Linéaires non Réciproques*, Lausanne: Presses Polytechnique Romands, 1989.
10. M. Hasler, On the solution of nonlinear resistive networks, *J. Commun*. (*Budapest, Hungary*), special issue on nonlinear circuits, July 1991.
11. T. Parker, M.P. Kennedy, Y. Lioa, and L.O. Chua, Qualitative analysis of nonlinear circuits using computers, *IEEE Trans. Circuits Syst*., 33, 794–804, 1986.
12. M. Fosséprez and M. Hasler, Algorithms for the qualitative analysis of nonlinear resistive circuits, *IEEE ISCAS Proc*., 2165–2168, May 1989.
13. M. Fosséprez and M. Hasler, Resistive circuit topologies that admit several solutions, *Int. J. Circuit Theory Appl*., 18, 625–638, Nov. 1990.
14. M. Fosséprez, M. Hasler, and C. Schnetzler, On the number of solutions of piecewise linear circuits, *IEEE Trans. Circuits Syst*., CAS-36, 393–402, March 1989.
15. T. Nishi and Y. Kawane, On the number of solutions of nonlinear resistive circuits, *IEEE Trans*., E74, 479–487, 1991.
16. A.N. Willson, The no-gain property for networks containing three-terminal elements, *IEEE Trans. Circuits Syst*., 22, 678–687, 1975.
17. L.O. Chua, Dynamic nonlinear networks: State of the art, *IEEE Trans. Circuits Syst*., 27, 1059–1087, 1980.

18. R.K. Brayton and C.H. Tong, Stability of dynamical systems, *IEEE Trans. Circuits Syst*., 26, 224–234, 1979.
19. R.K. Brayton and C.H. Tong, Constructive stability and asymptotic stability of dynamical systems, *IEEE Trans. Circuits Syst*., 27, 1121–1130, 1980.
20. L. Vandenberghe and S. Boyd, A polynomial-time algorithm for determining quadratic Lyapunov functions for nonlinear systems, *Proc. ECCTD*-93, 1065–1068, 1993.
21. M. Hasler and Ph. Verburgh, Uniqueness of the steady state for small source amplitudes in nonlinear nonautonomous circuits, *Int. J. Circuit Theory Appl*., 13, 3–17, 1985.

8

Synthesis and Design of Nonlinear Circuits

8.1 Introduction **8**-1
8.2 Approximation Issues **8**-3
Unidimensional Functions • Piecewise-Linear and Piecewise-Polynomial Approximants • Gaussian and Bell-Shaped Basis Functions • Multidimensional Functions
8.3 Aggregation, Scaling, and Transformation Circuits **8**-11
Transformation Circuits • Scaling and Aggregation Circuitry
8.4 Piecewise-Linear Circuitry **8**-18
Current Transfer Piecewise-Linear Circuitry • Transresistance Piecewise-Linear Circuitry • Piecewise-Linear Shaping of Voltage-to-Charge Transfer Characteristics
8.5 Polynomials, Rational, and Piecewise-Polynomial Functions **8**-22
Concepts and Techniques for Polynomic and Rational Functions • Multiplication Circuitry • Multipliers Based on Nonlinear Devices
8.6 Sigmoids, Bells, and Collective Computation Circuits **8**-29
Sigmoidal Characteristics • Bell-Like Shapes • Collective Computation Circuitry
8.7 Extension to Dynamic Systems **8**-33
Appendix A: Catalog of Primitives **8**-34
Appendix B: Value and Slope Hermite Basis Functions **8**-35
References **8**-35

Angel Rodríguez-Vázquez
University of Seville

Manuel Delgado-Restituto
National Center of Microelectronics

Jose L. Huertas
National Center of Microelectronics

F. Vidal
University of Málaga

8.1 Introduction

Nonlinear synthesis and design can be informally defined as a constructive procedure to interconnect components from a catalog of available primitives, and to assign values to their constitutive parameters to meet a specific nonlinear relationship among electrical variables. This relationship is represented as an implicit integrodifferential operator, although we primarily focus on the synthesis of *explicit algebraic* functions,

$$y = f(\mathbf{x}) \tag{8.1}$$

where

y is voltage or current
$f(\cdot)$ is a nonlinear real-valued function
$\mathbf{x}$ is a vector with components that include voltages and currents

This synthesis problem is found in two different circuit-related areas: device modeling [8,76] and analog computation [26]. The former uses ideal circuit elements as primitives to build computer models of real circuits and devices (see Chapter 7). The latter uses real circuit components, available either off the shelf or integrable in a given fabrication technology, to realize hardware for nonlinear signal processing tasks. We focus on this second area, and intend to outline systematic approaches to devise electronic function generators. Synthesis relies upon hierarchical decomposition, conceptually shown in Figure 8.1, which encompasses several subproblems listed from top to bottom:

- Realization of nonlinear operators (multiplication, division, squaring, square rooting, logarithms, exponentials, sign, absolute value, etc.) through the interconnection of primitive components (transistors, diodes, operational amplifiers, etc.)
- Realization of elementary functions (polynomials, truncated polynomials, Gaussian functions, etc.) as the interconnection of the circuit blocks devised to build nonlinear operators
- Approximation of the target as a combination of elementary functions and its realization as the interconnection of the circuit blocks associated with these functions

Figure 8.1 illustrates this hierarchical decomposition of the synthesis problem through an example in which the function is approximated as a linear combination of truncated polynomials [30], where realization involves analog multipliers, built by exploiting the nonlinearities of bipolar junction transistors (BJTs) [63]. Also note that the subproblems cited above are closely interrelated and, depending on the availability of primitives and the nature of the nonlinear function, some of these phases can be bypassed. For instance, a logarithmic function can be realized exactly using BJTs [63], but requires approximation if our catalog includes only field-effect transistors whose nonlinearities are polynomic [44].

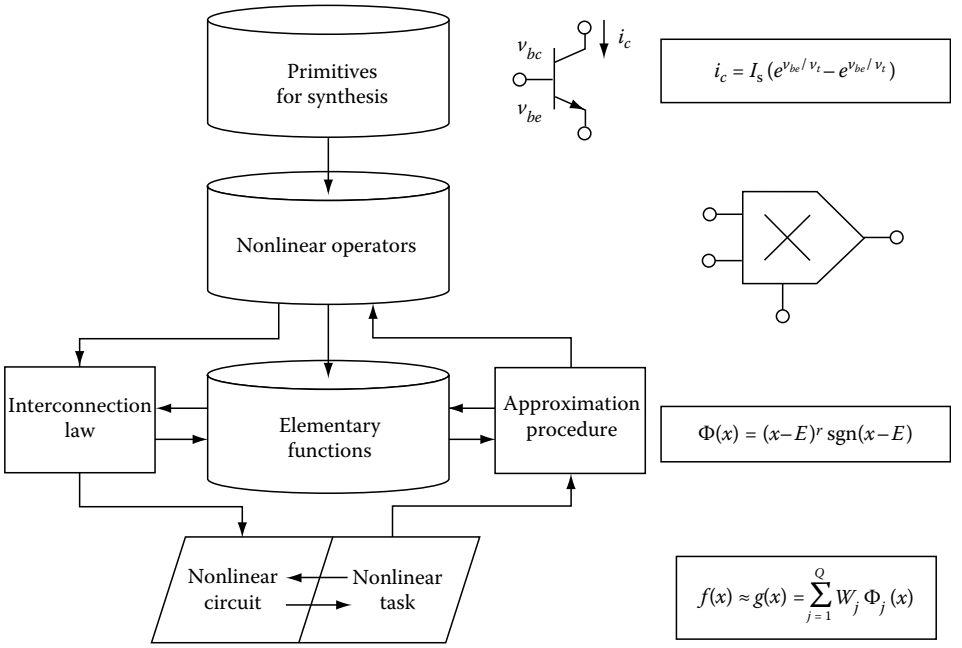


FIGURE 8.1 Hierarchical decomposition of the synthesis problem.

The technical literature contains excellent contributions to the solution of all these problems. These contributions can hardly be summarized or even quoted in just one section. Many authors follow a block-based approach which relies on the pervasive voltage operational amplifier (or op-amp), the rectification properties of junction diodes, and the availability of voltage multipliers, in the tradition of classical analog computation (e.g., Refs. [7,59,80]). Remarkable contributions have been made which focus on qualitative features such as negative resistance or hysteresis, rather than the realization of well-defined approximating functions [9,20,67]. Other contributions focus on the realization of nonlinear operators in the form of IC units. Translinear circuits, BJTs [23,62], and MOSFETs [79] are particularly well suited to realize algebraic functions in IC form. This IC orientation is shared by recent developments in analog VLSI computational and signal processing systems for neural networks [75], fuzzy logic [81], and other nonlinear signal processing paradigms [56,57,71].

This chapter is organized to fit the hierarchical approach in Figure 8.1. We review a wide range of approximation techniques and circuit design styles, for both discrete and monolithic circuits. It is based on the catalog of primitives shown in Appendix A. In addition to the classical op-amp-based continuous-time circuits, we include current-mode circuitry because nonlinear operators are realized simply and accurately by circuits that operate in *current domain* [23,57,62,79]. We also cover discrete-time circuits realized using analog dynamic techniques based on charge transfer, which is very significant for mixed-signal processing and computational microelectronic systems [27,72]. Section 8.2 is devoted to approximation issues and outlines different techniques for uni- and multidimensional functions, emphasizing hardware-oriented approaches. These techniques involve several nonlinear operators and the linear operations of scaling and aggregation (covered in Section 8.3, which also presents circuits to perform transformations among different kinds of characteristics). Sections 8.4 and 8.5 present circuits for piecewise-linear (PWL) and piecewise-polynomial (PWP) functions, Section 8.6 covers neural and fuzzy approximation techniques, and Section 8.7 outlines an extension to dynamic circuits.

8.2 Approximation Issues

8.2.1 Unidimensional Functions

Consider a target function, $f(x)$, given analytically or as a collection of measured data at discrete values of the independent variable. The approximation problem consists of finding a multiparameter function, $g(x, \mathbf{w})$, which yields proper fitting to the target, and implies solving two different subproblems: (1) which approximating functions to use, and (2) how to adjust the parameter vector, $\mathbf{w}$, to render optimum fitting. We only outline some issues related to this first point. Detailed coverage of both problems can be found in mathematics and optimization textbooks [73,78]. Other interesting views are found in circuit-related works [6,11,30], and the literature on neural and fuzzy networks [12,21,33,43,51].

An extended technique to design nonlinear electronic hardware for both discrete [63,80] and monolithic [35,62,79] design styles uses polynomial approximating functions,

$$g(x) = \sum_{j=0}^{Q} \alpha_j x^j \tag{8.2}$$

obtained through expansion by either Taylor series or orthogonal polynomials (Chebyshev, Legendre, or Laguerre) [26]. Other related approaches use rational functions,

$$g(x) = \frac{\sum_{j=0,Q} \alpha_j x^j}{\sum_{j=0,R} \beta_j x^j} \tag{8.3}$$

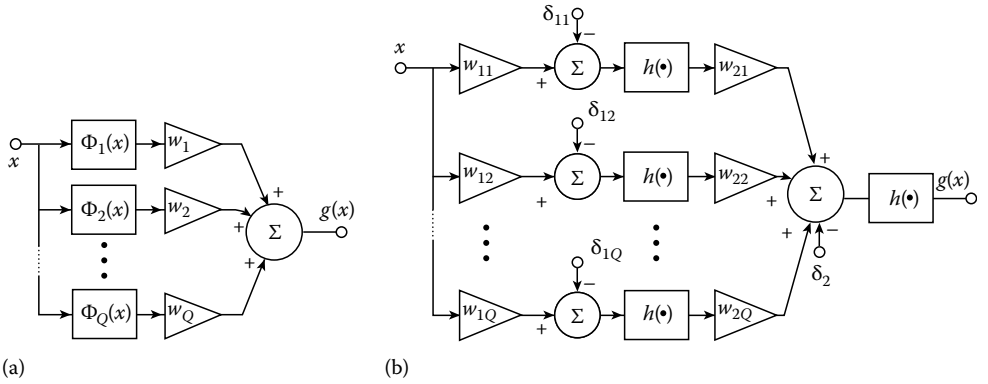


FIGURE 8.2 Block diagram for approximating function hardware. (a) Using linear combination of basis functions; (b) using two layers of nested sigmoids.

to improve accuracy in the approximation of certain classes of functions [14]. These can be realized by polynomial building blocks connected in feedback configuration [63]. In addition, Ref. [39] presents an elegant synthesis technique relying on linearly controlled resistors and conductors to take advantage of linear circuits synthesis methods (further extended in Ref. [28]).

From a more general point of view, hardware-oriented approximating functions can be classified into two major groups:

1. Those involving the linear combination of basis functions

$$g(x) = \sum_{j=1}^{Q} w_j \Phi_j(x) \tag{8.4}$$

   which include polynomial expansions. PWL and PWP interpolation and radial basis functions (RBF). The hardware for these functions consists of two layers, as shown in Figure 8.2a. The first layer contains Q nonlinear processing nodes to evaluate the basis functions; the second layer scales the output of these nodes and aggregates these scaled signals in a summing node.
2. Those involving a multilayer of nested sigmoids [51]; for instance, in the case of two layers [82],

$$g(x) = h\left[\left\{\sum_{j=1,Q} w_{2j} h\left(w_{1j}x - \delta_{1j}\right)\right\} - \delta_2\right] \tag{8.5}$$

   with the sigmoid function given by

$$h(x) = \frac{2}{1 + \exp(-\lambda x)} - 1 \tag{8.6}$$

   where $\lambda > 0$ determines the steepness of the sigmoid. Figure 8.2b shows a hardware concept for this approximating function, also consisting of two layers.

8.2.2 Piecewise-Linear and Piecewise-Polynomial Approximants

A drawback of polynomial and rational approximants is that their behavior in a small region determines their behavior in the whole region of interest [78]. Consequently, they are not appropriate to fit functions

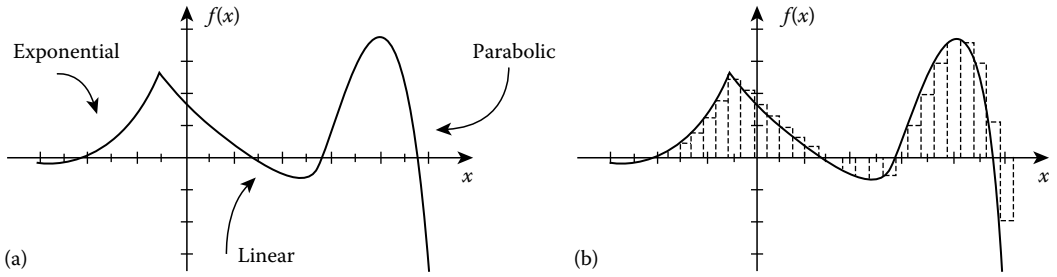


FIGURE 8.3 Example of nonuniform function. (a) A functions that is uniform throughout the whole region. (b) Interpolating the function by its samples associated to infinitely small subintervals.

that are uniform throughout the whole region (see Figure 8.3a). Another drawback is their lack of modularity, a consequence of the complicated dependence of each fitting parameter on multiple target data, which complicates the calculation of optimum parameter values. These drawbacks can be overcome by splitting the target definition interval into Q subintervals, and then expressing approximating function as a linear combination of basis functions, each having compact support over only one subinterval, i.e., zero value outside this subinterval. For the limiting case in which $Q \to \infty$, this corresponds to interpolating the function by its samples associated to infinitely small subintervals (Figure 8.3b). Such action is functionally equivalent to expressing a signal as its convolution with a delta of Dirac [10].

This splitting and subsequent approximation can be performed ad hoc, by using different functional dependences to fit each subregion. However, to support the systematic design of electronic hardware it is more convenient to rely on well-defined classes of approximating functions. In particular, Hermite PWPs provide large modularity by focusing on the interpolation of measured data taken from the target function. Any lack of flexibility as compared to the ad hoc approach may be absorbed in the splitting of the region.

Consider the more general case in which the function, $y = f(x)$, is defined inside a real interval $[\delta_0,\delta_{N+1}]$ and described as a collection of data measured at *knots* of a given interval partition, $\Delta = \{\delta_0, \delta_1, \delta_2, \ldots, \delta_N, \delta_{N+1}\}$. These data may include the function values at these points, as well as their derivatives, up to the $(M-1)$th order,

$$f^{(k)}(\delta_i) = \frac{\mathrm{d}^k}{\mathrm{d}x^k} f(x)\bigg|_{x=\delta_i} \qquad i = 0, 1, 2, \ldots, N, N+1 \tag{8.7}$$

where k denotes the order of the derivative and is zero for the function itself. These data can be interpolated by a linear combination of basis polynomials of degree $2M-1$,

$$g(x) = \sum_{i=0}^{N+1} \sum_{k=0}^{M-1} f^{(k)}(\delta_i)\Phi_{ik}(x) \tag{8.8}$$

where the expressions for these polynomials are derived from the interpolation data and continuity conditions [78]. Note that for a given basis function set and a given partition of the interval, each coefficient in Equation 8.8 corresponds to a single interpolation kust.

The simplest case uses linear basis functions to interpolate only the function values,

$$g(x) = \sum_{i=0}^{N+1} f(\delta_i) l_i(x) \tag{8.9}$$

with no function derivatives interpolated. Figure 8.4 shows the shape of the inner jth linear basis function, which equals 1 at δ_i and decreases to 0 at δ_{i-1} and δ_{i+1}. Figure 8.5a illustrates the representation

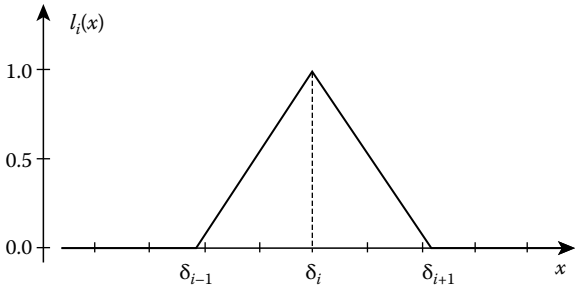


FIGURE 8.4 Hermite linear basis function.

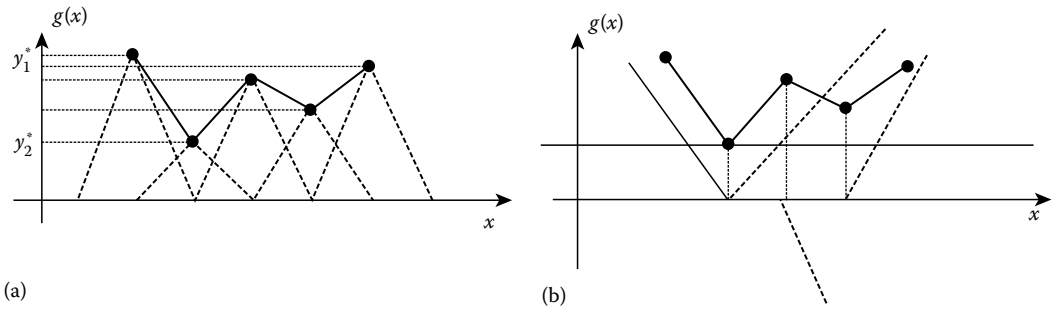


FIGURE 8.5 Decomposition of a PWL function using the extension operator. (a) Illustrating the representation in Equation 8.9. (b) Fitting some pieces from left to right and others from right to left.

in Equation 8.9. By increasing the degree of the polynomials, the function derivatives also can be interpolated. In particular, two sets of third-degree basis functions are needed to retain modularity in the interpolation of the function and its first derivative at the knots

$$g(x) = \sum_{i=0}^{N+1} f(\delta_i) v_i(x) + \sum_{i=0}^{N+1} f^{(1)}(\delta_i) s_i(x) \tag{8.10}$$

where Appendix B shows the shapes and expressions of the value, $v_i(x)$, and slope, $s_i(x)$, basis functions.

The modularity of Hermite polynomials is not free; their implementation is not cheapest in terms of components and, consequently, may not be optimal for application in which the target function is fixed. These applications are more conveniently handled by the so-called canonical representation of PWP functions. A key concept is the extension operator introduced in Ref. [6]; the basic idea behind this concept is to build the approximating function following an iterative procedure. At each iteration, the procedure starts from a function that fits the data on a subinterval, enclosing several pieces of the partition interval, and then adds new terms to also fit the data associated to the next piece. Generally, some pieces are fit from left to right and others from right to left, to yield

$$g(x) = g^0(x) + \sum_{i=1}^{N+} \Delta^+ g_i(x) + \sum_{i=-N_-}^{-1} \Delta^- g_i(x) \tag{8.11}$$

It is illustrated in Figure 8.5b. The functions in Equation 8.11 have the following general expressions

$$\begin{aligned}\Delta^{+}g(x) &= wu_{+}(x-\delta) \equiv w(x-\delta)\text{sgn}(x-\delta)\\ \Delta^{-}g(x) &= wu_{-}(x-\delta) \equiv w(x-\delta)\text{sgn}(\delta-x)\\ g^{0}(x) &= ax+b\end{aligned} \tag{8.12}$$

where sgn(·) denotes the *sign* function, defined as an application of the real axis onto the discrete set {0,1}.

This representation, based on the extension operator, is elaborated in Ref. [6] to obtain the following canonical representation for unidimensional PWL functions:

$$g(x) = ax + b + \sum_{i=1}^{N} w_i|x-\delta_i| \tag{8.13}$$

which has the remarkable feature of involving only one nonlinearity: the absolute value function.

The extension operator concept was applied in Ref. [30] to obtain canonical representations for cubic Hermite polynomials and B-splines. Consequently, it demonstrates that a PWP function admits a global expression consisting of a linear combination of powers of the input variable, plus truncated powers of shifted versions of this variable. For instance, the following expression is found for a cubic B-spline:

$$g(x) = \sum_{r=0}^{3} \alpha_r x^r + \sum_{i=1}^{N} \beta_i (x-\delta_i)^3 \text{sgn}(x-\delta_i) \tag{8.14}$$

with α_r and β_i obtainable through involved operations using the interpolation data. Other canonical PWP representations devised by these authors use

$$(x-\delta_i)^r \text{sgn}(x-\delta_i) = \frac{1}{2}\{|x-\delta_i| + (x-\delta_i)\}(x-\delta_i)^{r-1} \tag{8.15}$$

to involve the absolute value, instead of the sign function, in the expression of the function.

8.2.3 Gaussian and Bell-Shaped Basis Functions

The Gaussian basis function belongs to the general class of RBF [51,52], and has the following expression:

$$\Phi(x) = \exp\left(-\frac{(x-\delta)^2}{2\sigma^2}\right) \tag{8.16}$$

plotted in Figure 8.6. The function value is significant only for a small region of the real axis centered around its center, δ, and its shape is controlled by the *variance* parameter, σ^2. Thus, even though the

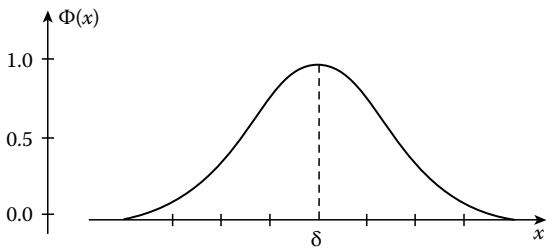


FIGURE 8.6 Guassian basis function.

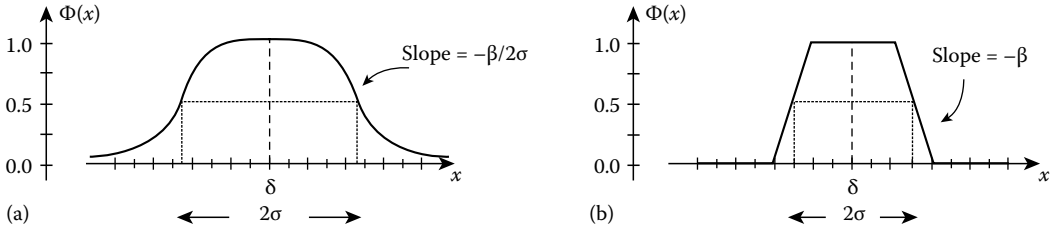


FIGURE 8.7 Fuzzy membership functions: (a) polynomial; (b) PWL.

support of Gaussian functions is not exactly compact, they are negligible except for well-defined local domains of the input values.

By linear combination of a proper number of Gaussians, and a proper choice of their centers and variances, as well as the weighting coefficients, it is possible to approximate nonlinear functions to any degree of accuracy [51]. Also, the local feature of these functions renders this adjustment process simpler than for multilayer networks composed of nested sigmoids, whose components are global [43,50].

A similar interpolation strategy arises in the framework of fuzzy reasoning, which is based on local membership functions whose shape resembles a Gaussian. For instance, in the ANFIS system proposed by Jang [33]

$$\Phi(x) = \frac{1}{1 + \left[\left(\frac{x-\delta}{\sigma}\right)^2\right]^{\beta}} \tag{8.17}$$

as plotted in Figure 8.7a where the shape is controlled by β and σ, and the position is controlled by δ. Other authors, for instance, Yamakawa [81], use the PWL membership function shape of Figure 8.7b, which is similar to the Hermite linear basis function of Figure 8.4. From a more general point of view, cubic B-splines [78] used to build hardware [59] and for device modeling [76] also can be considered to be members of this class of functions.

8.2.4 Multidimensional Functions

Approximation techniques for multidimensional functions are informally classified into five groups:

1. Sectionwise PWP functions [6,30]
2. Canonical PWL representations [11]
3. Neurofuzzy interpolation [33,81]
4. Radial basis functions [51,52]
5. Multilayers of nested sigmoids [82]

8.2.4.1 Sectionwise Piecewise-Polynomial Functions

This technique reduces the multidimensional function to a sum of products of functions of only one variable:

$$g(\mathbf{x}) = \sum_{k_1=1}^{M_1} \sum_{k_2=1}^{M_2} \cdots \sum_{k_P=1}^{M_P} \alpha(k_1, k_2, \ldots, k_P) \prod_{j=1}^{P} \Phi_{k_j}(x_j) \tag{8.18}$$

where $\alpha(k_1, k_2, \ldots, k_P)$ denotes a constant coefficient. These function representations were originally proposed by Chua and Kang for the PWL case [6] where

$$\Phi_1(x_j) = 1 \quad \Phi_2(x_j) = x_j \quad \Phi_3(x_j) = \left|x_j - \delta_{j1}\right| \cdots$$
$$\Phi_{M_P}(x_j) = \left|x_j - \delta_{jM_P-2}\right| \tag{8.19}$$

Similar to the unidimensional case, the only nonlinearity involved in these basis functions is the absolute value. However, multidimensional functions not only require weighted summations, but also multiplications. The extension of Equation 8.18 to PWP functions was covered in Ref. [30], and involves the same kind of nonlinearities as Equations 8.14 and 8.15.

8.2.4.2 Canonical Piecewise Linear Representations

The canonical PWL representation of Equation 8.13 can be extended to the multidimensional case, based on the following representation:

$$g(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{x} + b + \sum_{i=1}^{Q} c_i \left|\mathbf{w}_i^{\mathrm{T}}\mathbf{x} - \delta_i\right| \tag{8.20}$$

where

a and $\mathbf{w}_i$ are P-vectors
b, c_i, and δ_i are scalars

Q represents the number of hyperplanes that divide the whole space R^P into a finite number of polyhedral regions where $g(\cdot)$ can be expressed as an affine representation.

Note that Equation 8.20 avoids the use of multipliers. Thus, $g(\cdot)$ in Equation 8.20 can be realized through the block diagram of Figure 8.8, consisting of Q absolute value nonlinearities and weighted summers.

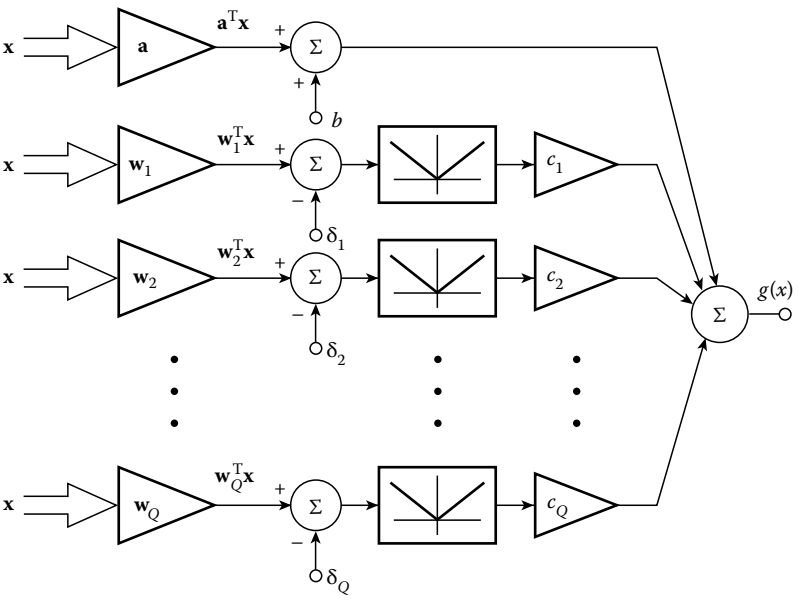


FIGURE 8.8 Canonical block diagram for a canonical PWL function.

8.2.4.3 Radial Basis Functions

The idea behind RBF expansion is to represent the function at each point of the input space as a linear combination of kernel functions whose arguments are the radial distance of the input point to a selected number of centers

$$g(\mathbf{x}) = \sum_{j=1}^{Q} w_j \Phi_j(\|\mathbf{x} - \boldsymbol{\delta}_j\|) \tag{8.21}$$

where $||\cdot||$ denotes a norm imposed on R^P, usually assumed Euclidean. The most common basis function is a Gaussian kernel similar to Equation 8.16,

$$\Phi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\delta}\|^2}{2\sigma^2}\right) \tag{8.22}$$

although many other alternatives are available [51], for instance,

$$\Phi(r) = (\sigma^2 + r^2)^{-\alpha}, \quad \Phi(r) = r, \quad \alpha \geq -1 \tag{8.23}$$

where r is the radial distance to the center of the basis function, $r \equiv ||\mathbf{x} - \delta||$. Micchelli [42] demonstrated that any function where the first derivative is monotonic qualifies as a RBF. As an example, as Equation 8.23 displays, the identity function $\Phi(r) = r$ falls into this category, which enables connecting the representation by RBF to the canonical PWL representation [40]. Figure 8.9 is a block diagram for the hardware realization of the RBF model.

8.2.4.4 Neurofuzzy Interpolation

This technique exploits the interpolation capabilities of fuzzy inference, and can be viewed as the multidimensional extension of the use of linear combination of bell-shaped basis functions to approximate nonlinear functions of a single variable (see Equations 8.4 and 8.17). Apart from its connection to approximate reasoning and artificial intelligence, this extension exhibits features similar to the sectionwise PWP representation, namely, it relies on a well-defined class of unidimensional functions. However, neurofuzzy interpolation may be advantageous for hardware implementation because it requires easy-to-build collective computation operators instead of multiplications.

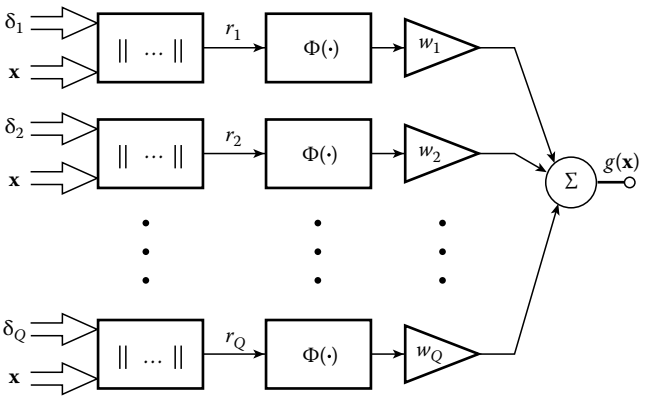


FIGURE 8.9 Concept of RBF hardware.

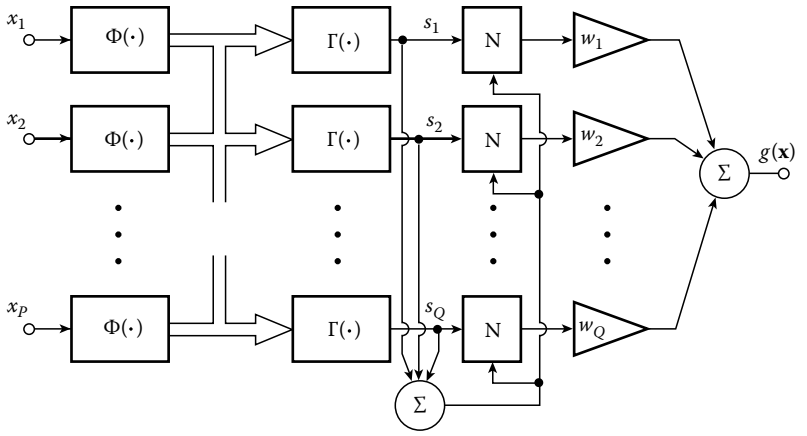


FIGURE 8.10 Conceptual architecture of a neurofuzzy interpolator.

Figure 8.10 depicts the block diagram of a neurofuzzy interpolator for the simplest case in which inference is performed using the singleton algorithm [33] to obtain

$$g(\mathbf{x}) = \sum_{j=1}^{Q} w_j \frac{s_j(\mathbf{x})}{\sum\limits_{i=1,Q} s_i(\mathbf{x})} \tag{8.24}$$

where the functions $s_i(\mathbf{x})$, called activities of the fuzzy rules, are given as

$$s_j(\mathbf{x}) = \Gamma\left\{\Phi_{j1}(x_1), \Phi_{j2}(x_2), \ldots, \Phi_{jP}(x_P)\right\} \tag{8.25}$$

where

$\Gamma(\cdot)$ is any T-norm operator, for instance, the *minimum*
$\Phi(\cdot)$ has a bell-like shape (see Figure 8.7)

8.2.4.5 Multilayer Perceptron

Similar to Equation 8.5, but consists of the more general case of several layers, with the input to each nonlinear block given as a linear combination of the multidimensional input vector [82].

8.3 Aggregation, Scaling, and Transformation Circuits

The mathematical techniques presented in Section 8.2 require several nonlinear operators and the linear operators of scaling and aggregation (covered for completeness in this section). This section also covers transformation circuits. This is because in many practical situations we aim to exploit some nonlinear mechanism which intrinsically involves a particular kind of characteristics. For instance, a MOS transistor has inherent square-law transconductance, while a diode exhibits an exponential driving-point. Similarly, many nonlinear operators are naturally realized in current-mode domain and involve currents at both the input and the output. Thus, transformation circuits are needed to exploit these mechanisms for other types of characteristics.

8.3.1 Transformation Circuits

Two basic problems encountered in the design of transformation circuits are how to convert a voltage node into a current node and vice versa. We know no unique way to realize these functions. Instead, there

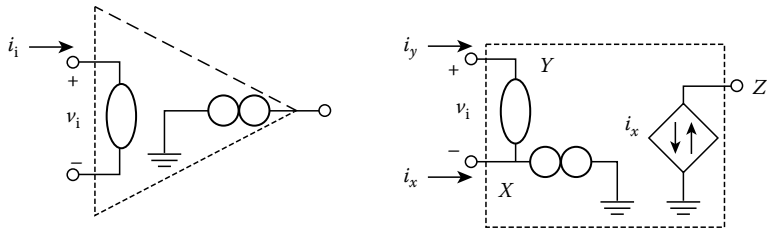


FIGURE 8.11 First-order models for voltage op-amps and CCIIs using nullators and norators.

are many alternatives which depend on which active component from Appendix A is used. The OTA can be represented to a first-order model as a voltage-controlled current source (VCCS) with linear transconductance parameter g_m. Regarding the op-amp and CCII, it is convenient to represent them by the first-order models of Figure 8.11, which contain nullators and norators.* A common appealing feature of both models is the virtual ground created by the input nullator. It enables us to sense the current drawn by nodes with fixed voltage—fully exploitable to design transformation circuits.

8.3.1.1 Voltage-to-Current Transformation

A straightforward technique for voltage-to-current conversion exploits the operation of the OTA as a VCCS (see Figure 8.12a) to obtain $i_0 = g_m v_i$, where g_m is the OTA transconductance parameter [22]. A drawback is that its operation is linear only over a limited range of the input voltage. Also, the scaling factor is inaccurate and strongly dependent on temperature and technology. Consequently,

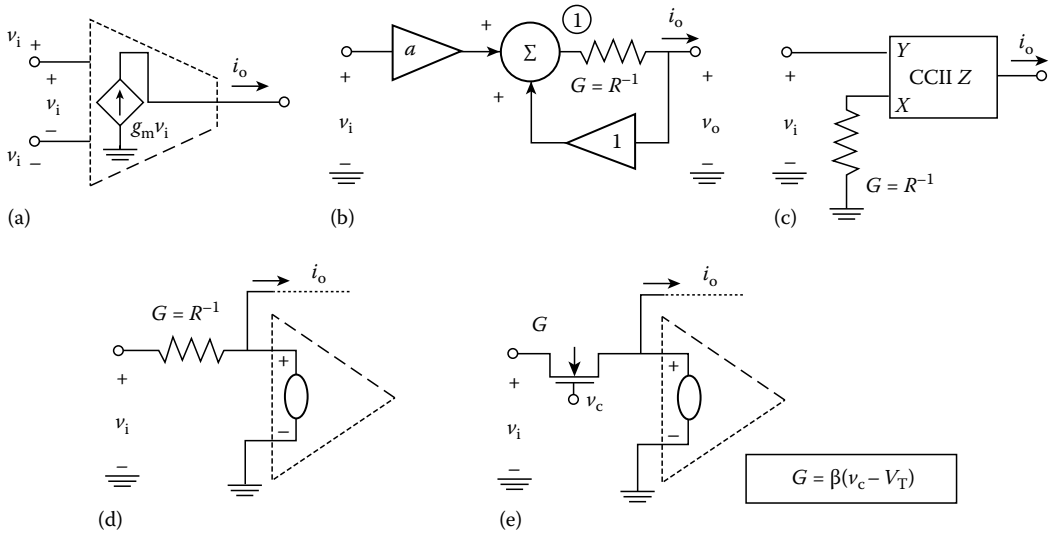


FIGURE 8.12 Voltage-to-current transformation: (a) using an OTA; (b) using voltage feedback; (c) using a current conveyor; (d) using virtual ground of an op-amp; (e) same as Figure 8.12d, but with active resistors.

* A nullator simultaneously yields a short circuit and an open circuit, while the voltage and the current at a norator are determined by the external circuitry. The use of a nullator to model the input port of an op amp is valid only if the component is embedded in a negative feedback configuration. With regard to the CCII, the required feedback is created by the internal circuitry.

voltage-to-current conversion using this approach requires circuit strategies to increase the OTA linear operation range [17,70], and tuning circuits to render the scaling parameter accurate and stable [70]. As counterparts, the value of the scaling factor is continuously adjustable through a bias voltage or current. Also, because the OTA operates in open loop, its operation speed is not restricted by feedback-induced pole displacements.

The use of feedback attenuates the linearity problem of Figure 8.12a by making the conversion rely on the constitutive equation of a passive resistor. Figure 8.12b illustrates a concept commonly found in op-amp-based voltage-mode circuits [29,59]. The idea is to make the voltage at node ① of the resistor change linearly with v_o, $v_1 = v_o + av_i$, and thus render the output current independent of v_o, to obtain $i_o = G(v_o + av_i - v_o) = aGv_i$. The summing node in Figure 8.12b is customarily realized using op-amps and resistors, which is very costly in the more general case in which the summing inputs have high impedance. The circuits of Figure 8.12c and d reduce this cost by direct exploitation of the virtual ground at the input of current conveyors (Figure 8.12c) and op-amps (Figure 8.12d). For both circuits, the virtual ground forces the input voltage v_i across the resistor. The resulting current is then sensed at the virtual ground node and routed to the output node of the conveyor, or made to circulate through the feedback circuitry of the op-amp, to obtain $i_o = Gv_i$.

Those implementations of Figure 8.12b through d that use off-the-shelf passive resistors overcome the accuracy problems of Figure 8.12a. However, the values of monolithic components are poorly controlled. Also, resistors may be problematic for standard VLSI technologies, where high-resistivity layers are not available and consequently, passive resistors occupy a large area. A common IC-oriented alternative uses the ohmic region of the MOS transistor to realize an active resistor [69] (Figure 8.12e). Tuning and linearity problems are similar to those for the OTA. Circuit strategies to overcome the latter are ground in Refs. [13,32,66,69].

8.3.1.2 Current-to-Voltage Transformation

The most straightforward strategy consists of a single resistor to draw the input current. It may be passive (Figure 8.13a) or active (Figure 8.13b). Its drawback is that the node impedance coincides with the resistor value, and thus makes difficult impedance matching to driving and loading stages. These matching problems are overcome by Figure 8.13c, which obtains low impedances at both the input and the output ports. On the other hand, Figure 8.13d obtains low impedance at only the input terminal, but maintains the output impedance equal to the resistor value. All circuits in Figure 8.13 obtain $v_o = Ri_i$, where $R = g_m^{-1}$ for the OTA.

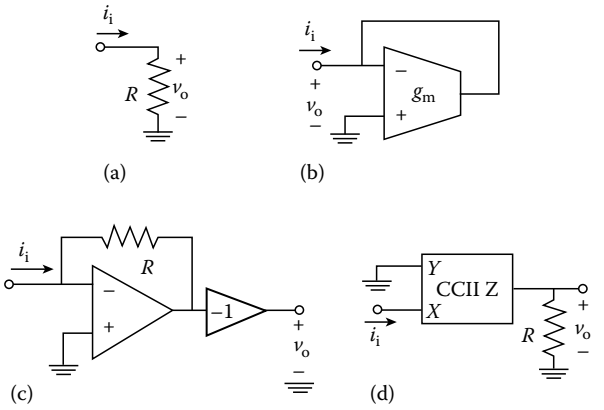


FIGURE 8.13 Current-to-voltage transformation: (a) using a resistor; (b) using a feedback OTA; (c) using op-amps; (d) using current conveyors.

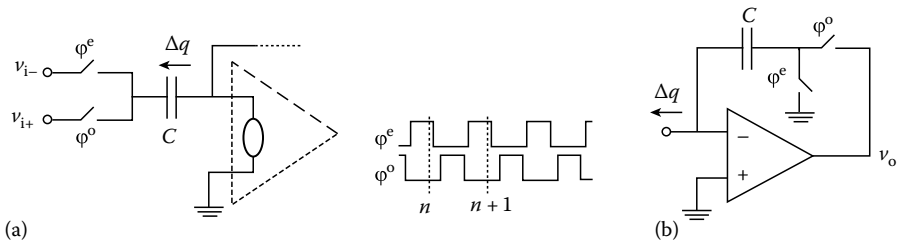


FIGURE 8.14 Transformations for sampled-data circuits: (a) voltage to charge; (b) charge to voltage.

8.3.1.3 Voltage/Charge Domain Transformations for Sampled-Data Circuits

The linearity and tuning problems of previous IC-related transformation approaches are overcome through the use of dynamic circuit design techniques based on switched-capacitors [72]. The price is that the operation is no longer asynchronous: relationships among variables are only valid for a discrete set of time instants. Variables involved are voltage and charge, instead of current, and the circuits use capacitors, switches, and op-amps.

Figure 8.14a is for voltage-to-charge transformation, while Figure 8.14b is for charge-to-voltage transformation. The switches in Figure 8.14a are controlled by nonoverlapping clock signals, so that the structure delivers the following incremental charge to the op-amp virtual ground node:

$$\Delta q^e = C(v_{i^+} - v_{i^-}) = -\Delta q^o \tag{8.26}$$

where the superscript denotes the clock phase during which the charge is delivered. Complementarily, the structure of Figure 8.14b initializes the capacitor during the even clock phase, and senses the incremental charge that circulates through the virtual ground of the op-amp during the odd clock phase. Thus, it obtains

$$v_0^o = C(\Delta q^o) \tag{8.27}$$

References [45,46,68] contain alternative circuits for the realization of the scaling function. Such circuits have superior performance in the presence of parasitics of actual monolithic op-amps and capacitors.

8.3.1.4 Transformation among Transfer Characteristics

Figure 8.15 depicts the general architecture needed to convert one kind of transfer characteristics, e.g., voltage transfer, into another, e.g., current transfer. Variables x' and y' of the original characteristics can be either voltage or current, and the same occurs for x and y of the converted characteristic. The figure depicts the more general case, which also involves a linear transformation of the characteristics themselves:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x' \\ y' \end{bmatrix} \tag{8.28}$$

For example, Figure 8.15 encloses the matrices to rotate the characteristics by an angle θ, and to reflect the characteristics with respect to an edge with angle θ. This concept of linear transformation converters and its applications in the synthesis of nonlinear networks was proposed initially by Chua [5] for driving-point characteristics, and further extended by Glover [24] and Huertas [29].

In the simplest case, in which the nondiagonal entries in Equation 8.28 are zero, the transformation performed over the characteristics is scaling, and the circuits of Figures 8.12 and 8.13 can be used directly

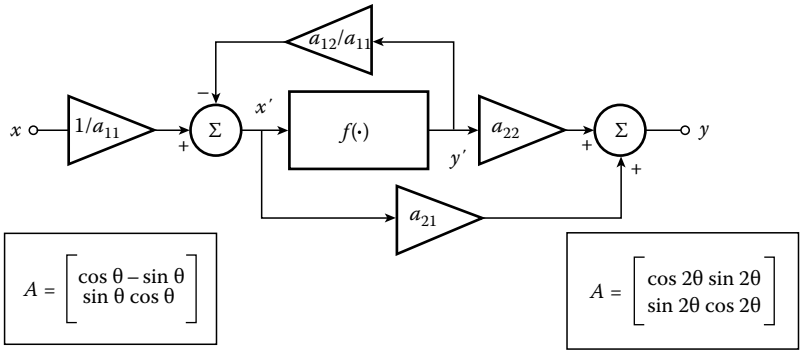


FIGURE 8.15 Concept of linear transformation converter for transfer characteristics: general architecture, and transformation matrices for rotation (left) and reflection (right).

to convert x into x' at the input front-end, and y' at the output front-end. Otherwise, aggregation operation is also required, which can be realized using the circuits described elsewhere.

8.3.1.5 From Driving Point to Transfer and Vice Versa

Figure 8.16 illustrates circuits to transform driving-point characteristics into related transfer characteristics. Figure 8.16a and b uses the same principle as Figure 8.12c and d to transform a voltage-controlled *driving-point* characteristic, $i_i = f(v_i)$, into a *transconductance* characteristics. On the other hand Figure 8.16c operates similarly to Figure 8.13c to transform a current-controlled driving-point characteristic, $v_i = f(i_i)$, into a transimpedance characteristic. If the resistance characteristics of the resistor in Figure 8.16a and b, or the conductance characteristic of the resistor in Figure 8.16c, is invertible, these circuits serve to invert nonlinear functions [63]. For instance, using a common base BJT in Figure 8.16c obtains a logarithmic function from the BJT exponential transconductance. Also, the use of a MOST operating in the ohmic region serves to realize a division operation.

Lastly, let us consider how to obtain driving-point characteristics from related transfer characteristics. Figure 8.17a and b corresponds to the common situation found in op-amp-based circuits, where the transfer is between voltages. Figure 8.17a is for the voltage-controlled case and Figure 8.17b is for the current-controlled case. They use feedback strategies similar to Figure 8.17b to render either the input voltage or the input current independent of the linear contributions of the other port variable. A general theory for this kind of transformation converter can be found in Ref. [29].

Note that these figures rely on a Thévenin representation. Similar concepts based on Norton representations allow us to transform current transfer characteristics into driving-point characteristics. However, careful design is needed to preserve the input current while sensing it.

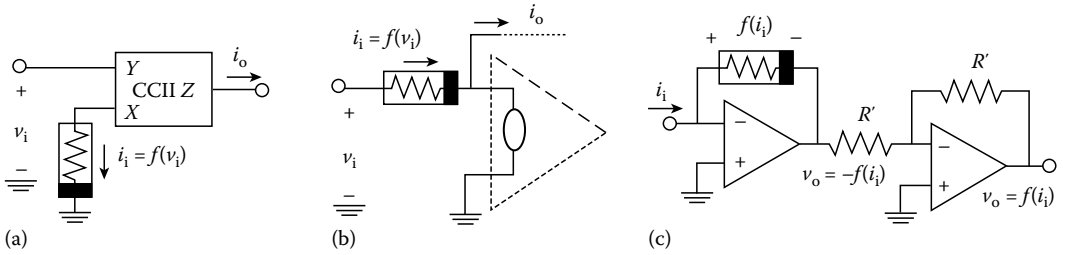


FIGURE 8.16 From driving point to transfer characteristics: (a) and (b) transconductance from voltage-controlled driving-point; (c) transimpedance from current-controlled driving-point.

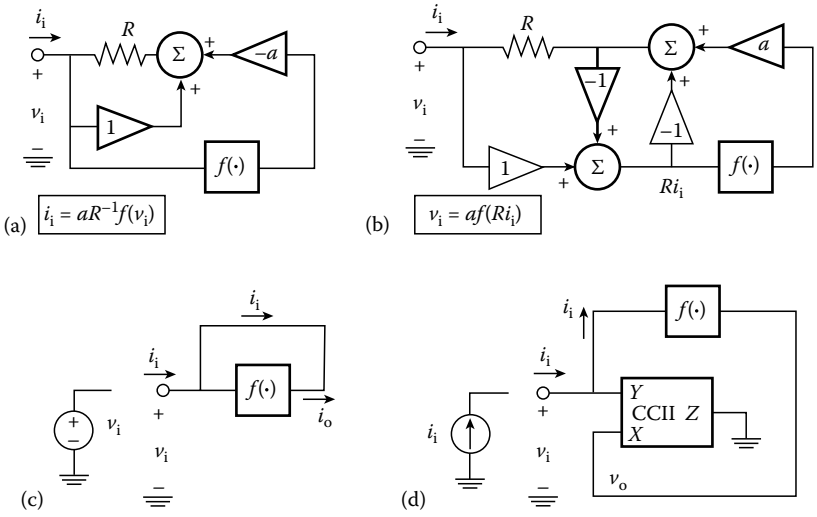


FIGURE 8.17 From transfer to driving-point characteristics. (a) Voltage-controlled case. (b) The current-controlled case. (c) A transconductor. (d) A current-controlled resistor.

Other interesting transformation circuits are depicted in Figure 8.17c and d. The block in Figure 8.17c is a transconductor that obtains $i_o = -f(v_i)$ with very large input impedance. Then, application of feedback around it obtains a voltage-controlled resistor, $i_o = f(v_i)$. Figure 8.17d obtains a current-controlled resistor, $v_i = f(i_i)$, using a current conveyor to sense the input current and feedback the output voltage of a transimpedance device with $v_o = f(i_i)$.

8.3.2 Scaling and Aggregation Circuitry

8.3.2.1 Scaling Operation

Whenever the weights are larger than unity, or are negatives, the operation of scaling requires active devices. Also, because any active device acts basically as a transconductor, the scaling of voltages is performed usually through the transformation of the input voltage into an intermediate current and the subsequent transformation of this current into the output voltage. Figure 8.18 illustrates this for an op-amp-based amplifier and an OTA-based amplifier. The input voltage is first scaled and transformed in i_o, and then this current is scaled again and transformed into the output voltage. Thus, the scaling factor depends on two design parameters. Extra control is achieved by also scaling the intermediate current.

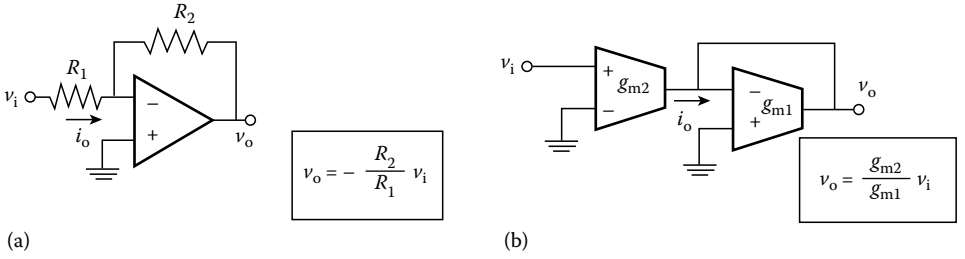


FIGURE 8.18 Mechanisms for voltage scaling. (a) An op-amp-based amplifier. (b) An OTA-based amplifier.

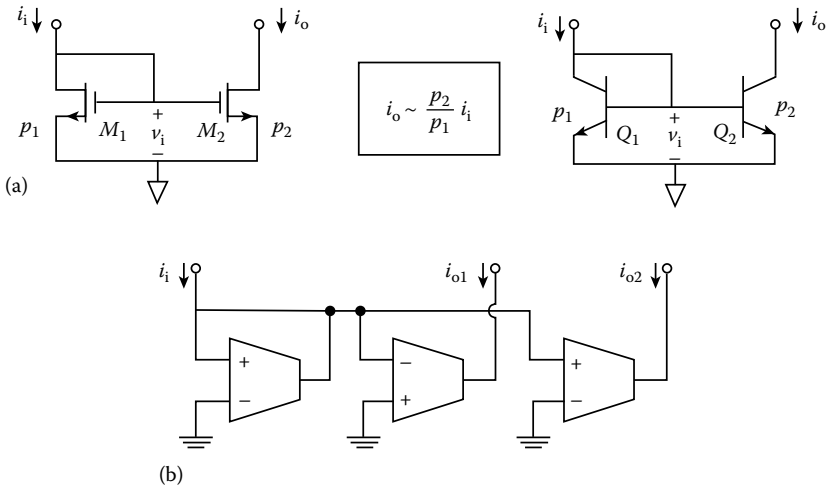


FIGURE 8.19 Current scaling using current mirrors. (a) Two matched transistors. (b) Noninverting amplification.

Let us now consider how to scale currents. The most convenient strategy uses a current mirror, whose simplest structure consists of two matched transistors connected as shown in Figure 8.19a [25]. Its operating principle relies on functional cancellation of the transistor nonlinearities to yield a linear relationship

$$i_o = p_2 f(v_i) = p_2 f\left[f^{-1}\left(\frac{i_i}{p_1}\right)\right] = \frac{p_2}{p_1} i_i \tag{8.29}$$

where p_1 and p_2 are parameters with value that can be designer controlled; for instance, β of the MOST or I_s of the BJT (see Appendix A and Ref. [44]). The input and output currents in Figure 8.19a must be positive. Driving the input and output nodes with bias currents I_B and $(p_2/p_1)I_B$, respectively, one obtains $i_i = i_i' + I_B$ and $i_o = I_o' + (p_2 + p_1)I_B$, and this enables bilateral operation on i_i' and i_o'.

In practical circuits, this simple design concept must be combined with circuit strategies to reduce errors due to nonnegligible input current of BJTs, DC voltage mismatch between input and output terminals, finite input resistance, and finite output resistance. Examples of these strategies can be found in Refs. [25,56,77]. On the other hand, sizing and layout strategies for other problems related to random mismatches between input and output devices are found in Ref. [41,48], which are applicable to most matching problems in MOS IC design.

The current mirror concept is extensible to any pair of matched transconductors, provided their transconductance characteristics are invertible and parameterized by a designer-controlled scale factor p, and that the dependence of the output current with the output voltage is negligible. In particular, the use of differential transconductors enables us to obtain bilateral operation simply, requiring no current-shifted biasing at the input and output nodes. It also simplifies achieving noninverting amplification (that is, positive scale factors), as Figure 8.19b illustrates. This figure also serves to illustrate the extension of the mirror concept to multiple current outputs. Note that except for loading considerations, no other limitations exist on the number of output transconductors that can share the input voltage. Also, because fan-out of a current source is strictly one, this replication capability is needed to enable several nodes to be excited by a common current. On the other hand, the fact that the different current output replicas can be scaled independently provides additional adjusting capability for circuit design.

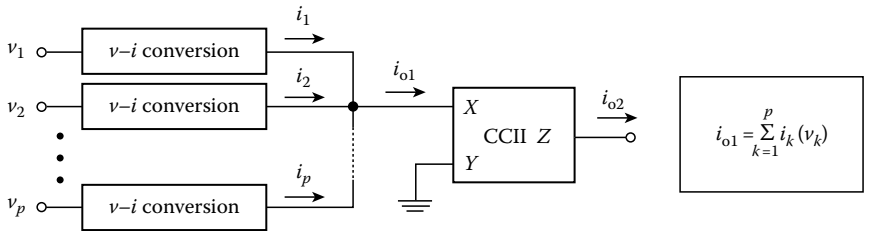


FIGURE 8.20 Aggregation of voltages through intermediate currents and current conveyor.

8.3.2.2 Signal Aggregation

As for the scaling operation, aggregation circuitry operates in current domain, based on Kirchhoff's current law (KCL). Thus, the aggregation of voltages requires that first they be transformed into currents (equivalently, charge packets in the switched-capacitor circuitry) and then added through KCL, while currents and incremental charges are added by routing all the components to a common node. If the number of components is large, the output impedance of the driving nodes is not large enough, and/or the input importance of the load is not small enough, this operation will encompass significant loading errors due to variations of the voltage at the summing node. This is overcome by clamping the voltage of this node using a virtual ground, which in practical circuits is realized by using either the input port of an op-amp, or terminals *X* and *Y* of a current conveyor. Figure 8.20 illustrates the current conveyor case.

8.4 Piecewise-Linear Circuitry

Consider the elementary PWL function that arise in connection with the different methods of representation covered in Section 8.2:

- Two-piece concave and convex characteristics (see Equation 8.12)
- Hermite linear basis function (see Figure 8.4 and Appendix B)
- Absolute value (see Equation 8.13)

where rectification is the only nonlinear operator involved. The circuit primitives in Appendix A exhibit several mechanisms which are exploitable in order to realize rectification:

- Cutoff of diodes and transistors—specifically, current through a diode negligible for negative voltage, output current of BJTs, and MOSTs negligible under proper biasing
- Very large resistance and zero offset voltage of an analog switch for negative biasing of the control terminal
- Digital encoding of the sign of a differential voltage signal using a comparator

Similar to scaling and aggregation operations, rectification is performed in current domain, using the mechanisms listed previously to make the current through a branch negligible under certain conditions. Three techniques are presented, which use current transfer in a transistor-based circuit, current-to-voltage transfer using diodes and op-amp, and charge transfer using switches and comparators, respectively.

8.4.1 Current Transfer Piecewise-Linear Circuitry

Figure 8.21a and b presents the simplest technique to rectify the current transferred from node ① to node ②. They exploit the feature of diodes and diode-connected transistors to support only positive

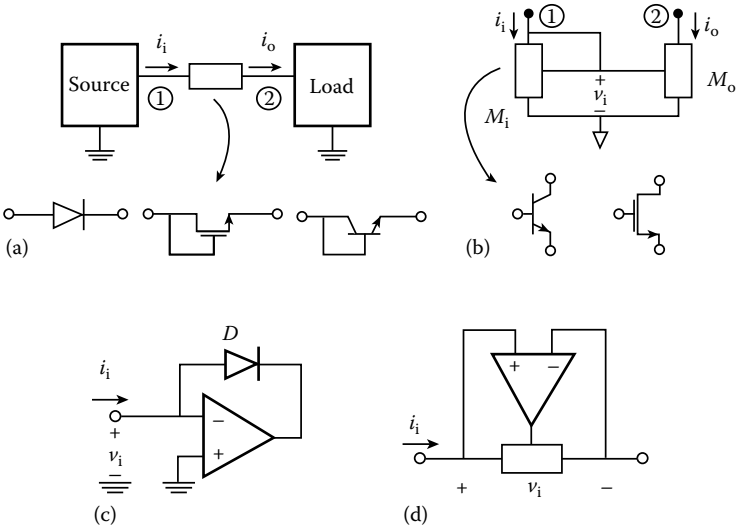


FIGURE 8.21 (a) and (b) Circuit techniques for current rectification; (c) and (d) superdiodes.

currents. Figure 8.21a operates by precluding negative currents to circulate from node ① to node ②, while Figure 8.21b also involves the nonlinear transconductance of the output transistor M_o; negative currents driving the node ① force v_i to become smaller than the cut-in voltage and, consequently, the output current becomes negligible. A drawback to both circuits is that they do not provide a path for negative input currents, which accumulates spurious charge at the input node and forces the driving stage to operate outside its linear operating regime. Solutions to these problems can be found in Refs. [57,61]. Also, Figure 8.21a produces a voltage displacement equal to the cut-in voltage of the rectifying device, which may be problematic for applications in which the voltage at node ① bears information. A common strategy to reduce the voltage displacements uses feedback to create superdiodes (shown in Figure 8.21c for the grounded case and Figure 8.21d for the floating case), and where the reduction of the voltage displacement is proportional to the DC gain of the amplifier.

Figure 8.22a, called a current switch, provides paths for positive and negative currents entering node ①, and obtains both kinds of elementary PWL characteristics exploiting cutoff of either BJTs or MOSTs. It consists of two complementary devices: *npn* (top) and *pnp* BJTs, or *n*-channel (top) and *p*-channel MOSTs. Its operation is very simple: any positive input current increases the input voltage, turning the bottom device ON. Because both devices share the input voltage, the top device becomes OFF. Similarly, the input voltage decreases for negative input currents, so that the top device becomes ON and the bottom OFF. In sum, positive input currents are drawn to the bottom device, while negative currents are drawn to the top device.

An inconvenience of Figure 8.22a is the dead zone exhibited by its input driving-point characteristics, which is very wide for MOSTs. It may produce errors due to nonlinear loading of the circuitry that drives the input node. Figure 8.22b overcomes this by using a circuit strategy similar to that of the superdiodes. The virtual ground at the op-amp input renders the dead zone centered around the voltage level E, and its amplitude is reduced by a factor proportional to the amplifier DC gain. Some considerations related to the realization of this amplifier are found in Ref. [58].

Proper routing and scaling of the currents i_p and i_n in Figure 8.22a gives us the concave and convex basic characteristics with full control of the knot and position and the slope in the conducting region. Figure 8.22c is the associated circuit, in which the input bias current controls the knot position, and the slope in the conducting region is given by the gain of the current mirrors. Note that this circuit also obtains the absolute value characteristics, while Figure 8.22d obtains the Hermite linear basis function.

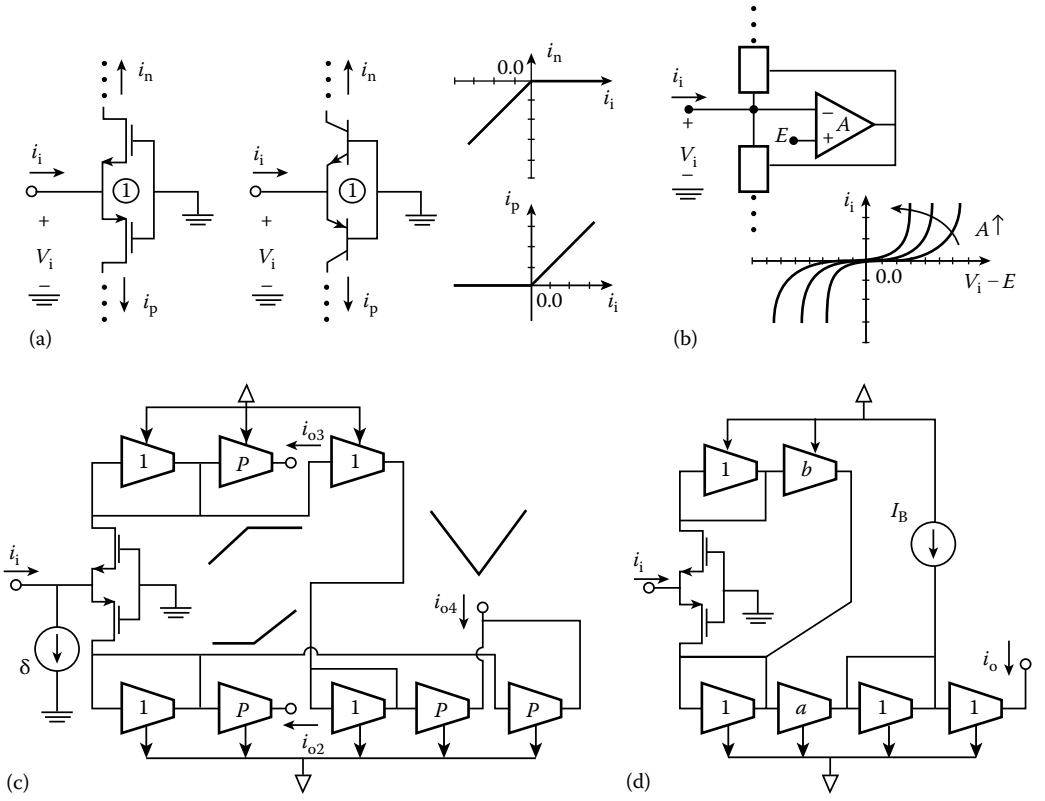


FIGURE 8.22 Current switch and its application for different basic PWL curves. (a) Dead zone exhibited by its input driving-point characteristics. (b) A circuit strategy similar to that of the superdiodes. (c) The associated circuit of (a). (d) A Hermite linear basis function circuit.

The way to obtain the PWL fuzzy membership function from this latter circuit is straightforward, and can be found in Ref. [58].

8.4.2 Transresistance Piecewise-Linear Circuitry

The circuit strategies involved in PWL current transfer can be combined in different ways with the transformation circuits discussed previously to obtain transconductance and voltage-transfer PWL circuits. In many cases design ingenuity enables optimum merging of the components and consequently, simpler circuits. Figure 8.23a depicts what constitutes the most extended strategy to realize the elementary PWL functions using off-the-shelf components [63,80]. The input current is split by the feedback circuitry around the op-amp to make negative currents circulate across D_n and positive currents circulate across D_p. Consequently, this feedback renders the input node of the op-amp a virtual ground and thus reduces errors due to finite diode cut-in voltage in the transresistance characteristics. Similar to Figure 8.22, the position of the knot in these elementary characteristics is directly controlled by an input bias current. Also note that the virtual ground can be exploited to achieve voltage-to-current transformation using the strategy of Figure 8.12d and thus, voltage-transfer operation.

Algebraic combination of the elementary curves provided by Figure 8.23a requires transforming the voltages v_{on} and v_{op} into currents and then aggregating these currents by KCL. For example, Figure 8.23b is the circuit for the absolute value and Figure 8.23c presents a possible implementation of the Hermite basis function.

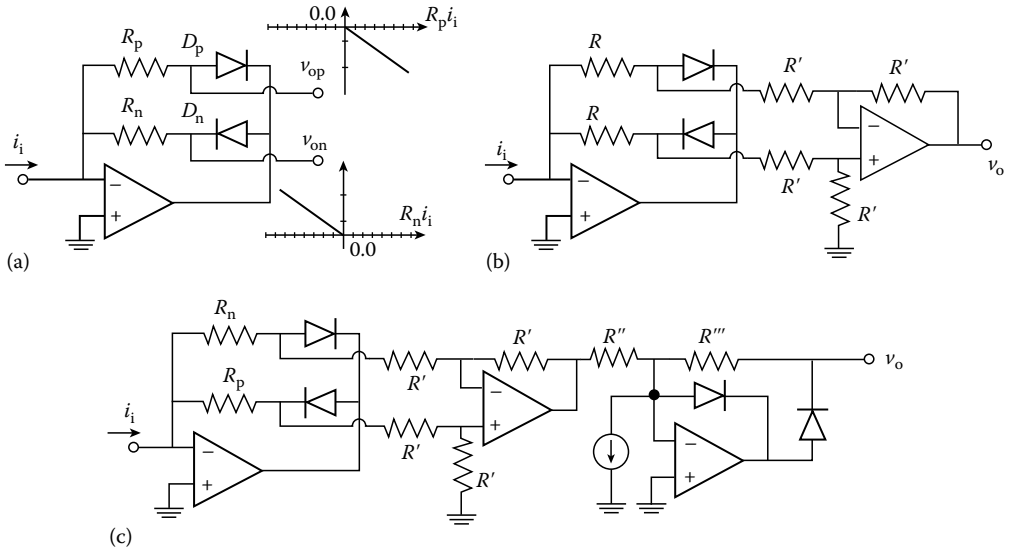


FIGURE 8.23 PWL transimpedance circuits. (a) Circuit for algebraic combination of the elementary curves. (b) Circuit for the absolute value. (c) Circuit for a possible implementation of the Hermite basis function.

Other related contributions found in the literature focus on the systematic realization of PWL driving-point resistors, and can be found in Refs. [7,10].

8.4.3 Piecewise-Linear Shaping of Voltage-to-Charge Transfer Characteristics

The realization of PWL relationships among sampled-data signals is based on nonlinear voltage-to-charge transfer and uses analog switches and comparators. Figure 8.24a is a circuit structure, where one of the capacitor terminals is connected to virtual ground and the other to a switching block. Assume that nodes ① and ② are both grounded. Note that for $(v-\delta)>0$ the switch arrangement set node ④ to δ, while node ⑤ is set to v. For $(v-\delta)<0$, nodes ④ and ⑤ are both grounded. Consequently, voltage at node ③ in this latter situation does not change from one clock phase to the next, and consequently, the incremental charge becomes null for $(v-\delta)<0$. On the other hand, for $(v-\delta)>0$, the voltage at node ③ changes from one clock phase to the next, and generates an incremental charge

$$\Delta q^{e} = C(v-\delta) = -\Delta q^{o} \tag{8.30}$$

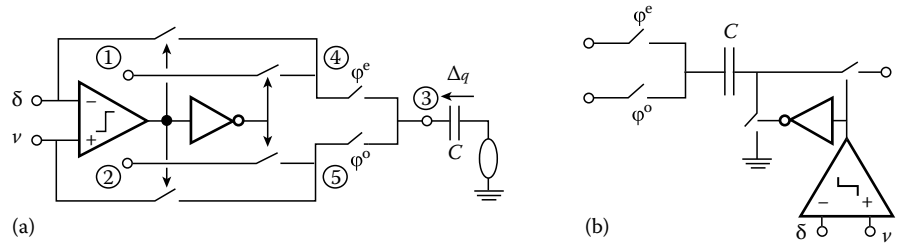


FIGURE 8.24 Circuits for rectification in voltage-to-charge domain. (a) Circuit structure where one of the capacitor terminals is connected to virtual ground and the other to a switching block. (b) Circuitry using series rectification of the circulating charge through a comparator-controlled switch.

which enables us to obtain negative and positive slopes using the same circuit, as shown in Figure 8.24a. To make the characteristics null for $(v - \delta) > 0$, it suffices to interchange the comparator inputs. Also, the technique is easily extended to the absolute value operation by connecting terminal ① to v, and terminal ② to δ. The realization of the Hermite linear basis function is straightforward and can be found in Ref. [55].

Other approaches to the realization of PWL switched-capacitor circuitry use series rectification of the circulating charge through a comparator-controlled switch (Figure 8.24b), and can be found in Refs. [16,31]. The latter also discusses exploitation of these switched-capacitor circuits to realize continuous-time driving-point characteristics, the associated transformation circuits, and the dynamic problematics.

8.5 Polynomials, Rational, and Piecewise-Polynomial Functions

These functions use rectification (required for truncation operation in the PWP case) and analog *multiplication*,

$$z = \frac{xy}{\alpha} \tag{8.31}$$

as basic nonlinear operators.* Joining the two inputs of the multiplier realizes the square function. Analog division is realized by applying feedback around a multiplier, illustrated at the conceptual level in Figure 8.25a; the multiplier obtains $e = (zy)/\alpha$, and for $A \to \infty$, the feedback forces $x = e$. Thus, if $y \neq 0$, the circuit obtains $z = \alpha(x/y)$. Joining y and z terminals, the circuit realizes the square root, $z = (\alpha x)^{1/2}$. This concept of division is applicable regardless of the physical nature of the variables involved. In the special case in which e and x are current and z is a voltage, the division can be accomplished using KCL to yield $x = e$. Figure 8.25b shows a circuit for the case in which the multiplication is in voltage domain, and Figure 8.25c is for the case in which multiplication is performed in transconductance domain. The transconductance gain for input z in the latter case must be negative to guarantee stability.

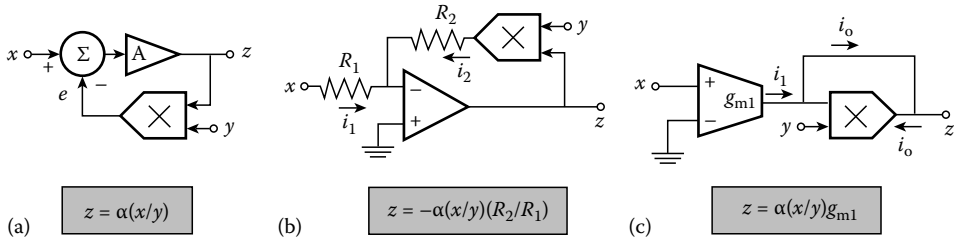


FIGURE 8.25 Division operator using a feedback multiplier: (a) concept; (b) with voltage multiplier and op-amp; (c) with transconductance multiplier and OTA.

* Scale factor α in Equation 1.31 must be chosen to guarantee linear operation in the full variation range of inputs and outputs.

8.5.1 Concepts and Techniques for Polynomic and Rational Functions

Figure 8.26 illustrates conceptual hardware for several polynomials up to the fifth degree. Any larger degree is realized similarly. Figure 8.27 uses polynomials and analog division to realize rational functions

$$g(x) = \frac{\sum\limits_{i=0,Q} \alpha_j x^j}{\sum\limits_{j=0,R} \beta_j x^j} \tag{8.32}$$

For simplicity, we have assumed that the internal scaling factors of the multipliers in Figures 8.26 and 8.27 equal one.

An alternative technique to realize rational functions is based on linearly controlled resistors, described as $v = (Lx)i$, and linearly controlled conductors, $i = (Cx)v$, where L and C are real parameters. This technique exploits the similarity between these characteristics and those which describe inductors and capacitors in the frequency domain, to take advantage of the synthesis techniques for rational transfer function in the s-plane through interconnection of these linear components [28,39] (Figure 8.28). As for the previous cases, realization of linearly controlled resistors and conductors require only multipliers and,

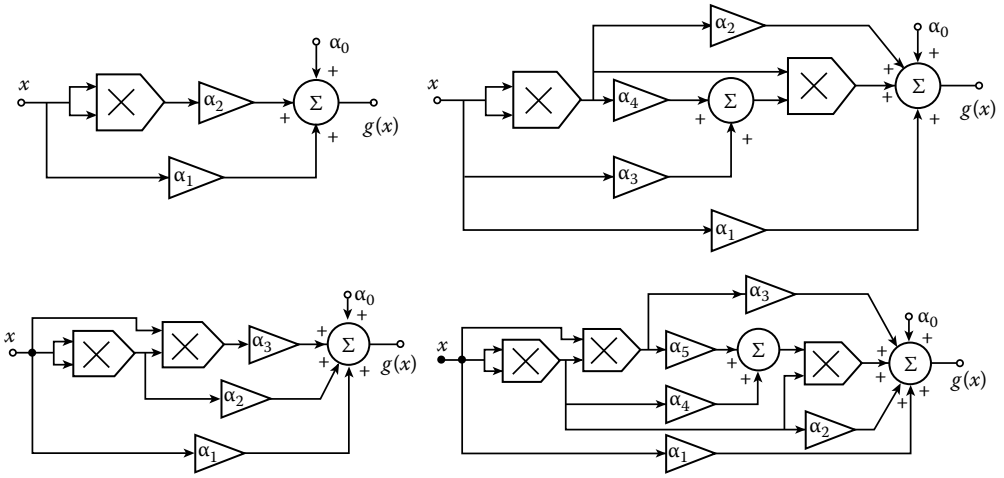


FIGURE 8.26 Conceptual hardware for polynomial functions.

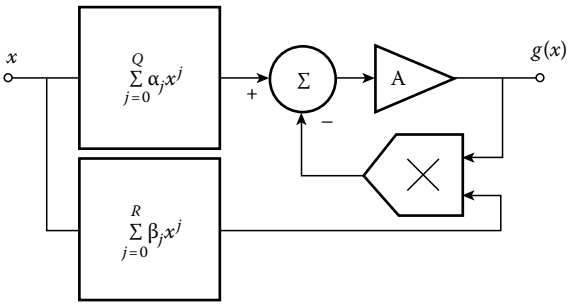


FIGURE 8.27 Rational function generation through feedback division.

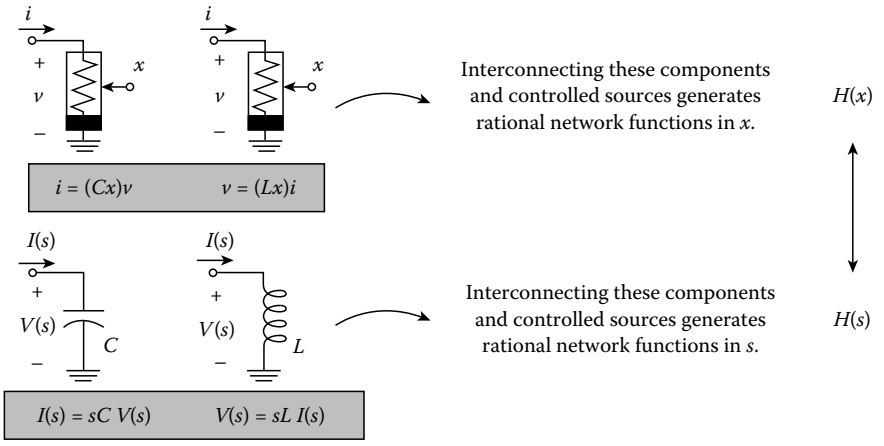


FIGURE 8.28 Usage of linearly controlled resistors to synthesize rational network functions.

depending upon the nature of the variables involved in the multipliers, voltage-to-current and current-to-voltage transformation circuits.

8.5.2 Multiplication Circuitry

Two basic strategies realize multiplication circuitry: using signal processing and exploiting some nonlinear mechanism of the primitive components. Signal processing multipliers rely on the generation of a pulsed signal whose amplitude is determined by one of the multiplicands and its duty cycle by the other, so that the area is proportional to the result of the multiplication operation. Figure 8.29a presents an implementation concept based on averaging. This is performed by a low-pass filter where the input is a pulse train with amplitude proportional to x and duty cycle proportional to y. The latter proportionality is achieved through nonlinear sampling by comparing y with a time reference sawtooth signal. Thus, the

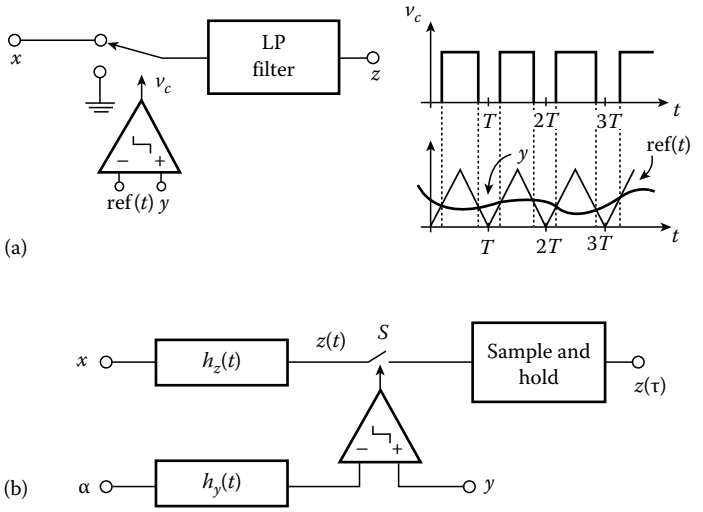


FIGURE 8.29 Signal processing multipliers by (a) averaging; (b) shaping in time domain.

area under each pulse in the train is the product of $x \times y$, extracted by the low-pass filter. This implementation concept is discussed in further detail in classical texts on analog computation [63], and applied more recently to analog VLSI signal processing [72].

Figure 8.29b is an alternative implementation concept based on signal shaping in the time domain. It uses two linear blocks with normalized unit step response given as $h_z(t)$ and $h_y(t)$. The first is driven by level x to obtain

$$z(t) = xh_z(t), \quad 0 \leq t < \tau \tag{8.33}$$

where τ denotes the duration of the time interval during which the switch S remains closed. The other is driven by a references level α, to render τ given by

$$\tau = h_y^{-1}\left(\frac{y}{\alpha}\right) \tag{8.34}$$

Assuming both linear blocks are identical and the time function invertible, one obtains the steady-state value of z, $z(\tau)$, as the product of levels x and y.

The simplest implementation of Figure 8.29 uses integrators, i.e., $h(t) = t$, as linear blocks (see Figure 8.41b). Also note that the principle can be extended to the generation of powers of an input signal by higher-order shaping in time domain. In this case, both linear blocks are driven by reference levels. The block $h_y(t)$ consists of a single integrator, $\tau = y/\alpha$. The other consists of the cascade of P integrators, and obtains $z(t) = \beta t^p$. Thus, $z(t) = \beta(y/\alpha)^p$. Realizations suitable for integrated circuits are found in Refs. [34,55].

8.5.3 Multipliers Based on Nonlinear Devices

The primitives in Appendix A display several mechanisms that are exploitable to realize analog multipliers:

- Exponential functionals associated to the large-signal transconductance of BJTS, and the possibility of obtaining logarithmic dependencies using feedback inversion
- Square-law functionals associated to the large-signal transconductance of the MOS transistor operating in saturation region
- Small-signal transconductance of a BJT in active region as a linear function of collector current
- Small-signal transconductance of a MOST in saturation as a linear function of gate voltage
- Small-signal self-conductance of a MOS transistor in ohmic region as a linear function of gate voltage

These and related mechanisms have been explained in different ways and have resulted in a huge catalog of practical circuits. To quote all the related published material is beyond the scope of this section. The references listed at the end were selected because of their significance, and their cross-references contain a complete view of the state of the art. Also, many of the reported structures can be grouped according to the theory of *translinear circuits*, which provides a unified framework to realize nonlinear algebraic functions through circuits [23,62,79].

8.5.3.1 Log–Antilog Multipliers

Based on the exponential large-signal transconductance of the BJT, and the following relationships,

$$\begin{aligned} z' &= \ln(x) + \ln(y) = \ln(xy) \\ z &= e^{z'} = e^{\ln(xy)} = xy \end{aligned} \tag{8.35}$$

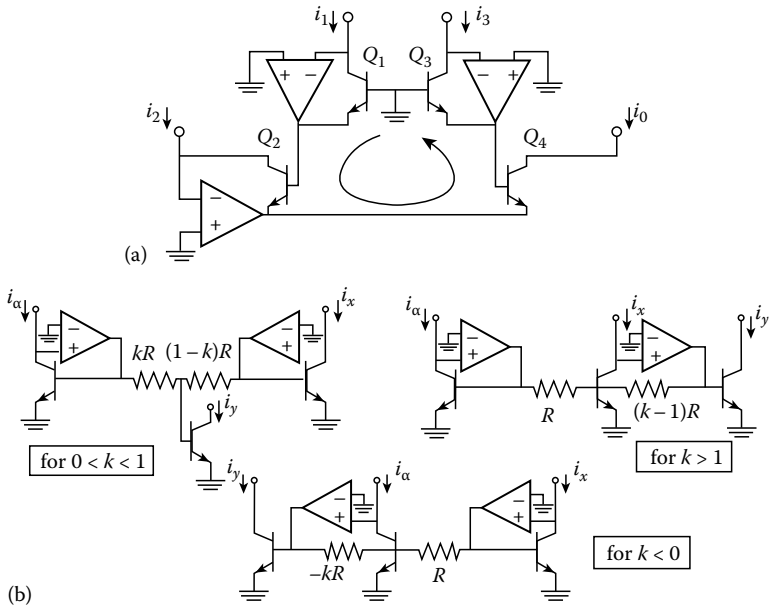


FIGURE 8.30 (a) Core block of a log–antilog multiplier; (b) circuits to elevate to a power.

which can be realized as illustrated in Figure 8.30a [65]. This circuit operates on positive terminal currents to obtain $i_0 = (i_1 i_2)/i_3$, which can be understood from translinear circuit principles by noting that the four base-to-emitter voltages define a translinear loop,

$$\begin{aligned} 0 &= v_{be1} + v_{be2} - v_{be3} - v_{be4} \\ &= \ln\left(\frac{i_1}{I_s}\right) + \ln\left(\frac{i_2}{I_s}\right) - \ln\left(\frac{i_3}{I_s}\right) - \ln\left(\frac{i_o}{I_s}\right) \end{aligned} \tag{8.36}$$

The circuit can be made to operate in four-quadrant mode, though restricted to currents larger than $-I_B$, by driving each terminal with a bias current source of value I_B. Also, because all input terminals are virtual ground the circuit can be made to operate on voltages by using the voltage-to-current transformation concept of Figure 8.12d. Similarly, the output current can be transformed into a voltage by using an extra op-amp and the current-to-voltage transformation concept of Figure 8.13c. Extension of this circuit structure to generate arbitrary powers is discussed in Ref. [23]. Figure 8.30b [1] uses similar techniques, based on introducing scaling factors in the translinear loop, to obtain

$$i_y = i_\alpha^{1-k} i_x^k \tag{8.37}$$

8.5.3.2 Square-Law Multipliers

Square-law multipliers are based on the algebraic properties of the square function, most typically

$$z = \frac{1}{4}\left[(x+y)^2 - (x-y)^2\right] = xy \tag{8.38}$$

shown conceptually in Figure 8.31a, and the possibility of obtaining the square of a signal using circuits, typically consisting of a few MOS transistors operating in saturation region. Figure 8.31b through f depict some squarer circuits reported in the literature.

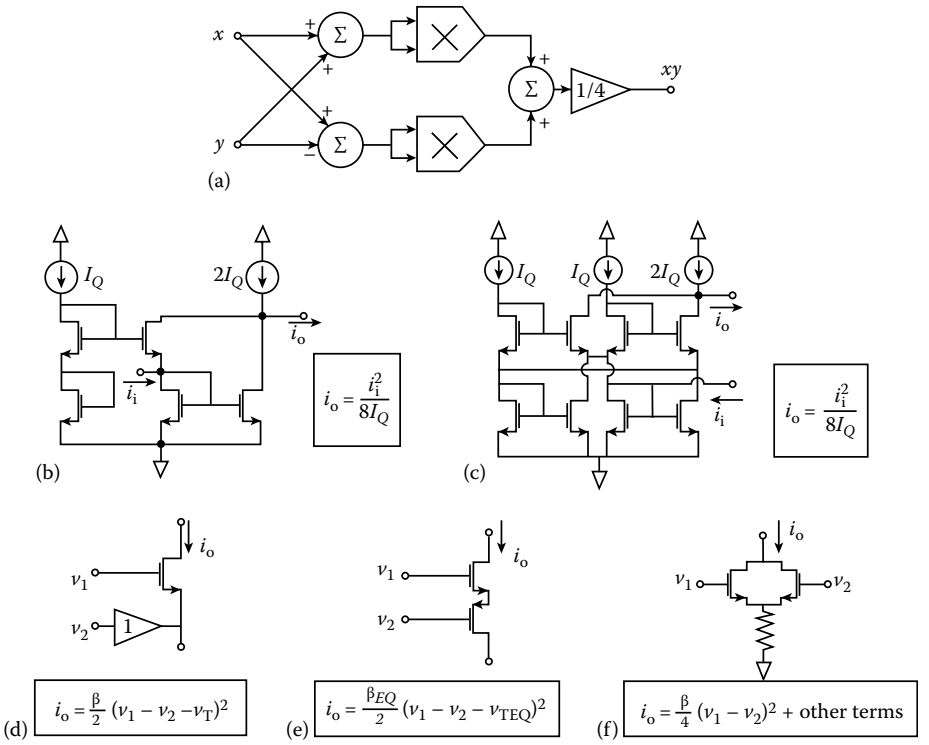


FIGURE 8.31 (a) Block diagram of the quarter-square multiplier; (b) current-mode squarer circuit in Ref. [3]; (c) current-mode squarer circuit in Ref. [79]; (d) voltage-mode squarer circuit in Ref. [36]; (e) voltage-mode squarer circuit in Ref. [60]; (f) voltage-mode squarer circuit in Ref. [49].

The completeness of square-law operators for the realization of nonlinear circuits was demonstrated from a more general point of view in Ref. [47], and their exploitation has evolved into systematic circuit design methodologies to perform both linear and nonlinear functions [3].

8.5.3.3 Transconductance Multipliers

A direct, straightforward technique to realize the multiplication function exploits the possibility of controlling the transconductance of transistors through an electrical variable (current or voltage). Although this feature is exhibited also by unilateral amplifiers, most practical realizations use differential amplifiers to reduce offset problems and enhance linearity [25]. Figure 8.32 presents a generic schematic for a differential amplifier, consisting of two identical three-terminal active devices with common bias current. The expressions on the right display its associated transconductance characteristics for *npn*-BJTs and *n*-channel MOSTs, respectively [25]. These characteristics are approximated to a first-order model as

$$i_{z\mathrm{BJT}} \approx \frac{i_y}{4U_t} v_x, \quad i_{z\mathrm{MOST}} \approx \left(\sqrt{\beta i_y}\right) v_x \tag{8.39}$$

which clearly displays the multiplication operation, although restricted to a rather small linearity range. Practical circuits based on this idea focus mainly on increasing this range of linearity, and follow different design strategies. Figure 8.33 gives an example known as the Gilbert cell or Gilbert multiplier [23]. Corresponding realizations using MOS transistors are discussed in Refs. [2,53]. Sánchez-Sinencio et al. [61] present circuits to realize this multiplication function using OTA blocks. On the other

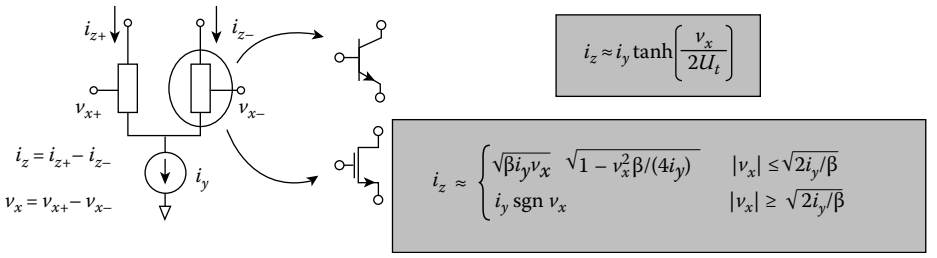


FIGURE 8.32 Differential amplifiers and their associated large-signal transconductances.

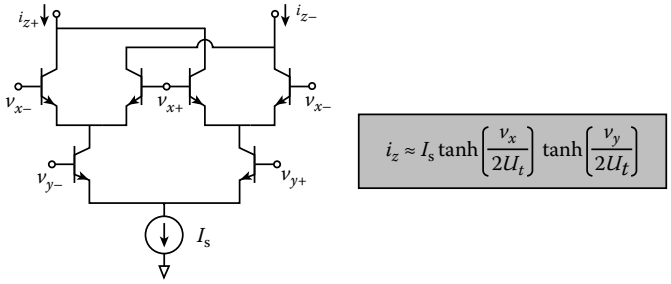


FIGURE 8.33 Bipolar Gilbert cell.

hand, Ref. [17] presents a tutorial discussion of different linearization techniques for MOS differential amplifiers.

8.5.3.4 Multiple Based in the Ohmic Region of MOS Transistors

The ohmic region of JFETs has been used to realize amplifiers with controllable gain for automatic gain control [54]. It is based on controlling the equivalent resistance of the JFET transistor in its ohmic region through a bias voltage. More recently, MOS transistors operating in the ohmic region were used to realize linear [69,70] and nonlinear [35] signal processing tasks in VLSI chips. There exist many ingenious circuits to eliminate second and higher-order nonlinearities in the equivalent resistance characteristics.

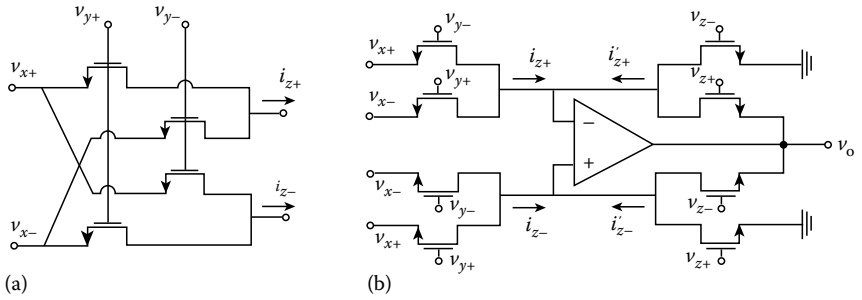


FIGURE 8.34 Four-quadrant multipliers based on MOS transistors in the ohmic region. (a) Circuit achieving very good nonlinear cancellation through cross-coupling and fully differential operation. (b) A more general view showing the conductance as well as the resistance of the MOS ohmic region used to obtain a versatile amplifier-divider building block.

The circuit in Figure 8.34a achieves very good nonlinearity cancellation through cross-coupling and fully differential operation, obtaining

$$i_{z+} - i_{z-} = 2\beta(v_{x+} - v_{x-})(v_{y+} - v_{y-}) \tag{8.40}$$

and its use in multiplication circuits is discussed in Refs. [35,66]. A more general view is presented in Figure 8.34b [35], where the conductance as well as the resistance of the MOS ohmic region are used to obtain a versatile amplifier-divider building block. Enomoto and Yasumoto [18] report another interesting multiplier that combines the ohmic region of the MOS transistor and sampled-data circuits.

8.6 Sigmoids, Bells, and Collective Computation Circuits

8.6.1 Sigmoidal Characteristics

As Equation 8.5 illustrates, approximating a nonlinear function through a multilayer perceptron requires the realization of sigmoidal functions, with arguments given as linear combinations of several variables. The exact shape of the sigmodial is not critical for the approximation itself, although it may play an important role in fitting [82]. Figure 8.35 depicts two shapes used in practice. Figure 8.35a, the hard limiter, has an inner piece of large (ideally infinite) slope, while for Figure 8.35b, the soft limiter, this slope is smaller and can be used as a fitting parameter.

Most amplifiers have large-signal transfer characteristics whose shape is a sigmoid or an inverted sigmoid. We present only those circuits whose inputs are currents because this simplifies the circuitry needed to obtain these inputs as linear combinations of other variables. The op-amp circuit of Figure 8.36a realizes the soft limiter characteristics in transimpedance form. The center is set by the input bias current and the slope through the resistor ($\beta = R$). If the branch composed of the two Zener diodes is eliminated, the saturation levels E_+ and E_- are determined through the internal op-amp circuitry, inappropriate for accurate control. (Otherwise, they are determined through the Zener breakdown voltages.) On the other hand, Figure 8.36b also realizes the hard sigmoid in transimpedance domain [58]. The output saturation levels for this structure are $E_+ = V_{Tn}$ and $E_- = |V_{Tp}|$, where V_{Tn} and V_{Tp} are the threshold voltages of the NMOS transistor and the PMOS transistor, respectively. To obtain the output represented by a current, one can use voltage-to-current transformation circuits. References [15,57,58] discuss simpler alternatives operating directly in current domain. For instance, Figure 8.36c and d depicts circuits for the soft limiter characteristics and the hard limiter characteristics.

With regard to the calculation of the input to the sigmoid as a linear combination of variables, note that the input node of all circuits in Figure 8.36 is virtual ground. Consequently, the input current can be obtained as a linear combination of voltages or currents using the techniques for signal scaling and aggregation presented in Section 8.3.

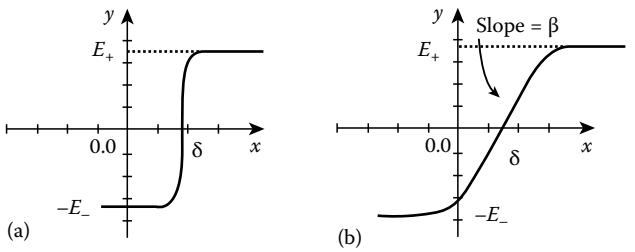


FIGURE 8.35 Typical sigmoidal shapes: (a) hard limiter; (b) soft limiter.

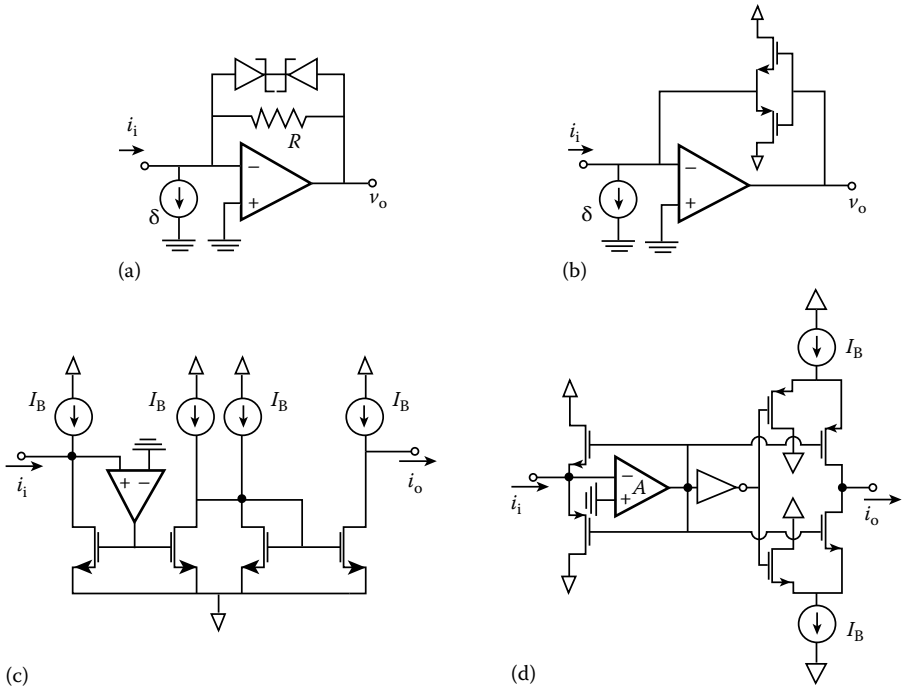


FIGURE 8.36 Realization of sigmoidal characteristics with input current: (a) transimpedance soft limiter; (b) transimpedance hard limiter; (c) and (d) soft and hard limiters in current transfer domain.

8.6.2 Bell-Like Shapes

The exact shapes of Equations 8.16 and 8.17 involve the interconnection of squarers, together with blocks to elevate to power, and exponential blocks—all realizable using techniques previously discussed in this chapter. However, these exact shapes are not required in many applications, and can be approximated using simpler circuits. Thus, let us consider the differential amplifier of Figure 8.32, and define $v_i = v_x$, $I_B = i_y$, and $i_o = i_z$ for convenience. The expressions for the large-signal transconductance displayed along with the figures show that they are sigmoids with saturation levels at I_B and $-I_B$. They are centered at $v_i = 0$, with the slope at this center point given by Equation 8.39. The center can be shifted by making $v_i = v_{x+}$ and $\delta = v_{x-}$.

Similar to the differential amplifier, most OTAs exhibit sigmoid-like characteristics under large-signal operation, exploitable to realize nonlinear functions [19,37,56,61,71]. This may rely on the mathematical techniques behind multilayer perceptrons, or on those behind RBF and fuzzy interpolation.

Figure 8.37a obtains a bell-shaped transconductance through a linear, KCL combination of the two sigmoidal characteristics, one of negative slope and the other of positive slope. The width and center of the bell (see Figure 8.7) are given respectively by

$$2\sigma = \delta_2 - \delta_1, \quad \delta = \frac{\delta_2 + \delta_1}{2} \tag{8.41}$$

controlled by the designer. The slope of the bell at the cross-over points is also controlled through the transconductance of the OTAs.

For simpler circuit realizations, this technique can be used directly with differential amplifiers, as shown in Figure 8.37b. The differential output current provided by the circuit can be transformed into a

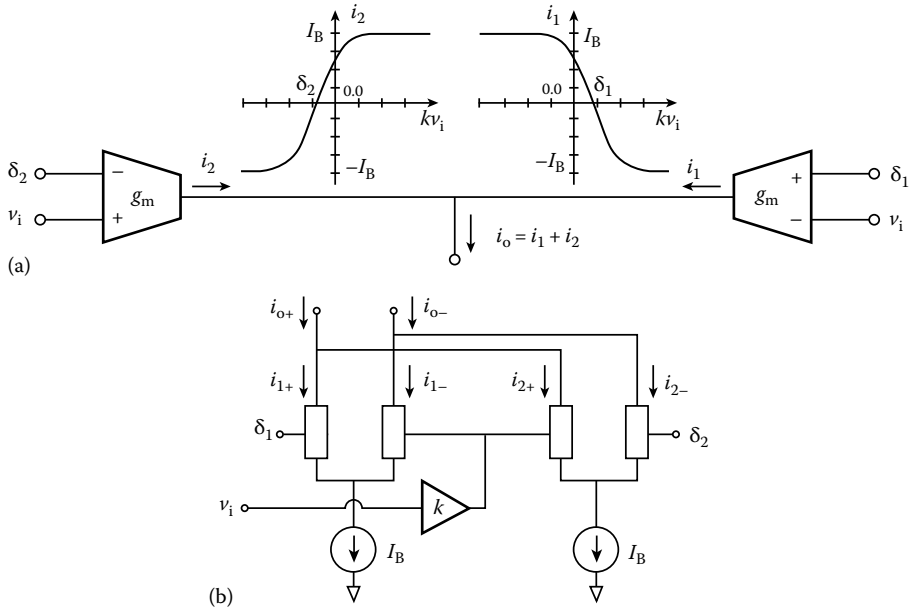


FIGURE 8.37 Transconductance circuits for bell-shaped function: (a) using OTAs; (b) using differential amplifiers.

unilateral one using a *p*-channel current mirror. Equation 8.41 also applies for this circuit, and the slope at the cross-overs is

$$\text{slope}_{\text{MOST}} = k\sqrt{\beta I_B}, \quad \text{slope}_{\text{BJT}} = \frac{kI_B}{4U_t} \tag{8.42}$$

Note that the control of this slope through the bias current changes the height of the bell. It motivates the use of a voltage gain block in Figure 8.37. Thus, the slope can be changed through its gain parameter k. The slope can also be changed through β for the MOSTs. Practical realizations of this concept are found in Refs. [4,71,74]. The voltage amplifier block can be realized using the techniques presented in this chapter. Simpler circuits based on MOS transistors are found in Ref. [53].

8.6.3 Collective Computation Circuitry

RBF and fuzzy inference require multidimensional operators to calculate radial distances in the case of RBF, and to normalize vectors and calculate T-norms in the case of fuzzy inference. These operators can be expressed as the interconnection of the nonlinear blocks discussed previously, or realized in a simpler manner through dedicated collective computation circuitry. Most of these circuits operate intrinsically in current domain and are worth mentioning because of this simplicity and relevance for parallel information processing systems.

8.6.3.1 Euclidean Distance

Figure 8.38 [38] presents a current-mode circuit to compute

$$i_y = \sqrt{\sum_{k=1,P} i_{xk}^2} \tag{8.43}$$

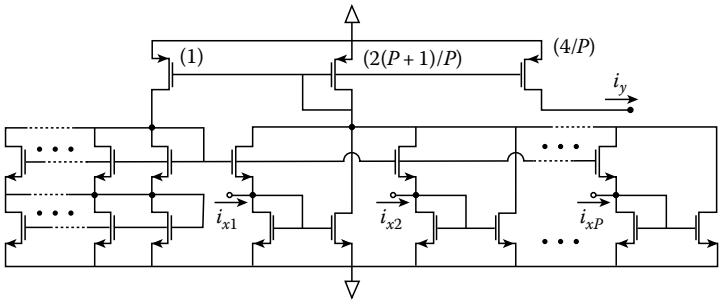


FIGURE 8.38 CMOS self-biased Euclidean distance circuit. (From Landolt, O., Vittoz, E., and Heim, P., *Electr. Lett.*, 28, 352, 1992. With permission.)

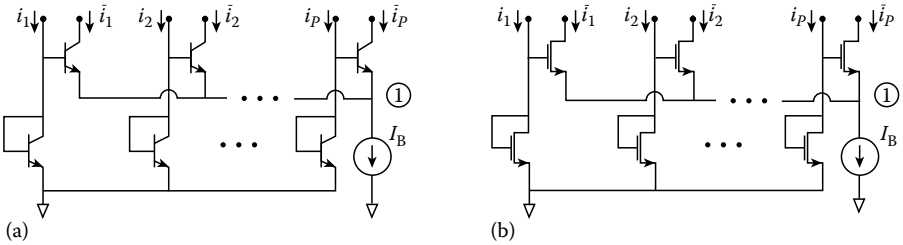


FIGURE 8.39 Current-mode normalization circuits: (a) BJT; (b) CMOS.

based on the square-law of MOS transistors in the saturation region. If the current i_k at each terminal is shifted through a bias current of value δ_k, the circuit serves to compute the Euclidean distance between the vector of input currents and the vector $\boldsymbol{\delta}$.

8.6.3.2 Normalization Operation

Figure 8.39 depicts circuits to normalize an input current vector, for the BJT [23] and the CMOS [74] cases, respectively. Their operation is based on KCL and the current mirror principle. Kirchhoff's circuit law forces the sum of the output currents at node ① to be constant. On the other hand, the current mirror operation forces a functional dependency between each pair of input and output currents. Thus, they obtain

$$\bar{i_k} \approx \frac{i_k}{\sum\limits_{j=1,P} i_j} \tag{8.44}$$

for each current component.

8.6.3.3 T-Norm Operator

The calculation of the minimum of an input vector $\mathbf{x}$ is functionally equivalent to obtaining the complement of the maximum of the complements of its components. Figure 8.40a illustrates a classical approach used in analog computation to calculate the maximum of an input vector $\mathbf{x}$. It is based on the following steady-state equation:

$$-y + \sum_{k=1,P} u_{-1}(A(x_k - y)) = 0 \tag{8.45}$$

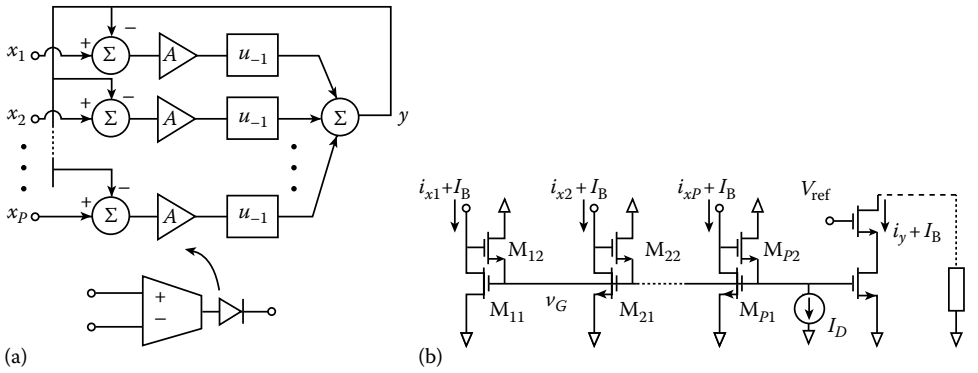


FIGURE 8.40 Concept for maximum operator and current-mode realization. (a) A classical approach used in analog computation to calculate the maximum of an input vector. (b) A CMOS current-mode realization.

where A is large. This concept can be realized in practice using OTAs, op-amps, or diodes. Both of these have voltage input and output. Alternatively, Figure 8.40b shows a CMOS current-mode realization [74]. In this circuit the maximum current determines the value of the common gate voltage, v_G. The only input transistor operating in the saturation region is that which is driven by maximum input current; the rest operate in the ohmic region.

8.7 Extension to Dynamic Systems

A dynamic system with state vector **x** and dynamics represented as

$$T_k \frac{\mathrm{d}x_k}{\mathrm{d}t} = f_k(\mathbf{X}), \quad 1 \le k \le P \tag{8.46}$$

can be mapped on the block diagram of Figure 8.41a, and realized by the interconnection of nonlinear resistive blocks and integrators. This approach is similar to that followed in classical analog computation

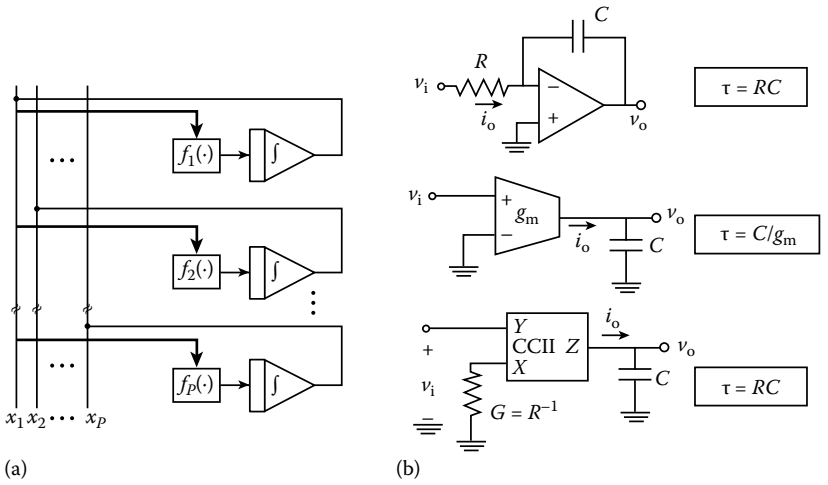


FIGURE 8.41 Conceptual state-variable block diagram of dynamic systems integrator circuits. (a) Block diagram realized by the interconnection of nonlinear resistive blocks and integrators. (b) Several integrated circuits.

[26] and has integrators as key components. Figure 8.41b illustrates several integrator circuits. Combining these circuits with the circuitry for nonlinear functions provides systematic approaches to synthesize nonlinear dynamic systems based on the approximations presented in this chapter [56]. On the other hand, Rodríguez-Vázquez and Delgado-Restituto [57] discuss related techniques to synthesize nonlinear systems described by finite-difference equations.

Appendix A: Catalog of Primitives

Figure 8.42 outlines our catalog of primitive components, all of which are available off-the-shelf, and, depending on the fabrication technology, can be realized on a common semiconductor substrate [44]. Generally, the catalog differs between individual technologies; for instance, no *npn*-BJTs are available in a CMOS *n*-well technology. The use of linear capacitors may appear surprising because we constrain ourselves to cover only static characteristics. However, we will not exploit their dynamic *i*–*v* relationship, but instead their constitutive equation in the charge–voltage plane, which is algebraic.

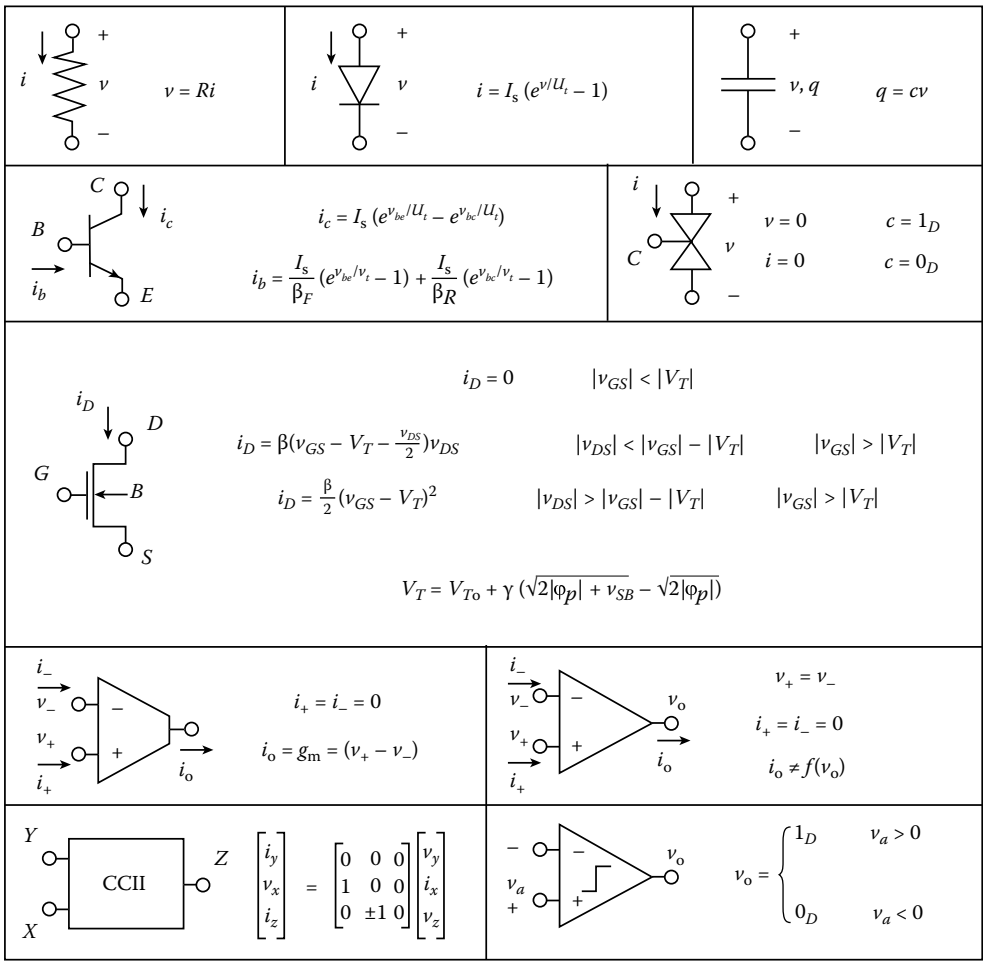
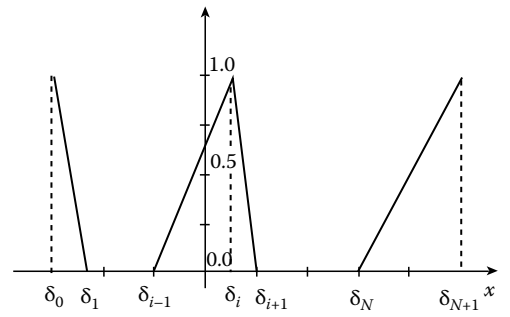


FIGURE 8.42 Section catalog of primitive circuit components.

Appendix B: Value and Slope Hermite Basis Functions

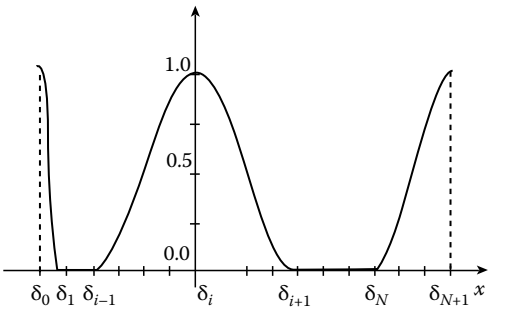


(a)

$$I_0(x) = \begin{cases} (\delta_1 - x)/\Delta_0 & \delta_0 \le x \le \delta_1 \\ 0 & \delta_1 \le x \le \delta_{N+1} \end{cases}$$

$$I_i(x) = \begin{cases} 0 & \delta_0 \le x \le \delta_{N+1} \\ (x - \delta_{i-1})/\Delta_{i-1} & \delta_{i-1} \le x \le \delta_i \\ (\delta_{i+1} - x)/\Delta_i & \delta_i \le x \le \delta_{i+1} \\ 0 & \delta_{i+1} \le x \le \delta_{N+1} \end{cases}$$

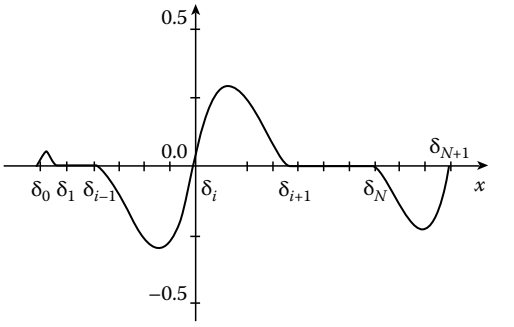
$$I_{N+1}(x) = \begin{cases} (x - \delta_N)/\Delta_N & \delta_N \le x \le \delta_{N+1} \\ 0 & \delta_0 \le x \le \delta_N \end{cases}$$



$$v_0(x) = \begin{cases} 1 - 2\beta_0^3 + 3\beta_0^2 & \delta_0 \le x \le \delta_1 \\ 0 & \delta_1 \le x \le \delta_{N+1} \end{cases}$$

$$v_i(x) = \begin{cases} 0 & \delta_0 \le x \le \delta_{N+1} \\ 1 - 3\alpha_i^2 - 2\alpha_i^3 & \delta_{i-1} \le x \le \delta_i \\ 1 - 3\beta_i^2 - 2\beta_i^3 & \delta_i \le x \le \delta_{i+1} \\ 0 & \delta_{i+1} \le x \le \delta_{N+1} \end{cases}$$

$$v_{N+1}(x) = \begin{cases} 1 - 3\alpha_{N+1}^2 - 2\alpha_{N+1}^3 & \delta_N \le x \le \delta_{N+1} \\ 0 & \delta_0 \le x \le \delta_N \end{cases}$$



$$s_0(x) = \begin{cases} \Delta_0\beta_0(1 - \beta_0)^2 & \delta_0 \le x \le \delta_1 \\ 0 & \delta_1 \le x \le \delta_{N+1} \end{cases}$$

$$s_i(x) = \begin{cases} 0 & \delta_0 \le x \le \delta_{N+1} \\ \Delta_{i-1}\alpha_i(1 + \alpha_i)^2 & \delta_{i-1} \le x \le \delta_i \\ \Delta_i\beta_i(1 - \beta_i)^2 & \delta_i \le x \le \delta_{i+1} \\ 0 & \delta_{i+1} \le x \le \delta_{N+1} \end{cases}$$

$$s_{N+1}(x) = \begin{cases} \Delta_N\alpha_{N+1}(1 + \alpha_{N+1})^2 & \delta_N \le x \le \delta_{N+1} \\ 0 & \delta_0 \le x \le \delta_N \end{cases}$$

$$\Delta_i = \delta_{i+1} - \delta_i \qquad \alpha_i = (x - \delta_i)/\Delta_{i-1} \qquad \beta_i = (x - \delta_i)/\Delta_i$$

(b)

FIGURE 8.43 Hermite basis functions: (a) PWL case; (b) PWC case.

References

1. X. Arreguit, E. Vittoz, and M. Merz, Precision compressor gain controller in CMOS technology, *IEEE J. Solid-State Circuits*, 22, 442–445, 1987.
2. J. N. Babanezhad and G. C. Temes, A 20-V four quadrant CMOS analog multiplier, *IEEE J. Solid-State Circuits*, 20, 1158–1168, 1985.

3. K. Bult and H. Wallinga, A class of analog CMOS circuits based on the square-law characteristics of an MOS transistor in saturation, *IEEE J. Solid-State Circuits*, 22, 357–365, 1987.
4. J. Choi, B. J. Sheu, and J. C. F. Change, A Gaussian synapse circuit for analog VLSI neural network, *IEEE Trans. Very Large Scale Integr. Syst*., 2, 129–133, 1994.
5. L. O. Chua, The linear transformation converter and its applications to the synthesis of nonlinear networks, *IEEE Trans. Circuit Theory*, 17, 584–594, 1970.
6. L. O. Chua and S. M. Kang, Section-wise piecewise-linear functions: Canonical representation, properties and applications, *Proc. IEEE*, 67, 915–929, 1977.
7. L. O. Chua and S. Wong, Synthesis of piecewise-linear networks, *Elect. Circ. Syst*., 2, 102–108, 1978.
8. L. O. Chua, Nonlinear circuits, *IEEE Trans. Circuits Syst*., 31, 69–87, 1984.
9. L. O. Chua and G. Zhong, Negative resistance curve tracer, *IEEE Trans. Circuits Syst*., 32, 569–582, 1985.
10. L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*, New York: McGraw-Hill, 1987.
11. L. O. Chua and A. C. Deng, Canonical piecewise linear representation, *IEEE Trans. Circuits Syst*., 35, 101–111, 1988.
12. G. Cybenko, Approximation by superposition of a sigmoidal function, *Math. Control. Syst. Signals*, 2, 303–314, 1989.
13. Z. Czarnul, Modification of Banu–Tsividis continuous-time integrator structure, *IEEE Trans. Circuits Syst*., 33, 714–716, 1986.
14. R. W. Daniels, *Approximation Methods for Electronic Filter Design*, New York: McGraw-Hill, 1974.
15. M. Delgado-Restituto and A. Rodríguez-Vázquez, Switched-current chaotic neutrons, *Electr. Lett*., 30, 429–430, 1994.
16. M. Delgado-Restituto, A chaotic switched-capacitor circuit for $1/f$ noise generation, *IEEE Trans. Circuits Syst*., 39, 325–328, 1992.
17. S. T. Dupuie and M. Ismail, High-frequency CMOS transconductors, in *Analogue IC Design: The Current-Mode Approach*, C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds., London: Peter Peregrinus Ltd., 1990.
18. T. Enomoto and M. Yasumoto, Integrated MOS four-quardrant analog multiplier using switched capacitor technology for analog signal processor IC's, *IEEE J. Solid-State Circuits*, 20, 852–859, 1985.
19. J. W. Fattaruso and R. G. Meyer, MOS analog function synthesis, *IEEE J. Solid-State Circuits*, 22, 1059–1063, 1987.
20. I. M. Filanovsky and H. Baltes, CMOS Schmitt trigger design, *IEEE Trans. Circuits Syst*., 41, 46–49, 1994.
21. K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks*, 2, 183–192, 1989.
22. R. L. Geiger and E. Sánchez-Sinencio, Active filter design using operational transconductance amplifiers: A tutorial, *IEEE Circuits Devices Mag*., 1, 20–32, 1985.
23. B. Gilbert, Current-mode circuits from a translinear view point: A tutorial, in *Analogue IC Design: The Current-Mode Approach*, C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds., London: Peter Peregrinus Ltd., 1990.
24. J. Glover, Basic T matrix patterns for 2-port linear transformation networks in the real domain, *IEEE Trans. Circuit Theory*, 10, 495–497, 1974.
25. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed., New York: John Wiley & Sons, 1993.
26. A. Hausner, *Analog and Analog/Hybrid Computer Programming*, Englewood Cliffs, NJ: Prentice-Hall, 1971.
27. B. J. Hosticka, W. Brockherde, U. Kleine, and R. Schweer, Design of nonlinear analog switched-capacitor circuits using building blocks, *IEEE Trans. Circuits Syst*., 31, 354–368, 1984.
28. J. L. Huertas, J. A. Acha, and A. Gago, Design of general voltage or current controlled resistive elements and their applications to the synthesis of nonlinear networks, *IEEE Trans. Circuits Syst*., 27, 92–103, 1980.

29. J. L. Huertas, DT-adaptor: Applications to the design of nonlinear *n*-ports, *Int. J. Circuit Theory Appl.*, 8, 273–290, 1980.
30. J. L. Huertas and A. Rueda, Sectionwise piecewise polynomial function: Applications to the analysis and synthesis on nonlinear *n*-ports networks, *IEEE Trans. Circuits Syst.*, 31, 897–905, 1984.
31. J. L. Huertas, L. O. Chua, A. Rodríguez-Vázquez, and A. Rueda, Nonlinear switched-capacitor networks: Basic principles and piecewise-linear design, *IEEE Trans. Circuits Syst.*, 32, 305–319, 1985.
32. M. Ismail, Four-transistor continuous-time MOS transconductance, *Electr. Lett.*, 23, 1099–1100, 1987.
33. J. S. Jang, Neuro-fuzzy modeling: Architectures, analyses and applications, Ph. D. dissertation, University of California, Berkeley, CA, 1992.
34. C. Jansson, K. Chen, and C. Svensson, Linear, polynomial and exponential ramp generators with automatic slope adjustments, *IEEE Trans. Circuits Syst.*, 41, 181–185, 1994.
35. N. I. Khachab and M. Ismail, Linearization techniques for *n*th-order sensor models in MOS VLSI technology, *IEEE Trans. Circuits Syst.*, 38, 1439–1449, 1991.
36. Y. H. Kim and S. B. Park, Four-quadrant CMOS analog multiplier, *Electr. Lett.*, 28, 649–650, 1992.
37. K. Kimura, Some circuit design techniques using two cross-coupled, emitter-coupled pairs, *IEEE Trans. Circuits Syst.*, 41, 411–423, 1994.
38. O. Landolt, E. Vittoz, and P. Heim, CMOS self-biased Euclidean distance computing circuit with high dynamic range, *Electr. Lett.*, 28, 352–354, 1992.
39. N. R. Malik, G. L. Jackson, and Y. S. Kim, Theory and applications of resistor, linear controlled resistor, linear controlled conductor networks, *IEEE Trans. Circuits Syst.*, 31, 222–228, 1976.
40. A. I. Mees, M. F. Jackson, and L. O. Chua, Device modeling by radial basis functions, *IEEE Trans. Circuits Syst.*, 39, 19–27, 1992.
41. C. Michael and M. Ismail, *Statistical Modeling for Computer-Aided Design of MOS VLSI Circuits*, Boston, MA: Kluwer Academic, 1993.
42. C. A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, *Constr. Approx.*, 2, 11–22, 1986.
43. J. Moody and C. Darken, Fast learning in networks of locally-tuned processing units, *Neutral Comput.*, 1, 281–294, 1989.
44. R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, New York: John Wiley & Sons, 1986.
45. K. Nagaraj, K. Singhal, T. R. Viswanathan, and J. Vlach, Reduction of finite-gain effect in switched-capacitor filters, *Electr. Lett.*, 21, 644–645, 1985.
46. K. Nagaraj, A parasitic-insensitive area-efficient approach to realizing very large time constants in switched-capacitor circuits, *IEEE Trans. Circuits Syst.*, 36, 1210–1216, 1989.
47. R. W. Newcomb, Nonlinear differential systems: A canonic multivariable theory, *Proc. IEEE*, 65, 930–935, 1977.
48. M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, Matching properties of MOS transistors, *IEEE J. Solid-State Circuits*, 24, 1433–1440, 1989.
49. J. S. Peña-Finol and J. A. Connelly, A MOS four-quadrant analog multiplier using the quarter-square technique, *IEEE J. Solid-State Circuits*, 22, 1064–1073, 1987.
50. J. Platt, A resource allocating neural network for function interpolation, *Neural Comput.*, 3, 213–215, 1991.
51. T. Poggio and F. Girosi, Networks for approximation and learning, *Proc. IEEE*, 78, 1481–1497, 1990.
52. M. J. D. Powell, Radial basis functions for multivariate interpolation: A review, Rep 1985/NA12, Department Of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, UK, 1985.
53. S. Qin and R. L. Geiger, A 5-V CMOS analog multiplier, *IEEE J. Solid-State Circuits*, 22, 1143–1146, 1987.
54. J. K. Roberge, *Operational Amplifiers: Theory and Practice*, New York: John Wiley, 1975.

55. A. Rodríguez-Vázquez, J. L. Huertas, and A. Rueda, Low-order polynomial curve fitting using switched-capacitor circuits, *Proc. IEEE Int. Symp. Circuits Syst*., May, 40, 1123–1125, 1984.
56. A. Rodríguez-Vázquez and M. Delgado-Restituto, CMOS design of chaotic oscillators using state variables: A monolithic Chua's circuit, *IEEE Trans. Circuits Syst*., 40, 596–613, 1993.
57. A. Rodríguez-Vázquez and M. Delgado-Restituto, Generation of chaotic signals using current-mode techniques, *J. Intell. Fuzzy Syst*., 2, 15–37, 1994.
58. A. Rodríguez-Vázquez, R. Domínguez-Castro, F. Mederio, and M. Delgado-Restituto, High-resolution CMOS current comparators: Design and applications to current-mode functions generation, *Analog Integr. Circuits Signal Process*., 7(2), 149–165, 1995.
59. A. Rueda, J. L. Huertas, and A. Rodríguez-Vázquez, Basic circuit structures for the synthesis of piecewise polynomial one-port resistors, *IEEE Proc*., 132(Part G), 123–129, 1985.
60. S. Sakurai and M. Ismail, High frequency wide range CMOS analogue multiplier, *Electr. Lett*., 28, 2228–2229, 1992.
61. E. Sánchez-Sinencio, J. Ramírez-Angulo, B. Linares-Barranco, and A. Rodríguez-Vázquez, Operational transconductance amplifier-based nonlinear function synthesis, *IEEE J. Solid-State Circuits*, 24, 1576–1586, 1989.
62. E. Seevinck, *Analysis and Synthesis of Translinear Integrated Circuits*, Amsterdam: Elsevier, 1988.
63. D. H. Sheingold, *Nonlinear Circuits Handbook*, Norwood, CA: Analog Devices Inc., 1976.
64. J. Silva-Martínez and E. Sánchez-Sinencio, Analog OTA multiplier without input voltage swing restrictions, and temperature compensated, *Electr. Lett*., 22, 559–560, 1986.
65. S. Soclof, *Analog Integrated Circuits*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
66. B. Song, CMOS RF circuits for data communication applications, *IEEE J. Solid-State Circuits*, 21, 310–317, 1986.
67. L. Strauss, *Wave Generation and Shaping*, New York: McGraw-Hill, 1970.
68. G. C. Temes and K. Haug, Improved offset compensation schemes for switched-capacitor circuits, *Electr. Lett*., 20, 508–509, 1984.
69. Y. P. Tsividis, M. Banu, and J. Khoury, Continuous-time MOSFET-C filters in VLSI, *IEEE J. Solid-State Circuits*, 21, 15–30, 1986.
70. Y. P. Tsividis and J. O. Voorman, Eds., *Integrated Continuous-Time Filters*, New York: IEEE Press, 1993.
71. C. Turchetti and M. Conti, A general approach to nonlinear synthesis with MOS analog circuits, *IEEE Trans. Circuits Syst*., 40, 608–612, 1993.
72. R. Unbehauen and A. Cichocki, *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems*, Berlin: Springer-Verlag, 1989.
73. G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design: with Applications*, New York: McGraw-Hill, 1984.
74. F. Vidal and A. Rodríguez-Vázquez, A basic building block approach to CMOS design of analog neuro/fuzzy systems, *Proc. IEEE Int. Conf. Fuzzy Syst*., 1, 118–123, 1994.
75. E. Vittoz, Analog VLSI signal processing why, where and how? *Analog Integr. Circuits Signal Process*., 6, 27–44, 1994.
76. J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd ed., New York: Van Nostrand Reinhold, 1994.
77. Z. Wang, Analytical determination of output resistance and DC matching errors in MOS current mirrors, *IEEE Proc*., 137(Part G), 397–404, 1990.
78. G. A. Watson, *Approximation Theory and Numerical Methods*, Chichester, UK: John Wiley & Sons, 1980.
79. R. Wiegerink, *Analysis and Synthesis of MOS Translinear Circuits*, Boston: Kluwer Academic, 1993.
80. Y. J. Wong and W. E. Ott, *Function Circuits; Design and Applications*, New York: McGraw-Hill, 1976.
81. T. Yamakawa, A fuzzy inference engineer in nonlinear analog mode and its application to fuzzy control, *IEEE Trans. Neural Networks*, 4, 496–522, 1993.
82. J. M. Zurada, *Introduction to Artificial Neural Systems*, St. Paul, MN: West Publishing, 1992.

9

Representation, Approximation, and Identification

9.1 Introduction **9**-1
9.2 Representation **9**-2
Differential Equation and State-Space Representations • Input–Output Representation • Volterra Series Representation
9.3 Approximation **9**-10
Best Approximation of Systems (Operators) • Best (Uniform) Approximation of Signals (Functions) • Best Approximation of Linear Functionals • Artificial Neural Network for Approximation
9.4 Identification **9**-26
Linear Systems Identification • Nonlinear Systems Identification • Nonlinear Dynamic Systems Identification from Time Series
References **9**-31

Guanrong Chen
City University of Hong Kong

9.1 Introduction

Representation, approximation, and identification of physical systems, linear or nonlinear, deterministic or random, or even chaotic, are three fundamental issues in systems theory and engineering. To describe a physical system, such as a circuit or a microprocessor, we need a mathematical formula or equation that can represent the system both qualitatively and quantitatively. Such a formulation is what we call a mathematical representation of the physical system. If the physical system is so simple that the mathematical formula or equation, or the like, can describe it perfectly without error, then the representation is ideal and ready to use for analysis, computation, and synthesis of the system. An ideal representation of a real system is generally impossible, so that system approximation becomes necessary in practice. Intuitively, approximation is always possible. However, the key issues are what kind of approximation is good, where the sense of ''goodness'' must first be defined, of course, and how to find such a good approximation. On the other hand, when looking for either an ideal or a approximate mathematical representation for a physical system, one must know the system structure (the form of the linearity or nonlinearity) and parameters (their values). If some of these are unknown, then one must identify them, leading to the problem of system identification.

This chapter is devoted to a brief description of mathematical representation, approximation, and identification of, in most cases, nonlinear systems. As usual, a linear system is considered to be a special case of a nonlinear system, but we do not focus on linear systems in this chapter on nonlinear circuits. It is

known that a signal, continuous or discrete, is represented by a function of time. Hence, a signal can be approximated by other functions and also may be identified using its sampled data. These are within the context of ''representation, approximation, and identification,'' but at a lower level—one is dealing with functions. A system, in contrast, transforms input signals to output signals, namely, maps functions to functions, and is therefore at a higher level—it can only be represented by an operator (i.e., a mapping). Hence, while talking about representation, approximation, and identification in this chapter, we essentially refer to operators. However, we notice that two systems are considered to be equivalent over a set of input signals if and only if (iff) they map the same input signal from the set to the same output signal, regardless of the distinct structures of the two systems. From this point of view, one system is a good approximation of the other if the same input produces outputs that are approximately the same under certain measure. For this reason, we also briefly discuss the classical function approximation theory in this chapter.

The issue of system representation is addressed in Section 9.2, while approximation (for both operators and functions) is discussed in Section 9.3, leaving the system identification problem to Section 9.4. Limited by space, we can discuss only deterministic systems. Topics on stochastic systems are hence referred to some standard textbooks [13,17].

It is impossible to cover all the important subjects and to mention many significant results in the field in this short and sketchy chapter. The selections made only touch upon the very elementary theories, commonly used methods, and basic results related to the central topics of the chapter, reflecting the author's personal preference. In order to simplify the presentation, we elected to cite only those closely related references known to us, which may or may not be the original sources. From our citations, the reader should be able to find more references for further reading.

9.2 Representation

The scientific term ''representation'' as used here refers to a mathematical description of a physical system. The fundamental issue in representing a physical system by a mathematical formulation, called a mathematical model, is its correct symbolization, accurate quantization, and strong ability to illustrate and reproduce important properties of the original system.

A circuit consisting of some capacitor(s), inductor(s), and/or resistors(s), and possibly driven by a voltage source or a current source, is a physical system. In order to describe this system mathematically for the purpose of analysis, design, and/or synthesis, a mathematical model is needed. Any mathematical model, which can correctly describe the physical behavior of the circuit, is considered a mathematical representation of the circuit. A lower level mathematical representation of a circuit can, for instance, be a signal flow chart or a circuit diagram like the nonlinear Chua's circuit shown in Figure 9.1, which is discussed next.

A circuit, such as that shown in Figure 9.1, can be used to describe a physical system, including its components and its internal as well as external connections. However, it is not convenient for carrying out theoretical analysis or numerical computations. This is because no qualitative or quantitative description exists about the relations among the circuit elements and their dynamic behavior. Hence, a higher level mathematical model is needed to provide a qualitative and quantitative representation of the real physical circuit.

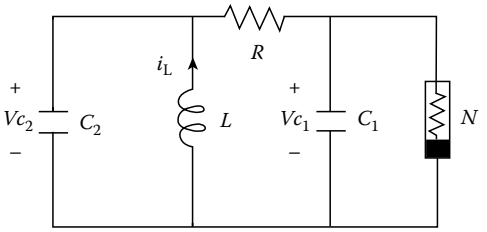


FIGURE 9.1 Chua's circuit.

Among several commonly used mathematical modeling approaches for various physical systems, differential equations, state-space formulations, I–O mappings, and functional series (particularly, the Volterra series) are the most important and useful, which have been very popular in the field of circuits

and systems engineering. In the following, we introduce these mathematical representation methods, along with some brief discussions of other related issues. Limited by space, detailed derivations are omitted.

9.2.1 Differential Equation and State-Space Representations

Mathematical modeling via differential equations and via state-space descriptions are the most basic mathematical representation methods. We illustrate the concept of mathematical modeling and the two representation methods by a simple, yet representative example: the nonlinear circuit in Figure 9.1. This circuit consists of one inductor L, two capacitors C_1 and C_2, one linear resistor R, and one nonlinear resistor, N, which is a nonlinear function of the voltage across its two terminals: $N = N(V_{C_1}(t))$. Let $i_L(t)$ be the current through the inductor L, and $V_{c_1}(t)$ and $V_{c_2}(t)$ be the voltages across C_1 and C_2, respectively. For the time being, let us remove the nonlinear resistor N from Figure 9.1 and consider the remaining linear circuit. This nonlinear resistor N is readded to the circuit with detailed discussions in Equation 9.6.

For this linear circuit without the resistor N, it follows from Kirchhoff's laws that

$$C_1 \frac{\mathrm{d}}{\mathrm{d}t} V_{C_1}(t) = \frac{1}{R}[V_{C_2}(t) - V_{C_1}(t)] \tag{9.1}$$

$$C_2 \frac{\mathrm{d}}{\mathrm{d}t} V_{C_2}(t) = \frac{1}{R}[V_{C_1}(t) - V_{C_2}(t)] + i_L(t) \tag{9.2}$$

$$L \frac{\mathrm{d}}{\mathrm{d}t} i_L(t) = -V_{C_2}(t) \tag{9.3}$$

By simple calculation we can eliminate both $V_{c_2}(t)$ and i_L, leaving a single ordinary differential equation on the unknown voltage $V_{c_1}(t)$ as follows:

$$\frac{\mathrm{d}^3}{\mathrm{d}t^3} V_{C_1}(t) + \frac{1}{R}\left(\frac{1}{C_1} + \frac{1}{C_2}\right) \frac{\mathrm{d}^2}{\mathrm{d}t^2} V_{C_1}(t) + \frac{1}{C_2 L} \frac{\mathrm{d}}{\mathrm{d}t} V_{C_1}(t) + \frac{1}{C_1 C_2 R L} V_{C_1}(t) = 0 \tag{9.4}$$

Once $V_{c_1}(t)$ is obtained from Equation 9.4, based on certain initial conditions, the other two unknowns, $V_{c_2}(t)$ and i_L, can be obtained by using Equations 9.1 and 9.3, successively. Hence, this third-order ordinary differential equation describes both qualitatively and quantitatively the circuit shown in Figure 9.1 (without the nonlinear resistor N). For this reason, Equation 9.4 is considered to be a mathematical representation, called a differential equation representation, of the physical linear circuit.

Very often, a higher-order, single-variable ordinary differential equation similar to Equation 9.4 is not as convenient as a first-order multivariable system of ordinary differential equations as is the original system of Equations 9.1 through 9.3, even when an analytic formulation of the solution is desired. Hence, a more suitable way for modeling a physical system is to introduce the concept of system state variables, which leads to a first-order higher dimensional system of ordinary differential equations.

If we introduce three state variables in Equations 9.1 through 9.3:

$$x_1(t) = V_{C_1}(t), \quad x_2(t) = V_{C_2}(t), \quad x_3(t) = i_L(t)$$

then we can rewrite those equations in the following vector form:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), & t \geq 0 \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \tag{9.5}$$

with an initial value $\mathbf{x}_0$ (usually given), where

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} -\dfrac{1}{RC_1} & \dfrac{1}{RC_1} & 0 \\ \dfrac{1}{RC_2} & -\dfrac{1}{RC_2} & \dfrac{1}{C_2} \\ 0 & -\dfrac{1}{L} & 0 \end{bmatrix}$$

in which $\mathbf{x}(t)$ is called the state vector of the system. Here, to be more general and for convenience in the discussions following, we formally added the term $B\mathbf{u}(t)$ to the system, in which B is a constant matrix and $\mathbf{u}(t)$ is called the control input of the system. In the present case, of course, $\mathbf{u} = 0$ and it is not important to specify B. However, note that $\mathbf{u}$ can be a nonzero external input to the circuit [19], which is discussed in more detail below.

This first-order, vector-valued linear ordinary differential equation is equivalent to the third-order differential equation representation (Equation 9.4) of the same physical circuit. A special feature of this state vector formulation is that with different initial state vectors and with zero control inputs, all the possible system state vectors together constitute a linear space of the same dimension [31]. Hence, Equation 9.5 is also called a linear state-space representation (or, a linear state-space description) for the circuit.

A few important remarks are in order. First, if the circuit is nonlinear, its state vectors do not constitute a linear space in general. Hence, its mathematical model in the state vector form should not be called a "state-space" representation. Note, however, that some of the linear system terminology such as state variables and state vectors usually make physical sense for nonlinear systems. Therefore, we use the term nonlinear state-variable representation to describe a first-order, vector-valued nonlinear ordinary differential equation of the form $\cdot\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$, where $\mathbf{f}(\cdot, \cdot, t)$ is generally a vector-valued nonlinear function. This is illustrated in more detail shortly.

Second, a linear state-space representation for a given physical system is not unique because one can choose different state variables. For example, in Equations 9.1 through 9.3 if we instead define $x_1 = V_{c_2}(t)$ and $x_2 = V_{c_1}(t)$, we arrive at a different linear state-space representation of the same circuit. However, we should note that if a linear nonsingular transformation of state vectors can map one state-space representation to another, then these two seemingly different representations are actually equivalent in the sense that the same initial values and control inputs will generate the same outputs (perhaps in different forms) through these two representations. Also worth noting is that not every circuit element can be used as a state variable, particularly for nonlinear systems. A basic requirement is that all the chosen state variables must be "linearly independent" in that the first-order, vector-valued ordinary differential equation has a unique solution (in terms of the control input) for any given initial values of the chosen state variables.

Finally, because A and B in the state-space representation (Equation 9.5) are both constant (independent of time), the representation is called a linear time-invariant system. If A or B is a matrix-valued function of time, then it will be called a linear time-varying system. Clearly, a time-invariant system is a special case of a time-varying system.

Now, let us return to the nonlinear circuit, with the nonlinear resistor N being connected to the circuit, as illustrated in Figure 9.1. Similar to Equations 9.1 through 9.3, we have the following circuit equations:

$$C_1 \frac{\mathrm{d}}{\mathrm{d}t} V_{C_1}(t) = \frac{1}{R}[V_{C_2}(t) - V_{C_1}(t)] - N(V_{C_1}(t)) \tag{9.6}$$

$$C_2 \frac{\mathrm{d}}{\mathrm{d}t} V_{C_2}(t) = \frac{1}{R}[V_{C_1}(t) - V_{C_2}(t)] + i_{\mathrm{L}}(t) \tag{9.7}$$

$$L \frac{\mathrm{d}}{\mathrm{d}t} i_{\mathrm{L}}(t) = -V_{C_2}(t) \tag{9.8}$$

Note that if the nonlinear resistor N is given by

$$\begin{aligned} N(V_{C_1}(t)) &= N(V_{C_1}(t); m_0, m_1) \\ &= m_0 V_{C_1}(t) + \frac{1}{2}(m_1 - m_0)\Big(\big|V_{C_1}(t) + 1\big| - \big|V_{C_1}(t) - 1\big|\Big) \end{aligned} \tag{9.9}$$

with $m_0 < 0$ and $m_1 < 0$ being two appropriately chosen constant parameters, then this nonlinear circuit is the well-known Chua's circuit [24].

It is clear that compared with the linear case, it would be rather difficult to eliminate two unknowns, particularly $V_{c_1}(t)$, in order to obtain a simple third-order, nonlinear differential equation that describes the nonlinear circuit. That is, it would often be inconvenient to use a higher-order, single-variable differential equation representation for a nonlinear physical system in general. By introducing suitable state variables, however, one can easily obtain a nonlinear state-variable representation in a first-order, vector-valued, nonlinear differential equation form. For instance, we may choose the following state variables:

$$x(\tau) = V_{C_1}(t), \quad y(\tau) = V_{C_2}(t), \quad \text{and} \quad z(\tau) = Ri_{\mathrm{L}}(t) \quad \text{with} \quad \tau = t/RC_2$$

where the new variable $z(\tau) = Ri_{\mathrm{L}}(t)$ and the rescaled time variable $\tau = t/RC_2$ are introduced to simplify the resulting representation of this particular circuit. Under this nonsingular linear transform, the previous circuit equations are converted to the following state-variable representation:

$$\begin{cases} \dot{x}(\tau) = p\big[-x(\tau) + y(\tau) - \tilde{N}(x(\tau))\big] \\ \dot{y}(\tau) = x(\tau) - y(\tau) + z(\tau) \\ \dot{z}(\tau) = -qy(\tau) \end{cases} \tag{9.10}$$

where

$p = C_2/C_1$
$q = R^2 C_2/L$

and

$$\begin{aligned} \tilde{N}(x(\tau)) &= N(x(\tau); \tilde{m}_0, \tilde{m}_1) \\ &= \tilde{m}_0 x(\tau) + \frac{1}{2}(\tilde{m}_1 - \tilde{m}_0)\big(\big|x(\tau) + 1\big| - \big|x(\tau) - 1\big|\big) \end{aligned} \tag{9.11}$$

with $\tilde{m}_0 = Rm_0$ and $\tilde{m}_1 = Rm_1$.

It is easy to see that this state-variable representation can be written as a special case in the following form, known as a canonical representation of Chua's circuit family:

$$\dot{\mathbf{x}}(\tau) = \mathbf{a} + A\mathbf{x}(\tau) + \sum_{i=1}^{k} \big|\mathbf{h}_i^{\mathrm{T}}\mathbf{x}(\tau) - \beta_i\big|\mathbf{c}_i + B\mathbf{u}(\tau) \tag{9.12}$$

namely, with $\mathbf{a} = 0$, $k = 2$, $\mathbf{h}_1 = \mathbf{h}_2 = [1\ 0\ 0]^{\mathrm{T}}$, $\beta_1 = -\beta_2 = -1$, $\mathbf{c}_1 = -\mathbf{c}_2 = \mathbf{H}(\tilde{m}_1 - \tilde{m}_0)$, $B\mathbf{u}(\tau)$ being a possible control input to the circuit [19], and

$$A = \begin{bmatrix} -\tilde{m}_0 - p & p & 0 \\ 1 & -1 & 1 \\ 0 & -q & 0 \end{bmatrix}$$

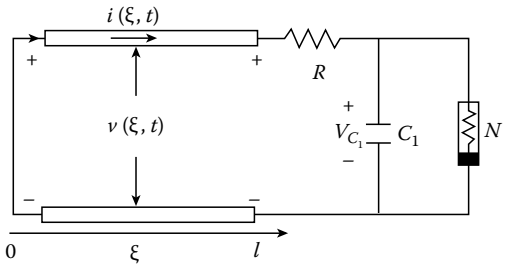


FIGURE 9.2 Time-delayed Chua's circuit.

The canonical (piecewise-linear) representation given by Equation 9.12 describes a large class of circuits that have very rich nonlinear dynamics [25,55].

Now, we return to Equations 9.6 through 9.8 and Figure 9.1. If we replace the $L - C_2$ part of Chua's circuit by a lossless transmission line (with the spatial variable ξ) of length l terminated on its left-hand side (at $\xi = 0$) by a short circuit, as depicted in Figure 9.2, then we obtain a time-delayed Chua's circuit [89]. This circuit has a partial differential equation representation of the form:

$$\begin{cases} \partial v/\partial \xi = -L\partial i(\xi,t)/\partial t \\ \partial i(\xi,t)/\partial \xi = -C_1\partial v(\xi,t)/\partial t \\ v(0,t) = 0 \\ i(l,t) = N(v(l,t) - e - Ri(l,t)) + C_1\partial[v(l,t) - Ri(l,t)]/\partial t \end{cases} \tag{9.13}$$

where $v(\xi, t)$ and $i(\xi, t)$ are the voltage and current, respectively, at the point $\xi \in [0, l]$ at time t, and $V_{c1} = e > 0$ is a constant, with the nonlinear resistor N satisfying

$$N(V_{C_1} - e) = \begin{cases} m_0(V_{C_1} - e) & |V_{C_1} - e| < 1 \\ m_1(V_{C_1} - e) - (m_1 - m_0)\mathrm{sgn}(V_{C_1} - e) & |V_{C_1} - e| \geq 1 \end{cases}$$

In general, systems that are described by (linear or nonlinear) partial differential equations, with initial-boundary value conditions, are studied under a unified framework of (linear or nonlinear) operator semigroup theory, and are considered to have an infinite-dimensional system representation [7].

9.2.2 Input–Output Representation

A state-variable representation of a nonlinear physical system generally can be written as

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad t \geq 0 \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \tag{9.14}$$

where

$\mathbf{f}(\cdot, \cdot, t)$ is a nonlinear, vector-valued function
$\mathbf{x}_0$ is a (given) initial value for the state vector $\mathbf{x}$ at $t = 0$
$\mathbf{u}$ is a control input to the system

Because not all state variables in the state vector $\mathbf{x}$ can be measured (observed) in a physical system, let us suppose that what can be measured is only part of $\mathbf{x}$, or a mixture of its components, expressed by a vector-valued function of $\mathbf{x}$ in the form

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), t), \quad t \geq 0 \tag{9.15}$$

where

$\mathbf{y}$ is called a (measurement or observation) output of the physical system
$\mathbf{g}$ is in general a lower dimensional vector-valued nonlinear function

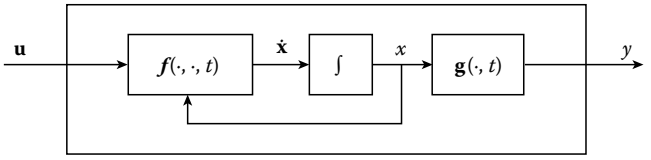


FIGURE 9.3 System I–O relationship.

As a particular case, **g** can be linear or, even more so, be $\mathbf{g}(\mathbf{x}(t), t) = \mathbf{x}(t)$ when all the components of the state vector are directly measurable.

If both $\mathbf{f} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ and $\mathbf{g} = \mathbf{g}(\mathbf{x}(t))$ are not explicit functions of the independent time variable t, the corresponding state-variable representation (Equations 9.14 and 9.15) is said to be autonomous.

It is clear that with both the system input **u** and output **y**, one can simply represent the overall physical system by its input-output (I–O) relationship, as illustrated in Figure 9.3.

Now, under certain mild conditions on the nonlinear function **f**, for a given control input **u**, and an initial value $\mathbf{x}_0$, the state-variable representation (Equation 9.14) has a unique solution, **x**, which depends on both **u** and $\mathbf{x}_0$. If we denote the solution as

$$\mathbf{x}(t) = \mathcal{F}(t; \mathbf{u}(t), \mathbf{x}_0) \tag{9.16}$$

where $\mathcal{F}$ is called an input-state mapping, then the overall I–O relationship shown in Figure 9.3 can be formulated as

$$\mathbf{y}(t) = \mathbf{g}(\mathcal{F}(t; \mathbf{u}(t), \mathbf{x}_0), t) \tag{9.17}$$

This is an I–O representation of the physical system having the state-variable representation (Equations 9.14 and 9.15).

As a simple example, let us consider the linear state-space representation (Equation 9.5), with a special linear measurement equation of the form $\mathbf{y}(t) = C\mathbf{x}(t)$, where C is a constant matrix. It is well known [31] that

$$\mathbf{y}(t) = C\mathcal{F}(t; \mathbf{u}(t), \mathbf{x}_0) = C\left\{e^{tA}\mathbf{x}_0 + \int_0^t e^{(t-\tau)A}B\mathbf{u}(\tau)\mathrm{d}\tau\right\}, \quad t \geq 0 \tag{9.18}$$

yielding an explicit representation formula for the I–O relationship of the linear circuit (together with the assumed measurement equation).

Note that because the state-variable representation (Equation 9.14) is not unique, as mentioned previously, this I–O representation is not unique in general. However, we note that if two state-variable representations are equivalent, then their corresponding I–O relationships also will be equivalent.

It is also important to note that although the above I–O relationship is formulated for a finite-dimensional open-loop system, it can also be applied to infinite-dimensional [7] and closed-loop systems [39]. In particular, similar to linear systems, many finite-dimensional, closed-loop nonlinear systems possess an elegant coprime factorization representation. The (left or right) coprime factorization representation of a nonlinear feedback system is a general I–O relationship that can be used as a fundamental framework, particularly suitable for studies of stabilization, tracking, and disturbance rejection. The problem is briefly described as follows. Let a nonlinear system (mapping) P be given, not necessarily stable, and assume that it has a right-coprime factorization $P = ND^{-1}$, where both N and D are stable (D^{-1} usually has the same stability as P). One is looking for two stable, nonlinear subsystems (mappings),

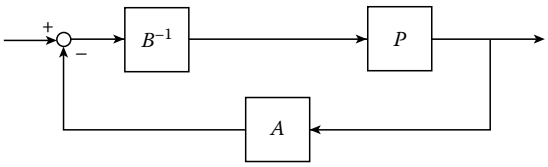


FIGURE 9.4 Right-coprime factorization of a nonlinear feedback system.

A and B^{-1}, representing feedback and feed-forward controllers, respectively, satisfying the Bezout identity

$$AN + BD = I$$

which are connected as shown in Figure 9.4, where B is also stable. If two controllers, A and B, can be found to satisfy such conditions, then even with an unstable P, the resulting closed-loop control system will be I–O, as well as internally, stable. In this sense, A and B together stabilize P.

For the left-coprime factorization, one simply uses formulas $P = D^{-1}N$ and $NA + DB = I$ instead and interchanges the two blocks of A and B^{-1} in Figure 9.4.

Taking into account causality and well-posedness of the overall closed-loop system, it is a technical issue as to how to construct the four subsystems A, B, D, and N, such that the preceding requirements can be satisfied. Some characterization results and construction methods are available in the literature [38,45,51,95].

9.2.3 Volterra Series Representation

Recall from the fundamental theory of ordinary differential equations that an explicit I–O representation of the overall system still can be found, even if the linear state-space representation (Equation 9.5) is time varying, via the state transition matrix $\Phi(t, \tau)$ determined by

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\Phi(t,\tau) = A(t)\Phi(t,\tau), & t \geq \tau \\ \Phi(\tau,\tau) = I \end{cases} \tag{9.19}$$

where I is the identity matrix. The formula, for the simple case $\mathbf{y}(t) = C(t)\mathbf{x}(t)$, is

$$\mathbf{y}(t) = C(t)\left\{\Phi(t,0)\mathbf{x}_0 + \int_0^t \Phi(t,\tau)B(\tau)\mathbf{u}(\tau)\mathrm{d}\tau\right\}, \quad t \geq 0 \tag{9.20}$$

For linear time-invariant systems, we actually have $\Phi(t, \tau) = e^{(t-\tau)A}$, so that Equation 9.20 reduces to the explicit formula (Equation 9.18).

For a nonlinear system, a simple explicit I–O representation with a single integral of the form (Equation 9.18 or Equation 9.20) is generally impossible. A natural generalization of such an integral formulation is the Volterra series representation. For simplicity, let us consider the one-dimensional case in which $y(t) = g(x(t), t) = x(t)$ below. A Volterra series representation for a nonlinear I–O relationship $\mathcal{F}(\cdot)$, convergent in some measure, is an infinite sum of integrals in the following form:

$$\mathcal{F}(t,u(t)) = \phi_0(t;x_0) + \int_0^t \phi_1(t,\tau_1)u(\tau_1)\mathrm{d}\tau_1 + \cdots + \int_0^t \cdots \int_0^{\tau_2} \phi_n(t,\tau_1,\ldots,\tau_n)u(\tau_1)\cdots u(\tau_n)\mathrm{d}\tau_1 \cdots \mathrm{d}\tau_n + \cdots \tag{9.21}$$

where $\{\phi_n\}_{n=0}^{\infty}$ are called the Volterra kernels of the series. Here, we note that this Volterra series representation can be extended easily to higher-dimensional systems.

For some representations $\mathcal{F}$, the corresponding Volterra series may have only finitely many nonzero terms in the above infinite sum. In this case, it is called a Volterra polynomial, which does not have convergence problem for bounded inputs, provided that all the integrals exist. In particular, when $\mathcal{F}$ is affine (or linear, if initial conditions are zero, so that $\phi_0 = 0$), its Volterra series has at most two nonzero terms, as given by Equations 9.18 and 9.20, and is called a first-order Volterra polynomial. In general, however, the Volterra series (Equation 9.21) is an infinite sum. Hence, the convergence of a Volterra series is a crucial issue in formulating such a representation for a given nonlinear I–O relationship [5,12,59,85].

In order to state a fundamental result about the convergence of a Volterra series, we must first recall that a mapping that takes a function to a (real or complex) value is called a functional and a mapping that takes a function to another function is called an operator. A functional may be considered to be a special operator if one views a value as a constant function in the image of the mapping. Clearly, the I–O relationship (Equation 9.17) and the Volterra series (Equation 9.21), including Volterra polynomials, are nonlinear operators. Recall also that an operator $\mathcal{T}: X \to Y$, where X and Y are normed linear spaces, is said to be continuous at $x \in X$ if $||x_n - x||_x \to 0$ implies $||\mathcal{T}(x_n) - \mathcal{T}(x)||_y \to 0$ as $n \to \infty$. Note that for a linear operator, if it is continuous at a point, then it is also continuous on its entire domain [34], but this is not necessarily true for nonlinear operators.

As usual, we denote by $C[0,T]$ and $L_p[0,T]$, respectively, the space of continuous functions defined on $[0,T]$ and the space of measurable functions f satisfying $\int_0^T |f(t)|^p \mathrm{d}t < \infty$ for $1 \le p < \infty$ or $\sup_{t\in[0,T]}|f(t)| < \infty$ for $p = \infty$. The following result [5] is an extension of the classical Stone–Weierstrass theorem [22,36,40].

THEOREM 9.1

Let X be either $C[0, T]$ or $L_p[0,T]$, with $1 \le p < \infty$, and Ω be a compact subset in X. Then, for any continuous operator $\mathcal{F}: \Omega \to L_q[0,T]$, where $(1/p) + (1/q) = 1$, and for any $\varepsilon > 0$, a Volterra polynomial $P_n(\cdot)$ exists, with n determined by ε, such that

$$\sup_{x\in\Omega} ||\mathcal{F}(x) - P_n(x)|_{L_q} < \varepsilon$$

In other words, $P_n \to \mathcal{F}$ uniformly on the compact subset $\Omega \subset X$ as $n \to \infty$.

In the literature, many variants of this fundamental convergence theorem exist under various conditions in different forms, including the $L_\infty[0,T]$ case [45,59,84,85]. We may also find different methods for constructing the Volterra kernels $\{\phi_n\}_{n=0}^{\infty}$ for $\mathcal{F}$ [83]. In addition, specially structured Volterra series representations abound for nonlinear systems, such as the Volterra series with finite memory [5], approximately finite memory [86], and fading memory [10].

Finally, it should be mentioned that in a more general manner, a few abstract functional series representations exist, including the generating power series representation for certain nonlinear systems [48], from which the Volterra series can be derived. Briefly, an important result is the following theorem [6,54,71,91].

THEOREM 9.2

Consider a nonlinear control system of the form

$$\begin{cases} \dot{x}(t) = g_0(x(t)) + \sum_{k=1}^{m} g_k(x(t))u_k(t), \quad t \in [0, T] \\ y(t) = h(x(t)) \end{cases}$$

where $h(\cdot)$ and $\{g_i(\cdot)\}_{i=0}^{m}$ are sufficiently smooth functionals, with an initial state x_0. If the control inputs satisfy $\max_0 \le \tau \le T\ |u_k(t)| < 1$, *then the corresponding output of this nonlinear system has a convergent functional series of the form*

$$y(t) = h(x_0) + \sum_{i=0}^{\infty} \sum_{k_0,\ldots,k_1=0}^{m} L_{g_{k_0}} \cdots L_{g_{k_i}} h(x_0) \int_0^t d\xi_{k_i} \cdots d\xi_{k_0} \tag{9.22}$$

where $L_g h(x_0) := [\partial h/\partial x] g(x)|_{x=x0}$ and ξ_k are defined by

$$\xi_0(t) = t \quad \xi_k(t) = \int_0^t u_k(\tau) d\tau, \quad k = 1, \ldots, m$$

with the notation

$$\int_0^t d\xi_{k_i} \cdots d\xi_{k_0} := \int_0^t d\xi_{k_i}(\tau) \int_0^t d\xi_{k_{i-1}} \cdots d\xi_{k_0}$$

Note that in order to guarantee the convergence of the functional series (Equation 9.22), in many cases it may be necessary for T to be sufficiently small.

Analogous to the classical Taylor series of smooth functions, a fairly general series representation for some nonlinear systems is still possible using polynomial operators, or the like [90]. As usual, however, the more general the presentation is, the less concrete the results. Moreover, a very general series expansion is likely to be very local, and its convergence is difficult to analyze.

9.3 Approximation

The mathematical term ''approximation'' used here refers to the theory and methodology of function (functional or operator) approximation. Mathematical approximation theory and techniques are important in engineering when one seeks to represent a set of discrete data by a continuous function, to replace a complicated signal by a simpler one, or to approximate an infinite-dimensional system by a finite-dimensional model, etc., under certain optimality criteria.

Approximation is widely used in system modeling, reduction, and identification, as well as in many other areas of control systems and signal processing [32]. A Volterra polynomial as a truncation of the infinite Volterra series (discussed earlier) serves as a good example of system (or operator) approximation, where the question ''In what sense is this approximation good?'' must be addressed further.

9.3.1 Best Approximation of Systems (Operators)

Intuitively, approximation is always possible. However, two key issues are the quality of the approximation and the efficiency of its computation (or implementation). Whenever possible, one would like to

have the best (or optimal) approximation, based on the available conditions and subject to all the requirements.

A commonly used criterion for best (or optimal) approximations is to achieve a minimum norm of the approximation error using a norm that is meaningful to the problem. Best approximations of systems (operators) include the familiar least-squares technique, and various other uniform approximations.

9.3.1.1 Least-Squares Approximation and Projections

Let us start with the most popular "best approximation" technique (the least-squares method), which can also be thought of as a projection, and a special min–max approximation discussed in the Section 9.3.1.2. Discrete data fitting by a continuous function is perhaps the best-known example of least-squares. The special structure of Hilbert space, a complete inner-product space of functions, provides a general and convenient framework for exploring the common feature of various least-squares approximation techniques. Because we are concerned with approximation of nonlinear systems rather than functions, a higher-level framework, the Hilbert space of operators, is needed. We illustrate such least-squares system (or operator) approximations with the following two examples.

First, we consider the linear space, H, of certain nonlinear systems that have a convergent Volterra series representation (Equation 9.21) mapping an input space X to an output space Y. Note that although a nontrivial Volterra series is a nonlinear operator, together they constitute a linear space just like nonlinear functions.

To form a Hilbert space, we first need an inner product between any two Volterra series. One way to introduce an inner product structure into this space is as follows. Suppose that all the Volterra series, $\mathcal{F}$: $X \to Y$, where both X and Y are Hilbert spaces of real-valued functions, have bounded admissible inputs from the set

$$\Omega = \{x \in X \big| \|x\|_X \leq \gamma < \infty\}$$

For any two convergent Volterra series of the form (Equation 9.21), say $\mathcal{F}$ and $\mathcal{G}$, with the corresponding Volterra kernel sequences $\{\phi_n\}$ and $\{\psi_n\}$, respectively, we can define an inner product between them via the convergent series formulation

$$\langle \mathcal{F}, \mathcal{G} \rangle_H := \sum_{n=0}^{\infty} \frac{\rho_n}{n!} |\phi_n \psi_n|$$

with the induced norm $||\mathcal{F}||_H = \langle \mathcal{F}, \mathcal{F} \rangle_H^{1/2}$, where the weights $\{\rho_n\}$ satisfy

$$\sum_{n=0}^{\infty} \frac{1}{\rho_n} \frac{\gamma^{2n}}{n!} < \infty$$

Recall also that a reproducing kernel Hilbert space $\tilde{H}$ is a Hilbert space (of real-valued functions or operators) defined on a set S, with a reproducing kernel $K(x, y)$, which belongs to $\tilde{H}$ for each fixed x or y in S and has the property

$$\langle K(x, y), \mathcal{F}(y) \rangle_{\tilde{H}} = \mathcal{F}(x) \quad \forall \mathcal{F} \in \tilde{H} \text{ and } \forall x, y \in S$$

Using the notation defined above, the following useful result was established [43,45] and is useful for nonlinear systems identification (see Theorem 9.23).

THEOREM 9.3

The family of all the convergent Volterra series of the form (Equation 9.21) *that maps the bounded input set* Ω *to Y constitutes a reproducing kernel Hilbert space with the reproducing kernel*

$$K(x,y) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{1}{\rho_n} \langle x,y\rangle_X^n, \quad x,y \in \Omega \subset X \tag{9.23}$$

The reproducing kernel Hilbert space H defined above is called a generalized Fock space [46]. For the special case in which $\rho_n \equiv 1$, its reproducing kernel has a nice closed-form formula as an exponential operator $K(x,y) = e^{\langle xy\rangle}$.

Now, suppose that a nonlinear system $\mathcal{F}$ is given, which has a convergent Volterra series representation (Equation 9.21) with infinitely many nonzero terms in the series. For a fixed integer $n \geq 0$, if we want to find an nth-order Volterra polynomial, denoted $V_n^\star$, from the Hilbert space H such that

$$\left\|\mathcal{F} - V_n^\star\right\|_{\mathrm{H}} = \inf_{V_n \in H} \left\|\mathcal{F} - V_n\right\|_{\mathrm{H}} \tag{9.24}$$

then we have a best approximation problem in the least-squares sense. To solve this optimization problem is to find the best Volterra kernels $\{\phi_k(t)\}_{k=0}^n$ over all the possible kernels that define the Volterra polynomial V_n, such that the minimization (Equation 9.24) is achieved.

Note that, if we view the optimal solution $V_n^\star$ as the projection of $\mathcal{F}$ onto the $(n+1)$-dimensional subspace of H, then this least-squares minimization is indeed a projection approximation. It is then clear, even from the Hilbert space geometry (see Figure 9.5), that such an optimal solution, called a best approximant, always exists due to the norm-completeness of Hilbert space and is unique by the convexity of inner product space.

As a second example, let H be a Hilbert space consisting of all the linear and nonlinear systems that have an nth-order Taylor series representation of the form

$$P_n(\,\cdot\,) = \sum_{k=0}^{n} \alpha_k(t) M_k(\cdot) = \alpha_0(t) + \alpha_1(t)(\cdot)(t) + \cdots + \alpha_n(t)(\cdot)^n(t) \tag{9.25}$$

where

$M_k(\cdot) := (\cdot)^k$ is the monomial operator of degree k

$\{\alpha_k\}_{k=0}^n$ are continuous real-valued functions satisfying certain conditions arising from some basic properties of both the domain and the range of the operator

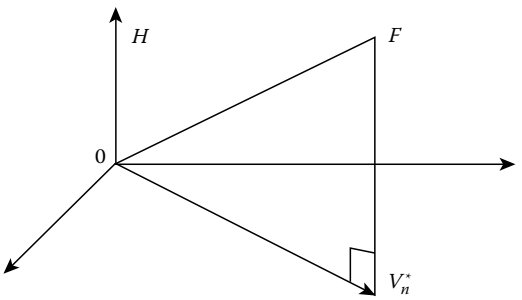


FIGURE 9.5 Projection in a Hilbert space.

Suppose that the monomial operators $\{M_k\}_{k=0}^\infty$ are orthogonal under the inner product of H. Given an lth-order polynomial operator P_l with $\alpha_l(t) \neq 0$ almost everywhere, if for a fixed integer $n < l$ we want to find an nth-order polynomial operator $P_n^\star$ of the form (Equation 9.25) from H, such that

$$\left\|P_l - P_n^\star\right\|_{\mathrm{H}} = \inf_{P_n \in H} \left\|P_l - P_n\right\|_{\mathrm{H}} \tag{9.26}$$

then we have a best approximation problem in the least-squares sense. To solve this optimization problem is to find the best coefficient

functions $\{\alpha_k(t)\}_{k=0}^n$ over all possible functions that define the polynomial operator P_n. Again, because the optimal solution is the projection of P_l onto the $(n+1)$-dimensional subspace H of a Hilbert space to which P_l belongs, it always exists and is unique.

We now state a general result of least-squares approximation for systems, which is a straightforward generalization of the classical result of least-squares approximation for functions [22,36].

THEOREM 9.4

Let H be a Hilbert space of nonlinear operators, and let H_n be its n-dimensional subspace. Then, given an $\mathcal{F} \in H$, the least-squares approximation problem

$$\left\|\mathcal{F}-\mathcal{N}_n^*\right\|_{\mathrm{H}} = \inf_{\mathcal{N}_n \in H_n} \left\|\mathcal{F}-\mathcal{N}_n\right\|_{\mathrm{H}}$$

is always uniquely solvable, with the optimal solution given by

$$\mathcal{N}^*(\cdot) = \sum_{k=1}^{n} \langle \mathcal{F}, h_k \rangle_H h_k(\cdot)$$

where $\{h_k)\}_{k=1}^n$ is an orthonormal basis of H_n.

A more general setting is to replace the Hilber space H by a Banach space (a complete normed linear space, such as L_1 and L_∞, which may not have an inner product structure). This extension includes the Hilbert space setting as a special case, but generally does not have so many special features. Even the existence and uniqueness of best approximants cannot be taken for granted in general—not even for the simpler case of best approximation of real-valued functions—if a Banach (non-Hilbert) space is considered [73]. Nevertheless, the following result is still convenient to use [22].

THEOREM 9.5

Let B be a uniformly convex Banach space and Ω be a closed convex set in B. Then, for any given $\mathcal{F} \in B$, the optimal approximation problem

$$\left\|\mathcal{F}-\omega^*\right\|_B = \inf_{\omega \in \Omega} \left\|\mathcal{F}-\omega\right\|_B$$

has a unique solution.

Here, a space (or subset) B is said to be uniformly convex if, for any $\varepsilon > 0$, there exists a $\|f\|_B = \|g\|_B = 1$ and $\|1/2(f+g)\|_B > 1-\delta$ together imply $\|f-g\|_B < \varepsilon$. Geometrically, a disk is uniformly convex while a triangle is only convex, but not uniformly so. It is then intuitively clear that for a given point outside (or inside) a disk, only a single point exists in the disk that has the shortest distance to the given point. However, this is not always true for a nonuniform case. In fact, a best approximation problem in the general Banach space setting has either a unique solution or has infinitely many solutions (if it is solvable), as can be seen from the next result [32].

THEOREM 9.6

Let Ω be a closed convex set in a Banach space B, and $\omega_1^\star$ and $\omega_2^\star$ be two optimal solutions of the best approximation problem

$$\left\|\mathcal{F} - \omega^\star\right\|_B = \inf_{\omega\in\Omega}\left\|\mathcal{F} - \omega\right\|_B$$

Then, any convex combination of $\omega_1^\star$ and $\omega_2^\star$ in the form

$$\omega^\star = a\omega_1^\star + (1-a)\omega_2^\star, \quad 0 \le a \le 1$$

is also an optimal solution of the problem.

Usually, a best approximant (if it exists) for an optimal approximation problem in a Banach space is also called a (minimal) projection of the given operator from a higher-dimensional subspace onto a lower-dimensional subspace. In this extension, the projection has no simple geometric meaning of ''orthonormality'' due to the lack of an inner product structure. However, a projection operator with a unity norm in the Banach space setting is a natural generalization of the orthonormal projection in the Hilbert space framework.

9.3.1.2 Min–Max (Uniform) Approximation

It is clear from the least-squares approximation formulation that if the given nonlinear representation (operator) $\mathcal{F}$ and the lower-order approximant (used to approximate $\mathcal{F}$) do not have the same structure (the same type of series), then the least-squares approximation cannot be applied directly or efficiently.

To introduce another approach, we first recall that for two given normed linear spaces X and Y and for a given bounded subset Ω of X, with $0 \in \Omega$, the operator norm of a nonlinear operator $\mathcal{N}$: $\Omega \to Y$ satisfying $\mathcal{N}(0) = 0$, can be defined as

$$\left|\left\|\mathcal{N}\right\|\right| = \sup_{\substack{x,y\in\Omega \\ x\neq y}} \frac{\left\|\mathcal{N}(x) - \mathcal{N}(y)\right\|_y}{\left\|x - y\right\|_x} \tag{9.27}$$

Thus, given a norm-bounded nonlinear operator $\mathcal{F}$, representing a given physical system, we may consider the problem of finding another norm-bounded nonlinear operator $\mathcal{N}^\star$ from a certain class 1 of desired nonlinear operators (systems), not necessarily having the same structure as $\mathcal{F}$, to best approximate $\mathcal{F}$ in the sense that

$$\left|\left\|\mathcal{F} - \mathcal{N}^\star\right\|\right| = \inf_{\mathcal{N}\in N}\left|\left\|\mathcal{F} - \mathcal{N}\right\|\right| \tag{9.28}$$

For example, N can be the family of nth-order Volterra polynomilas or nth-order polynomial operators discussed previously. Commonly used function spaces X and Y include the space of all continuous functions, the standard L_p space (or l_p for the discrete case), and the Hardy space H_p (for complex-variable functions [32]), with $1 \le p \le \infty$.

Because the nonlinear operator norm defined by Equation 9.27 is a sup (max) norm and this optimization is an inf (min) operation, the best approximation problem (Equation 9.28) is called a min–max approximation. Note also that because the nonlinear operator norm (Equation 9.27) is defined over all the bounded inputs in the set Ω, this approximation is uniform, and thus independent of each individual input function of the set Ω. For this reason, this approximation is also called a uniform approximation, indicating that the best approximant is the optimal solution over *all* input functions.

It should be noted that both existence and uniqueness of best approximation solutions to the min–max approximation problem (Equation 9.28) must be investigated according to the choice of the operator family N and the I–O spaces X and Y, which generally cannot be taken for granted, as previously discussed.

An important and useful class of nonlinear operators which can be put into a Banach space setting with great potential in systems and control engineering is the family of generalized Lipschitz operators [45]. To introduce this concept, we first need some notation. Let X be a Banach space of real-valued functions defined on $[0, \infty)$ and, for any $f \in X$ and any $T[0, \infty)$, define

$$[f]_{\mathrm{T}}(t) = \begin{cases} f(t), & t < T \\ 0, & t > T \end{cases}$$

Then, form a normed linear space X^{e}, called the extended linear space associated with $\boldsymbol{X}$, by

$$X^{\mathrm{e}} = \left\{ f \in X \middle| \left\| [f]_{\mathrm{T}} \right\|_{\mathrm{X}} < \infty, \forall T < \infty \right\}$$

For a subset $D \subseteq X^{\mathrm{e}}$, any (linear or nonlinear) operator $\mathcal{G}$: $D \to Y^{\mathrm{e}}$ satisfying

$$\left\| [\mathcal{G}(x_1)]_{\mathrm{T}} - [\mathcal{G}(x_2)]_{\mathrm{T}} \right\|_{\mathrm{Y}} \leq L \left\| [x_1]_{\mathrm{T}} - [x_2]_{\mathrm{T}} \right\|_{\mathrm{X}}, \quad \forall x_1, x_2 \in D, \quad \forall T \in [0, \infty)$$

for some constant $L < \infty$, is called a generalized Lipschitz operator defined on D. The least of such constants L is given by the seminorm of the operator $\mathcal{G}$:

$$\left\| \mathcal{G} \right\| : \sup_{T \in [0,\infty)} \sup_{\substack{x_1, x_2 \in D \\ [x_1]_{\mathrm{T}} \neq [x_2]_{\mathrm{T}}}} \frac{\left\| [\mathcal{G}(x_1)]_{\mathrm{T}} - [\mathcal{G}(x_2)]_{\mathrm{T}} \right\|_{Y}}{\left\| [x_1]_{\mathrm{T}} - [x_2]_{\mathrm{T}} \right\|_{X}}$$

and the operator norm of $\mathcal{G}$ is defined via this seminorm by

$$\left\| \mathcal{G} \right\|_{\mathrm{Lip}} = \left\| \mathcal{G}(x_0) \right\|_{Y} + \left\| \mathcal{G} \right\|$$

for an arbitrarily chosen and fixed $x_0 \in D$. The following result has been established [45].

THEOREM 9.7

The family of generalized Lipschitz operators

$$\mathrm{Lip}(D, Y^{\mathrm{e}}) = \{ \mathcal{G}: D \subseteq X^{\mathrm{e}} \to Y^{\mathrm{e}} \Big| \left\| \mathcal{G} \right\|_{\mathrm{Lip}} < \infty \text{ on } D \}$$

is a Banach space.

Based on this theorem, a best approximation problem for generalized Lipschitz operators can be similarly formulated, and many fundamental approximation results can be obtained. In addition, generalized Lipschitz operators provide a self-unified framework for both left and right coprime factorization representations of nonlinear feedback systems. Under this framework, the overall closed-loop system shown in Figure 9.4 can have a causal, stable, and well-posed coprime factorization representation, which can be applied to optimal designs such as tracking and disturbance rejection [45].

We now discuss briefly a different kind of min–max (uniform) approximation: the best Hankel norm approximation, where the norm (Equation 9.27) is replaced by the operator norm of a Hankel operator defined as follows [32,77]. Consider, for instance, the transfer function

$$H(z) = \alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \cdots$$

of a discrete time linear time-invariant system. The Hankel operator associated with this series is defined as the infinite matrix

$$\Gamma_\alpha := \left[\alpha_{|i-j|}\right] = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \cdots \\ \alpha_1 & \alpha_2 & \cdots & \\ \alpha_2 & \cdots & & \\ \vdots & & & \end{bmatrix}$$

which is a linear operator on a normed linear space of sequences. The operator norm of Γ_α over the l_2 space is called the Hankel norm of Γ_α.

One important feature of the Hankel operators is reflected in the following theorem [32,77].

THEOREM 9.8

An infinite Hankel matrix has a finite rank iff its corresponding functional series is rational (*it sums up to a rational function*); *and this is true iff the rational series corresponds to a finite-dimensional bilinear system.*

Another useful property of Hankel operators in system approximation is represented in the following theorem [28].

THEOREM 9.9

The family of compact Hankel operators is an M-ideal in the space of Hankel operators that are defined on a Hilbert space of real-valued functions.

Here, a compact operator is one that maps bounded sets to compact closures and an M-ideal is a closed subspace X of a Banach space Z such that $X^\perp$, the orthogonal complemental subspace of X in Z, is the range of the projection P from the dual space Z^* to $X^\perp$ that has the property

$$\|f\| = \|P(f)\| + \|f - P(f)\| \quad \forall f \in Z^*$$

The importance of the M-ideal is that it is a proximinal subspace with certain useful approximation characteristics, where the proximinal property is defined as follows. Let $L(X)$ and $C(X)$ be the classes of bounded linear operators and compact operators, respectively, both defined on a Banach space X. If every $\mathcal{L} \in L(X)$ has at least one best approximant from $C(X)$, then $C(X)$ is said to be proximinal in $L(X)$. A typical result would be the following: for any $1 < p < \infty$, $C(l_p)$ is proximinal in $L(l_p)$. However, $C(X)$ is not proximinal in $L(X)$ if $X = C[a, b]$, the space of continuous functions defined on $[a, b]$, or $X = L_p[a, b]$ for all $1 < p < \infty$ except $p = 2$.

9.3.2 Best (Uniform) Approximation of Signals (Functions)

Best approximations of signals for circuits and systems are also important. For example, two (different) systems (e.g., circuits) are considered to be equivalent over a set Ω of admissible input signals iff the same input from Ω yields the same outputs through the two systems. Thus, the problem of using a system to best approximate another may be converted, in many cases, to the best approximation problem for their output signals.

A signal is a function of time, usually real valued and one-dimensional. The most general formulation for best approximation of functions can be stated as follows. Let X be a normed linear space of real-valued functions and Ω be a subset of X. For a given f in X but not in Ω, find a $g^\star \in \Omega$ such that

$$\left\|f - g^\star\right\|_X = \inf_{g\in\Omega} \left\|f - g\right\|_X \tag{9.29}$$

In particular, if $X = L_\infty$, l_∞, or H_∞, the optimal solution is the best result for the worst case.

If such a $g^\star$ exists, then it is called a best approximant of f from the subset Ω. In particular, if $\Omega_1 \subset \Omega_2 \subset \cdots$ is a sequence of subspaces in X, such that $\overline{\cup\Omega_n} = X$, an important practical problem is to find a sequence of best approximants $g_n^\star \in \Omega_n$ satisfying the requirement (Equation 9.29) for each $n = 1, 2, \ldots$, such that $||g_n^\star - g^\star||_X \to 0$ as $n \to \infty$. In this way, for each n, one may be able to construct a simple approximant $g_n^\star$ for a complicated (even unknown) function f, which is optimal in the sense of the min–max approximation (Equation 9.29).

Existence of a solution is the first question about this best approximation. The fundamental result is the following [22,36].

THEOREM 9.10

For any $f \in X$, a best approximant $g^\star$ of f in Ω always exists, if Ω is a compact subset of X; or Ω is a finite-dimensional subspace of X.

Uniqueness of a solution is the second question in approximation theory, but it is not as important as the existence issue in engineering applications. Instead, characterization of a best approximant for a specific problem is significant in that it is often useful for constructing a best approximant.

As a special case, the preceding best approximation reduces to the least-squares approximation if X is a Hilbert space. The basic result is the following (compare it with Theorem 9.4, and see Figure 9.5).

THEOREM 9.11

Let H be a Hilbert space of real-valued functions, and let H_n be its n-dimenstional subspace. Then, given an $f \in H$, the least-squares approximation problem

$$\left\|f - h_n^\star\right\|_{\mathrm{H}} = \inf_{h_n\in H_n} \left\|f - h_n\right\|_{\mathrm{H}}$$

is always uniquely solvable, with the optimal solution given by

$$h_n^\star(t) = \sum_{k=1}^{n} \langle f, h_k\rangle_{\mathrm{H}} h_k(t)$$

where $\{h_k\}_{k=1}^n$ is an orthonormal basis of H_n.

Here, the orthonormal basis of H_n is a Chebyshev system, a system of functions which satisfy the Haar condition that the determinant of the matrix $[h_i(t_j)]$ is nonzero at n distinct points $t_1 < \cdots < n_n$ in the domain. Chebyshev systems include many commonly used functions, such as algebraic and trigonometric polynomials, splines, and radial functions. Best approximation by these functions is discussed in more detail below.

We remark that the least-squares solution shown in Theorem 9.11 is very general, which includes the familiar truncations of the Fourier series [36] and the wavelet series [29] as best approximation.

9.3.2.1 Polynomial and Rational Approximations

Let π_n be the space of all algebraic polynomials $p_n(t)$ of degree not greater than n. For any continuous function $f(t)$ defined on $[a, b]$, one is typically looking for a best approximant $g_n^* p_n^* \in \pi_n$ for a fixed n, such that

$$\left\| f - p_n^* \right\|_{L_\infty[a,b]} = \min_{p_n \in \pi_n} \left\| f - p_n \right\|_{L_\infty[a,b]} \tag{9.30}$$

This is a best (min–max and uniform) algebraic polynomial approximation problem. Replacing the algebraic polynomials by the nth-order trigonometrix polynomials of the form $\sum_{k=0}^{n}(a_k \cos(kt) + b_k \sin(kt))$ changes the problem to the best trigonometric polynomial approximation, in the same sense as the best algebraic polynomial approximation, for a given function $f \in C[-\pi, \pi]$. This can be much further extended to any Chebyshev system, such as the radial basis functions and polynomial spline functions, which are discussed later. According to the second part of Theorem 9.10, the best uniform polynomial approximation problem (Equation 9.30) always has a solution that, in this case, is unique. Moreover, this best approximant is characterized by the following important sign-alternation theorem. This theorem is also valid for the best uniform approximation from any other Chebyshev system [22,36].

THEOREM 9.12

The algebraic polynomial p_n^* is a best uniform approximant of $f \in C[a, b]$ from π_n iff there exist $n + 2$ points $a \le t_0 < \cdots < t_{n+1} \le b$ such that

$$f(t_k) = p_n^*(t_{k-} = c(-1)^k \left\| f - p_n^* \right\|_{L_\infty[a,b]}, \quad k = 0, 1, \ldots, n+1$$

where $c = 1$ or -1.

An efficient Remes (exchange) algorithm is available for constructing such a best approximant [79].

Another type of function is related to algebraic polynomials: the algebraic rational functions of the form $r_{n,m}(t) = p_n(t)/q_m(t)$, which has finite values on $[a, b]$ with coprime $p_n \in \pi_n$ and $q_m \in \pi_m$. We denote by $R_{n,m}$ the family of all such rational functions, or a subset of them, with fixed integers $n \ge 0$ and $m \ge 1$. Although $R_{n,m}$ is not a compact set or a linear space, the following result can be established [22].

THEOREM 9.13

For any given function $f \in C[a, b]$, there exists a unique $r_{n,m}^*(t) \in R_{n,m}$ such that

$$\left\| f - r_{n,m}^* \right\|_{L_\infty[a,b]} = \inf_{r_{n,m} \varepsilon R_{n,m}} \left\| f - r_{n,m} \right\|_{L\infty[a,b]} \tag{9.31}$$

The optimal solution $r_{n,m}^*(t)$ of Equation 9.31 is called the best uniform rational approximant of $f(t)$ on $[a, b]$ from $R_{n,m}$.

Note that the unique best rational approximant may have different expressions unless it is coprime, as assumed previously. The following theorem [22] characterizes such a best approximant, in which we use $d(p_n)$ to denote the actual degree of p_n, $0 \le d(p_n) \le n$.

THEOREM 9.14

A rational function $r^\star_{n,m} = p^\star_n / q^\star_m$ is a best uniform approximant of $f \in C[a, b]$ from $R_{n,m}$ iff there exist s points $a \le t_1 < \cdots < t_s \le b$, with $s = 2 + min\{n + d(q_m), m + d(p_n)\}$, such that

$$f(t_k) - r^\star_{n,m}(t_k) = c(-1)^k \left\| f - r^\star_{n,m} \right\| L_{\infty[a,b]}, \quad k = 1, \ldots, s$$

where $c = 1$ or -1.

The Remes (exchange) algorithm [79] also can be used for constructing a best rational approximant.

An important type of function approximation, which utilizes rational functions, is the Padé approximation. Given a formal power series of the form

$$f(t) = c_0 + c_1 t + c_2 t^2 + \cdots, \quad t \in [-1, 1]$$

not necessarily convergent, the question is to find a rational function $p_n(t)/q_m(t)$, where n and m are both fixed, to best approximate $f(t)$ on $[-1, 1]$, in the sense that

$$\left| f(t) - \frac{p_n(t)}{q_m(t)} \right| \le c|t|^l, \quad t \in [-1, 1] \tag{9.32}$$

for a "largest possible" integer l. It turns out that normally the largest possible integer is $l = n + m + 1$. If such a rational function exists, it is called the $[n, m]$th-order Padé approximant of $f(t)$ on $[-1, 1]$. The following result is important [22].

THEOREM 9.15

If $f(t)$ is $(n + m + 1)$ times continuously differentiable in a neighborhood of $t = 0$, then the $[n, m]$th-order Padé approximant of $f(t)$ exists, with $l > n$. If $l \le n + m + 1$, then the coefficients $\{a_k\}_{k=0}^n$ and $\{b_k\}_{k=0}^m$ of $p_n(t)$ and $q_m(t)$ are determined by the following linear system of algebraic equations:

$$\sum_{j=0}^{i} \frac{f^j(0)}{j!} b_{i-j} = a_i, \quad i = 0, 1, \ldots, l - 1$$

with $a_{n+j} = b_{m+j} = 0$ for all $j = 1, 2, \ldots$. Moreover, if p_n/q_m is the $[n, m]$th-order Padé approximant of $f(t) = \sum^\infty \{a_k\}_{k=0}^n f_k t^k$, then the approximation error is given by

$$\left| f(t) - \frac{p_n(t)}{q_m(t)} \right| = \sum_{k=n+1}^{\infty} \left(\sum_{j=0}^{m} f_{k-j} b_j \right) \frac{t^k}{q_m(t)}, \quad t \in [-1, 1]$$

Padé approximation can be extended from algebraic polynomials to any other Chebyshev systems [22].

9.3.2.2 Approximation via Splines and Radial Functions

Roughly speaking, spline functions, or simply splines, are piecewise smooth functions that are structurally connected and satisfy some special properties. The most elementary and useful splines are polynomial splines, which are piecewise algebraic polynomials, usually continuous, with a certain degree of smoothness at the connections. More precisely, let

$$a = t_0 < t_1 < \cdots < t_n < t_{n+1} = b$$

be a partition of interval $[a, b]$. The polynomial spline of degree m with knots $\{t_k\}_{k=1}^{n}$ on $[a, b]$ is defined to be the piecewise polynomial $g_{\mathrm{m}}(t)$ that is a regular algebraic polynomial of degree m on each subinterval $[t_k, t_{k+1}]$, $k = 0, \ldots, n$, and is $(m-1)$ times continuously differentiable at all knots [41,88]. We denote the family of these algebraic polynomial splines by $S_m(t_1, \ldots, t_n)$, which is an $(n+m+1)$-dimensional linear space.

Given a continuous function $f(t)$ on $[a, b]$, the best uniform spline approximation problem is to find a $g_m^{\star} \in S_m(t_1, \ldots, t_n)$ such that

$$\left\| f - g_{\mathrm{m}}^{\star} \right\|_{L_\infty[a,b]} = \inf_{g_{\mathrm{m}} \in S_{\mathrm{m}}} \left\| f - g_{\mathrm{m}} \right\|_{L_\infty[a,b]} \tag{9.33}$$

According to the second part of Theorem 9.10, this best uniform approximation problem always has a solution. A best spline approximant can be characterized by the following sign-alteration theorem [72], which is a generalization of Theorem 9.12, from polynomials to polynomial splines.

THEOREM 9.16

The polynomial spline $g_m^{\star}(t)$ is a best uniform approximant of $f \in C\,[a, b]$ from $S_m(t_1, \ldots, t_n)$ iff there exists a subinterval $[t_r, t_{r+s}] \subset [a, b]$, with integers r and $s \leq 1$, such that the maximal number γ of sign-alteration points on this subinterval $[t_r, t_{r+s}]$, namely,

$$f(t_k) - g_{\mathrm{m}}(t_k) = c(-1)^k \left\| f - g_{\mathrm{m}} \right\|_{L_\infty[a,b]}, \quad t_k \in [t_r, t_{r+s}], \quad k = 1, \ldots, \gamma$$

satisfies $\gamma \geq m + s + 1$, where $c = 1$ or -1.

Polynomial splines can be used for least-squares approximation, just like regular polynomials, if the L_∞-norm is replaced by the L_2-norm in Equation 9.33. For example, B-splines, i.e., basic splines with a compact support, are very efficient in least-squares approximation. The spline quasi-interpolant provides another type of efficient approximation, which has the following structure

$$g_{\mathrm{m}}(t) = \sum_k f(t_k) \phi_k^m(t) \tag{9.34}$$

and can achieve the optimal approximation order, where $\{\phi_k^m\}$ is a certain linear combination of B-splines of order m [18].

Spline functions have many variants and generalizations, including natural splines, perfect splines, various multivariate splines, and some generalized splines defined by linear ordinary or partial differntial operators with initial-boundary conditions [27,41,42,44,88].

Splines are essentially local, in the sense of having compact supports, perhaps with the exception perhaps of the thin-plate splines [94], where the domains do not have a boundary.

Radial functions are global, with the property $\phi(r) \to \infty$ as $r \to \infty$ and, normally, $\phi(0) = 0$. Well-conditioned radial functions include $|r|^{2m+1}$, $r^{2m}\log(r)$, $(r^2 + a^2)^{\pm 1/2}$, $0 < a \ll 1$, etc. [80]. Many radial functions are good candidates for modeling nonlinear circuits and systems [63,64]. For example, for l distinct points $\mathbf{t}_1, \ldots, \mathbf{t}_l$ in R^n, the radial functions $\{\phi(|\mathbf{t} - \mathbf{t}_k|)\}_{k=1}^{l}$ are linearly independent, and thus the minimization

$$\min_{\{c_k\}} \left| f(\mathbf{t}) - \sum_{k=1}^{l} c_k \phi\left(\left|\mathbf{t} - \mathbf{t}_k\right|\right) \right|^2 \tag{9.35}$$

at some scattered points can yield a best least-squares approximant for a given function $f(t)$, with some especially desirable features [81]. In particular, an affine plus radial function in the form

$$\mathbf{a} \cdot \mathbf{t} + b + \sum_{k=1}^{l} c_k \phi(\left|\mathbf{t} - \mathbf{t}_k\right|), \quad \mathbf{t} \in R^n \tag{9.36}$$

where $\mathbf{a}$, b, $\{c_k\}_{k=1}^{l}$ are constants, provides a good modeling framework for the canonical piecewise linear representation (Equation 9.12) of a nonlinear circuit [63].

9.3.2.3 Approximation by Means of Interpolation

Interpolation plays a central role in function approximation theory. The main theme of interpolation is this: suppose that an unknown function exists for which we are given some measurement data such as its function values, and perhaps some values of its derivatives, at some discrete points in the domain. How can we use this information to construct a new function that interpolates these values at the given points as an approximant of the unknown function, preferably in an optimal sense? Constructing such a function, called an interpolant, is usually not a difficult problem, but the technical issue that remains is what kind of functions should be used as the interpolant so that a certain meaningful and optimal objective is attained?

Algebraic polynomial interpolation is the simplest approach for the following Lagrange interpolation problem [22,36].

THEOREM 9.17

For arbitrarily given $n+1$ distinct points $0 \le t_0 < t_1 < \cdots < t_n \le 1$ and $n+1$ real values $v_0, v_1, \ldots, v_n$, there exists a unique polynomial $p_n(t)$ of degree n, which satisfies

$$p_n(t_k) = v_k, \quad k = 0, 1, \ldots, n$$

This polynomial is given by

$$p_n(t) = \sum_{k=0}^{n} v_k L_k(t)$$

with the Lagrange basis polynomials

$$L_k(t) := \frac{(t - t_0)\cdots(t - t_{k-1})(t - t_{k+1})(t - t_n)}{(t_k - t_0)\cdots(t_k - t_{k-1})(t_k - t_{k+1})(t_k - t_n)}, \quad k = 0, \ldots, n$$

Moreover, if $f(t)$ is l ($\leq n+1$) times continuously differentiable on $[a, b]$, then the interpolation error is bounded by

$$\left\|f - p_n\right\|_{L_\infty[0,1]} \leq \frac{1}{n!}\left\|f^{(l)}\right\|_{L_\infty[0,1]}\left\|h\right\|_{L_\infty[0,1]}$$

where $h(t) = \prod_{k=0}^{n-1}(t - t_k)$, and $||h||_{L_\infty[0,[1]]}$ attains its minimum at the Chebyshev points $t_k = \cos(2k+1)\pi/2(n+1))$, $k = 0, 1, \ldots, n$.

Note that the set $\{L_k(t)\}_{k=0}^n$ is a Chebyshev system on the interval $[t_0, t_n]$, which guarantees the existence and uniqueness of the solution. This set of basis functions can be replaced by any other Chebyshev system to obtain a unique interpolant.

If not only functional values, but also derivative values, are available and required to be interpolated by the polynomial,

$$p_n^{(i_k)}(t_k) = v_{k,i_k} \quad i_k = 0, \ldots, m_k, \quad k = 0, 1, \ldots, n$$

then we have a Hermite interpolation problem. An algebraic polynomial of degree $d = n + \sum_{k=0}^n m_k$ always exists as a Hermite interpolant. An explicit closed-form formula for the Hermite interpolant also can be constructed. For example, if only the functional values $\{v_k\}_{k=0}^n$ and the first derivative values $\{w_k\}_{k=0}^n$ are given and required to be interpolated, then the Hermite interpolant is given by

$$p_{2n}(t) = \sum_{k=0}^{n}\{v_k A_k(t) + w_k B_k(t)\}$$

where, with notation $L_k'(t_k) := (\mathrm{d}/\mathrm{d}t)L_k(t)|_{t=t_k}$,

$$A_k(t) = \left[1 - 2(t - t_k)L_k'(t_k)\right]L_k^2(t) \quad \text{and} \quad B_k(t) = (t - t_k)L_k^2(t)$$

in which $L_k(t)$ are Lagrange basis polynomials, $k = 0, 1, \ldots, n$.

However, if those derivative values are not consecutively given, we have a Hermite–Birkhoff interpolation problem, which is not always uniquely solvable [61].

The preceding discussions did not take into consideration any optimality. The unique algebraic polynomial interpolant obtained previously may not be a good result in many cases. A well-known example is provided by Runge, in interpolating the continuous and smooth function $f(t) = 1/(1 + 25t^2)$ at $n+1$ equally spaced points on the interval $-[1, 1]$. The polynomial interpolant $p_n(t)$ shows extremely high oscillations near the two end-points ($|t| > 0.726, \ldots$). Hence, it is important to impose an additional optimality requirement (e.g., a uniform approximation requirement) on the interpolant. In this concern, the following result is useful [36].

THEOREM 9.18

Given a continuous function $f \in C - [1, 1]$, let $\{t_k\}_{k=1}^n$ be the Chebyshev points on $-[1, 1]$; namely, $t_k = \cos((2k-1)\pi/(2n))$, $k = 1, \ldots, n$. Let also $P_{2n-1}(t)$ be the polynomial of degree $2n-1$ that satisfies the following special Hermite interpolation conditions: $P_{2n-1}(t_k) = f(t_k)$ and $P_{2n-1}(t_k) = f(t_k)$ and $P'_{2n-1}(t_k)$ $t(k) = 0$, $k = 1, \ldots, n$. Then, the interpolant $P_{2n}(t)$ has the uniform approximation property

$$\left\|f - P_{2n-1}\right\|_{L_\infty[-1,1]} \to 0 \quad \text{as} \quad n \to \infty$$

Because polynomial splines are piecewise algebraic polynomials, similar uniform approximation results for polynomial spline interpolants may be established [41,72,88].

Finally, a simultaneous interpolation and uniform approximation for a polynomial of a finite (and fixed) degree may be very desirable in engineering applications. The problem is that given and $f \in C[a, b]$ with $n+1$ points $a \leq t_0 < t_1 < \cdots < t_n \leq b$ and a given $\varepsilon > 0$, find a polynomial $p(t)$ of finite degree (usually, larger than n) that satisfies both

$$\|f - p\|_{L_\infty[a,b]} < \varepsilon \quad \text{and} \quad p(t_k) = f(t_k), \quad k = 0, 1, \ldots, n$$

The answer to this question is the Walsh theorem, which states that this is always possible, even for complex polynomials [36]. Note that natural splines can also solve this simultaneous interpolation and uniform-approximation problem.

9.3.3 Best Approximation of Linear Functionals

As already mentioned, a functional is a mapping that maps functions to values. Definite integrals, derivatives evaluated at some points, and interpolation formulas are good examples of linear functionals.

The best approximation problem for a given bounded linear functional, L, by a linear combination of n independent and bounded linear functionals $L_1, \ldots, L_n$, all defined on the same normed linear space X of functions, can be similarly stated as follows: determine n constant coefficients $\{a_k^\star\}_{k=1}^n$ such that

$$\left\| L - \sum_{k=1}^{n} a_k^\star L_k \right\|_{X^\star} = \min_{\{a_k\}} \left\| L - \sum_{k=1}^{n} a_k L_k \right\|_{X^\star} \tag{9.37}$$

where $X^\star$ is the dual space of X, which is also a normed linear space. A basic result is described by the following theorem [36].

THEOREM 9.19

If X is a Hilbert space, then the best approximation problem (Equation 9.37) *is uniquely solvable. Moreover, if r and $\{r_k\}_{k=1}^n$, are the functional representors of L and $\{L_k\}_{k=1}^n$, respectively, then*

$$\left\| r - \sum_{k=1}^{n} a_k r_k \right\|_{X^\star} = \min \Rightarrow \left\| L - \sum_{k=1}^{n} a_k L_k \right\|_{X^\star} = \min$$

It is important to note that for linear functionals, we have an interpolation problem: given bounded linear functionals L and $\{L_k\}_{k=1}^n$, all defined on a normed linear space X, where the last n functionals are linearly independent on X, and given also n points $x_k \in X$, $k = 1, \ldots, n$, determine n constant coefficients $\{a_k\}_{k=1}^n$, such that

$$\sum_{k=1}^{n} a_k L_k(x_i) = L(x_i), \quad i = 1, \ldots, n$$

Obviously, this problem is uniquely solvable. Depending on the specific formulation of the linear functionals, a bulk of the approximation formulas in the field of numerical analysis can be derived from this general interpolation formulation.

Finally, convergence problems also can be formulated and discussed for bounded linear functionals in a manner similar to interpolation and approximation of functions. The following result is significant [36].

THEOREM 9.20

Let L and $\{L_k\}_{k=1}^{\infty}$ be bounded linear functionals defined on a Banach space X. A necessary and sufficient condition for

$$\lim_{k\to\infty} \|L_k - L\|_{X^*} = 0$$

is that $\{L_k\}_{k=1}^{\infty}$ are uniformly bounded:

$$\|L_k\|_{X^*} \le M < \infty \quad \forall k = 1, 2, \ldots$$

and there is a convergent sequence $\{x_i\}_{i=1}^{\infty} \in X$, such that

$$\lim_{k\to\infty} L_k(x_i) = L(x_i) \quad \text{for each } i = 1, 2, \ldots$$

9.3.4 Artificial Neural Network for Approximation

Artificial neural networks offer a useful framework for signal and system approximations, including approximation of continuous and smooth functions of multivariables. Due to its usually mutilayered structure with many weights, an artificial neural network can be ''trained,'' and hence has a certain ''learning'' capability in data processing. For this reason, artificial neural networks can be very efficient in performing various approximations. The main concern with a large-scale artificial neural network is its demand on computational speed and computer memory.

Both parametrized and nonparametrized approaches to approximations use artificial neural networks. In the parametrized approach the activation function, basic function, and network topology are all predetermined; hence, the entire network structure is fixed, leaving only a set of parameters (weights) to be adjusted to best fit the available data. In this way, the network with optimal weights becomes a best approximant, usually in the least-squares sense, to a nonlinear system. Determining the weights from the data is called a training process. Back-propagation multilayered artificial neural networks are a typical example of the parametrized framework. The nonparametrized approach requires that the activation and/or basic functions also be determined, which turns out to be difficult in general.

To illustrate how an artificial neural network can be used as a system or signal approximant, we first describe the structure of a network. The term neuron used here refers to an operator or processing unit, which maps R''–R, with the mathematical expression

$$o_i = f_a(f_b(\mathbf{i}_i, \mathbf{w}_i)) \tag{9.38}$$

where $\mathrm{i}_i = [i_1, \ldots, i_n]^T$ is the input vector, $\mathbf{w}_i = [w_{i1}, \ldots, w_{in}]^T$ is the weight vector associated with the ith neuron, o_i the output of the ith neuron, f_a the activation function (usually sigmoidal or Gaussian), and

f_b the basic function (which can be linear, affine, or radial). For example, if an affine basic function is used, Equation 9.38 takes on the form

$$o_i = f_a(\mathbf{i}_i \cdot \mathbf{w}_i + b_i) \tag{9.39}$$

where b_i is a constant.

A fully connected feed-forward artificial neural network is generally a multi-input/multi-output network, where the output from each neuron of each layer is an input to each neuron of the next layer. Such a network, arranged in one input layer, multiple hidden layers, and one output layer, can be constructed as follows (see Figure 9.6). Suppose we have n-inputs, n_L-outputs and $L-1$ hidden layers, and a linear basic function is used with a sigmoidal activation function $f_a(t) = \sigma(t)$:

$$\sigma(t) \to \begin{cases} 1 & \text{as } t \to +\infty \\ 0 & \text{as } t \to -\infty \end{cases}$$

Also, let $o_{l,i}$ be the output of the ith neuron at the lth layer and $w_{l,i} = [w_{l,i,1} \cdots w_{l,i,s}]^T$ be the weight vector associated with the same neuron connected to the neurons at the $(l-1)$st layer. Then, we have

$$o_{l,i} = \sigma\left(\sum_{j=1}^{n_l} 0_{l-1,j} w_{l,i,j} + w_{l,i,0}\right) \tag{9.40}$$

Inductively, the output of the ith neuron in the last (the Lth) layer is given by

$$o_{L,i} = \sigma\left(\sum_{j=1}^{n_{L-1}} w_{L,i,j}\left(\cdots \sigma\left(\sum_{q=1}^{n_i} w_{1,p,q} i_{n_0} + w_{1,p,o}\right) + w_{2,p,0}\right) + \cdots + w_{L,i,o}\right) \tag{9.41}$$

where $i = 1, \ldots, n_L$.

The following best uniform approximation property of an artificial neural network is a fundamental result in neural-network approximation [35].

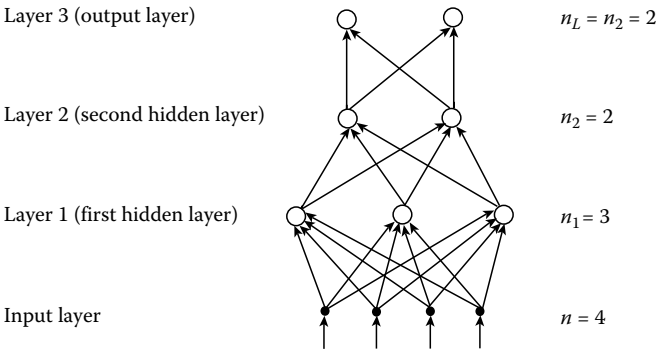


FIGURE 9.6 A two-hidden layer, feed-forward artificial neural network.

THEOREM 9.21

Let $f(t)$ be a continuous function defined on a compact subset $\Omega \subset R^n$. Then, for any $\in > 0$, there exists an integer $m \geq 1$ and real parameters $\{c_k, w_{ki}, b_k\}_{k=1}^{m}$ such that using any nonconstant, bounded, and monotonically increasing continuous function f_a as the activation function, the artificial neural network can uniformly approximate f on Ω, in the sense that

$$\|f - N\|_{L_\infty(\Omega)} < \varepsilon$$

where the network has the form

$$N(\mathbf{t}) = \sum_{k=1}^{m} c_k f_a\left(\sum_{i=1}^{n} w_{k,i} t_i + b_k\right), \quad \mathbf{t} = [t_1 \cdots t_n]^T \in \Omega$$

Neural networks can also provide approximation for a mapping together with its derivatives [52]. On the other hand, neural networks can provide localized approximation, which is advantageous in that if a certain portion of the data is perturbed, only a few weights in the network need to be retrained. It was demonstrated that a single hidden layered network cannot provide localized approximation of continuous functions on any compact set of a Euclidean space with dimension higher than one; however, two hidden layers are sufficient for the purpose [33].

As mentioned previously, the basic function f_b in a network need not be linear. An artificial neural network, using a radial function for f_b, can also give very good approximation results [76]. Also, as a system approximation framework, stability of a network is very important [68]. Finally, a major issue that must be addressed in designing a large-scale network is the computer memory, which requires some special realization techniques [67].

9.4 Identification

System identification is a problem of finding a good mathematical model, preferably optimal in some sense, for an unknown physical system, using some available measurement data. These data usually include system outputs and sometimes also inputs. Very often, the available data are discrete, but the system to be identified is continuous [97].

A general formulation of the system identification problem can be described as follows. Let S be the family of systems under consideration (linear or nonlinear, deterministic or stochastic, or even chaotic), with input u and output γ, and let $R_{(u,y)}$ be the set of I–O data. Define a mapping M: $S \rightarrow R_{(u,y)}$. Then, a system $\mathcal{F} \in S$ is said to be (exactly) identifiable if the mapping M is invertible, and the problem is to find the $\mathcal{F} = M^{-1}(\tilde{u},\tilde{y})$ using the available data $(\tilde{u}, \tilde{y}) \in R_{(u,y)}$. Here, how to define the mapping M, linear or not, is the key to the identification problem. Usually, we also want M^{-1} to be causal for the implementation purpose.

The first question about system identification is of course the identifiability [82]. Not all systems, not even linear deterministic systems, are exactly identifiable [21]. Because many physical systems are not exactly identifiable, system identification in a weaker sense is more realistic.

Suppose that some inputs and their corresponding outputs of an unknown system, $\mathcal{S}_1$, are given. We want to identify this unknown system by an approximate model, $\mathcal{S}_2$, using the available I–O data, such that the corresponding outputs produced by any input through $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, are "very close" under certain meaningful measure. If the structure of $\mathcal{S}_1$ (hence, $\mathcal{S}_2$) is known *a priori*, then what we need is to identify some system parameters. If the structure of $\mathcal{S}_1$ is not clear, the task becomes much more difficult because we must determine what kind of model to choose in approximating the unknown system [50]. This includes many crucial issues such as the linearity and dimension (or order) of the

model used. In particular, if the system is nonlinear and contains uncertainties, special techniques from set-valued mapping and differential inclusion theories may be needed [58].

Usually, the basic requirement is that $\mathcal{S}_2$ should be a best approximant of $\mathcal{S}_1$ from a desired class of simple and realizable models under a suitably chosen criterion. For example, the least-squares operator approximation discussed previously can be thought of as an identification scheme. For this reason, identification in the weak sense is traditionally considered to be one of the typical best approximation problems in mathematics. If a minimal, worst-case model-matching error bound is required, the approximation is known as the optimal recovery problem, for either functions or functionals [65,66], or for operators [15,45]. In system engineering it usually refers to system identification or reconstruction, with an emphasis on obtaining an identified model or a reconstruction scheme.

Generally speaking, system identification is a difficult problem, often leading to nonunique solutions when it is solvable. This is typically true for nonlinear circuits and systems. In systems and control engineering, an unknown system is identified by a desired model such that they can produce "close enough" outputs from the same input, measured by a norm in the signal space, such as L_p, l_p, or H_p $(1 \le p \le \infty)$. For dynamic systems, however, this norm-measure is generally not a good choice because one is concerned with nonlinear dynamics of the unknown system, such as limit cycles, attractors, bifurcations, and chaos. Hence, it is preferable to have an identified model that preserves the same dynamic behavior. This is a very challenging research topic; its fundamental theories and methodologies are still open for further exploration.

9.4.1 Linear Systems Identification

Compared to nonlinear systems, linear systems, either autoregressive with moving-average (ARMA) or state-space models, can be relatively easily identified, especially when the system dimension (order) is fixed. The mainstream theory of linear system identification has the following characteristics [37]:

1. The model class consists of linear, causal, stable, finite-dimensional systems with constant parameters.
2. Both system inputs and their corresponding outputs are available as discrete or continuous data.
3. Noise, if any, is stationary and ergodic (usually with rational spectral densities), white and uncorrelated with state vectors in the past.
4. Criteria for measuring the closeness in model-matching are of least-squares type (in the deterministic case) or of maximum likelihood type (in the stochastic case).
5. Large-scale linear systems are decomposed into lower-dimensional subsystems, and nonlinear systems are decomposed into linear and simple (e.g., memoryless) nonlinear subsystems.

Because for linear systems, ARMA models and state-space models are equivalent under a nonsingular linear transformation [17,32], we discuss only ARMA models here.

An (n, m, l)th-order ARMAX model (an ARMA model with exogenous noisy inputs) has the general form

$$a\left(z^{-1}\right)\mathbf{y}(t) = b\left(z^{-1}\right)\mathbf{u}(t)c\left(z^{-1}\right)\varepsilon(t), \quad t = \ldots, -1, 0, 1, \ldots \tag{9.42}$$

in which z^{-1} is the time-delay operator defined by $z^{-1}\mathbf{f}(t) = \mathbf{f}(t-l)$, and

$$a\left(z^{-1}\right) = \sum_{i=0}^{n} A_i z^{-i}, \quad b\left(z^{-1}\right) = \sum_{j=0}^{m} B_j z^{-j}, \quad c\left(z^{-1}\right) = \sum_{k=1}^{l} C_k z^{-k}$$

with constant coefficient matrices $\{A_i\}$, $\{B_j\}$, $\{C_k\}$ of appropriate dimensions, where $A_0 = I$ (or, is nonsingular). In the ARMAX model (Equation 9.42) $\mathbf{u}(t)$, $\mathbf{y}(t)$, and $\varepsilon(t)$ are considered to be system

input, output, and noise vectors, respectively, where the input can be either deterministic or random. In particular, if $l = 0$ and $n = 0$ (or $m = 0$), then Equation 9.42 reduces to a simple moving-average (MA) (or autoregressive, AR) model. Kolmogorov [56] proved that every linear system can be represented by an infinite-order AR model. It is also true that every nonlinear system with a Volterra series representation can be represented by a nonlinear AR model of infinite order [53].

The system identification problem for the ARMAX model (Equation 9.42) can now be described as follows. Given the system I–O data $(\mathbf{u}(t), \mathbf{y}(t))$ and the statistics of $\varepsilon(t)$, determine integers (n, m, l) (system-order determination) and constant coefficient matrices $\{A_i\}$, $\{B_j\}$, $\{C_k\}$ (system-parameter identification). While many successful methods exist for system parameter identification [3,23,49,60], system order determination is a difficult problem [47].

As already mentioned, the identifiability of an unknown ARMAX model using the given I–O data is a fundamental issue. We discuss the exact model identification problem here. The ARMAX model (Equation 9.42) is said to be exactly identifiable if $(\tilde{a}(z^{-1}), \tilde{b}(z^{-1}), \tilde{c}(z^{-1}))$ is an ARMAX model with $\tilde{n} \le n$, $\tilde{m} \le m$, and $\tilde{l} \le l$, such that

$$\begin{cases} [\tilde{a}(z^{-1})]^{-1}\tilde{b}(z^{-1}) = [a(z^{-1})]^{-1}b(z^{-1}) \\ [\tilde{a}(z^{-1})]^{-1}\tilde{c}(z^{-1}) = [a(z^{-1})]^{-1}c(z^{-1}) \end{cases}$$

Note that not all ARMAX models are exactly identifiable in this sense. A basic result about this identifiability is the following [21].

THEOREM 9.22

The ARMAX model (Equation 9.42) (*with* $t \ge 0$) *is exactly identifiable iff* $a(z^{-1})$, $b(z^{-1})$, *and* $c(z^{-1})$ *have no common left factor and the rank of the constant matrix* $[A_n, B_m, C_l]$, *consisting of the highest-order coefficient terms in* $a(z^{-1})$, $b(z^{-1})$, $c(z^{-1})$, *respectively, is equal to the dimension of the system output* $\mathbf{y}$.

Even if an unknown system is exactly identifiable and its identification is unique, how to find the identified system is still a very technical issue. For simple AR models, the well-known Levinson–Durbin algorithm is a good scheme for constructing the identified model; for MA models, one can use Trench–Zohar and Berlekamp–Massey algorithms. There exist some generalizations of these algorithms in the literature [23]. For stochastic models with significant exogenous noise inputs, various statistical criteria and estimation techniques, under different conditions, are available [82]. Various recursive least-squares schemes, such as the least-mean-square (LMS) algorithm [96], and various stochastic searching methods, such as the stochastic gradient algorithm [49], are popular. Because of their simplicity and efficiency, the successful (standard and extended) Kalman filtering algorithms [16,30] have also been widely applied in parameters identification for stochastic systems [13,17,62], with many real-world applications [92].

Finally, for linear systems, a new framework, called the behavioral approach, is proposed for mathematical system modeling and some other related topics [2,98].

9.4.2 Nonlinear Systems Identification

Identifying a nonlinear system is much more difficult than identifying a linear system in general, whether it is in the exact or in the weak sense, as is commonly known and can be seen from its information-based complexity analysis [15,45].

For some nonlinear systems with simple Volterra series representations, the least-squares approximation technique can be employed for the purpose of identification in the weak sense [8]. As a simple

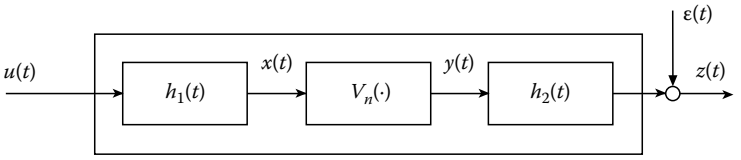


FIGURE 9.7 A cascaded linear–nonlinear system with noise input.

illustrative example, consider the cascaded nonlinear system with noise input shown in Figure 9.7. In this figure $h_1(t)$ and $h_2(t)$ are unit impulse responses of two linear subsystems, respectively, and $V_n(\cdot)$ is a memoryless nonlinear subsystem which is assumed to have an nth-order Volterra polynomial in the special form

$$y(t) = \sum_{k=1}^{n} c_k \int_0^t \cdots \int_0^t \phi_k(\tau_1, \ldots, \tau_k) x(\tau_1) \cdots x(\tau_k) \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_k \tag{9.43}$$

where all the Volterra kernels $\{\phi_k\}_{k=0}^n$ are assumed to be known, but the constant coefficients $\{c_k\}_{k=0}^n$ must be identified.

It is clear from Figure 9.7 that the output of the cascaded system can be expressed via convolution-type integrals as

$$z(t) = c_1 \left(\int h_2 \phi_1 h_1 u \right)(t) + \cdots + c_n \left(\int \cdots \int h_2 \phi_1 h_1 \cdots h_2 u \cdots u \right)(t) + \varepsilon(t) \tag{9.44}$$

Now, because all the integrals can be computed if the input function $u(t)$ is given, the standard least-squares technique can be used to determine the unknown constant coefficients $\{c_k\}_{k=0}^n$, using the measured system output $z(t)$.

A neural network implementation of Volterra series model identification is described in Ref. [1]. Neural network for system identification has been used in many different cases, as can also be seen from Ref. [70].

Finally, we consider one approach to nonlinear systems identification which combines the special structure of the generalized Fock space of Volterra series (Theorem 9.3) and the ''training'' idea from neural networks discussed previously (Theorem 9.21). For simplicity, consider the scalar nonlinear system

$$y(t) + f(y(t-1), y(t-2), \ldots, y(t-n)) = u(t), \quad t = 0, 1, \ldots \tag{9.45}$$

where n is a fixed integer, with the given initial conditions $y(-1) = y_1, \ldots, y(-n) = y_n$. Introducing a simple notation

$$y^{t-1} = (y(t-1), \ldots, y(t-n)) \tag{9.46}$$

we first rewrite this system as

$$\gamma(t) + f\left(\gamma^{t-1}\right) = u(t) \tag{9.47}$$

Then, we denote by E^n as the n-dimensional Euclidean space of continuous functions and let $u^1, \ldots, u^m \in E^n$, called the domain training samples, be given data vectors that are componentwise nonzero and

distinct, namely, $u_i^k \neq u_j^k$ if $i \neq j$ for all $k = 1, \ldots, m$, $1 \leq i, j \leq n$. Here, Equation 9.46 also has been used for these domain training samples. Also, let $r_1, \ldots, r_m$, be given real numbers, called the corresponding range training samples. The identification problem is to find an approximate system, $f^\star(\cdot)$, among all Volterra series representations from the generalized Fock space formulated in Theorem 9.3, such that $f^\star$ maps all the domain training samples to their corresponding range training samples:

$$f^\star(u^k) = r_k, \quad k = 1, \ldots, m \tag{9.48}$$

and $f^\star$ has the minimum operator-norm among all such candidates. The following theorem provides an answer to this problem [45].

THEOREM 9.23

There is a unique element $f^\star$ of minimum norm in the generalized Fock space defined in Theorem 9.3, *with the domain $\Omega = E^n$ therein, that satisfies the constraint* (Equation 9.48). *Moreover, $f^\star$ has the following expression*:

$$f^\star(v) = \sum_{k=1}^{m} a_k K\left(u^k, v\right) \quad \forall v \in E^n$$

where $K(\cdot, \cdot)$ is the reproducing kernel defined in Theorem 9.3, and the system parameters are determined by

$$\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} K(u^1, u^1) & \cdots & K(u^1, u^m) \\ \vdots & & \vdots \\ K(u^m, u^1) & \cdots & K(u^m, u^m) \end{bmatrix}^{-1} \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix}$$

Here, it should be noted that because K is a reproducing kernel, the set of functions $\{K(u^k, \cdot)\}_{k=1}^m$ are linearly independent, so that the above inverse matrix exists.

Also, note that this system identification method can be applied to higher-dimensional systems and the continuous-time setting [45].

9.4.3 Nonlinear Dynamic Systems Identification from Time Series

Measurement (observation) data obtained from an unknown system are often available in the form of time series. There are some successful techniques for identification of linear and nonlinear systems from time series if the time series is generated from Gaussian white noise. For example, for linear systems we have the Box–Jenkins scheme and for nonlinear systems, a statistical method using a nonlinear filter [75].

Concerned with nonlinear dynamic systems, however, statistical methods and the commonly used norm-measure criterion may not be capable of identifying the system dynamics in general. This is because the main issue in this concern is the nonlinear dynamic behavior of the unknown system, such as limit cycles, attractors, bifurcations, and chaos. Hence, it is preferable that the identified model can preserve the nonlinear dynamics of the unknown system. This turns out to be a very challenging task; many fundamental theories and methodologies for this task remain to be developed.

When an unknown nonlinear dynamic system is measured to produce a set of continuous or discrete data (a time series), a natural approach for studying its dynamics from the available time series is to take an integral transform of the series, so as to convert the problem from the time domain to the frequency domain. Then, some well-developed engineering frequency domain methods can be applied to perform analysis and computation of the nonlinear dynamics [69].

One common approach formulated in the time domain is the (delay-coordinate) embedding method that can be applied to reconstruct (identify) an unknown nonlinear dynamic model from which only a set of discrete measurement data (a time series) is available.

Let us consider the problem of identifying a periodic trajectory of an unknown, nonlinear dynamic system using only an experimental time series measured from the system. Let $\{r_k\}$ be the available data. The embedding theory guarantees this can be done in the space R^m with the embedding dimension $m \geq 2n+1$, where n is the dimension of the dynamic system 93], or $m \geq 2d_A$, where d_A is the dimension of the attractor [87]. A way to achieve this is to use the delay-coordinate technique, which approximates the unknown, nonlinear dynamics in R^m by introducing the embedding vector

$$\mathbf{r}_k = \left[r_k r_{k-\mu} \cdots r_{k-(m-1)\mu}\right]^{\mathrm{T}} \tag{9.49}$$

where μ is the time-delay step. This embedding vector provides enough information to characterize the essence of the system dynamics and can be used to obtain an experimental Poincaré map, which helps in understanding the dynamics. For example, one may let the map be the equation of the first component of the vector being equal to a constant: $r_{ki} = \text{constant}$. This procedure yields the successive points

$$\xi_i := \left[r_{k_i-\mu} \cdots r_{k_i-(m-1)\mu}\right]^{\mathrm{T}} \tag{9.50}$$

at the ith piercing of the map by the trajectory (or the vector r_k), where k_i is the time index at the ith piercing. Then, one can locate the periodic trajectories of the unknown system using the experimental data [4,14]. In this approach, however, determining a reasonably good time-delay step size, i.e., the real number μ in Equation 9.49, remains an open technical problem.

Finally, we note that the embedding method discussed previously has been applied to the control of chaotic circuits and systems [19,20,74].

References

1. A. Abdulaziz and M. Farsi, Non-linear system identification and control based on neural and self-tuning control, *Int. J. Adapt. Control Signal Process.*, 7(4), 297–307, 1993.
2. A. C. Antoulas and J. C. Willems, A behavioral approach to linear exact modeling, *IEEE Trans. Autom. Control*, 38(12), 1776–1802, 1993.
3. K. J. Aström and P. Eykhoff, System identification: A survey, *Automatica*, 7, 123–162, 1971.
4. H. Atmanspacher, H. Scheingraber, and W. Voges, Attractor reconstruction and dimensional analysis of chaotic signals, in *Data Analysis in Astronomy III*, V. Di Gesú et al., Eds., New York: Plenum Press, 1988, pp. 3–19.
5. I. Baesler and I. K. Daugavet, Approximation of nonlinear operators by Volterra polynomials, *Am. Math. Soc. Transl.*, 155(2), 47–57, 1993.
6. S. P. Banks, *Mathematical Theories of Nonlinear Systems*, New York: Prentice-Hall, 1988.
7. A. Bensoussan, G. de Prado, M. C. Felfour, and S. K. Mitter, *Representation and Control of Infinite Dimensional Systems*, Boston: Birkhäuser, 1992.
8. S. A. Billings, Identification of nonlinear systems—a survey, *IEEE Proc., Pt. D*, 127, 272–285, 1980.

9. S. Boyd and L. O. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series, *IEEE Trans. Circuits Syst*., 32(11), 1150–1161, 1985.
10. S. Boyd, L. O. Chua, and C. A. Desoer, Analytic foundations of Volterra series, *IMA J. Info. Control*, 1, 243–282, 1984.
11. D. Braess, *Nonlinear Approximation Theory*, New York: Springer-Verlag, 1986.
12. R. W. Brockett, Volterra series and geometric control theory, *Automatica*, 12, 167–176, 1976.
13. P. E. Caines, *Linear Stochastic Systems*, New York: John Wiley & Sons, 1988.
14. M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson, State space reconstruction in the presence of noise, *Physica D*, 51, 52–98, 1991.
15. G. Chen, Optimal recovery of certain nonlinear analytic mappings, *Optimal Recovery*, B. Bojanov and H. Wozniakowski, Eds., New York: Nova Science, 1992, pp. 141–144.
16. G. Chen, Ed., *Approximate Kalman Filtering*, New York: World Scientific, 1993.
17. G. Chen, G. Chen, and H. S. Hsu, *Linear Stochastic Control Systems*, Boca Raton, FL: CRC Press, 1995.
18. G. Chen, C. K. Chui, and M. J. Lai, Construction of real-time spline quasi-interpolation schemes, *Approx. Theory Appl*., 4(4), 61–75, 1988.
19. G. Chen and X. Dong, *From Chaos to Order: Methodologies*, *Perspectives, and Applications*, Singapore: World Scientific, 1998.
20. G. Chen and J. Moiola, An overview of bifurcation, chaos and nonlinear dynamics in control systems, *The Franklin Institute Journal*, 331B, 819–858, 1994.
21. H. F. Chen and J. F. Zhang, On identifiability for multidimensional ARMAX model, *Acta Math. Appl. Sin*., 9(1), 1–8, 1993.
22. E. W. Cheney, *Introduction to Approximation Theory*, New York: McGraw-Hill, 1966.
23. B. Choi, *ARMA Model Identification*, New York: Springer-Verlag, 1992.
24. L. O. Chua, The genesis of Chua's circuit, *AEU Int. J. Electr. Commun*., 46, 187–257, 1992.
25. L. O. Chua and A. C. Deng, Canonical piecewise-linear modeling, *IEEE Trans. Circuits Syst*., 33(5), 511–525, 1986.
26. L. O. Chua and A. C. Deng, Canonical piecewise-linear representation, *IEEE Trans. Circuits Syst*., 35(1), 101–111, 1988.
27. C. K. Chui, *Multi-Variate Splines*, Philadelphia, PA: SIAM, 1988.
28. C. K. Chui, Approximations and expansions, in *Encyclopedia of Physical Science and Technology*, vol. 2, New York: Academic Press, 1992, pp. 1–29.
29. C. K. Chui, *An Introduction to Wavelets*, New York: Academic Press, 1992.
30. C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*, New York: Springer-Verlag, 1987.
31. C. K. Chui and G. Chen, *Linear Systems and Optimal Control*, New York: Springer-Verlag, 1989.
32. C. K. Chui and G. Chen, *Signal Processing and Systems Theory: Selected Topics*, New York: Springer-Verlag, 1992.
33. C. K. Chui, X. Li, and H. N. Mhaskar, Neural networks for localized approximation, *Math. Comput*., 63, 607–623, 1994.
34. J. B. Conway, *A Course in Functional Analysis*, New York: Springer-Verlag, 1985.
35. G. Cybenko, Approximation by superposition of sigmoidal functions, *Math. Control, Signals Syst*., 2, 303–314, 1989.
36. P. J. Davis, *Interpolation and Approximation*, New York: Dover, 1975.
37. M. Deistler, Identification of linear systems (a survey), in *Stochastic Theory and Adaptive Control*, T. E. Duncan and B. P. Duncan, Eds., New York: Springer-Verlag, 1991, pp. 127–141.
38. C. A. Desoer and M. G. Kabuli, Right factorizations of a class of time-varying nonlinear systems, *IEEE Trans. Autom. Control*, 33, 755–757, 1988.
39. C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input–Output Properties*, New York: Academic Press, 1975.
40. R. A. DeVore and G. G. Lorentz, *Constructive Approximation*, New York: Springer-Verlag, 1993.

41. C. de Boor, *A Practical Guide to Splines*, New York: Springer-Verlag, 1978.
42. C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines*, New York: Springer-Verlag, 1993.
43. R. J. P. de Figueiredo and G. Chen, Optimal interpolation on a generalized Fock space of analytic functions, in *Approximation Theory VI*, C. K. Chui, L. L. Schumaker, and J. D. Ward, Eds., New York: Academic Press, 1989, pp. 247–250.
44. R. J. P. de Figueiredo and G. Chen, PDLG splines defined by partial differential operators with initial and boundary value conditions, *SIAM J. Numerical Anal.*, 27(2), 519–528, 1990.
45. R. J. P. de Figueiredo and G. Chen, *Nonlinear Feedback Control Systems: An Operator Theory Approach*, New York: Academic Press, 1993.
46. R. J. P. de Figueiredo and T. A. Dwyer, A best approximation framework and implementation for simulation of large scale nonlinear systems, *IEEE Trans. Circuits Syst.*, 27(11), 1005–1014, 1980.
47. J. G. de Gooijer, B. Abraham, A. Gould, and L. Robinson, Methods for determining the order of an autoregressive-moving average process: A survey, *Int. Statist. Rev.*, 53, 301–329, 1985.
48. M. Fliess, M. Lamnabhi, and F. L. Lagarrigue, An algebraic approach to nonlinear functional expansions, *IEEE Trans. Circuits Syst.*, 39(8), 554–570, 1983.
49. G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
50. R. Haber and H. Unbehauen, Structure identification of nonlinear dynamic systems—a survey on input/output approaches, *Automatica*, 26(4), 651–677, 1990.
51. J. Hammer, Fraction representations of nonlinear systems: A simplifed approach, *Int. J. Control*, 46, 455–472, 1987.
52. K. Hornik, M. Stinchcombe, and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feed-forward networks, *Neural Networks*, 3, 551–560, 1990.
53. L. R. Hunt, R. D. DeGroat, and D. A. Linebarger, Nonlinear AR modeling, *Circuits Syst. Signal Process.*, 14(5), 689–705, 1995.
54. A. Isidori, *Nonlinear Control Systems*, New York: Springer-Verlag, 1985.
55. C. Kahlert and L. O. Chua, A generalized canonical piecewise-linear representation, *IEEE Trans. Circuits Syst.*, 37(3), 373–383, 1990.
56. A. N. Kolmogorov, Interpolation and extrapolation von stationaren zufallingen folgen, *Bull. Acad. Sci. USSR Ser. Math.*, 5, 3–14, 1941.
57. S. Y. Kung, *Digital Neural Networks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
58. A. B. Kurzhanski, Identification—a theory of guaranteed estimates, in *From Data to Model*, J. C. Willems, Ed., New York: Springer-Verlag, 1989, pp. 135–214.
59. C. Lesiak and A. J. Krener, The existence and uniqueness of Volterra series for nonlinear systems, *IEEE Trans. Autom. Control*, 23(6), 1090–1095, 1978.
60. L. Ljung, *System Identification: Theory for the User*, Englewood Cliffs, NJ: Prentice-Hall, 1987.
61. G. G. Lorentz, K. Jetter, and S. D. Riemenschneider, *Birkhoff Interpolation*, Reading, MA: Addison-Wesley, 1983.
62. H. Lütkepohl, *Introduction to Multiple Time Series Analysis*, New York: Springer-Verlag, 1991.
63. A. I. Mees, M. F. Jackson, and L. O. Chua, Device modeling by radial basis functions, *IEEE Trans. Circuits Syst.*, 39(11), 19–27, 1992.
64. A. I. Mees, Parsimonious dynamical reconstruction, *Int. J. Bifurcation Chaos*, 3(3), 669–675, 1993.
65. C. A. Micchelli and T. J. Rivlin, A survey of optimal recovery, in *Optimal Estimation in Approximation Theory*, C. A. Micchelli and T. J. Rivlin, Eds., New York: Plenum Press, 1977, pp. 1–54.
66. C. A. Micchelli and T. J. Rivlin, Lectures on optimal recovery, in *Numerical Analysis*, P. R. Turner, Ed., New York: Springer-Verlag, 1984, pp. 21–93.
67. A. N. Michel, J. A. Farrell, and W. Porod, Associative memories via artificial neural networks, *IEEE Control Syst. Mag.*, pp. 6–17, Apr. 1990.
68. A. N. Michel, J. A. Farrell, and W. Porod, Qualitative analysis of neural networks, *IEEE Trans. Circuits Syst.*, 36(2), 229–243, 1989.

69. J. L. Moiola and G. Chen, *Hopf Bifurcation Analysis: A Frequency Domain Approach*, Singapore: World Scientific, 1996.
70. K. S. Narendra and K. Parthasarathy, Identification and control of dynamic systems using neural networks, *IEEE Trans. Neural Networks*, 1(1), 4–27, 1990.
71. H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems*, New York: Springer-Verlag, 1990.
72. G. Nürnberger, *Approximation by Spline Functions*, New York: Springer-Verlag, 1989.
73. W. Odyniec and G. Lewicki, *Minimal Projections in Banach Spaces*, New York: Springer-Verlag, 1990.
74. E. Ott, C. Grebogi, and J. A. Yorke, Controlling chaos, *Phys. Rev. Lett*., 64, 1196–1199, 1990.
75. T. Ozaki, Identification of nonlinearities and non-Gaussianities in time series, in *New Directions in Time Series Analysis, Part I*, D. Brillinger et al., Eds., New York: Springer-Verlag, 1993, pp. 227–264.
76. J. Park and I. W. Sandberg, Approximation and radial-basic-function networks, *Neural Comput*., 5, 305–316, 1993.
77. J. R. Partington, *An Introduction to Hankel Operators*, London: Cambridge University Press, 1988.
78. P. P. Petrushev and V. A. Popov, *Rational Approximation of Real Functions*, London: Cambridge University Press, 1987.
79. M. J. D. Powell, *Approximation Theory and Methods*, London: Cambridge University Press, 1981.
80. M. J. D. Powell, Radial basis functions for multi-variable interpolation: A review, *IMA Conference on Algorithms for Approximation of Functions and Data*, RMCS, Shrivenham, UK, 1985.
81. E. Quak, N. Sivakumar, and J. D. Ward, Least-squares approximation by radial functions, *SIAM J. Math. Anal*., 24(4), 1043–1066, 1993.
82. B. L. S. P. Rao, *Identifiability in Stochastic Models*, New York: Academic Press, 1992.
83. W. J. Rugh, *Nonlinear Systems Theory: The Volterra/Wiener Approach*, Baltimore: Johns Hopkins University Press, 1981.
84. I. W. Sandberg, A perspective on system theory, *IEEE Trans. Circuits Syst*., 31(1), 88–103, 1984.
85. I. W. Sandberg, Criteria for the global existence of functional expansions for input/output maps, *AT&T Tech. J*., 64(7), 1639–1658, 1985.
86. I. W. Sandberg, Approximately finite memory and input–output maps, *IEEE Trans. Circuits Syst*., 39(7), 549–556, 1992.
87. T. Sauer, J. A. Yorke, and M. Casdagli, Embedology, *J. Stat. Phys*., 63(3–4), 579–616, 1991.
88. L. L. Schumaker, *Spline Functions: Basic Theory*, New York: John Wiley & Sons, 1981.
89. A. N. Sharkovsky, Chaos from a time-delayed Chua's circuit, *IEEE Trans. Circuits Syst*., 40(10), 781–783, 1993.
90. E. D. Sontag, *Polynomial Response Maps*, New York: Springer-Verlag, 1979.
91. E. D. Sontag, *Mathematical Control Theory*, New York: Springer-Verlag, 1990.
92. H. W. Sorenson, Ed., *Kalman Filtering: Theory and Applications*, New York: IEEE Press, 1985.
93. F. Takens, Detecting strange attractors in turbulence, in *Lecture Notes in Mathematics*, vol. 898, D. A. Rand and L. S. Yong, Eds., New York: Springer-Verlag, 1981, pp. 366–381.
94. F. I. Utreras, Positive thin plate splines, *J. Approx. Theory Appl*., 1, 77–108, 1985.
95. M. S. Verma and L. R. Hunt, Right coprime factorizations and stabilization for nonlinear systems, *IEEE Trans. Autom. Control*, 38, 222–231, 1993.
96. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1985.
97. J. C. Willems, Ed., *From Data to Model*, New York: Springer-Verlag, 1989.
98. J. C. Willems, Paradigms and puzzles in the theory of dynamical systems, *IEEE Trans. Autom. Control*, 36, 259–294, 1991.

10

Transformation and Equivalence

10.1 Transformation of Nonlinear Dynamical Circuit Equations **10**-1
10.2 Normal Forms of Nonlinear Dynamical Circuit Equations **10**-5
10.3 Dimensionless Forms of Circuit Equations **10**-9
10.4 Equivalence of Nonlinear Resistive *n*-Ports **10**-16
10.5 Equivalence of Nonlinear Dynamical *n*-Ports **10**-17
10.6 Equivalence of Reciprocal Nonlinear Dynamic Circuits **10**-18
References **10**-23

Wolfgang Mathis
University of Hanover

10.1 Transformation of Nonlinear Dynamical Circuit Equations

One of the basic problems in studying linear and nonlinear dynamical electrical circuits is the analysis of the underlying descriptive equations and their solution manifold. In the case of linear or affine circuits, the constitutive relations of circuit elements are restricted to classes of linear or affine functions and, therefore, possess rather restricted types of solutions. In contrast, the solution manifold of nonlinear networks may consist of many different types. Naturally, it is useful to decompose nonlinear networks into classes that possess certain similarities. One approach, for example, is to consider the solution manifold and to decompose solutions into similar classes. Furthermore, if the descriptive differential equations of dynamic networks are considered to be mathematical sets, their decompositions will be of interest.

The technique of equivalence relations is the preferred method used to decompose a set of mathematical objects into certain classes. A well-known approach to define equivalence relations uses transformation groups. For example, real symmetric $n \times n$ matrices $\mathbb{R}_s^{n\times n}$ can be decomposed into equivalence classes by using the general linear transformation group $GL(n; \mathbb{R})$, and by applying the following similarity transformation:

$$\mathbf{M} \rightarrow \mathbf{U}^{-1}\mathbf{M}\mathbf{U}, \tag{10.1}$$

where $U \in GL(n; \mathbb{R})$. By applying $GL(n; \mathbb{R})$, the set $\mathbb{R}_s^{n\times n}$ is decomposed into similarity classes that are characterized by their eigenvalues. Furthermore, each class of $\mathbb{R}_s^{n\times n}$ contains a diagonal matrix $\mathbf{D}$ with these eigenvalues on the main diagonal [35]. These eigenvalues are invariants of the group and characterize different classes. These and other related results can be applied to classify linear and affine dynamical networks [20]. Thus, properties of the A-matrix of the state-space equations are used for the classification. Note that each linear and affine dynamical network can be described in state-space form.

We discuss the theory of equivalence of linear and affine dynamical networks only as special cases of nonlinear networks. An interesting reformulation of the classical material of the decomposition of real matrices by using similarity transformations in the framework of one-parameter groups in $GL(n; \mathbb{R})$ is given by Ref. [21].

A classification of the vector fields is needed in order to classify differential equations of the type $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^n$ is a vector field on $\mathbb{R}^n$. A first concept is established by a k-times differentiable change of coordinates that transforms a differentiable equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ into $\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y})$ by a function $\mathbf{h} \in C^k$. C^k is the set of k-times continuously differentiable functions $\mathbf{h}$: $\mathbb{R}^n \to \mathbb{R}^n$. In other words, two vector fields are called *C^k-conjugate* if there exists a C^k-diffeomorphism $\mathbf{h}$ ($k \geq 1$) such that $\mathbf{h} \circ \mathbf{f} = \mathbf{g} \circ \mathbf{h}$. An equivalent formulation uses the concept of flows associated with differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. A flow is a continuously differentiable function φ: $\mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ such that, for each $t \in \mathbb{R}$, the restriction $\varphi(t, \cdot) =: \varphi_t(\cdot)$ satisfies $\varphi_0 = \mathrm{id}_{\mathbb{R}^n}$ and $\varphi_t \circ \varphi_s = \varphi_{t+s}$ for all $t, s \in \mathbb{R}^n$. The relationship to a associated differential equation is given by

$$\mathbf{f}(\mathbf{x}) := \frac{\mathrm{d}\varphi_t}{\mathrm{d}t}(\mathbf{x})\bigg|_{t=0} = \lim_{\varepsilon \to 0} \left\{ \frac{\varphi(\varepsilon, \mathbf{x}) - \varphi(0, \mathbf{x})}{\varepsilon} \right\}. \tag{10.2}$$

For more details see, for example, Ref. [13]. Two flows φ_t and Ψ_t (associated with $\mathbf{f}$ and $\mathbf{g}$, respectively) are called C^k-conjugate, if there exists a C^k-diffeomorphism $\mathbf{h}$ ($k \geq 1$) such that $\mathbf{h} \circ \varphi_t = \Psi_t \circ \mathbf{h}$. In the case that $k = 0$, the term C^k-conjugate needs to be replaced by C^0 or topological conjugate and $\mathbf{h}$ is a homeomorphism. Clearly, differential equations, vector fields, and flows are only alternative ways of presenting the same dynamics.

By the previous definitions, equivalence relations can be generated and the set of differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ (as well as vector fields and flows) can be decomposed in certain classes of inequivalent differential equations and so on with different behavior (with respect to the equivalence relation). Although C^k-conjugate seems to be a natural concept for classifying differential equations, vector fields, and flows, this approach leads to a very refined classification (up to a diffeomorphism). In other words, too many systems become inequivalent. We consider two examples for illustrating this statement. Consider the nonlinear dynamical circuit (see Figure 10.1) with the descriptive equations (in a dimensionless form):

$$C\frac{\mathrm{d}v_C}{\mathrm{d}t} = i(v_C, i_L) - i_L, \tag{10.3}$$

$$L\frac{\mathrm{d}i_L}{\mathrm{d}t} = v_C + v(v_C, i_L), \tag{10.4}$$

where the nonlinear functions i: $\mathbb{R}^2 \to \mathbb{R}$ and v: $\mathbb{R}^2 \to \mathbb{R}$ characterize the nonlinear controlled sources. If these controlled sources are given in the following form:

$$i = i(v_C, i_L) := -\frac{1}{2}\left(\sqrt{v_C^2 + i_L^2 - 1}\right) v_C, \tag{10.5}$$

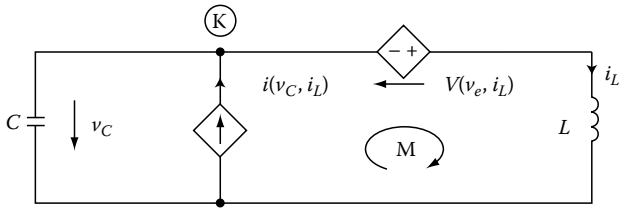


FIGURE 10.1 Circuit with nonlinear controlled sources.

$$v = v(v_C, i_L) = -\frac{1}{2}\left(\sqrt{v_C^2 + i_L^2 - 1}\right) i_L, \tag{10.6}$$

we obtain the following concrete descriptive equations for the circuits

$$C\frac{\mathrm{d}v_C}{\mathrm{d}t} = -\frac{1}{2}\left(\sqrt{v_C^2 + i_L^2 - 1}\right) v_C - i_L, \tag{10.7}$$

$$L\frac{\mathrm{d}i_L}{\mathrm{d}t} = v_C - \frac{1}{2}\left(\sqrt{v_C^2 + i_L^2 - 1}\right) i_L. \tag{10.8}$$

These equations can be transformed into the following form if we use polar coordinates (r, ϕ) by $v_C := r\cos(\phi t)$ and $i_L := r\sin(\phi t)$:

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{1}{2}(1 - r)r, \quad \dot{\phi} = 1, \tag{10.9}$$

where $C = 1$ and $L = 1$. If we consider the same circuit with the parameters $C = 1/2$ and $L = 1/2$ we obtain the slightly different descriptive equations:

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{1}{2}(1 - r)r, \quad \dot{\phi} = 2. \tag{10.10}$$

Note that both differential equations differ only by a time rescaling $t \to 2t$. The question arises whether these two sets of equations are C^1 conjugate.

It can be demonstrated (see Refs. [2,29]) that if a diffeomorphism converts a singular point of a vector field into a singular point of another vector field, then the derivative of the diffeomorphism converts the Jacobian matrix of the first vector field at its singular point into the Jacobian matrix of the second field at its singular point. Consequently, these two Jacobian matrices are in the same similarity class and therefore, have the same eigenvalues. In other words, the eigenvalues of the Jacobian matrices are invariants with respect to a diffeomorphism, and the corresponding decomposition of the set of vector fields (differential equations and flows) is continuous rather than discrete. Obviously, the eigenvalues of Equations 10.9 and 10.10 ($\lambda_1 = 1/2$, $\lambda_2 = 1$ and $\tilde{\lambda}_1 = 1$, $\tilde{\lambda}_2 = 2$, respectively) are different and, in conclusion, the two vector fields are not C^1 conjugate. Moreover, these two vector fields are not topologically or C^0 conjugate. A more "coarse" equivalence relation is needed in order to classify these vector fields, differential equations, and flows. As mentioned above, a time rescaling transforms the differential equations (Equations 10.9 and 10.10) into one another and therefore it should be no essential difference between these two differential equations. This motivates the following definition.

Definition 10.1: *Two flows φ_t and ψ_t are called C^k equivalent ($k \geq 1$) if there exists a C^k diffeomorphism* $\mathbf{h}$ *that takes each orbit of ψ_t into an orbit of ψ_t, preserving their orientation. In the case of $k = 0$, the flows are called C^0 or topologically equivalent.*

Because C^k equivalence preserves the orientation of orbits, the relation $\mathbf{h}(\varphi_t(\mathbf{x})) = \varphi_{\tau_{y(y)}}$ with $\mathbf{y} = \mathbf{h}(\mathbf{x})$ between φ_t and ψ_t is allowed, where τ_y is an increasing function of t for every $\mathbf{y}$.

It can be demonstrated (see Ref. [29]) that the eigenvalues of the Jacobian matrices of the two vector fields must be in the same ratio if a monotonic time rescaling is allowed. Therefore, the two vector fields (Equations 10.9 and 10.10) are C^1 equivalent. However, the two linear vector fields of the equations

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1+\varepsilon \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{10.11}$$

are not C^1 equivalent for any $\varepsilon \neq 0$, although the solutions of the differential equations are very close for small ε in a finite time interval. In conclusion, topological equivalence is the appropriate setting for classifying differential equations, vector fields, and flows. Note, that the decomposition of the set of linear vector fields into equivalence classes using the topological equivalence does not distinguish between nodes, improper nodes, and foci, but does distinguish between sinks, saddles, and sources. This suggests the following theorem [3]:

THEOREM 10.1

Let $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ ($\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{A} \in \mathbb{R}^{n \times n}$) *define a hyperbolic flow on* $\mathbb{R}^n$, *e.g., the eigenvalues of* $\mathbf{A}$ *have only nonzero parts, with* n_s *eigenvalues with a negative real part. Then,* $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ *is topological equivalent to the system* ($n_u := n - n_s$):

$$\dot{\mathbf{x}}_s = -x_s, \quad \mathbf{x}_s \in \mathbb{R}^{n_s}, \tag{10.12}$$

$$\dot{\mathbf{x}}_u = +x_s, \quad \mathbf{x}_u \in \mathbb{R}^{n_u}. \tag{10.13}$$

Therefore, it follows that hyperbolic linear flows can be classified in a finite number of types using topological equivalence.

A local generalization of this theorem to nonlinear differential equations is known as the theorem of Hartmann and Grobman; see, e.g., Ref. [3].

THEOREM 10.2

Let $\mathbf{x}^\star$ *be a hyperbolic fixed point of* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *with the flow* φ_t: $U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, *i.e., the eigenvalues of the Jacobian matrix* $\mathbf{J}_f(\mathbf{x}^\star)$ *have only nonzero real parts. Then, there is a neighborhood N of* $\mathbf{x}^\star$ *on which* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *is topologically equivalent to* $\dot{\mathbf{x}} = \mathbf{J}_f(\mathbf{x}^\star)\,\mathbf{x}$.

The combination of the two theorems implies that a very large set of differential equations can be classified in an isolated hyperbolic fixed point by a finite number of types (namely, n_s (or n_u)). The reason behind this interesting result is that the theorem of Hartman and Grobman, based on homeomorphisms, leads to a coarse decomposition of the set of vector fields under consideration.

As a consequence of the preceding theorems, the behavior of nonlinear differential equations near hyperbolic fixed points is equivalent up to a homeomorphism to the behavior of a simple system of linear differential equations. In the theory of nonlinear circuits, these mathematical results can be interpreted in the following way: The behavior of nonlinear circuits near an operational point where the Jacobian matrix of the descriptive equations (in statespace form) has only eigenvalues with nonzero real parts is ''similar'' to that of a corresponding linear dynamical circuit. Therefore, the analysis of these nonlinear circuits can be decomposed into two steps: (1) calculation of the operational points and the eigenvalues of

their Jacobian matrix and (2) restricting to the hyperbolic operational points we consider the behavior of the dynamics near these operational points; that is called small-signal behavior.

In the following section, we discuss more general methods for the analysis of vector fields that have at least one nonhyperbolic fixed point.

10.2 Normal Forms of Nonlinear Dynamical Circuit Equations

In Section 10.1, we present theorems useful for classifying the "local" behavior of nonlinear differential equations near hyperbolic fixed points by using "global" results from the theory of linear differential equations. An essential remaining problem is to calculate a homeomorphic transformation **h** in concrete cases. An alternative way to circumvent some of the difficulties is to apply the theory of normal forms that goes back to the beginning of the twentieth century and is based on classical ideas of Poincaré and Dulac. Detailed investigations of this subject are beyond the scope of this section and, therefore, an interested reader should consult the monographs of Refs. [2,3,10] as well as Ref. [29], where also further references of the theory of normal forms can be found. In this section, we present only the main ideas to illustrate its areas of applications.

In contrast to the theory described in Section 10.1, which is dedicated to hyperbolic cases, the theory of normal forms applies diffeomorphisms instead of homeomorphisms. This is necessary in order to distinguish the dynamical behavior of differential equations in more detail. To classify the topological types of fixed points of nonlinear differential equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ one proceeds in two steps:

1. Construction of a "normal form" in which the nonlinear terms of the vector field **f** take their "most simple" form
2. Determination of the topological type of the fixed point (under consideration) from the normal form

We present the main aspects of this "algorithm" without a proof. First, we suppose that the vector field $\mathbf{f}(\mathbf{x})$ of the nonlinear differential equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ satisfies $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ (otherwise, using a suitable transformation) and that it is represented by a formal Taylor expansion:

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \tilde{\mathbf{f}}(\mathbf{x}), \tag{10.14}$$

where

$\mathbf{A} = \mathbf{J}_f(\mathbf{x})$
$\tilde{\mathbf{f}}(\mathbf{x}) = O(\|\mathbf{x}\|^2)$ is of class C^r

Power (Taylor) series with no assumptions about convergence are called *formal series*. In practice, we begin with the formal series and then we determine the corresponding region of convergence (if available). Otherwise we use the framework of the theory of the so-called asymptotic series where a convergence in a strict sense is not necessary. In perturbation theory of differential equations most of the power series have to be considered as asymptotic series since a convergence in a strict sense is not available. More details about asymptotic series and perturbation theory can be, e.g., found in Refs. [28,45,57].

If a power series representation is derived, we apply a diffeomorphic C^r change of coordinates $\mathbf{h}$: $\mathbb{R}^n \to \mathbb{R}^n$ with $\mathbf{y} \mapsto \mathbf{x} = \mathbf{h}(\mathbf{y})$ ($\mathbf{h}(\mathbf{0}) = \mathbf{0}$) in the form of a near identity transformation:

$$\mathbf{h}(\mathbf{y}) := \mathbf{y} + \mathbf{h}^k(\mathbf{y}), \tag{10.15}$$

where $\mathbf{h}^k(\mathbf{y})$ is a homogeneous polynomial of order k in $\mathbf{y}$ ($k \geq 2$). The result of the transformation is

$$\dot{\mathbf{y}} = \{id + \mathbf{J}_h(\mathbf{y})\}^{-1} A\big(\mathbf{y} + \mathbf{h}^k(\mathbf{y})\big) + \{id + \mathbf{J}_h(\mathbf{y})\}^{-1} \tilde{\mathbf{f}}\big(\mathbf{y} + \mathbf{h}^k(\mathbf{y})\big) \tag{10.16}$$

$$= \mathbf{A}\mathbf{y} + \mathbf{g}(\mathbf{y}) \tag{10.17}$$

where

$\mathbf{J}_h(\mathbf{y})$ is the Jacobian matrix of $\mathbf{h}^k$ with respect to $\mathbf{y}$
$\mathbf{g}(\mathbf{h}) = O(\|\mathbf{y}\|^2)$ is a class C^r

For the following discussion it is useful to define $\mathbf{f}_T^k$ of a sufficient differentiable function $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^n$ as the "truncated" Taylor series in $\mathbf{0}$ expansion that have a degree less or equal k; the ith-order term of the Taylor series of $\mathbf{f}$ is denoted by $\mathbf{f}_i$. In a more abstract setting these "truncated" Taylor series are called *k-jets*; see Ref. [38] for more details. The set of $\mathbf{f}_T^k$'s forms a real vector space of functions and is denoted by H_k^n where the components are homogeneous polynomials in n variables of degree k. Using the k-jet notation and expanding $\mathbf{g}$ into a (formal) Taylor series, Equation 10.17 can be reformulated as

$$\dot{\mathbf{y}} = \mathbf{g}_T^{k-1}(\mathbf{y}) + \mathbf{g}_R^k(\mathbf{y}), \tag{10.18}$$

where $\mathbf{g}_R^k$ contains all terms of degree k or higher.

Expanding $\tilde{\mathbf{f}}$ of Equation 10.14 into a Taylor series

$$\tilde{\mathbf{f}} = \tilde{\mathbf{f}}_2 + \tilde{\mathbf{f}}_3 + \cdots \tag{10.19}$$

and using it to represent $\mathbf{g}_R^k$ of Equation 10.18 we can get

$$\dot{\mathbf{y}} = \mathbf{g}_T^{k-1}(\mathbf{y}) + \left\{\mathbf{f}_k - \left[\mathbf{A}\mathbf{y}, \mathbf{h}^k\right]\right\} + O\left(\|y\|^{k+1}\right), \tag{10.20}$$

where in the sense of Equation 10.17 the so-called Lie bracket $[\mathbf{A}\mathbf{y}, \mathbf{h}^k]$ of the linear vector field $\mathbf{A}\mathbf{y}$ and of $\mathbf{h}^k(\mathbf{y})$ is introduced by

$$\left[\mathbf{A}\mathbf{y}, \mathbf{h}^k\right](\mathbf{y}) := \mathbf{J}_h(\mathbf{y})\mathbf{A}\mathbf{y} - \mathbf{A}\mathbf{h}^k(\mathbf{y}). \tag{10.21}$$

Now we define the linear operator $L_A^k: H_k^n \to H_k^n$ by

$$L_A^k\mathbf{h}^k: \mathbf{y} \to \left[\mathbf{A}\mathbf{y}, \mathbf{h}^k\right](\mathbf{y}) \tag{10.22}$$

with the range $\mathcal{R}_k$, and let C_k ba any complementary subspace to $\mathcal{R}_k$ in H_k^n, i.e., $H_k^n = \mathcal{R}_k \oplus C_k (k \geq 2)$. Then, the following theorem implies a simplification of a nonlinear differential equation $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, which can be interpreted as an equivalent representation of the associated dynamical circuit equations.

THEOREM 10.3

Let $\mathbf{f}$: $\mathbb{R}^n \to \mathbb{R}^n$ *be a* C^r *vector field with* $\mathbf{f}(\mathbf{0}) = \mathbf{0}$ *and* $\mathbf{A} \in \mathbb{R}^{n \times n}$, *and let the decomposition* $\mathcal{R}_k \oplus C_k$ *of* H_k^n *be given. Then, there exists a series of near identity transformations* $\mathbf{x} = \mathbf{y} + \mathbf{h}^k(\mathbf{y}) \in \Omega$ *for* $k = 2, 3, \ldots, r$, *where* $\Omega \subset R^n$ *is a neighborhood of the origin and* $\mathbf{h}^k \in H_k^n$, *such that the equation* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ *is transformed to*

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{g}_2(\mathbf{y}) + \mathbf{g}_3(\mathbf{y}) + \cdots + \mathbf{g}_k(\mathbf{y}) + O(\|\mathbf{y}\|^{r+1}), \quad \mathbf{y} \in \Omega, \tag{10.23}$$

where $\mathbf{g}_k \in C_k$ *for* $k = 2, 3, \ldots$.

A proof of this theorem and the following definition can be found in, e.g., Ref. [4].

Definition 10.2: *Let $\mathcal{R}_k \oplus C_k$ be decompositions of H_k^n for $k = 2, 3, \ldots, r$. The "truncated" equation of (Equation* 10.23*)*

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y} + \mathbf{g}_2(\mathbf{y}) + \mathbf{g}_3(\mathbf{y}) + \cdots + \mathbf{g}_k(\mathbf{y}), \tag{10.24}$$

where $\mathbf{g}_k \in C_k$ $(k = 2, 3, \ldots, r)$ is called normal form of $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{f}}(\mathbf{x})$ associated with matrix A up to order $r \geq 2$ (with respect to the decomposition where $\mathbf{g}_k \in C_k$ for $k = 2, 3, \ldots, \mathcal{R}_k \oplus C_k$).

Theorem 10.3 suggests an equivalence relation in the set of vector fields $\mathbf{f}$ that decomposes the set into equivalence classes. Each class can be represented by using the definition of normal forms. Because a concrete normal form depends on the choice of complementary subspaces C_k, it is not unique. In practical problems a constructive method of finding these subspaces is needed. An elegant way to find these subspaces is to start with the introduction of a suitable inner product $\langle \cdot \mid \cdot \rangle_n$ in H_k^n that is needed to define the adjoint operator $(L_A^k)^\star$ of L_A^k (in a sense of linear algebra) by

$$\langle \eta | L_A^k(\xi) \rangle_n := \langle (L_A^k)^\star(\eta) | \xi \rangle_n, \quad \text{for all } \eta, \xi \in H_k^n. \tag{10.25}$$

It can be shown that $\left(L_A^k\right)^\star = L_{A^\star}^k$ where $\boldsymbol{A}^\star = \boldsymbol{A}^{\mathrm{T}}$ is the transposed matrix of the matrix $\boldsymbol{A}$. The desired construction is available as an application of the following theorem.

THEOREM 10.4

Vector space ker $\left\{L_{A^\star}^k\right\}$ that is the solution space of the equation $L_{A^\star}^k \xi = 0$ is a complementary subspace of $\mathcal{R}_k$ in H_k^n, i.e.,

$$H_k^n = \mathcal{R}_k \oplus ker\left\{L_{A^\star}^k\right\}. \tag{10.26}$$

The interesting reader is referred to Ref. [4] for a detailed discussion of this subject. As a consequence, finding a normal form in the above sense up to the order r requires solving the partial differential equation $L_{A^\star}^k \xi = 0$ with definition (Equation 10.22). From an algebraic point of view this means that a base of ker $\{L_{A^\star}^k\}$ has to be chosen, but this can be done, again, with some degrees of freedom. For example, the two sets of differential equations are distinct normal form of $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \tilde{\mathbf{f}}(\mathbf{x})$ associated with the same matrix $\mathbf{A}$ (see Ref. [4]):

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} ax_1^2 \\ bx_2^2 + ax_1x_2 \end{pmatrix}, \tag{10.27}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ bx_1^2 + bx_1x_2 \end{pmatrix}. \tag{10.28}$$

To reduce the number of nonlinear monomials in the normal forms, a more useful base of C_k must be determined. If a nonlinear differential equation of the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \tilde{\mathbf{f}}(\mathbf{x})$ is given with an arbitrary matrix $\mathbf{A}$, several partial differential equations need to be solved. This, in general, is not an easy task. If $\mathbf{A}$ is diagonal or has an upper triangular form, methods for constructing a base are available. For this purpose we introduce the following definition:

Definition 10.3: *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ possesses the eigenvalues $\lambda_1, \ldots, \lambda_n$ and let $x_1^{\alpha_1} x_1^{\alpha_2} \cdots x_1^{\alpha} x_1^{\alpha_n} \mathbf{e}_j$ be a monomial in n variables. It is called a resonant monomial if the so-called resonant condition*:

$$\alpha \cdot \lambda - \lambda_j = 0 \tag{10.29}$$

is satisfied ($\alpha^T := (\alpha_1, \ldots, \alpha_n)$, $\lambda^T := (\lambda_1, \ldots, \lambda_n)$). *If the resonant condition holds for some* $\sqrt{\alpha^T \alpha} \geq 2$ *and some* $j \in \{1, \ldots, n\}$, *we say that* **A** *has a resonant set of eigenvalues.*

The importance of this definition is that the following statement can be shown: a monomial $x_1^{\alpha_1} x_1^{\alpha_2} \cdots x_1^{\alpha} x_1^{\alpha_n} \mathbf{e}_j$ is an element of ker $\{L_{A^*}^k\}$ if and only if $\alpha^T \cdot \lambda - \lambda_j = 0$ [4]. The next theorem proves that if **A** is diagonal, a minimal normal form exists (in certain sense).

THEOREM 10.5

Let $\boldsymbol{A} = diag(\lambda_1, \ldots, \lambda_n)$. *Then an A-normal form equation up to order r can be chosen to contain all resonant monomials up to order r.*

If some eigenvalues of $\boldsymbol{A}$ are complex, a linear change to complex coordinates is needed to apply this theorem. Furthermore, theorems and definitions need to be modified to such complex cases. In the case of differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ +1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + O\left(\|\mathbf{x}\|^2\right) \tag{10.30}$$

that can be used to describe oscillator circuits, the coordinates are transformed to

$$\dot{z}_1 := x_1 + jx_2, \tag{10.31}$$

$$\dot{z}_2 = x_1 - jx_2 \tag{10.32}$$

with the resonant set of eigenvalues $\{-j, +j\}$ that can be written in a complex normal form (z represents z_1 and z_2, respectively):

$$\dot{z} = j + a_1|z|^2 z + \cdots + a_k|z|_z^{2k}. \tag{10.33}$$

This normal form equation—Poincaré normal form—is used intensively in the theory of the Poincaré–Andronov–Hopf bifurcation. The monograph of Hale and Koc̀ak [30] is worth reading for the illustration of this phenomenon in nonlinear dynamical systems. A detailed proof of the Poincaré–Andronov–Hopf theorem can be found, e.g., in Ref. [32]. These authors emphasize that further preparation of the system of differential equations is needed before the normal form theorem is applicable. In general, the linearized part of $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{f}}(\mathbf{x})$ (in a certain fixed point of the vector field $\mathbf{f}$) has two classes of eigenvalues: (a) *central* eigenvalues $\lambda_i \in j\,\mathbb{R} \subset \mathbb{C}$ and (b) *noncentral* eigenvalues $\lambda_i \in \mathbb{C} \setminus j\,\mathbb{R}$. The dynamic behavior of a system associated with the noncentral eigenvalues is governed by the theorem of Grobman and Hartman (see Section 10.1) and therefore it is oriented in generic sense at the exponential behavior of linear systems. If a system has no central eigenvalues it is called a *hyperbolic system*. In circuit applications network, models with a hyperbolic behavior are related to, e.g., amplifiers and active filters; see, e.g., Ref. [44].

A more complex behavior is only possible if the theorem of Grobman and Hartman cannot be applied, i.e., we have systems with central eigenvalues see Ref. [49]. Therefore in the design of oscillator circuits we have to derive conditions to get central eigenvalues, e.g., Barkhausen criterion. The noncentral eigenvalues in these systems are related with the relaxation behavior and should be eliminated if we are interested in the dominant asymptotic behavior. A systematic procedure to eliminate noncentral eigenvalues from the system is based on the so-called center manifold theorem (see Ref. [11]) that is used also in the paper of Hassard and Wan [32]. A numerical algorithm is presented by Hassard et al. [31]. A more detailed discussion of this essential theorem goes beyond the scope of this section; however, more about the theory of normal forms of vector fields and its applications in circuit analysis can be found

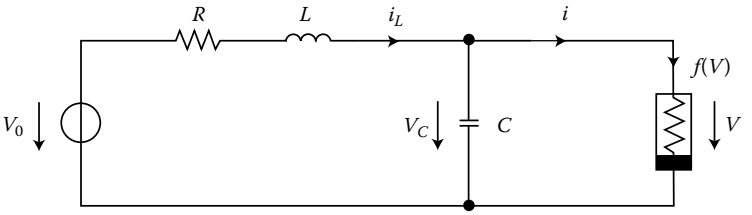


FIGURE 10.2 Nonlinear circuit with tunnel diode.

in Refs. [21,22]. Until now, we have discussed differential equations of the type $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. Descriptive equations of circuits consist, in general, of linear and nonlinear algebraic equations, as well as differential equations. Therefore, the so-called constrained differential equations need to be considered for applications in circuit theory. A typical example is the well-known circuit shown in Figure 10.2, containing a model of a tunnel diode (see Refs. [9,53]). Circuit equations can be written as

$$C\frac{dv_C}{dt} = i_L - f(v_C), \tag{10.34}$$

$$L\frac{di_L}{dt} = -v_C - Ri_L + V_0, \tag{10.35}$$

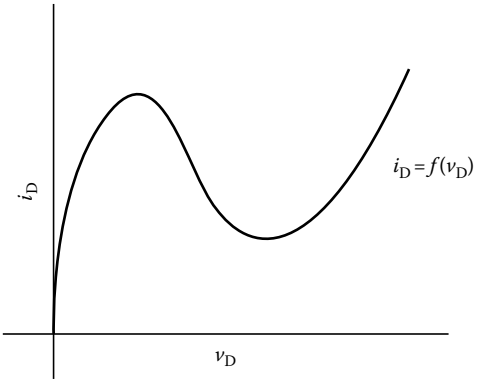


FIGURE 10.3 Characteristic of a tunnel diode.

where $i_D = f(v_D)$ is the characteristic of the tunnel diode (see Figure 10.3).

If the behavior of the circuit is considered with respect to the timescale $\tau := RC$, then the differential equations (Equations 10.34 and 10.35) can be reformulated as

$$\frac{dv}{d\theta} = i - F(v), \tag{10.36}$$

$$\varepsilon\frac{di}{d\theta} = -v - i + 1, \tag{10.37}$$

where

$\theta : t/\tau$
$i := (R/V_0)\, i_L$
$v := v_C/V_0$
$\varepsilon : L/(CR^2)$
$F(v) := (R/V_0)\, f(V_0 \cdot v)$

The behavior of the idealized circuit, where $\varepsilon = 0$ is often of interest. These types of descriptive equations are called constrained equations or differential-algebraic equations (DAEs). The above discussed theory of normal forms for vector fields cannot be applied directly to DAEs; however, a generalized theory is available, as presented (with applications) in Refs. [24,25].

10.3 Dimensionless Forms of Circuit Equations

In Section 10.1, the problem of equivalence is considered in a rather complex manner by adopting the mathematical point of view of dynamical systems. We use this approach in Section 10.6 to classify a

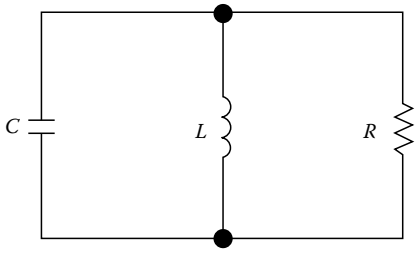


FIGURE 10.4 LCR circuit.

special class of nonlinear dynamical circuits. In contrast to this approach, another kind of equivalence of descriptive equations for linear time-invariant circuits and for certain nonlinear circuits is well known. It is illustrated by means of a simple example. The *RLC* parallel circuit in Figure 10.4, analyzed in the frequency domain, can be described by its impedance

$$Z(j\omega) = \frac{1}{1/R + j(\omega C - \frac{1}{\omega L})}. \tag{10.38}$$

In the case of small damping, this formula can be simplified using the so-called Q-factor that is defined as the ratio of the current flowing through L (or C) to the current flowing through the whole circuit in the case of resonance. The case of small damping is characterized by the condition $R/(2L) \ll \omega_0$, where $\omega_0 := 1 / \sqrt{LC}$ (Thompson's formula for LC circuits with $R = 0$). The Q-factor is given by $Q := 1/(\omega_0 \sqrt{RC})$. Simple calculations lead to a so-called normalized impedance

$$\tilde{Z} := \frac{Z}{R} = \left[1 - jQ^{-1}\left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega}\right)\right]^{-1} \tag{10.39}$$

that contains only two instead of three parameters. By using this method, a whole class of *RLC* circuits may be described by the same normalized impedance and which can be "de-normalized" at a certain frequency if a special filter application is considered. Therefore, an equivalence relation is defined in this manner. Handbooks of filter design written for practical electrical engineers contain diagrams of those and similar frequency curves of normalized impedances as well as admittances. Nor that the formula (Equation 10.39) is exact, although the interpretation of the parameters Q and ω_0 depends on the condition $R/(2L) \ll \omega_0$.

Methods for normalizing descriptive equations of circuits and for reducing the number of parameters are known in linear and nonlinear circuit theory; see, e.g., Ref. [20]. Unfortunately, these methods are stated without a presentation of their mathematisch reasoning. The main ideas for justification of normalization procedures are based on the so-called dimensional analysis. Their first applications in physics and the development of their mathematical foundations can be traced to the end of the nineteenth century. In this section, we discuss only a few aspects of this subject. Interested readers may find more details about the theory and applications of dimensional analysis as well as further references in a paper of Mathis and Chua [50]. In this paper they demonstrated that for a complete mathematical discussion of physical quantities, several algebraic concepts (e.g., Lie groups, etc.) are needed. In this section, a concise introduction into dimensional analysis is preferred and therefore, an intuitive introduction based on multiparameter Lie groups is presented. We use main ideas from the monograph of Ovsiannikov [55].

For describing physical systems, suitable descriptive physical quantities are required that can be measured. To perform a physical measurement of one of these quantities, we need at least one measuring instrument that provides us with corresponding measuring values on its scale. This scale is calibrated with respect to a standard cell. Therefore, an intuitive mathematical model of a physical quantity ϕ consists of two parts: a real number $|\phi|$ that characterizes its value, and a symbol E_ϕ that is due to the standard cell. In general, a measuring instrument is composed of elementary measuring instruments to evaluate, e.g., time or frequency, length, voltage, and charge. Each elementary instrument is calibrated with a corresponding standard cell and, therefore, associated to the instrument. Entirely, an arbitrary physical quantity ϕ is defined by

$$\phi := |\phi| E_1^{\lambda_1} E_2^{\lambda_2} \cdots E_r^{\lambda_r}, \tag{10.40}$$

where

r is the number of elementary instruments for measuring ϕ

λ_k determine how many times an elementary instrument have to be applied and whether the value on the scale needs to multiplied or divided

The dimensionality of a physical quantity ϕ is defined by

$$[\phi] := E_1^{\lambda_1} E_2^{\lambda_2} \cdots E_r^{\lambda_r}, \tag{10.41}$$

where $r \geq n$ such that a physical quantities is fully described by $\phi = |\ \phi\ | [\phi]$.
A physical quantity ϕ is called dimensionless if its dimensionality is null, that is if $\lambda_1 = 0$, $\lambda_2 = 0, \ldots,$ $\lambda_r = 0$. Moreover, a set of n physical quantities $\phi_1, \phi_2, \ldots, \phi_n$ is called dependent (in the sense of their dimensionality) if there exist integers $\chi_1, \chi_2, \ldots, \chi_n$ (not all equal zero) such that the product of these quantities

$$\phi_1^{\chi_1} \phi_2^{\chi_2} \cdots \phi_n^{\chi_n} \tag{10.42}$$

is dimensionless. Otherwise $\phi_1\ \phi_2 \cdots \phi_n$ are called independent.

The main problem of (physical) dimension theory is to determine how many independent physical quantities are included in a set of n describing quantities, to find them, and then express the other quantities in the set in terms of these independent quantities. As an application, a systematic procedure to normalize physical descriptive equations can be given—instead a intuitive manner as considered above.

In order to solve the main problem, the change of measuring instruments and measuring scales needs to be introduced and modeled in a mathematical manner. Obviously, in terms of modeling physical quantities, the same ϕ can be represented in different ways. Using two sets of measuring instruments, denoted by $E_1, E_2, \ldots, E_r$ and $\tilde{E}_1, \tilde{E}_2, \ldots, \tilde{E}_r$, respectively, ϕ is given by

$$|\phi| E_1^{\lambda_1} E_2^{\lambda_2} \cdots E_r^{\lambda_r} = |\tilde{\phi}| \tilde{E}_1^{\tilde{\lambda}_1} \tilde{E}_2^{\tilde{\lambda}_2} \cdots \tilde{E}_r^{\tilde{\lambda}_r}, \tag{10.43}$$

where in general $r \neq p$. This suggests the so-called analogy transformation (see Ref. [50]):

$$E_k = a_1^{\alpha_1} \cdots a_k^{\alpha_k} \tilde{E}_1^{\alpha_1} \cdots \tilde{E}_k^{\alpha_k}, \quad k = 1, \ldots, r. \tag{10.44}$$

Transformations of the scales of measuring instruments—in the following denoted as *scale transformations*—are special analogy transformations:

$$E_k = a_k \tilde{E}_k, \quad k = 1, \ldots, r. \tag{10.45}$$

It can be demonstrated that analogy transformations of dimension theory are special cases of the so-called extension groups. These groups belong to the r-parameter Lie groups and, subsequently, all results of dimension theory can be interpreted by theorems from this mathematical theory. Further details are contained in Ref. [55]. To introduce the main ideas, we consider only scale transformations in this section.

Let $\mathcal{Z} := \mathbb{R}^n \times \mathbb{R}^m$ be the Cartesian product of the set of n-column vectors $\mathbf{x}$ and m-column vectors $\mathbf{y}$, and let $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $\{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$, be, respectively, arbitrary (but fixed) bases of these vector spaces. Endowing $\mathcal{Z}$ with the structure of a direct sum $\mathbb{R}^n \oplus \mathbb{R}^m$, each $z \in \mathcal{Z}$ can be represented by (with respect to these bases)

$$z = \sum_{i=1}^{n} x^i \mathbf{e}_i + \sum_{k=1}^{m} y^k \mathbf{f}_k. \tag{10.46}$$

An extension of $\mathcal{Z}$ (with respect to these bases) is defined by the transformation:

$$h: z \to \sum_{i=1}^{n} c^i x^i \mathbf{e}_i + \sum_{k=1}^{m} d^k y^k \mathbf{f}_k \quad (c^i >, \, d^k > 0). \tag{10.47}$$

Obviously, the set of all extensions generates an Abelian group of transformations on $\mathcal{Z}$ that is an $(n+m)$-parameter Lie group. this group is denoted by diag$\{\mathbf{e}_i, \mathbf{f}_k\}$. Any subgroup $H \subset$ diag$\{\mathbf{e}_i, \mathbf{f}_k\}$ is called an extension group of $\mathcal{Z}$. We now consider extension groups H^r, with $0 < r \leq n+m$.

Ovsiannikov [55] demonstrated that extensions of H^r can be represented, choosing a parametric group, in the form:

$$\tilde{x}^i = x^i \prod_{\alpha=1}^{r} (a_\alpha)^{\lambda_\alpha^i}, \quad \tilde{y}^k = y^k \prod_{\alpha=1}^{r} (a_\alpha)^{\mu_\alpha^k}, \tag{10.48}$$

where

$i = 1, \ldots, n$

$k = 1, \ldots, m$

The main property of transformation groups is that they induce equivalence relations decomposing the subjects on which the group acts into equivalence classes. If h_p acts on elements $x \in X$, and let $\mathbf{p} \in \mathbb{R}^p$ the vector of parameters, an *orbit* $U(x)$ *of a point* $x \in X$ is defined by the set $U(x) := \{\xi \in X | \xi = h_p(x, \mathbf{p})$, for all $\mathbf{p} \in \mathbb{R}^p\}$. In this sense, the points of an orbit can be identified by a transformation group. A transformation group acts transitive on X if there exists an orbit $U(x)$ that is an open subset of X, with $\bar{U} = X$.

To study the so-called local Lie groups with actions that are defined near a null neighborhood of the parameter space (including the vector $\mathbf{0}$), we can discuss the Lie algebra that characterizes the local behavior of the associated local Lie group. In finite dimensional parameter spaces, a Lie algebra is generated by certain partial differential operators. Using the representations (Equation 10.48) of H^r, the operators are of the form:

$$\sum_{i=1}^{n} \lambda_\alpha^i x^i \frac{\partial}{\partial x^i} + \sum_{i=1}^{m} \mu_\alpha^k y^k \frac{\partial}{\partial y^k}, \tag{10.49}$$

where

$i = 1, \ldots, n$

$k = 1, \ldots, m$

These operators can be represented in a matrix form:

$$\mathbf{M}(z) := \mathbf{M}_1 \circ \text{diag}\{x^1, \ldots, x^n; y^1, \ldots, y^m\}, \tag{10.50}$$

where

$$\mathbf{M}_1 := \begin{pmatrix} \lambda_1^1 & \cdots & \lambda_1^n, \mu_1^1 & \cdots & \mu_1^m \\ \vdots & \cdots & & \cdots & \vdots \\ \lambda_r^1 & \cdots & \lambda_r^n, \mu_r^1 & \cdots & \mu_r^m \end{pmatrix}. \tag{10.51}$$

Obviously, H^r is not transitive if $r < n+m$.

In order to solve the main problem of dimension theory, we need to introduce invariants of a Lie group. Let $F: X \to Y$ be a function on X and let transformations h_p of a transformation group act on X, then F is an invariant of the group if $F(h_p(x)) = F(x)$ holds for any $x \in X$ and $\mathbf{p} \in \mathbb{R}^p$. The invariant $J: X \to Y$ is called a universal invariant if there exists, for any invariant $F: X \to Y$ of the group, a function Φ such that $F = \Phi \circ J$. The following main theorem can be proved for the extension group [55].

THEOREM 10.6

For the extension group H^r on $\mathcal{Z}$, there exists a universal invariant $J: \mathcal{Z} \to \mathbb{R}^{n+m-r}$ if the condition $r < n + m$ is satisfied. The independent components of J have the monomial form:

$$J^\tau(Z) = \prod_{i=1}^{n} \left(x^i\right)^{\theta_k^\tau} \cdot \prod_{k=1}^{m} \left(y^k\right)^{\sigma_k^\tau}, \tag{10.52}$$

where $\tau = 1, \ldots, n + m - r$.

If dimensional analysis considers only scale transformations (Equation 10.45), this theorem contains the essential result of the so-called *Pi-theorem*. For this purpose we present a connection between the dimensionalities and the extension group H^r (see Ref. [55]). The group H^r of the space $\mathbb{R}^n$, defined only be the dimensions of the physical quantities ϕ_k with respect to the set of symbols $\{E_\alpha\}$, has a one-to-one correspondence with every finite set $\{\phi_k\}$ of n physical quantities, which can be measured in the system of symbols $\{E_\alpha\}$ consisting of r independent measurement units; see Equation 10.43. The transformations belonging to the group H^r give the rule of change in the form

$$|\tilde{\phi}| = |\phi| \prod_{\alpha=1}^{r} (a^\alpha)^{\lambda_\alpha} \tag{10.53}$$

of the numerical values $|\phi_k|$ as a result of the transition from the units $\{E_\alpha\}$ to $\{\tilde{E}_\alpha\}$ by means of Equation 10.45.

As a consequence of this relationship, a quantity ϕ is dimensionless if and only if its numerical value is an invariant of the group H^r. Thus, the problem to determine the independent physical quantities of a given set of quantities is solved by the construction of a universal invariant of H^r stated by the Pi-theorem; see also Ref. [7]. Normalization as well as the popular method of dimension comparison, are consequences of the invariance of physical equations with respect to the group of analogy (scale) transformations. In applications of dimensional theory, a normal form that has certain advantageous properties is desired. For example, it is useful to reduce the number of parameters in physical equations. Normal forms of this type are used very often in practical calculations, but with no clarification of their mathematical foundations.

Circuit equations, similar to other physical equations, contain numberous parameters. In applications, it is often desired to suppress some of these parameters and moreover, they should be replaced by the numerical value 1.

For this purpose, Desloge [26] chooses a new system of units $\{E_\alpha\}$. A theory of Desloge's method, based on analogy transformation (Equation 10.44) instead of scale transformation (Equation 10.45), was presented by Mathis and Chua [50]. The main idea behind this method is that, beside the foundation units time $[T]$, voltage $[E]$, and charge $[Q]$ that are useful in circuit theory, the units of other parameters are considered as foundational units. We denote the units by $[A_\alpha]$ instead of E_α. For example, in the case of the tunnel-diode circuit (see Figure 10.2), $[T]$, $[E]$, and $[Q]$, as well as $[R]$, $[C]$, and $[L]$, need to be

discussed. As a consequence of Desloge's method, three of the four parameters can be suppressed and the other variables will be normalized. The method works in the case of linear as well as nonlinear circuits.

The method is illustrated using the tunnel-diode circuit equations (Equations 10.34 and 10.35). At first, the dimensional matrix is determined by

$$\begin{array}{c} \\ [R] \\ [L] \\ [C] \end{array} \begin{array}{c} \begin{array}{ccc} [T] & [E] & [Q] \end{array} \\ \begin{pmatrix} 1 & 1 & -1 \\ 2 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \end{array} \tag{10.54}$$

that characterizes the relation between the dimensions of *t, v, q, R, C,* and *L*.

Now, Desloge considers another set of power independent dimensional scalars A_1, A_2, and A_3 with

$$[A_i] = [T]^{a_i^1}[E]^{a_i^1 2}[Q]^{a_i^3}, \quad i = 1, 2, 3. \tag{10.55}$$

These relations are interpreted as an analogy transformation (Equation 10.44). Applying the map $L(\cdot)$ that has the same properties as the logarithmic function (see Ref. [50]) to Equation 10.55, the symbols $L([A_1])$, $L([A_2])$, and $L([A_3])$ are represented by linear combinations of $L([T])$, $L([E])$, $L([Q])$. The coefficient matrix in Equation 10.55 is regular and contains the exponents. Solving these linear equations using "antilog," the $[T]$, $[E]$, and $[Q]$ are products of powers of $[A_1]$, $[A_2]$, and $[A_3]$. In this manner, dimensionsless versions of differential equations of the tunnel-diode circuit can be derived.

By using the independent units $A_1 := L$, $A_2 := C$, and $A_3 := V_0$ to replace their values $|V_0|$, $|L|$, and $|C|$ by 1 (with respect to the new units), the following equation is derived by Desloge's approach sketched previously

$$\begin{array}{c} \\ [V_0] \\ [L] \\ [C] \end{array} \begin{array}{c} \begin{array}{ccc} [T] & [E] & [Q] \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 2 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \end{array} \begin{pmatrix} \ln([T]) \\ \ln([E]) \\ \ln([Q]) \end{pmatrix} = \begin{pmatrix} \ln([V_0]) \\ \ln([L]) \\ \ln([C]) \end{pmatrix}. \tag{10.56}$$

Multiplying Equation 10.56 by the inverse of the dimensional matrix

$$\begin{array}{c} \\ [T] \\ [E] \\ [Q] \end{array} \begin{array}{c} \begin{array}{ccc} [V_0] & [L] & [C] \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \tag{10.57}$$

and applying "antilog" to the result, we obtain

$$[T] = [L]^{1/2}[C]^{-1/2}, \quad [E] = [V_0], \quad [Q] = [V_0]\,[C]. \tag{10.58}$$

From these equations, the relations between the old and new units can be derived; see Ref. [50]. *T, E,* and *Q* are expressed by the new units *L, C,* and V_0 and the parameters and variables in Equations 10.34 and 10.35 can be reformulated if the numerical values of V_0, *L,* and *C* are added

$$T = |L|^{-1/2}|C|^{-1/2}L^{1/2}C^{1/2}, \quad E = |V_0|^{-1}V_0, \quad Q = |V_0|^{-1}|C|^{-1}V_0C. \tag{10.59}$$

These relations represent parameters and variables of the tunnel-diode circuit with respect to the new quantities

$$R = \frac{|R||C|^{1/2}}{|L|^{1/2}} L^{1/2} C^{-1/2}, \quad V_0 = 1 \cdot V_0, \quad L = 1 \cdot L, \quad C = 1 \cdot C, \tag{10.60}$$

$$i_L = \frac{|i_L||L|^{1/2}}{|V_0||C|^{1/2}} V_0 L^{-1/2} C^{1/2}, \quad v_C = \frac{|v_C|}{|V_0|} V_0, \quad t = \frac{|t|}{|L|^{1/2}|C|^{1/2}} L^{1/2} C^{1/2}. \tag{10.61}$$

The dimensional exponents for these quantities can be found by using the inverse dimensional matrix (Equation 10.57):

1. *T, E, Q*: their exponents correspond the associated rows of Equation 10.57
2. V_0, *L, C, R*: premultiply Equation 10.57 with the corresponding row of Equation 10.54

For example, taking $[C] \triangleq (0 - 11)$ results in

$$[C] \begin{array}{c} \begin{array}{ccc} [T] & [E] & [Q] \end{array} \\ \begin{pmatrix} 0 & -1 & 1 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{c} [T] \\ [E] \\ [Q] \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} [V_0] & [L] & [C] \end{array} \\ = [C] \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \end{array} \tag{10.62}$$

or with the corresponding row $[R] \triangleq (11 - 1)$ of Equation 10.54

$$[R] \begin{array}{c} \begin{array}{ccc} [T] & [E] & [Q] \end{array} \\ \begin{pmatrix} 1 & 1 & -1 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{c} [T] \\ [E] \\ [Q] \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{ccc} [V_0] & [L] & [C] \end{array} \\ = [R] \begin{pmatrix} 0 & 1/2 & -1/2 \end{pmatrix}. \end{array} \tag{10.63}$$

With these representations of the dimensional quantities, we can obtain a dimensionless representation of Equations 10.34 and 10.35

$$\frac{\mathrm{d}\bar{v}_C}{\mathrm{d}\bar{t}} = \bar{i}_L - \bar{f}(\bar{v}_C), \tag{10.64}$$

$$\frac{\mathrm{d}\bar{i}_L}{\mathrm{d}\bar{t}} = 1 - \sqrt{\varepsilon}\bar{i}_L - \bar{v}_C, \tag{10.65}$$

where

$$\bar{v}_C := \frac{|v_C|}{|V_0|}, \quad \bar{t} := \frac{|t|}{\sqrt{|L||C|}}, \quad \sqrt{\varepsilon} := \frac{|R||C|^{1/2}}{|L|^{1/2}}, \quad \bar{i}_L := \frac{|i_L||L|^{1/2}}{|V_0||C|^{1/2}}. \tag{10.66}$$

Furthermore, the dimensionless tunnel-diode current $\bar{f}$ is defined by

$$\bar{f}(\bar{v}_C) := V_0^{-1} L^{1/2} C^{-1/2} f(V_0 \bar{v}_C). \tag{10.67}$$

The associated dimensionless form of Equations 10.36 and 10.37 can be derived by another scaling of the current $\hat{i}_L := \sqrt{\varepsilon}\bar{i}_L$. Obviously, the dimensionless normal form is not unique.

The classical dimensional analysis shows that R^2C/L is the only dimensionless constant of the quantities in Equations 10.34 and 10.35. Because the describing equations of the parallel *LCR* circuit

include the same variables constants, the results of the previous dimensional analysis of the tunnel-diode circuit can be used to normalize Equation 10.38.

Further interesting applications of Desloge's approach of suppressing superfluous parameters in the describing equations can be found in the theory of singular perturbation analysis. The reader is referred, e.g., to the monograph of Smith [57] for further details. Miranker [53] demonstrated that the differential equations of the tunnel-diode circuit can be studied on three timescales $\tau_1 := L/R$, $\tau_2 := RC$, and $\tau_3 := \sqrt{LC}$ where different phenomena arise. If these phenomena are known, corresponding timescales can be written down. However, since all dimensionless equations can be derived in a systematic manner using Desloge's approach also the corresponding timescales result. In this way, e.g., all normalized differential equations describing Chua's circuit (see Ref. [42]) can be obtained but other representations of these differential are possible using dimensional analysis.

10.4 Equivalence of Nonlinear Resistive *n*-Ports

In this section, we only consider equivalence of nonlinear resistive n-ports; however, resistive circuits without accessible ports will not be included in the discussion. Although the explanations that follow are restricted to resistive n-ports, this theory can be easily extended to pure capacitive and inductive n-ports; see Ref. [23]. In Section 10.5, we give a definition of those n-ports.

At first, we consider linear resistive 1-ports that contain Ohmic resistors described by $v_k = R_k i_k$ or/and $i_k = G_k u_k$, and independent current and voltage sources $v_k = V_0^k$ and $i_k = I_0^k$. We can use Helmholtz–Thevenin's or Mayer–Norton's theorem [40,51] to compare any two of those 1-ports and reduce a complex 1-port to a more simple ''normal'' form. Therefore, two of those 1-ports are called equivalent if they have the same Helmholtz–Thevenin or Mayer–Norton resistor-source 1-port representation. Clearly, by this approach, an equivalence relation is defined in the set of linear resistive 1-ports and it is decomposed into ''rich'' classes of 1-ports. To calculate these normal forms, Δ-Y and/or Y-Δ transformations are needed; see Refs. [20,51]. It is known that this approach is not applicable to nonlinear resistive networks because, e.g., Δ-Y and/or Y-Δ transformations generally do not exist for nonlinear networks. This was observed by Millar [52] for the first time. Certain circuits where these transformations can be performed were presented by Chua [14]. More recently, Boyd and Chua [5,6] clarified the reasons behind this difficulty from the point of view of Volterra series. As a conclusion, the set of nonlinear resistive 1-ports can be decomposed into equivalence classes, but no reasonable large classes of equivalent 1-ports exist. More general studies of this subject are based on the well-known substitution theorem, which can be used successfully in a certain class of nonlinear circuits. A detailed discussion of the classical substitution theorem can be found in Refs. [27,57] and in an abstract setting in Refs. [33,56]. Some results applicable to 1-ports can be generalized to linear resistive n-ports (extraction of independent sources), but this point of view is not suited for nonlinear resistive n-ports.

Better understanding of nonlinear resistive n-ports and the problem of equivalence cannot be based on the ''operational'' approach mentioned earlier. Instead, a geometrical approach that was developed by Brayton and Moser [9] is more useful. These authors (see also Ref. [8]) characterize a resistive n-port in a generic manner by n independent relations between the $2n$ port variables, n-port currents $i_1, \ldots, i_n$, and n-port voltages $u_1, \ldots, u_n$. Geometrically, this means that in the $2n$-dimensional space of port variables, the external behavior of a resistive n-port can be represented generically by an n-dimensional surface. The classical approach formulates a system of equations by

$$\begin{aligned} y_1 - f_1(x_1, \ldots, x_n) &= 0, \\ &\vdots \\ y_n - f_n(x_1, \ldots, x_n) &= 0, \end{aligned} \tag{10.68}$$

where x's and y's are the port variables. The zero set of equations (Equation 10.69) corresponds to the n-dimensional surface. Therefore, two n-ports are called equivalent if they are different parametrizations of the same n-port surface. As an application of this point of view, Brayton and Moser [9] demonstrated that a 2-port consisting of a Y-circuit and a circuit consisting of a Δ-circuit cannot be equivalent, in general. For example, they proved by means of Legendre transformations that a Y-circuit with two Ohmic resistors and a third resistor can be equivalent to a Δ-circuit if and only if the third resistor is also linear. Therefore, the operational approach is not a very useful equivalence concept for nonlinear n-ports.

The subject of synthesizing a prescribed input–output behavior of nonlinear resistive n-ports is closely related to the problem of of equivalence. Several results were published in this area using ideal diodes, resistors with a concave and convex characteristic, dc voltage and current sources, ideal op-amps, and controlled sources. Therefore, we give a short review of some of these results. On the other hand we do not consider the synthesis of resistive n-ports.

Although the synthesis of nonlinear resistive n-ports was of interest to many circuit designers since the beginning of twentieth century, the first systematic studies of this subject based on previous studies of Millar [52] were published by Chua [13,14]; see, e.g., Itoh [37] for more recent aspects of nonlinear circuit synthesis. Chua's synthesis approach based on the introduction of new linear 2-ports (R-rotators, R-reflectors, and scalors) as well as their electronic realizations. Now, curves in the i–v space of port current i and port voltage v that characterize a (nonlinear) resistive 1-port can be reflected and scaled in a certain manner. Chua suggested that a prescribed behavior of an active or passive nonlinear resistive 1-port can be reduced essentially to the realization of passive i–v curves. Piecewise-linear approximations of characteristics of different types of diodes, as well as the previously mentioned 2-ports, are used to realize a piecewise-linear approximation of any prescribed passive i–v curve. In another article, Chua [15] discussed a unified procedure to synthesize a nonlinear dc circuit mode that represents a prescribed family of input and output curves of any strongly passive 3-terminal device (e.g., a transistor). It was assumed that the desired curves are piecewise-linear. Since then, this research area has grown very rapidly and piecewise-linear synthesis and modeling has become an essential tool in the simulation of nonlinear circuits; see Refs. [19,34,39,41].

10.5 Equivalence of Nonlinear Dynamical *n*-Ports

In this section, we consider more general n-ports that can be used for device modeling; see Ref. [16]. Although many different *lumped* multiterminal and multiport circuits are used, a decomposition into two mutually exclusive classes is possible: algebraic and dynamic multiterminal and multiports. Adopting the definition of Chua [16], an $(n+1)$-terminal of n-port circuit is called an algebraic element if and only if its constitutive relations can be expressed symbolically by algebraic relationships involving at most two dynamically independent variables for each port. In the case of an 1-port, a so-called memristor is described by flux and charge, a resistor by voltage and current, an inductor by flux and current, and a capacitor by voltage and charge. An element is called a dynamic element if and only if it is not an algebraic element.

Despite the fact that the class of all dynamic elements is much larger than that of algebraic ones, the following theorem of Chua [16] suggests that resistive multiports are essential for dynamic elements, too.

THEOREM 10.7

Every lumped $(n+1)$-terminal or n-port element can be synthesized using only a finite number n of linear 2-terminal capacitors (or inductors) and one (generally nonlinear) $(n+m)$-port resistor with n accessible ports and m ports for the capacitors.

This theorem demonstrates that any n-port made of lumped multiterminal and/or multiport elements is equivalent to a multiterminal network where all of its nonlinear elements are *memoryless*. This fact offers a possibility to classify $(n+1)$-terminal and n-port elements in an operational manner.

The proof of this theorem provides the answer of a fundamental question: what constitutes a minimal set of network elements from which all lumped elements can be synthesized?

THEOREM 10.8

The following set $\mathcal{M}$ of network elements constitutes the minimal basic building blocks in the senses that any lumped multiterminal or multiport element described by a continuous constitutive relation on any closed and bounded set can be synthesized using only a finite number of elements of $\mathcal{M}$, and that this statement is false if even one element is deleted from $\mathcal{M}$:

1. *Linear 2-terminal capacitors (or inductors)*
2. *Nonlinear 2-terminal resistors*
3. *Linear 2-port current-controlled voltage sources (CCVS) defined by $v_1 = 0$ and $v_2 = ki_1$*
4. *Linear 2-port current-controlled current sources (CCCS) defined by $i_1 = 0$ and $i_2 = \tilde{k}v_1$*

The proof of Theorem 10.8 (see Ref. [16]) is based on a remarkable theorem of Kolmogoroff, which asserts that a continuous function $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}$ can always be decomposed over the unit cube of $\mathbb{R}^n$ into a certain sum of functions of a single variable. Although the proof of Theorem 10.8 is constructive, it is mainly of theoretical interest since the number of controlled sources needed in the realization is often excessive.

10.6 Equivalence of Reciprocal Nonlinear Dynamic Circuits

As already mentioned in Section 10.1, a certain set of circuits can be decomposed into classes of equivalent circuits by some type of equivalence relation. Such equivalence relations are introduced in a direct manner with respect to the descriptive equations, using a transformation group of classifying the behavior of the solution of the descriptive equations. In the last few sections, several useful ideas for defining equivalence relations were discussed that can be suitable for circuit theory. In this section equivalent dynamical circuits are considered in more detail. It should be emphasized again that equivalence has a different meaning depending of the applied equivalence relation.

As the so-called state-space equations in circuit and system theory arose in the early 1960s, a first type of equivalence was defined because various circuits can be described by the same state-space equations. Of course, from this observation an equivalence relation is induced; see Ref. [61] for further references. Although this approach is interesting, in some cases different choices of variables for describing nonlinear circuits exist that need not lead to equivalent state-space equations; see, e.g., Ref [17]. In other words, the transformations of coordinates are not well conditioned. This approach was applied also to nonlinear input–output systems.

If we consider nonlinear reciprocal circuits, the Brayton–Moser approach [9] to formulate the describing equations for this class of circuits is very suitable; see also Ref. [48]. In order to formulate a generalized type of state-space equations, Brayton and Moser used a so-called mixed-potential function P. P depends on the capacitor voltages $\mathbf{v}$ and the inductor currents $\mathbf{i}$ where the vector field of the circuit equations can be derived by partial derivatives. The Brayton–Moser equations can be formulated in the following manner:

$$\mathbf{C}(\mathbf{v})\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\frac{\partial P}{\partial \mathbf{v}}(\mathbf{v},\mathbf{i}), \quad -\mathbf{L}(\mathbf{i})\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t} = -\frac{\partial P}{\partial \mathbf{i}}(\mathbf{v},\mathbf{i}). \tag{10.69}$$

A discussion of the Brayton–Moser equations from a point of view of constrained differential equation in a differential geometric setting can be found in Ref. [46]. A very general concept of constructing mixed-potential functions is presented by Weiss et al. [60]. In the following, we consider the equivalence concept for reciprocal nonlinear dynamical circuits based in Brayton–Moser equations where Varaiya and Verma [58] included external ports (input–output description). For this purpose they use a more compact form of the Brayton–Moser where the generalized state vector $\mathbf{x} := (\mathbf{v}, \mathbf{i})$ and the coefficient matrix $\mathbf{A} := \text{diag}(\mathbf{C}(\mathbf{v}), -\mathbf{L}(\mathbf{i}))$ is introduced. It should be noted that another geometric approach for nonlinear dynamical input–output circuits is available based on Hamiltonian equations with external ports—the so-called port Hamiltonian equations—where a so-called Dirac structure is used for energy preserving interconnections. With respect to this concept, the reader is referred to the literature; see recent publications of Maschke [47] and can der Schaft [12]) and the cited older publications.

The input–output circuit description of Verma [59] is formulated in the following manner: Let $\mathbf{x} \in \mathbb{R}^n$ the state-space vector, $\mathbf{u} \in \mathbb{R}^m$ the input vector, and $\mathbf{e} \in \mathbb{R}^m$ the output vector, then the Brayton–Moser type state-space equations can be generated by a matrix-valued function $\mathbf{A}(\mathbf{x})$: $\mathbb{R}^n \to \mathbb{R}^{n\times n}$ and a real-valued function P: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$

$$\mathbf{A}(\mathbf{x})\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\frac{\partial P}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u}), \tag{10.70}$$

$$\mathbf{e} = \frac{\partial P}{\partial \mathbf{u}}(\mathbf{x}, \mathbf{u}). \tag{10.71}$$

For two such circuits $N_1 = \{\mathbf{A}_1, P_1\}$ and $N_2 = \{\mathbf{A}_2, P_2\}$, Varaiya and Verma [58] defined the following equivalence concept.

Definition 10.4: *Two Brayton–Moser type circuits N_1 and N_2 with the outputs $\mathbf{e}_1 = \partial P_1/\partial \mathbf{u}$ and $\mathbf{e}_2 = \partial P_2/\partial \mathbf{u}$ are input–output equivalent if there exists a diffeomorphism Θ with $\mathbf{y} = \Theta(\mathbf{x})$, such that for all $\mathbf{x}_0 \in \mathbb{R}^n$, all input functions $\mathbf{u}$, and all $t \geq 0$ the following assumptions are satisfied*

1. $\Theta(x(t, \mathbf{x}_0, \mathbf{u})) = \mathbf{y}(t, \Theta(\mathbf{x}_0), \mathbf{u})$
2. $\mathbf{e}_1(t, \mathbf{x}_0, \mathbf{u}) = \mathbf{e}_2(t, \Theta(\mathbf{x}_0), \mathbf{u})$

The diffeomorphism Θ is called the equivalence map.

Thus, two circuits are equivalent if their external behavior is identical, i.e., if for the same input and corresponding states they yield the same output. It is clear that this definition results an equivalence relation on the set of all dynamical circuits under consideration. In their paper, Varaiya and Verma [58] showed that, under the additional assumption of controllability, the diffeomorphism Θ establishes an isometry between the manifold with the (local) pseudo-Riemannian metric $(\mathrm{d}\mathbf{x}, \mathrm{d}\mathbf{x}) := \langle \mathrm{d}\mathbf{x}, \mathbf{A}_1\, \mathrm{d}\mathbf{x}\rangle$ and the manifold with the (local) pseudo-Riemannian metric $(\mathrm{d}\mathbf{y}, \mathrm{d}\mathbf{y}) := \langle \mathrm{d}\mathbf{y}, \mathbf{A}_2\, \mathrm{d}\mathbf{y}\rangle$ in many interesting cases of nonlinear reciprocal circuits. This statement has an interesting interpretation in the circuit context. It can be proven that Θ must relate the reactive parts of the circuits N_1 and N_2 in such a way that, if N_1 is in the state $\mathbf{x}$ and N_2 is in the state $\mathbf{y} = \Theta(\mathbf{x})$, and if the input $\mathbf{u}$ is applied, then

$$\left\langle \frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t}, \mathbf{L}(\mathbf{i})\frac{\mathrm{d}\mathbf{i}}{\mathrm{d}t}\right\rangle - \left\langle \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}, \mathbf{C}(\mathbf{v})\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right\rangle = \left\langle \frac{\mathrm{d}\tilde{\mathbf{i}}}{\mathrm{d}t}, \tilde{\mathbf{L}}(\tilde{\mathbf{i}})\frac{\mathrm{d}\tilde{\mathbf{i}}}{\mathrm{d}t}\right\rangle - \left\langle \frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t}, \tilde{\mathbf{C}}(\tilde{\mathbf{v}})\frac{\mathrm{d}\tilde{\mathbf{v}}}{\mathrm{d}t}\right\rangle. \tag{10.72}$$

The concept of equivalence defined in a certain subset of nonlinear dynamic circuits with input and output terminals given by Varaiya and Verma [58] is based on diffeomorphic coordinate transformations

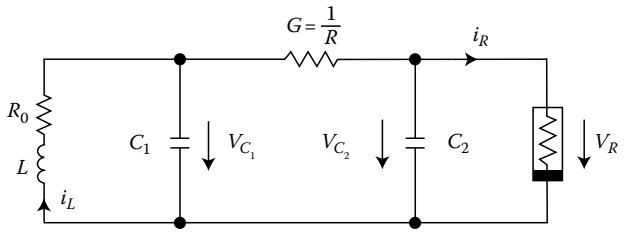


FIGURE 10.5 Modified Chua's circuit.

(the transformation group of diffeomorphisms). Unfortunately, the authors present no ideas about the kind of "coarse graining" produced in this set of circuits by their equivalence relation. However, a comparison to C^k conjugacy or C^k equivalence of vector fields in Section 10.1 implies that input–output equivalence leads to a "fine" decomposition in the set of Brayton–Moser input–output circuits. To classify the main features of the dynamics of circuits, the concept of topological equivalence (the transformation group of homeomorphisms) is useful. On the other hand, in the case of circuits with nonhyperbolic fixed points, the group of diffeomorphisms is needed to distinguish the interesting features. An interesting application of C^1 equivalence of vector fields is given by Chua [18]. To compare nonlinear circuits that generate chaotic signals, Chua applied the concept of equivalence relation and concluded that the class of circuits and systems that are C^1 equivalent to Chua's circuit (see Figure 10.5) is relative small. The nonlinearity in this circuit is described by a piecewise-linear i–v characteristic; see Ref. [42] for further details. The equations describing Chua's circuit are

$$\frac{dv_{C_1}}{dt} = \frac{1}{C_1}[G(v_{C_2} - v_{C_1}) - f(v_{C_1})], \tag{10.73}$$

$$\frac{dv_{C_2}}{dt} = \frac{1}{C_2}[G(v_{C_1} - v_{C_2}) + i_L)], \tag{10.74}$$

$$\frac{di_L}{dt} = \frac{1}{L}[v_{C_2} - R_0 i_L)], \tag{10.75}$$

where $R_0 = 0$ and the piecewise-linear function is defined by (G_a, G_b, E suitable constants)

$$f(v_{C_1}) := G_b v_{C_1} + \frac{1}{2}(G_a - G_b)(|v_{C_1} + E| - |v_{C_1} - E|). \tag{10.76}$$

Chua's extended approach to study the set of the piecewise-linear circuits that includes Chua's circuit introduces the concept of global unfoldings. This concept can be considered as an analogy to the theory of "local unfoldings" of nonhyperbolic systems in a small neighborhood of singularities [3,30]. Heuristically, a minimum number of parameters in a given nonhyperbolic system is obtained. Chua demonstrated that Chua's circuit with arbitrary $R_0 \neq 0$ can be considered as an "unfolding" of the original circuit. Furthermore, he proved that a class of circuits that can be described without any loss of generality by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}, \quad x_1 \leq -1 \tag{10.77}$$

$$= \mathbf{A}\mathbf{x}, \quad -1 \leq x_1 \leq +1 \tag{10.78}$$

$$= \mathbf{A}\mathbf{x} + \mathbf{b}, \quad x_1 \geq +1 \tag{10.79}$$

is equivalent to the unfolded Chua's circuit if certain conditions are satisfied. In the associated parameter space, these conditions defined a set of measure zero. The proof of this theorem as well as some applications are included in Ref. [18].

The ideas of normal forms presented in Section 10.2 can be applied to nonlinear circuits with hyperbolic and nonhyperbolic fixed points. A similar theory of normal forms of maps can be used to study limit cycles, but this subject is beyond our scope; see Ref. [3] for further details. In any case the vector field has to be reduced to lower dimensions and that can be achieved by the application of the so-called center manifold theorem. Altman [1] illustrated this approach by calculating the center manifold of Chua's circuit equations and its normal form in a tutorial style. To perform the analytical computations the piecewise nonlinearity (Equation 10.78) is replaced by a cubic function $f(x) = c_0 x + c_1 x^3$. Based on this normal form, Altman studied bifurcations of Chua's circuits.

In the following, we describe applications of normal form theory from Section 10.2 to decompose nonlinear dynamical circuits at an arbitrary fixed point into nondynamical and dynamical parts; a sketch of this concept is presented by Keidies and Mathis [43]. In this section we restrict ourself to nonlinear dynamical circuits with constant sources where the describing equations are formulated in a state-space form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n, \tag{10.80}$$

where all nonlinear reactances are replaced by linear reactances, nonlinear resistors, and linear controlled sources; see, e.g., Ref. [20]. It is assumed that all nonlinearities are polynomial functions and can be interpreted as nonlinear controlled sources. It can be assumed that the circuit is decomposed into a linear part that consists of linear reactances and resistive elements, and the nonlinear controlled sources. In other words, the RHS **f** of Equation 10.82 can be reformulated in the form $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \tilde{\mathbf{f}}(\mathbf{x})$; a block diagram is shown in Figure 10.6. Now, the normal form theorem is applied to transform the nonlinear controlled sources to the input. It is known from Section 10.2 that all nonresonant parts of $\tilde{\mathbf{f}}(\mathbf{x})$ can be eliminated until a prescribed order k if the associated homological equation is satisfied. Therefore, after a nearly identity transformation (Equation 10.15) the normal form of the nonresonant part can be described by (see Figure 10.6)

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}, \quad \mathbf{x} = \mathbf{y} + \mathbf{h}(\mathbf{y}). \tag{10.81}$$

For this decomposition we have to define the nonresonant and resonant terms of the vector field where the eigenvalues of the linear part **A** of **f** and the degree of the polynomial nonlinearities must be studied.

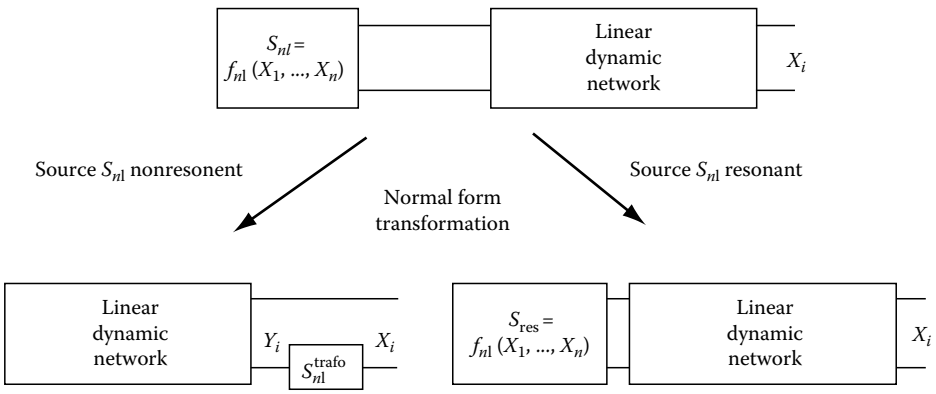


FIGURE 10.6 Decomposition of nonlinear dynamic circuits.

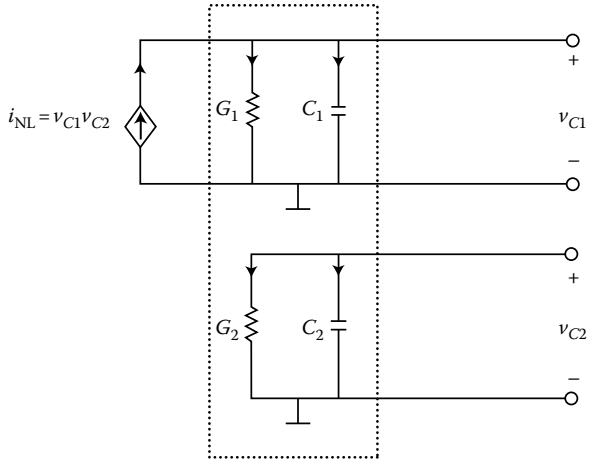


FIGURE 10.7 Decomposition of a simple nonlinear dynamic circuits.

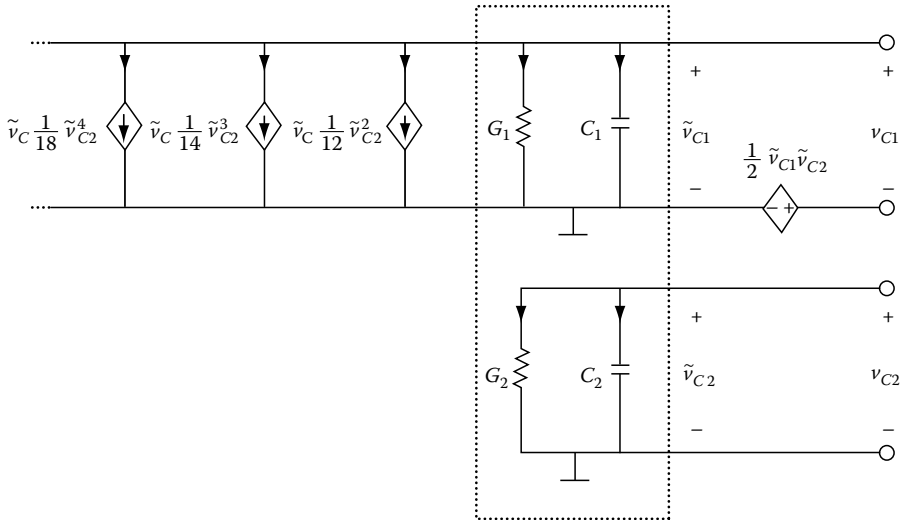


FIGURE 10.8 Decomposition

Under certain conditions, a finite recursive process exists, such that all nonlinear controlled sources can be transformed to the input of the linear part of the circuit under consideration. In these cases, the circuits are described by Equation 10.83 and the corresponding block diagram is shown on the left-hand side in Figure 10.6. If resonant terms occur, a number of additional sources are generated by means of a recursive process. In these cases the controlled cannot transforme to the input what is shown on the right-hand side of Figure 10.6. In order to illustrate these statements, a simple example is presented in Figure 10.7. After a nearly identity transformation, a reduction of the nonresonant terms, and the recursive process with respect to the resonant terms, the decomposed circuit is shown in Figure 10.8. Therefore, this application of normal form theorems in circuit analysis can be interpreted as a kind of extraction of nonlinear controlled sources from a nonlinear dynamic circuit. Finally it should be mentioned that this decomposition based on the normal form theorem is related in a certain sense to the so-called exact linearization that is studied in the theory of nonlinear control systems; see, e.g., Refs. [36,54].

References

1. E. J. Altman Bifurcations analysis of Chua's circuit with application for low-level visual sensing, in: R. N. Madan (Ed.), *Chua's Circuit: A Paradigm for Chaos*, Singapore: World Scientific, 1993.
2. V. I. Arnold, *Geometrical Methods in the Theory of Ordinary Differential Equations*, Berlin-New York: Springer-Verlag, 1983.
3. D. K. Arrowsmith and C. M. Place, *An introduction to Dynamical Systems*, Cambridge: Cambridge University Press, 1993.
4. M. Ashkenazi and S.-N. Chow, Normal forms near critical points for differential equations and maps, *IEEE Trans. Circuits Syst.,* 35, 850–862, 1988.
5. S. Boyd and L. O. Chua, Uniqueness of a basic nonlinear structure, *IEEE Trans. Circuits Syst.,* 30, 648–651, 1983.
6. S. Boyd and L. O. Chua, Uniqueness of circuits and systems containing one nonlinearity, *IEEE Trans. Autom. Control,* 30, 674–681, 1985.
7. G. W. Bluman and S. Kumei, *Symmetries and Differential Equations*, New York: Springer-Verlag, 1989.
8. R. K. Brayton, Nonlinear reciprocal networks, in: H. S. Wilf and F. Harary (Eds.), *Mathematical Aspects of Electrical Network Analysis*, Providence, RI: AMS, 1971.
9. R. K. Brayton and J. K. Moser, Nonlinear networks, I, II, *Quart. Appl. Math.,* 23, 1–33, 81–104, 1964.
10. A. D. Bruno, *Local Methods in Nonlinear Differential Equations*, Berlin-New York: Springer-Verlag, 1981.
11. J. Carr, *Applications of Center Manifold Theorem*, Berlin-New York: Springer-Verlag, 1981.
12. J. Cervera, A. J. van der Schaft, and A. Banos, Interconnection of port-Hamiltonian systems and composition of Dirac structures, *Automatica,* 43, 212–225, 2007.
13. L. O. Chua, The rotator—A new network element, *Proc. IEEE,* 55, 1566–1577, 1967.
14. L. O. Chua, Synthesis of new nonlinear network elements, *Proc. IEEE,* 56, 1325–1340, 1968.
15. L. O. Chua, Modelling of three terminals devices: A black box approach, *IEEE Trans. Circuit Theory,* 19, 555–562, 1972.
16. L. O. Chua, Device modeling via basic nonlinear circuit elements, *IEEE Trans. Circuits Syst.,* 27, 1014–1044, 1980.
17. L. O. Chua, Dynamical nonlinear networks: State of the Art, *IEEE Trans. Circuits Syst.,* 27, 1059–1087, 1980.
18. L. O. Chua, Global unfolding of Chua's circuit, *IEEE Trans. Fundam. E76-A,* 704–734, 1993.
19. L. O. Chua and A. C. Deng, Canonical piecewise linear representation, *IEEE Trans. Circuits Syst.,* 33, 101–111, 1988.
20. L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*, New York: McGraw-Hill, 1987.
21. L. O. Chua and H. Kokukbo, Normal forms for nonlinear vector fields—Part I: Theory and algorithms, *IEEE Trans. Circuits Syst.,* 35, 863–880, 1988.
22. L. O. Chua and H. Kokukbo, Normal forms for nonlinear vector fields—Part II: Applications, *IEEE Trans. Circuits Syst.,* 36, 51–70, 1989.
23. L. O. Chua and Y.-F. Lam, A theory of algebraic *n*-ports, *IEEE Trans. Circuit Theory,* 20, 370–382, 1973.
24. L. O. Chua and H. Oka, Normal forms of constrained nonlinear differential equations—Part I: Theory and algorithm'', *IEEE Trans. Circuits Syst.,* 35, 881–901, 1988.
25. L. O. Chua and H. Oka, Normal forms of constrained nonlinear differential equations—Part II: Bifurcations, *IEEE Trans. Circuits Syst.,* 36, 71–88, 1989.
26. E. A. Desloge, Suppression and restoration of constants in physical equations, *Am J. Phys.,* 52, 312–316, 1984.
27. C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, New York: McGraw-Hill, 1969.

28. N. Fenichel, Geometric singular perturbation theory for ordinary differential equations, *J. Diff. Eq.,* 31, 5398, 1979.
29. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Berlin-New York: Springer-Verlag 1990.
30. S. Hale and H. Koċak, *Dynamics and Bifurcation*, Berlin-New York: Springer-Verlag, 1991.
31. B. D. Hassard, N. D. Kazarinoff, and Y.-H. Wan, *Theory and Applications of the Hopf Bifurcation*, Cambridge: Cambridge University Press, 1980.
32. B. D. Hassard and Y.-H. Wan, Bifurcation formulae derived from center manifold theorem, *J. Math. Anal. Appl.,* 63, 297–312, 1978.
33. J. Haase, On generalizations and applications of the subtitution theorems, *Proc. ECCTD'85*, Praha, Czech., Sept. 2–6, 1985. pp. 220–223.
34. W. P. M. H. Heemels, M. K. Camlibel, and J. M. H. Schumacher, On the dynamic analysis of piecewise-linear networks, *IEEE Trans. Circuits Syst. -I: Fundamental Theory and Applications*, 49, 315–327, 2002.
35. R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge: Cambridge University Press, 1992.
36. A. Isidori, *Nonlinear Control Systems*, Berlin-New York: Springer-Verlag, 1989.
37. M. Itoh, Synthesis of electronic circuits for simulating nonlinear dynamcis, *International Journal of Bifurcation and Chaos,* 11(3), 605–653, 2001.
38. V. G. Ivancevic and T. T. Ivancevic, *Geometric Dynamics of Complex Systems*, Dordrecht, The Netherlands: Springer-Verlag 2006.
39. J. Jess, Piecewise linear models for nonlinear dynamic systems, *Frequenz*, 42, 71–78, 1988.
40. D. H. Johnson, Scanning our past—Origins of the equivalent circuit concept: The current-source equivalent, *Proc. IEEE,* 91, 817–821, 2003.
41. T. A. M. Kevenaar and D. M. W. Leenaerts, A comparison of piecewise-linear model descriptions, *IEEE Trans. Circuits Syst. I,* 39, 996–1004, 1992.
42. R. N. Madan, Ed., *Chua's Circuit: A Paradigm for Chaos*, Singapore: World Scientific, 1993.
43. C. Keidies and W. Mathis, Applications of normal forms to the analysis of nonlinear circuits, *Proc. 1993 Int. Symp. Nonlinear Theory and Its Appl.* (NOLTA), Hawaii, pp. 853–858, Dec. 1993.
44. G. M. Maggio, O. De Feo, and M. P. Kennedy, A general method to predict the amplitude of oscillation in nearly-sinusoidal oscillators, *IEEE Trans. Circuits Sys. I,* 5(8), 1586–1595, Aug 2004.
45. J. Kevorkian and J. D. Cole, *Multiple Scale and Singular Perturbation Methods*, Berlin-New York: Springer-Verlag, 1996.
46. W. Marten, W. Mathis, and L. O. Chua, On the geometrical meaning of pseudo hybrid content and mixed potential, *Int. J. Electronics Communications (AEÜ),* 46 (4), 305–309, 1992.
47. C. Valentin, M. Magos, and B. Maschke, A port-Hamiltonian formulation of physical switching systems with varying constraints, *Automatica,* 43, 1125–1133, 2007.
48. W. Mathis, *Theory of Nonlinear Networks (in German)*, Berlin-Heidelberg: Springer-Verlag, 1987.
49. W. Mathis, Nonlinear electronic circuits—An overview, *Proc. MIXDES 2000*, Gdynia, Poland, 2000, p. 1517.
50. W. Mathis and L. O. Chua, Applications of dimensional analysis to network theory, *Proc. ECCTD'91*, Copenhagen Denmark, pp. 1243–1252, Sept. 4–6, 1991.
51. W. Mathis and R. Pauli, Network theorems, in: J. G. Webster (Ed.), *Encyclopedia of Electrical and Electronics Engineering*, New York: John Wiley& Sons, Inc., Vol. 14, 1999, pp. 227–240.
52. W. Millar, The nonlinear resistive 3-pole: Some general concepts, *Proc. Symp. Nonlinear Circuit Anal.*, Institute of Brooklyn, New York: Interscience, 1957.
53. W. L. Miranker, *Numerical Methods for Stiff Equations and Singular Perturbation Problems*, Dordrecht: D. Reidel, 1981.
54. H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems*, New York: Springer-Verlag, 1990.

55. L. V. Ovsiannikov, *Group Analysis of Differential Equations*, New York: Academic Press, 1982.
56. A. Reibiger, W. Mathis, T, Nähring, and Lj. Trajkovic, Mathematical foundations of the TC-method for computing multiple DC-operating points, *Int. J. Appl. Electromagn. Mech.,* 17, 169–191, 2003.
57. D. R. Smith, *Singular-Perturbation Theory*, Cambridge: Cambridge University Press, 1985.
58. P. P. Varaiya and J. P. Verma, Equivalence of nonlinear reciprocal networks, *IEEE Trans. Circuit Theory,* 18, 214–217, 1971.
59. J. P. Verma, Equivalence of nonlinear networks, Ph.D. dissertation, University of California, Berkeley, 1969.
60. L. Weiss, W. Mathis, and L. Trajkovič, A generalization of Brayton-Moser's mixed potential function, *IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications,* 45, 423–427, 1998.
61. A. N. Willson, Jr., *Nonlinear Networks: Theory and Analysis*, New York: IEEE, 1975.

11

Piecewise-Linear Circuits and Piecewise-Linear Analysis

11.1 Introduction and Motivation **11**-1
11.2 Hierarchy of Piecewise-Linear Models and Their Representations **11**-3
11.3 Piecewise-Linear Models for Electronic Components... **11**-8
11.4 Structural Properties of Piecewise-Linear Resistive Circuits **11**-14
11.5 Analysis of Piecewise-Linear Resistive Circuits **11**-15
Theorem Canonical PWL (Tableau Analysis) • Algorithm
11.6 Piecewise-Linear Dynamic Circuits **11**-18
11.7 Efficient Computer-Aided Analysis of PWL Circuits **11**-19
Acknowledgment **11**-20
References **11**-20

Joos Vandewalle
Catholic University of Leuven

Lieven Vandenberghe
University of California, Los Angeles

11.1 Introduction and Motivation

In this chapter, we present a comprehensive description of the use of piecewise-linear (PWL) methods in modeling, analysis, and structural properties of nonlinear circuits. The main advantages of piecewise-linear circuits are fourfold. (1) Piecewise-linear circuits are the easiest in the class of nonlinear circuits to analyze exactly, because many methods for linear circuits can still be used. (2) The piecewise-linear approximation is an adequate approximation for most applications. Moreover, certain operational amplifier (op-amp), operational transconductance amplifier (OTA), diode and switch circuits are essentially piecewise linear. (3) Quite a number of methods exist to analyze piecewise-linear circuits. (4) Last, but not least, piecewise-linear circuits exhibit most of the phenomena of nonlinear circuits while still being manageable. Hence, PWL circuits provide unique insight in nonlinear circuits.

This chapter is divided into six sections. First, the piecewise-linear models will be presented and interrelated. A complete hierarchy of models and representations of models is presented. Rather than proving many relations, simple examples are given. Second, the piecewise-linear models for several important electronic components are presented. Third, since many PWL properties are preserved by interconnection, a short discussion on the structural properties of piecewise-linear circuits is given in Section 11.4. Fourth, analysis methods of PWL circuits are presented, ranging from the Katzenelson algorithm to the linear complementarity methods and the homotopy methods. Fifth, we discuss PWL dynamic circuits, such as the famous Chua circuit, which produces chaos. Finally, in Section 11.7, efficient

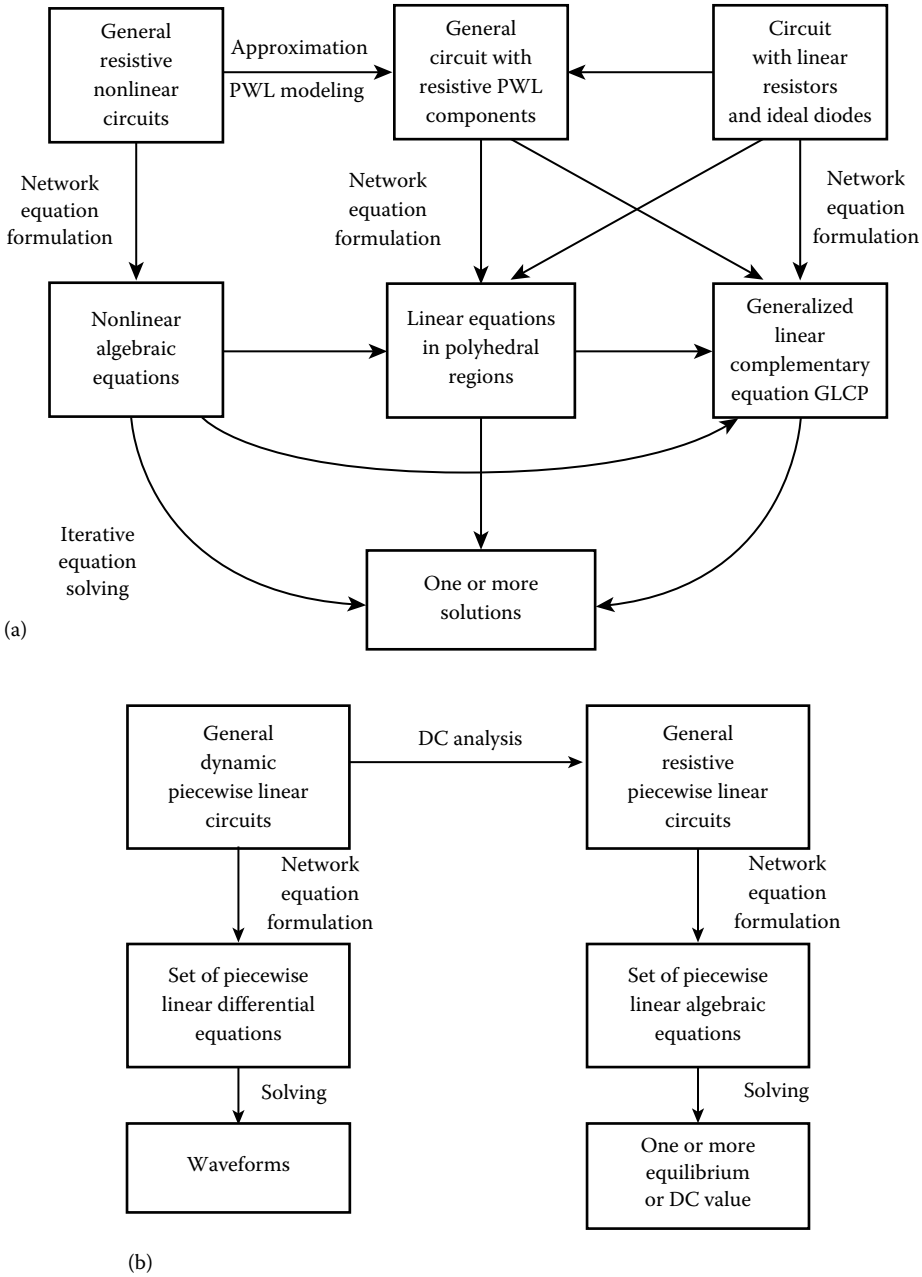


FIGURE 11.1 Interrelation of PWL circuit analysis methods: (a) resistive and (b) dynamic nonlinear circuits.

computer-aided analysis of PWL circuits and the hierarchical mixed-mode PWL analysis are described. A comprehensive reference list is included. For the synthesis of PWL circuits, we refer to Chapter 8.

In order to situate these subjects in the general framework of nonlinear circuits, it is instructive to interrelate the PWL circuit analysis methods (Figure 11.1). In the horizontal direction of the diagrams, one does the PWL approximation of the dc analysis from left to right. In the vertical direction, we show the conversion from a circuit to a set of equations by network equation formulation and the conversion

from equations to solutions (waveforms or dc values) by solution methods. The specific methods and names used in the figure are described in detail in the different parts.

11.2 Hierarchy of Piecewise-Linear Models and Their Representations

In the past 25 years, much progress has been achieved in the representations of piecewise-linear resistive multiports and their relationships (see Refs. [1,2]). From a practical point of view, a clear trade-off exists between the efficiency of a representation in terms of the number of parameters and the ease of evaluation (explicit versus implicit models) on the one hand and the generality or accuracy on the other hand. Here, we go from the easiest and most efficient to the most general representations.

We define here a resistive multiport (Figure 11.2) as an n-port whose port variables (the vector of port currents $i = [i_1, \ldots, i_n]^T$ and the vector of port voltages $v = [v_1, \ldots, v_n]^T$) are related by m algebraic equations called constitutive equations

$$\varphi(i, v) = 0 \tag{11.1}$$

where $i, v \in \mathbb{R}^n$ and $\phi(.,.)$ maps $\mathbb{R}^{2n}$ into $\mathbb{R}^m$.

For example, for a bipolar transistor (Figure 11.3), one obtains the explicit form $i_1 = f_1(v_1, v_2)$ and $i_2 = f_2(v_1, v_2)$, and $i = [i_1, i_2]^T$ and $v = [v_1, v_2]^T$. These relations can be measured with a curve tracer as dc characteristic curves. Clearly, here $\phi(.,.)$ is a map from $\mathbb{R}^4 \rightarrow \mathbb{R}^2$ in the form

$$i_1 - f_1(v_1, v_2) = 0 \tag{11.2}$$

$$i_2 - f_2(v_1, v_2) = 0 \tag{11.3}$$

It is easy to see that a complete table of these relationships would require an excessive amount of computer storage already for a transistor. Hence, it is quite natural to describe a resistive n-port with a piecewise-linear map f over polyhedral regions P_k by

$$v = f(i) = a_k + B_k i, \quad i \in P_k, \quad k \in \{0, 1, \ldots, 2^l - 1\} \tag{11.4}$$

where the Jacobian $B_k \in \mathbb{R}^{n \times n}$ and the offset vector $a_k \in \mathbb{R}^n$ are defined over the polyhedral region P_k, separated by hyperplanes $c_i^T x - d_i = 0$, $i = 1, \ldots, l$ and defined by

$$P_k = \left\{ x \in \mathbb{R}^n | c_j^T x - d_j \geq 0, j \in I_k, c_j^T x - d_j \leq 0, j \notin I_k \right\} \tag{11.5}$$

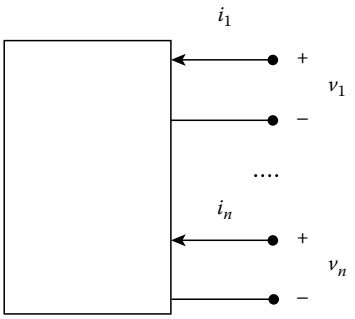


FIGURE 11.2 Resistive n-port.

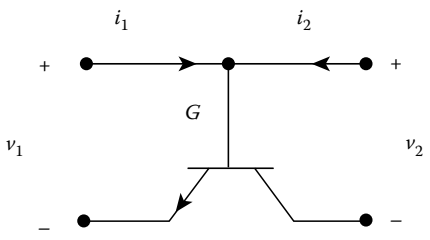


FIGURE 11.3 Two-port configuration of a bipolar transistor.

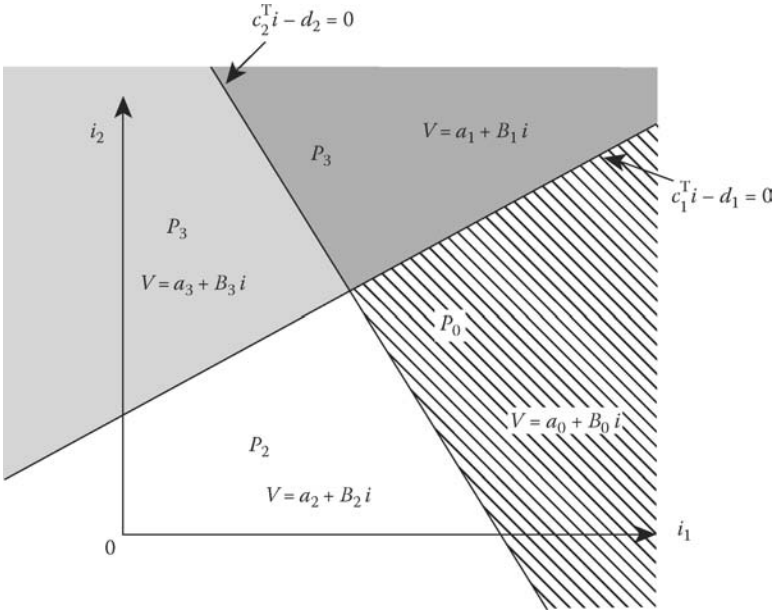


FIGURE 11.4 PWL function defined in four polyhedral regions in $\mathbb{R}^n$ defined by $c_1^T i - d_1 \lesseqgtr 0$ and $c_2^T i \gtreqless d_2 > 0$.

where

$k = \Sigma_{j \in Ik}\, 2^{j-1}$
$I_k \subseteq \{1, 2, \ldots, l\}$
$c_j \in \mathbb{R}^n$, $d_j \in \mathbb{R}^n$

In other words, the hyperplanes $c_i^T x - d_i = 0$, $i = 1, \ldots, l$ separate the space $\mathbb{R}^n$ into 2^l polyhedral regions P_k (see Figure 11.4) where the constitutive equations are linear.

The computer storage requirements for this representation is still quite large, especially for large multiports. A more fundamental problem with this rather intuitive representation is that it is not necessarily continuous at the boundaries between two polyhedral regions. In fact, the continuity of the nonlinear map is usually desirable for physical reasons and also in order to avoid problems in the analysis.

The canonical PWL representation [6] is a very simple, attractive, and explicit description for a resistive multiport that solves both problems:

$$v = f(i) = a + Bi + \sum_{j=1}^{l} e_j \left| c_j^T i - d_j \right| \qquad (11.6)$$

One can easily understand this equation by looking at the wedge form of the modulus map (see Figure 11.5). It has two linear regions: in the first $x \geq 0$ and $y = x$, while in the second $x \leq 0$ and $y = -x$. At the boundary the function is clearly continuous. Equation 11.6 is hence also continuous and is linear in each of the polyhedral regions P_k described by Equation 11.5. If l modulus terms are in Equation 11.6, there are 2^l polyhedral regions where the map Equation 11.6

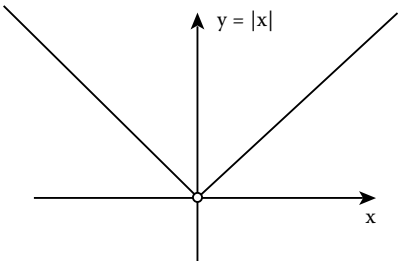


FIGURE 11.5 Absolute value function $y = |x|$.

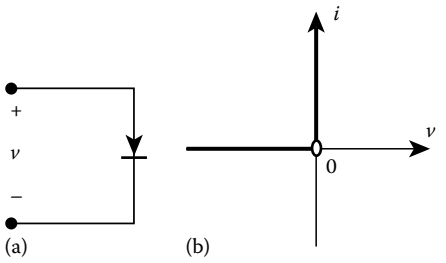


FIGURE 11.6 (a) The ideal diode and (b) the $(i-v)$ relation of an ideal diode.

is linear. Because the map is represented canonically with $n + n^2 + l(n+1)$ real parameters, this is a very compact and explicit representation.

Several examples of canonical PWL models for components are given in Section 11.3.

From Figure 11.5, it should be clear that the right and left derivative of $y = |x|$ at 0 are different, their difference being 2. Hence, the Jacobian J_+ and J_- of Equation 11.6 will be different on the boundary between the two neighboring polyhedral regions where $(c_j i - d_j) \geq 0$ and $(c_j i - d_j) \leq 0$.

$$J_+ - J_- = 2e_j c_j^{\mathrm{T}} \tag{11.7}$$

Observe that this difference is a rank 1 matrix, which is also called a dyadic or outer vector product of e_j and c_j. Moreover, this difference is independent of the location of the independent variable i on the boundary. This important observation is made in Ref. [24], and is called the consistent variation property [10] and essentially says that the variation of the Jacobian of a canonical PWL representation is independent of the place where the hyperplane $c_j i - d_j = 0$ is crossed. Of course, this implies that the canonical PWL representation (Equation 11.6) is not the most general description for a continuous explicit PWL map. In Refs. [26,29] two more general representations, which include nested absolute values, are presented. These are too complicated for our discussion.

Clearly, the canonical PWL representation (Equation 11.6) is valid only for single-valued functions. It can clearly not be used for an important component: the ideal diode (Figure 11.6) characterized by the multivalued (i, v) relation. It can be presented analytically by introducing a real scalar parameter ρ [31].

$$i = \frac{1}{2}(\rho + |\rho|) \tag{11.8}$$

$$v = \frac{1}{2}(\rho - |\rho|) \tag{11.9}$$

This parametric description can easily be seen to correspond to Figure 11.6b because $i = \rho$ and $v = 0$ for $\rho \geq 0$, while $i = 0$ and $v = \rho$ when $\rho \leq 0$. Such a parametric description $i = f(\rho)$ and $v = g(\rho)$ with f and g PWL can be obtained for a whole class of unicursal curves [6].

When we allow implicit representations between v and i for a multiport, we obtain a linear complementarity problem (LCP) model Equations 5.10 through 5.12 with an interesting state space like form [55]:

$$v = Ai + Bu + f \tag{11.10}$$

$$s = Ci + Du + g \tag{11.11}$$

$$u \geq 0, \quad s \geq 0, \quad u^{\mathrm{T}} s = 0 \tag{11.12}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times l}$, $f \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^{l \times n}$, $D \in \mathbb{R}^{l \times l}$ are the parameters that characterize the relationship between v and i. In the model, u and s are called the state vectors and we say that $u \geq 0$ when all its components are nonnegative. Clearly, Equation 11.12 dictates that all components of u and s should be nonnegative and that, whenever a component u_j satisfies $u_j > 0$, then $s_j = 0$ and, vice versa, when $s_j > 0$, then $u_j = 0$. This is called the linear complementarity property, which we have seen already in the ideal diode equation (Equations 11.8 and 11.9) where $i \geq 0$, $v \geq 0$ and $iv = 0$. Hence, an implicit or LCP model for the ideal diode equation (Equations 11.8 and 11.9) is

$$v = u \tag{11.13}$$

$$s = i \tag{11.14}$$

$$u \geq 0, \quad s \geq 0, \quad us = 0 \tag{11.15}$$

In order to understand that the general equations (Equations 11.10 through 11.12) describe a PWL relation such as Equations 11.4 and 11.5 between i and v over polyhedral regions, one should observe first that $v = Ai + f$ is linear when $u = 0$ and $s = Ci + g \geq 0$. Hence, the relation is linear in the polyhedral region determined by $Ci + g \geq 0$. In general, one can consider 2^l possibilities for u and s according to

$$(u_j \geq 0 \text{ and } s_j = 0) \text{ or } (u_j = 0 \text{ and } s_j = 0) \quad \text{for } j = 1, 2, \ldots, l$$

Denote sets of indexes U and S for certain values of u and s satisfying Equation 11.12

$$U = \{j | u_j \geq 0 \text{ and } s_j = 0\} \tag{11.16}$$

$$S = \{j | u_j = 0 \text{ and } s_j \geq 0\} \tag{11.17}$$

then, clearly, U and S are complementary subsets of $\{1, 2, \ldots, l\}$ when for any j, u_j, and s_j cannot be both zero. Clearly, each of these 2^l possibilities corresponds to a polyhedral region P_U in $\mathbb{R}^n$, which can be determined from

$$u_j \geq 0 \quad (Ci + Du + g)_j = 0 \quad \text{for } j \in U \tag{11.18}$$

$$u_j = 0 \quad (Ci + Du + g)_j \geq 0 \quad \text{for } j \in S \tag{11.19}$$

The PWL map in region P_U is determined by solving the u_j for $j \in U$ from Equation 11.18 and substituting these along with $u_j = 0$ for $j \in S$ into Equation 11.10. This generates, of course, a map that is linear in the region P_U.

When Equation 11.11 is replaced by the implicit equation

$$Es + Ci + Du + g\alpha = 0, \quad \alpha \geq 0$$

in Equations 11.10 through 11.13, we call the problem a generalized linear complementarity problem (GLCP).

A nontrivial example of an implicit PWL relation (LCP model) is the hysteresis one port resistor (see Figure 11.7). Its equations are

$$v = -i + [-\mathbf{1}\ \mathbf{1}] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + 1 \tag{11.20}$$

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} i + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{11.21}$$

$$s_1 \geq 0, \quad s_2 \geq 0, \quad u_1 \geq 0, \quad u_2 \geq 0, \quad u_1 s_1 + u_2 s_2 = 0 \tag{11.22}$$

In the first region P, we have

$$s_1 = -i + 1 \geq 0, \quad s_2 = i \geq 0, \quad \text{and} \quad v = -i + 1 \tag{11.23}$$

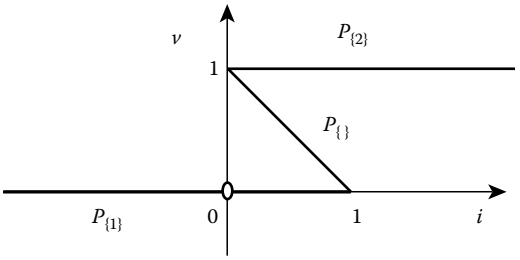


FIGURE 11.7 Hysteresis nonlinear resistor.

The region $P_{\{1,2\}}$, on the other hand, is empty because the following set of equations are contradictory:

$$\begin{gathered} s_1 = s_2 = 0, \quad -i - u_1 + u_2 + 1 = 0, \\ i + u_1 - u_2 = 0 \end{gathered} \tag{11.24}$$

The region $P_{[1]}$ is

$$u_1 \geq 0, \quad s_1 = -i - u_1 + 1 = 0, \quad u_2 = 0, \quad s_2 = i + u_1 \geq 0 \tag{11.25}$$

Hence, $u_1 = -i + 1$ and $s_2 = 1$ and $v = -i + i - 1 + 1 = 0$, while $i \leq 1$.

Finally, the region $P_{[2]}$ is

$$u_1 \geq 0, \quad s_1 = -i + u_2 + 1 \geq 0, \quad u_2 \geq 0, \quad s_2 = i - u_2 = 0$$

Hence

$$u_2 = i \quad \text{and} \quad s_1 = 1 \quad \text{and} \quad v = -i + i + 1 = 1, \quad \text{while } i \geq 0 \tag{11.26}$$

It is now easy to show in general that the canonical PWL representation is a special case of the LCP model. Just choose $u_j \geq 0$ and $s_j \geq 0$ for all j as follows:

$$\left| c_j^{\mathrm{T}} i - d_j \right| = \frac{1}{2}(u_j + s_j) \tag{11.27}$$

$$c_j^{\mathrm{T}} i - d_j = \frac{1}{2}(u_j - s_j) \tag{11.28}$$

then, u and s are complementary vectors, i.e.,

$$u \geq 0, \quad s \geq 0, \quad u^{\mathrm{T}} s = 0$$

Observe that the moduli in Equation 11.6 can be eliminated with Equation 11.27 to produce an equation of the form (Equation 11.10) and that (Equation 11.28) produces an equation of the form (Equation 11.11).

More generally, it has been proven [36] that the implicit model includes all explicit models. Because it also includes the parametric models, one obtains the general hierarchy of models as depicted in Figure 11.8.

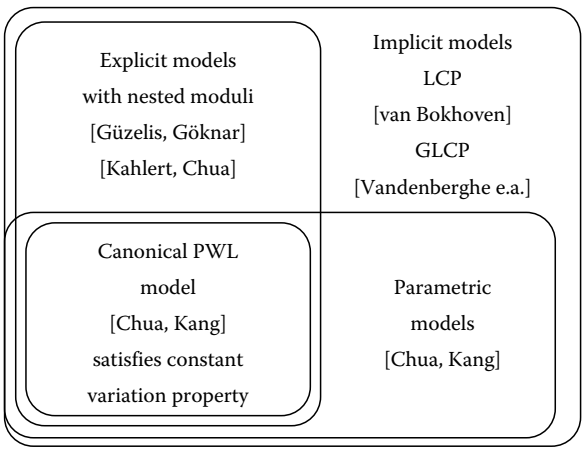


FIGURE 11.8 Interrelation of the PWL models.

A general remark should be made about all models that have been presented until now. Although the models have been given for resistive multiports where the voltages v at the ports are expressed in terms of the currents i, analogous equations can be given for the currents i in terms of the voltages, or hybrid variables. It can even be adapted for piecewise-linear capacitors, inductors, or memristors, where the variables are, respectively, q, v for capacitors, φ, i for inductors, and q, φ for memristors.

11.3 Piecewise-Linear Models for Electronic Components

In order to simulate nonlinear networks with a circuit or network simulator, the nonlinear behavior of the components must be modeled fist. During this modeling phase, properties of the component that are not considered important for the behavior of the system may be neglected. The nonlinear behavior is often important, therefore, nonlinear models have to be used. In typical simulators such as SPICE, nonlinear models often involve polynomials and transcendental functions for bipolar and MOS transistors. These consume a large part of the simulation time, so table lookup methods have been worked out. However, the table lookup methods need much storage for an accurate description of multiports and complex components.

The piecewise-linear models constitute an attractive alternative that is both efficient in memory use and in computation time. We discuss here the most important components. The derivation of a model usually requires two steps: first, the PWL approximation of constitutive equations, and second, the algebraic representation.

Two PWL models for an ideal diode (Figure 11.6) have been derived, that is, a parametric model (Equations 11.8 and 11.9) and an implicit model (Equations 11.13 through 11.15, while a canonical PWL model does not exist.

The piecewise-linear models for op-amps and OTAs are also simple and frequently used. The piecewise-line approximation of op-amps and OTAs of Figure 11.9 is quite accurate. It leads to the

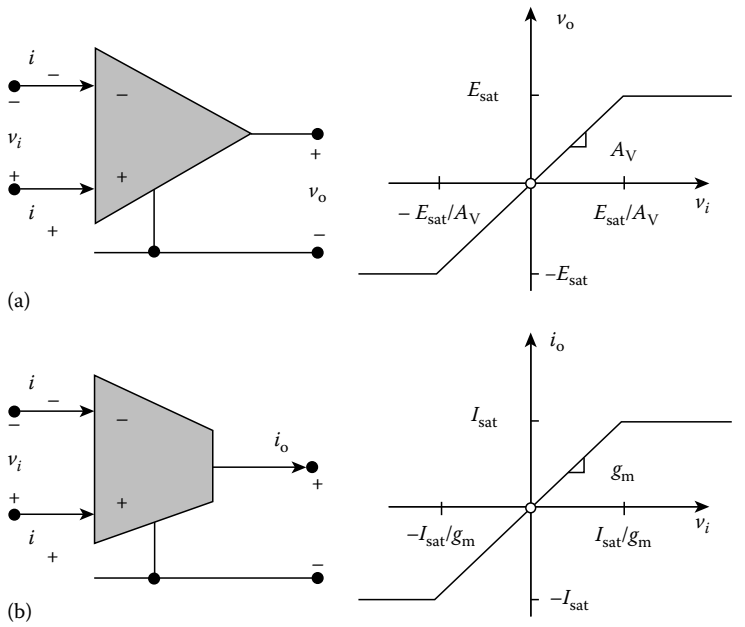


FIGURE 11.9 (a) Op-amp and PWL model and (b) OTA and PWL model.

following representation for the op-amp, which is in the linear region for $-E_{\text{sat}} \le v_0 \le E_{\text{sat}}$ with voltage amplification A_v and positive and negative saturation E_{sat} and $-E_{\text{sat}}$

$$v_0 = \frac{A_v}{2}\left(\left|v_i + \frac{E_{\text{sat}}}{A_v}\right| - \left|v_i - \frac{E_{\text{sat}}}{A_v}\right|\right) \tag{11.29}$$

$$i_- = i_+ = 0 \tag{11.30}$$

This is called the op-amp finite-gain model. In each of the three regions, the op-amp can be replaced by a linear circuit.

For the OTA, we have similarly in the linear region for $-I_{\text{sat}} \le i_0 \le I_{\text{sat}}$ with transconductance gain g_{m} and positive and negative saturation I_{sat} and $-I_{\text{sat}}$

$$i_0 = \frac{g_{\text{m}}}{2}\left(\left|v_i + \frac{I_{\text{sat}}}{g_{\text{m}}}\right| - \left|v_i - \frac{I_{\text{sat}}}{g_{\text{m}}}\right|\right) \tag{11.31}$$

$$i_- = i_+ = 0 \tag{11.32}$$

Next, for a tunnel diode, one can perform a piecewise-linear approximation for the tunnel-diode characteristic as shown in Figure 11.10. It clearly has three regions with conductances g_1, g_2, and g_3. This PWL characteristic can be realized by three components (Figure 11.10b) with conductances, voltage sources, and diodes. The three parameters G_0, G_1, and G_2 of Figure 11.10b must satisfy

$$\text{In Region 1: } G_0 = g_1 \tag{11.33}$$

$$\text{In Region 2: } G_0 + G_1 = g_2 \tag{11.34}$$

$$\text{In Region 3: } G_0 + G_1 + G_2 = g_3 \tag{11.35}$$

Thus, $G_0 = g_1$, $G_1 = -g_1 + g_2$, and $G_2 = -g_2 + g_3$. We can derive the canonical PWL representation as follows:

$$i = -\frac{1}{2}(G_1E_1 + G_2E_2) + \left(G_0 + \frac{1}{2}G_1 + \frac{1}{2}G_2\right)v + \frac{1}{2}G_1|v - E_1| + \frac{1}{2}G_2|v - E_2| \tag{11.36}$$

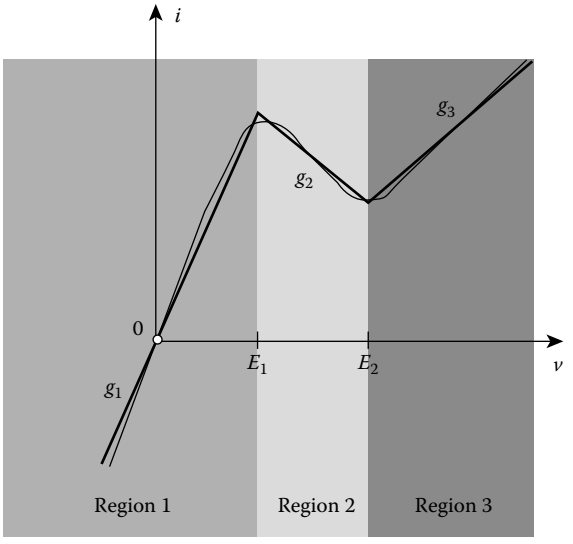
Next, we present a canonical piecewise-linear bipolar transistor model [12]. Assume a *npn*-bipolar transistor is connected in the common base configuration with $v_1 = v_{BE}$, $v_2 = v_{BC}$, $i_1 = i_E$, and $i_2 = i_C$, as shown in Figure 11.3. We consider data points in a square region defined by $0.4 \le v_1 \le 0.7$ and $0.4 \le v_2 \le 0.7$, and assume the terminal behavior of the transistor follows the Ebers–Moll equation; namely,

$$i_1 = \frac{I_{\text{s}}}{\alpha_{\text{f}}}\left(e^{v_1/V_{\text{T}}} - 1\right) - I_{\text{s}}\left(e^{v_2/V_{\text{T}}} - 1\right) \tag{11.37}$$

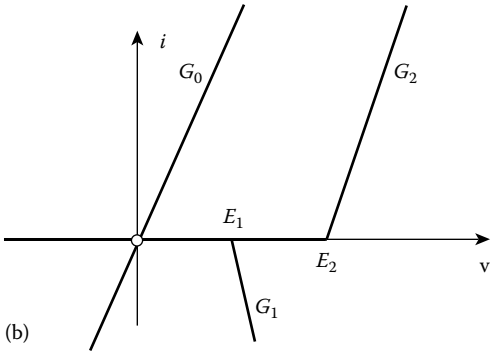
$$i_2 = \frac{I_{\text{s}}}{\alpha_{\text{r}}}\left(e^{v_2/V_{\text{T}}} - 1\right) - I_{\text{s}}\left(e^{v_1/V_{\text{T}}} - 1\right) \tag{11.38}$$

with $I_{\text{s}} = 10^{-14}$ A, $V_{\text{T}} = 26$ mV, $\alpha_{\text{f}} = 0.99$, and $\alpha_{\text{r}} = 0.5$. In Ref. [12], the following canonical piecewise-linear model is obtained, which optimally fits the data points (Figure 11.11)

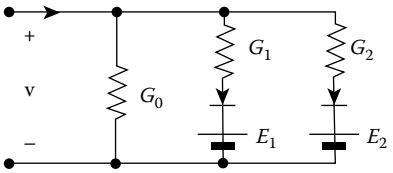
$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix}|m_1v_1 - v_2 + t_1| + \begin{bmatrix} c_{12} \\ c_{22} \end{bmatrix}|m_2v_1 - v_2 + t_2| + \begin{bmatrix} c_{13} \\ c_{23} \end{bmatrix}|m_3v_1 - v_2 + t_3| \tag{11.39}$$



(a)



(b)



(c)

FIGURE 11.10 (a) Piecewise-linear approximation of the tunnel-diode characteristic. The three-segment approximation defines the three regions indicated. (b) Decomposition of the piecewise-linear characteristic (a) into three components, and (c) the corresponding circuit.

where

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 5.8722 \times 10^{-3} \\ -3.2652 \times 10^{-2} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} = \begin{bmatrix} 3.2392 \times 10^{-2} \\ -3.2067 \times 10^{-2} \end{bmatrix}$$

$$\begin{bmatrix} b_{12} \\ b_{22} \end{bmatrix} = \begin{bmatrix} -4.0897 \times 10^{-2} \\ 8.1793 \times 10^{-2} \end{bmatrix} \begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix} = \begin{bmatrix} 3.1095 \times 10^{-6} \\ -3.0784 \times 10^{-6} \end{bmatrix}$$

$$\begin{bmatrix} c_{12} \\ c_{22} \end{bmatrix} = \begin{bmatrix} -9.9342 \times 10^{-3} \\ 1.9868 \times 10^{-2} \end{bmatrix} \begin{bmatrix} c_{13} \\ c_{23} \end{bmatrix} = \begin{bmatrix} -3.0471 \times 10^{-2} \\ 6.0943 \times 10^{-2} \end{bmatrix}$$

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = \begin{bmatrix} 1.002 \times 10^{4} \\ -1.4 \times 10^{-4} \\ 1.574 \times 10^{-6} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} -6472 \\ 0.61714 \\ 0.66355 \end{bmatrix}$$

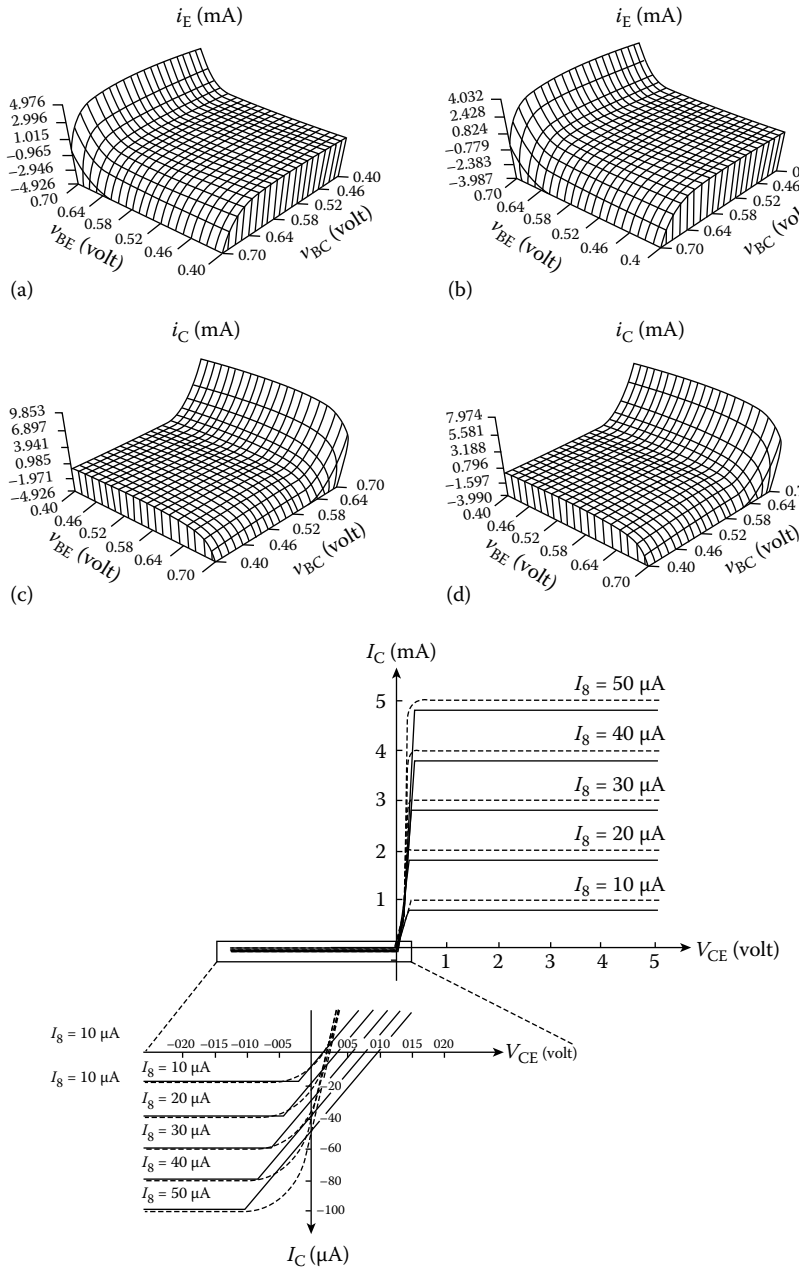


FIGURE 11.11 Three-dimensional plots for the emitter current in the Ebers–Moll model given by Equations 11.37 and 11.38. (b) Three-dimensional plot for the emitter current in the canonical piecewise-linear model given by Ref. [10, (B.1)] (low-voltage version). (c) Three-dimensional plot for the collector current in the Ebers–Moll model given by Equations 11.37 and 11.38. (d) Three-dimensional plot for the collector current in the canonical piecewise-linear model given by Ref. [10, (B.1)] (low-voltage version). (e) Comparison between the family of collector currents in the Ebers–Moll model (dashed line) and the canonical piecewise-linear model (solid line). (From Chua, L.O. and Deng, A., *IEEE Trans. Circuits Syst*., CAS-33, 519, 1986. With permission.)

Next, a canonical piecewise-linear MOS transistor model is presented. Assume the MOS transistor is connected in the common source configuration with $v_1 = v_{GS}$, $v_2 = v_{DS}$, $i_1 = i_G$, and $i_2 = i_D$, as illustrated, in Figure 11.12, where both v_1, v_2 are in volts, and i_1, i_2 are in microamperes. The data points are uniformly spaced in a grid within a rectangular region defined by $0 \le v_1 \le 5$, and $0 \le v_2 \le 5$. We assume the data points follow the Shichman–Hodges model, namely,

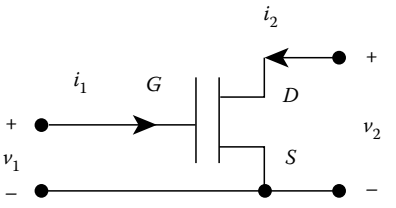


FIGURE 11.12 Two-port configuration of the MOSFET.

$$
\begin{aligned}
i_1 &= 0 \\
i_2 &= k\left[(v_1 - V_t)v_2 - 0.5v_2^2\right], \quad \text{if } v_1 - V_t \ge v_2
\end{aligned}
$$

or

$$
i_2 = 0.5k(v_1 - V_t)^2[1 + \lambda(v_2 - v_1 + V_t)], \quad \text{if } v_1 - V_t < v_2 \tag{11.40}
$$

with $k = 50\ \mu\text{A/V}^2$, $V_t = 1$ V, $\lambda = 0.02\ \text{V}^{-1}$. Applying the optimization algorithm of Ref. [11], we obtain the following canonical piecewise-linear model (see Figure 11.13):

$$
\begin{aligned}
i_2 = {} & a_2 + b_{21}v_1 + b_{22}v_2 + c_{21}|m_1v_1 - v_2 + t_1| \\
& + c_{22}|m_2v_1 - v_2 + t_2| + c_{23}|m_3v_1 - v_2 + t_3|
\end{aligned} \tag{11.41}
$$

where

$$
\begin{aligned}
a_2 &= -61.167, \quad b_{21} = 30.242, \quad b_{22} = 72.7925 \\
c_{21} &= -49.718, \quad c_{22} = -21.027, \quad c_{23} = 2.0348 \\
m_1 &= 0.8175, \quad m_2 = 1.0171, \quad m_3 = -23.406 \\
t_1 &= -2.1052, \quad t_2 = -1.4652, \quad t_3 = 69
\end{aligned}
$$

Finally, a canonical piecewise-linear model of GaAs FET is presented. The GaAs FET has become increasingly important in the development of microwave circuits and high-speed digital IC's due to its fast switching speed.

$$
\begin{aligned}
i_2 = {} & a_2 + b_{21}v_1 + b_{22}v_2 + c_{21}|m_1v_1 - v_2 + t_2| \\
& + c_{22}|m_2v_1 - v_2 + t_2| + c_{23}|m_3v_1 - v_2 + t_3|
\end{aligned} \tag{11.42}
$$

where $v_1 = v_{GS}$ (V), $v_2 = v_{DS}$ (V), $i_2 - i_D$ (mA), and

$$
\begin{aligned}
a_2 &= 6.3645, \quad b_{21} = 2.4961, \quad b_{22} = 32.339 \\
c_{21} &= 0.6008, \quad c_{22} = 0.9819, \quad c_{23} = -29.507 \\
m_1 &= -19.594, \quad m_2 = -6.0736, \quad m_3 = 0.6473 \\
t_1 &= -44.551, \quad t_2 = -8.9962, \quad t_3 = 1.3738
\end{aligned}
$$

Observe that this model requires only three absolute-value functions and 12 numerical coefficients and compares rather well to the analytical model (Figure 11.14).

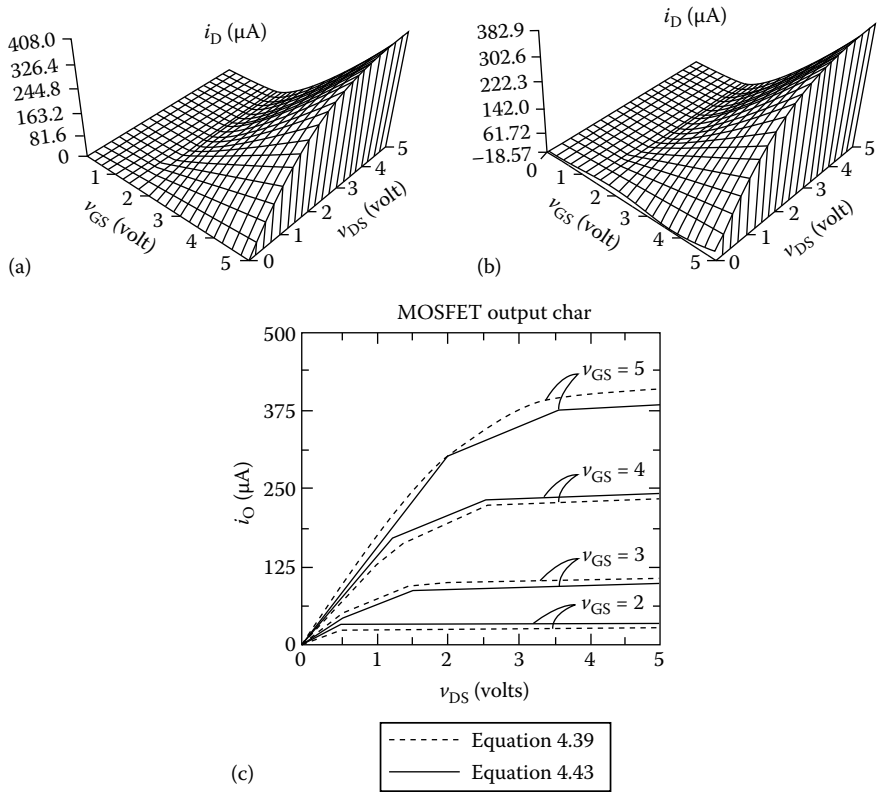


FIGURE 11.13 (a) Three-dimensional plot of drain current from the Shichman–Hodges model. (b) Three-dimensional plot of the drain current from the canonical piecewise-linear model. (c) Family of drain currents modeled by Equations 11.40 (dashed line) and 11.41 (solid line). (From Chua, L.O. and Deng, A., *IEEE Trans. Circuits Syst.*, CAS-33, 520, 1986. With permission.)

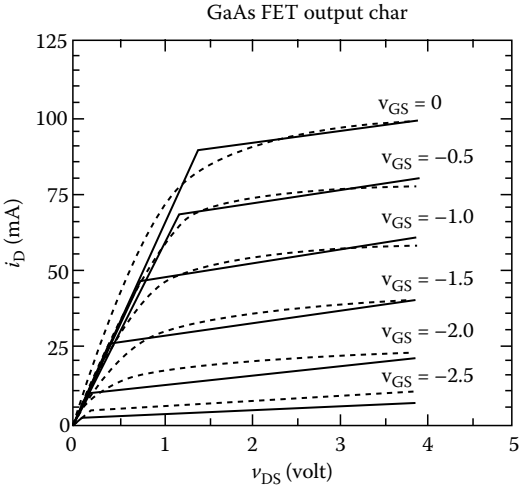


FIGURE 11.14 Comparison of the canonical piecewise linear described by Equation 11.42 (solid line) and the analytical model (dashed line) for the ion-implanted GaAs FET. (From Chua, L.O. and Deng, A., *IEEE Trans. Circuits Syst.*, CAS-33, 522, 1986. With permission.)

More piecewise-linear models for timing analysis of logic circuits can be found in Ref. [21]. In the context of analog computer design, even PWL models of other nonlinear relationships have been derived in Ref. [51].

11.4 Structural Properties of Piecewise-Linear Resistive Circuits

When considering interconnections of PWL resistors (components), it follows from the linearity of KVL and KCL that the resulting multiport is also a piecewise-linear resistor. However, if the components have a canonical PWL representation, the resulting multiport may not have a canonical PWL representation. This can be illustrated by graphically deriving the equivalent one port of the series connection of two tunnel diodes [3] (Figure 11.15). Both resistors have the same current, so we have to add the corresponding voltages $v = v_1 + v_2$ and obtain an $i-v$ plot with two unconnected parts. Values of i correspond to three values of v_1 for R_1 and three values of v_2 for R_2, and hence to nine values of the equivalent resistor (Figure 11.15d). This illustrates once more that nonlinear circuits may have more solutions than expected at first sight. Although the two tunnel diodes R_1 and R_2 have a canonical PWL representation, the equivalent one port of their series connection has neither a canonical PWL voltage description, nor a current one. It, however, has a GLCP description because KVL, KCL, and the LCP of R_1 and R_2 constitute a GLCP. If the v–i PWL relation is monotonic, the inverse i–v function exists and then some uniqueness properties hold.

These observations are, of course, also valid for the parallel connection of two PWL resistors and for more complicated interconnections.

In Section 11.3, we illustrated with an example how a PWL one-port resistor can be realized with linear resistors and ideal diodes. This can be proven in general. One essentially needs a diode for each breakpoint in the PWL characteristic. Conversely, each one port with diodes and resistors is a PWL one port resistor.

This brings us to an interesting class of circuits composed of linear resistors, independent sources, linear controlled sources, and ideal diodes. These circuits belong to the general class of circuits with PWL

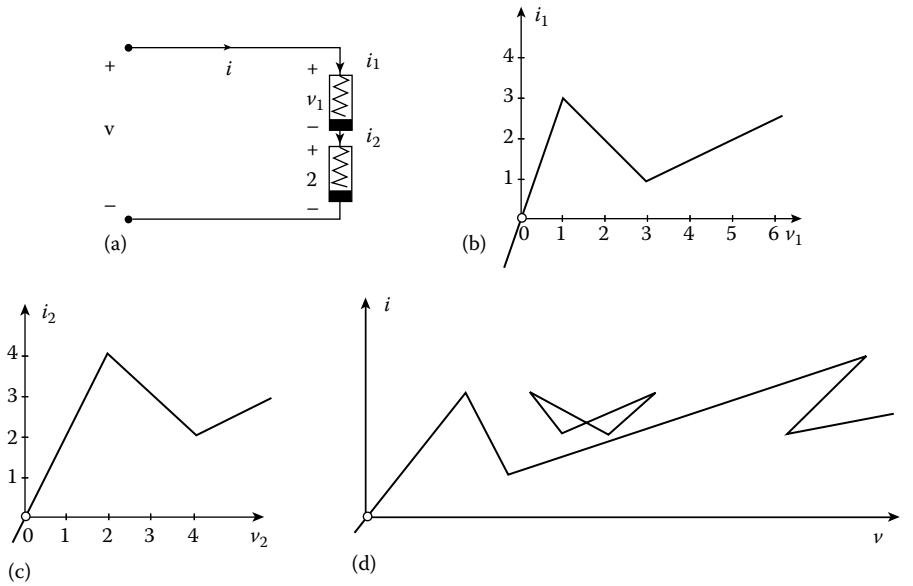


FIGURE 11.15 (a) The series connection of two tunnel diodes, (b) and (c), their $i-v$ characteristics, and (d) the composite $i-v$ plot, which consists of two unconnected parts.

components (see Figure 11.1a) and can be described by GLCP equations. Such networks have not only shown their importance in analysis but also in the topologic study of the number of solutions and more general qualitative properties. When only short-circuit and open-circuit branches are present, one independent voltage source with internal resistance and ideal diodes, an interesting loop cut set exclusion property holds that is also called the colored branch theorem or the arc coloring theorem (see Section 1.7 of *Fundamentals of Circuits and Filters*). It says that the voltage source either forms a conducting loop with forward-oriented diodes and some short circuits or there is a cut set of the voltage source, some open circuits, and blocking diodes. Such arguments have been used to obtain [23] topologic criteria for upper bounds of the number of solutions of PWL resistive circuits. In fact, diode resistor circuits have been used extensively in PWL function generators for analog computers [51]. These electrical analogs can also be used for mathematical programming problems (similar to linear programming) and have reappeared in the neural network literature.

11.5 Analysis of Piecewise-Linear Resistive Circuits

It is first demonstrated that all conventional network formulation methods (nodal, cut, set, hybrid, modified nodal, and tableau) can be used for PWL resistive circuits where the components are described with canonical or with LCP equations. These network equations may have one or more solutions. In order to find solutions, one can either search through all the polyhedral regions P_k by solving the linear equations for that region or by checking whether its solution is located inside that region P_k.

Because many regions often exist, this is a time-consuming method, but several methods can be used to reduce the search [28,61]. If one is interested in only one solution, one can use solution tracing methods, also called continuation methods or homotopy methods, of which the Katzenelson method is best known. If one is interested in all solutions, the problem is more complicated, but some algorithms exist.

11.5.1 Theorem Canonical PWL (Tableau Analysis)

Consider a connected resistive circuit N containing only linear two-terminal resistors, dc-independent sources, current-controlled and voltage-controlled piecewise-linear two-terminal resistors, linear- and piecewise-linear-controlled sources (all four types) and any linear multiterminal resistive elements. A composite branch of this circuit is given in Figure 11.16. If each piecewise-linear function is represented in the canonical form (Equation 11.6), then the *tableau formulation* also has the canonical PWL form

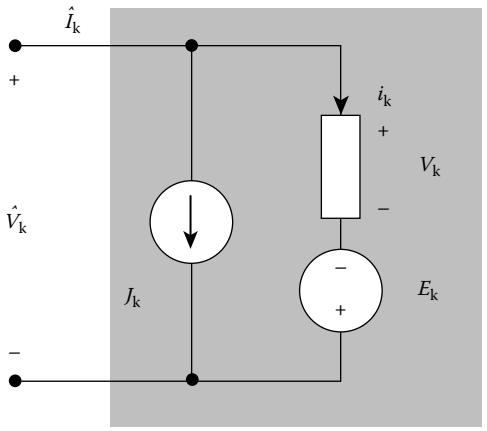


FIGURE 11.16 A composite branch.

$$f(x) = a + Bx + \sum_{i+1}^{p} c_i |\alpha_i^{\mathrm{T}} x - \beta_i| = 0 \qquad (11.43)$$

where $x = \left[i^{\mathrm{T}}, v^{\mathrm{T}}, v_n^{\mathrm{T}}\right]^{\mathrm{T}}$ and i, respectively v, is the branch current voltage vector (Figure 11.16) and v_n is the node-to-datum voltage vector.

PROOF. Let A be the reduced incidence matrix of N relative to some datum node, then KCL, KVL, and element constitutive relations give

$$Ai = AJ \qquad (11.44)$$

$$v = A^{\mathrm{T}} v_n + E \qquad (11.45)$$

$$f_1(i) + f_v(v) = S \qquad (11.46)$$

where we can express $f_{\mathrm{I}}(\cdot)$ and $f_{\mathrm{v}}(\cdot)$ in the canonical form (Equation 11.6)

$$f_I(i) = a_I + B_I i + C_I \mathrm{abs}\left(D_I^{\mathrm{T}} e - e_1\right) \tag{11.47}$$

$$f_{\mathrm{v}}(v) = a_{\mathrm{v}} + B_{\mathrm{v}} v + C_{\mathrm{v}} \mathrm{abs}\left(D_{\mathrm{v}}^{\mathrm{T}} v - e_{\mathrm{v}}\right) \tag{11.48}$$

Substituting Equations 11.47 and 11.48 into Equation 11.46, we obtain

$$\begin{bmatrix} -AJ \\ -E \\ a_I + a_{\mathrm{v}} - S \end{bmatrix} + \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & A^{\mathrm{T}} \\ B_I & B_{\mathrm{v}} & 0 \end{bmatrix} \begin{bmatrix} i \\ v \\ v_n \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ C_{\mathrm{I}} & C_{\mathrm{v}} & 0 \end{bmatrix}$$
$$\mathrm{abs}\left(\begin{bmatrix} D_{\mathrm{I}} & 0 & 0 \\ 0 & D_{\mathrm{V}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i \\ v \\ v_n \end{bmatrix} - \begin{bmatrix} e_{\mathrm{I}} \\ e_{\mathrm{V}} \\ 0 \end{bmatrix} \right) = 0 \tag{11.49}$$

Clearly, Equation 11.49 is in the canonical form of Equation 11.43.

Of course, an analogous theorem can be given when the PWL resistors are given in LCP form. Then the tableau constitute a GLCP. Moreover, completely in line with the section on circuit analysis (see Chapter 23 of *Fundamentals of Circuits and Filters*), one can derive nodal, cut set, loop, hybrid, and modified nodal analysis from the tableau analysis by eliminating certain variables. Alternatively, one can also directly derive these equations.

Whatever the description for the PWL components may be, one can always formulate the network equations as linear equations

$$0 = f(x) = a_k + B_k x, \quad x \in P_k \tag{11.50}$$

in the polyhedral region P_k defined by Equation 11.50. The map f is a continuous PWL map. A solution x of Equation 11.50 can then be computed in a finite number of steps with the Katzenelson algorithm [4,33], by tracing the map f from an initial point $(\mathrm{x}^{(1)}, \mathrm{y}^{(1)})$ to a value $(\mathrm{x}^*, 0)$ (see Figure 11.18).

11.5.2 Algorithm

STEP 1. Choose an initial point $x^{(1)}$ and determine its polyhedral region $P^{(1)}$, and compute

$$y^{(1)} = f\left(x^{(1)}\right) = a^{(1)} + B^{(1)} x \text{ and set} \quad j = 1$$

STEP 2. Compute

$$\hat{x} = x^{(j)} + \left(B^{(j)}\right)^{-1} \left(0 - y^{(j)}\right) \tag{11.51}$$

STEP 3. If $\hat{x} \in P^{(j)}$, we have obtained a solution $\hat{x}$ of $f(\hat{x}) = 0$. Stop.

STEP 4. Otherwise, compute

$$x^{(j+1)} = x^{(j)} + \lambda^{(j)} \left(\hat{x} - x^{(j)}\right) \tag{11.52}$$

where $\lambda^{(j)}$ is the largest number such that $x^{(j+1)} \in P^{(j)}$, i.e., $x^{(j+1)}$ is on the boundary between $P^{(j)}$ and $P^{(j+1)}$ (see Figure 11.17).

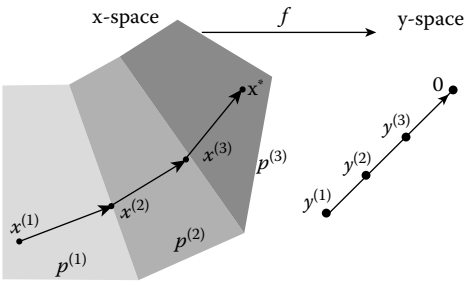


FIGURE 11.17 Iteration in the Katzenelson algorithm for solving $y = f(x) = 0$.

STEP 5. Identify $P^{(j+1)}$ and the linear map $y = a^{(j+1)} + B^{(j+1)}\,x$ in the polyhedral region $P^{(j+1)}$ and compute

$$y^{(j+1)} = y^{(j)} + \lambda^{(j)}\left(y^{\star} - y^{(j)}\right) \tag{11.53}$$

Set $j = j + 1$. Go to step 2.

This algorithm converges to a solution in a finite number of steps if the determinants of all matrices $B^{(j)}$ have the same sign. This condition is satisfied when the i–v curves for the PWL one port resistors are monotonic. The Katzenelson algorithm was extended in Ref. [45] by taking the sign of the determinants into account in Equations 11.52 and 11.53. This requires the PWL resistors to be globally coercive. If by accident in the iteration the point $x^{(j+1)}$ is not on a single boundary and instead is located on a corner, the region $P^{(j+1)}$ is not uniquely defined. However, with a small perturbation [1], one can avoid this corner and still be guaranteed to converge.

This algorithm was adapted to the canonical PWL Equation 11.49 in Ref. [8]. It can also be adapted to the GLCP. However, there exist circuits where this algorithm fails to converge. For the LCP problem, one can then use other algorithms [20,40,56]. One can also use other homotopy methods [43,57,60], which can be shown to converge based on eventual passivity arguments. In fact, this algorithm extends the rather natural method of source stepping, where the PWL circuit is solved by first making all sources zero and then tracing the solution for increasing (stepping up) the sources. It is instructive to observe here that these methods can be used successfully in another sequence of the steps in Figure 11.1a. Until now, we always first performed the horizontal step of PWL approximation or modeling and then the vertical step of network equation formulation. With these methods, one can first perform the network equation formulation and then the PWL approximation. The advantage is that one can use a coarser grid in the simplicial subdivision far away from the solution, and hence dynamically adapt the accuracy of the PWL approximation.

In any case, if all solutions are requested, all these homotopy-based methods are not adequate, because not all solutions can be found even if the homotopy method is started with many different $x^{(1)}$. Hence, special methods have been designed. It is beyond the scope of this text to give a complete algorithm [39,59], but the solution of the GLCP basically involves two parts. First, calculate the solution set of all nonnegative solutions to Equations 11.10 and 11.11. This is a polyhedral cone where extremal rays can be easily determined [44,54]. Second, this solution set is intersected with a hyperplane and the complementarity condition $u^{\mathrm{T}}s = 0$ implies the elimination of vertices (respectively, convex combinations) where these complementarity (respectively, cross complementarity) is not satisfied. This has allowed to systematically obtain the complete solution set for the circuit of Figure 11.15 and for circuits with infinitely many solutions.

A more recent method [46] covers the PWL i–v characteristic with a union of polyhedra and hierarchically solves the circuit with finer and finer polyhedra.

An important improvement in efficiency for the methods is possible when the PWL function $f(\cdot)$ is separable, i.e., there exist $f^i : \mathbb{R}^{\rightarrow} \mathbb{R}^n$ $i = 1, 2, \ldots, n$ such that

$$f(x) = \sum_{i=1}^{n} f^i(x_i) \tag{11.54}$$

This happens when there are only two terminal PWL resistors, linear resistors, and independent sources, and if the bipolar transistors are modeled by the Ebers–Moll model (see Equation 11.39). Then, the subdivision for x is rectangular and each rectangle is subdivided into simplices (see Figure 11.18). This

property can be used to eliminate certain polyhedral regions without solutions [62] and also to speed up the Katzenelson-type algorithm [60,62]. If there are MOS transistors, the map f is not separable but one can apply the extended concept of pairwise separable map [62].

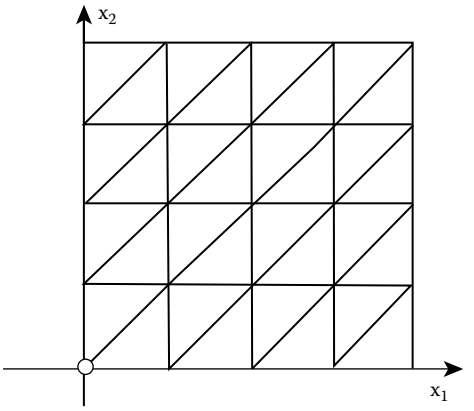


FIGURE 11.18 Simplicial subdivision.

11.6 Piecewise-Linear Dynamic Circuits

As mentioned at the end of Section 11.2, the piecewise-linear descriptions of Section 11.2 can be used also for PWL capacitors, respectively, inductors and memristors, by replacing the port voltages v and currents i by q, v, respectively, φ, i and φ, q. Whenever we have a network obtained by interconnecting linear and/or PWL resistors, inductors, capacitors, and memristors, we have a dynamic piecewise-linear circuits. Of course, such networks are often encountered because it includes the networks with linear R, L, C, and linear-dependent sources, diodes, switches, op-amps, and components such as bipolar and MOS transistors, and GaAs FETs with PWL resistive models. This includes several important and famous nonlinear circuits such as Chua's circuit [18,19], and the cellular neural networks (CNNs) [48], which are discussed in Chapter 13 and Section 14.2.

Of course, PWL dynamic circuits are much more interesting and much more complicated and can exhibit a much more complex behavior than resistive circuits and hence this subject is much less explored. It is clear from the definition of a PWL dynamic circuit that it can be described by linear differential equations over polyhedral regions. Hence, it can exhibit many different types of behavior. They may have many equilibria, which can essentially be determined by solving the resistive network (see Section 11.5 and Figure 11.1) obtained by opening the capacitive ports and short circuiting the inductive ports (dc analysis). When there is no input waveform, the circuit is said to be autonomous and has transients. Some transients may be periodic and are called limit cycles but they may also show chaotic behavior. Next, one may be interested in the behavior of the circuit for certain input waveforms (transient analysis). This can be performed by using integration rules in simulations.

For the analysis of limit cycles, chaos, and transients, one can of course use the general methods for nonlinear circuits, but some improvements can be made based on the PWL nature of the nonlinearities. Here, we only describe the methods briefly. If one is interested in the periodic behavior of a PWL dynamic circuit (autonomous or with a periodic input), then one can, for each PWL nonlinearity, make some approximations.

First, consider the case that one is only interested in the dc and fundamental sinusoidal contributions in all signals of the form $i(t) = A_0 + A_1 \cos \omega t$. The widely used describing function method [6] for PWL resistors $v = f(i)$ consists of approximating this resistor by an approximate resistor where $\hat{v}(t) = D_0 + D_1 \cos \omega t$ has only the dc and fundamental contribution of $v(t)$. This is often a good approximation since the remainder of the circuit often filters out all higher harmonics anyway. Using a Fourier series, one can then find D_0 and D_1 as

$$D_0(A_0, A_1) = \frac{1}{2\pi} \int_0^{2\pi} f(A_0 + A_1 \cos \phi) \mathrm{d}\phi$$

$$D_1(A_0, A_1) = \frac{1}{\pi A_1} \int_0^{2\pi} f(A_0 + A_1 \cos \phi) \mathrm{d}\phi$$

By replacing all PWL components by their describing functions, one can use linear methods to set up the network equations in the Laplace–Fourier domain. When this approximation is not sufficient, one can include more harmonics. Then, one obtains the famous harmonic balance method, because one is balancing more harmonic components.

Alternatively, one can calculate the periodic solution by simulating the circuit with a certain initial condition and considering the map F: $x_0 \rightarrow x_1$ from the initial condition x_0 to the state x_1 one period later. Of course, a fixed point $x^* = F(x^*)$ of the map corresponds to a periodic solution. It has been demonstrated [27] that the map F is differentiable for PWL circuits. This is very useful in setting up an efficient iterative search for a fixed point of F. This map is also useful in studying the eventual chaotic behavior and is then called the Poincaré return map.

In transient analysis of PWL circuits, one is often interested in the sensitivity of the solution to certain parameters in order to optimize the behavior. As a natural extension of the adjoint network for linear circuits in Ref. [22], the adjoint PWL circuit is defined and used to determine simple sensitivity calculations for transient analysis.

Another important issue is whether the PWL approximation of a nonlinear characteristic in a dynamic circuit has a serious impact on the transient behavior. In Ref. [63], error bounds were obtained on the differences of the waveforms.

11.7 Efficient Computer-Aided Analysis of PWL Circuits

Transient analysis and timing verification is an essential part of the VLSI system design process. The most reliable way of analyzing the timing performance of a design is to use analog circuit analysis methods. Here as well, a set of algebraic-differential equations has to be solved. This can be done by using implicit integration formulas that convert these equations into a set of algebraic equations, which can be solved by iterative techniques like Newton–Raphson (see Chapter 12). The computation time then becomes excessive for large circuits. It mainly consists of linearizations of the nonlinear component models and the solution of the linear equations. In addition, the design process can be facilitated substantially if this simulation tool can be used at many different levels from the top level of specifications over the logic and switch level to the circuit level. Such a hierarchical simulator can support the design from top to bottom and allow for mixtures of these levels. In limited space, we describe here the main simulation methods for improving the efficiency and supporting the hierarchy of models with piecewise-linear methods. We refer the reader to Chapter 8, *Computer Aided Design and Design Automation* for general simulation of VLSI circuits and to the literature for more details on the methods and for more descriptions on complete simulators.

It is clear from our previous discussion that PWL models and circuit descriptions can be used at many different levels. An op-amp, for example, can be described by the finite gain model (see Figure 11.9 and Equations 11.29 and 11.30), but when it is designed with a transistor circuit it can be described by PWL circuit equations as in Section 11.5. Hence, it is attractive to use a simulator that can support this top-down design process [35]. One can then even incorporate logic gates into the PWL models. One can organize the topological equations of the network hierarchically, so that it is easy to change the network topology. The separation between topological equations and model descriptions allows for an efficient updating of the model when moving from one polyhedral region into another. Several other efficiency issues can be built into a hierarchical PWL simulator.

An important reduction in computation time needed for solving the network equations can be obtained by using the consistent variation property. In fact, only a rank one difference exists between the matrices of two neighboring polyhedral regions, and hence, one inverse can be easily derived from the other [8,35]. In the same spirit, one can at the circuit level take advantage of the PWL transistor models (see Ref. [62] and separability discussion in Section 8.5). In Ref. [53], the circuit is partitioned dynamically into subcircuits during the solution process, depending on the transistor region of operation. Then, the subcircuits are dynamically ordered and solved with block Gauss–Seidel for minimal or no coupling among them.

Interesting savings can be obtained [34] by solving the linear differential equations in a polyhedral region with Laplace transformations and by partitioning the equations. However, the computation of the intersection between trajectories in neighboring polyhedral regions can be a disadvantage of this method.

Acknowledgment

This work was supported by the Research Council Kuleuven Project MEFISTO666GOA.

References

1. M. J. Chien, Piecewise-linear homeomorphic resistive networks, *IEEE Trans. Circuits Syst*., CAS-24, 118–127, Mar. 1977.
2. M. J. Chien and E. S. Kuh, Solving nonlinear resistive network using piecewise-linear analysis and simplicial subdivision, *IEEE Trans. Circuits Syst*., CAS-24, 305–317, 1977.
3. L. O. Chua, Analysis and synthesis of multivalued memoryless nonlinear networks, *IEEE Trans. Circuits Theory*, CT-14, 192–209, June 1967.
4. L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
5. L. O. Chua and P. M. Lin, A switching-parameter algorithm for finding multiple solutions of nonlinear resistive circuits, *Int. J. Circuit Theory Appl*., 4, 215–239, 1976.
6. L. O. Chua and S. M. Kang, Section-wise piecewise-linear functions: Canonical representation properties and applications, *Proc. IEEE*, 65, 915–929, June 1977.
7. L. O. Chua and D. J. Curtin, Reciprocal n-port resistor represented by continuous n-dimensional piecewise-linear function realized by circuit with 2-terminal piecewise-linear resistor and $p+g$ port transformer, *IEEE Trans. Circuits Syst*., CAS-27, 367–380, May 1980.
8. L. O. Chua and R. L. P. Ying, Finding all solutions of piecewise-linear circuits, *Int. J. Circuit Theory Appl*., 10, 201–229, 1982.
9. L. O. Chua and R. L. P. Ying, Canonical piecewise-linear analysis, *IEEE Trans. Circuits Syst*., CAS-30, 125–140, 1983.
10. L. O. Chua and A. C. Deng, Canonical piecewise-linear analysis—Part II: Tracing driving-point and transfer characteristics, *IEEE Trans. Circuits Syst*., CAS-32, 417–444, May 1985.
11. L. O. Chua and A. C. Deng, Canonical piecewise-linear modeling, unified parameter optimization algorithm: Application to pn junctions, bipolar transistors, MOSFETs, and GaAs FETs, *IEEE Trans. Circuits Syst*., CAS-33, 511–525, May 1986.
12. L. O. Chua and A. C. Deng, Canonical piecewise linear modeling, *IEEE Trans. Circuits Syst*., CAS-33, 511–525, May 1986.
13. L. O. Chua and A. C. Deng, Canonical piecewise linear analysis: Generalized breakpoint hopping algorithm, *Int. J. Circuit Theory Appl*., 14, 35–52, 1986.
14. L. O. Chua and A. C. Deng, Canonical piecewise linear representation, *IEEE Trans. Circuits Syst*., 35, 101–111, Jan. 1988.
15. L. O. Chua and A. C. Deng, Canonical piecewise-linear modeling, ERL Memo. UCB/ERL M85/35, University of California, Berkeley, Apr. 26, 1985.
16. L. O. Chua and G. Lin, Canonical realization of Chaua's circuit family, *IEEE Trans. Circuits Syst*., 37, 885–902, July 1990.
17. L. O. Chua and G. Lin, Intermittency in piecewise-linear circuit, *IEEE Trans. Circuits Syst*., 38, 510–520, May 1991.
18. L. O. Chua, The genesis of Chua's circuit, *Archiv Elektronik Übertragungstechnik*, 46(4), 250–257, 1992.
19. L. O. Chua, C.-W. Wu, A.-S. Huang, and G.-Q. Zhong, A universal circuit for studying and generating chaos, *IEEE Trans. Circuits Syst. I*, 40(10), 732–744, 745–761, Oct. 1993.

20. R. Cottle, J.-S. Pang, and R. Stone, *The Linear Complementarity Problem*, New York: Academic Press, 1992.
21. A. C. Deng, Piecewise-linear timing model for digital CMOS circuits, *IEEE Trans. Circuits Syst.*, 35, 1330–1334, Oct. 1988.
22. Y. Elcherif and P. Lin, Transient analysis and sensitivity computation in piecewise-linear circuits, *IEEE Trans. Circuits Syst.*, 36, 1525–1533, Dec. 1988.
23. M. Fossepréz, M. J. Hasler, and C. Schnetzler, On the number of solutions of piecewise-linear resistive circuits, *IEEE Trans. Circuits Syst.*, 36, 393–402, Mar. 1989.
24. T. Fujisawa and E. S. Kuh, Piecewise-linear theory of nonlinear networks, *SIAM J. Appl. Math.*, 22(2), 307–328, Mar. 1972.
25. T. Fujisawa, E. S. Kuh, and T. Ohtsuki, A sparse matrix method for analysis of piecewise-linear resistive circuits, *IEEE Trans. Circuit Theory*, 19, 571–584, Nov. 1972.
26. G. Güzelis and I. Göknar, A canonical representation for piecewise affine maps and its applications to circuit analysis, *IEEE Trans. Circuits Syst.*, 38, 1342–1354, Nov. 1991.
27. I. N. Hajj and S. Skelboe, Dynamic systems: Steady-state analysis, *IEEE Trans. Circuits Syst.*, CAS-28, 234–242, March 1981.
28. Q. Huang and R. W. Liu, A simple algorithm for finding all solutions of piecewise-linear networks, *IEEE Trans. Circuits Syst.*, 36, 600–609, Apr. 1989.
29. C. Kahlert and L. O. Chua, A generalized canonical piecewise-linear representation, *IEEE Trans. Circuits Syst.*, 37, 373–383, Mar. 1990.
30. C. Kahlert and L. O. Chua, Completed canonical piecewise-linear representation: Geometry of domain space, *IEEE Trans. Circuits Syst.*, 39, 222–236, Mar. 1992.
31. S. M. Kang and L. O. Chua, A global representation of multidimensional piecewise-linear functions with linear partitions, *IEEE Trans. Circuits Syst.*, CAS-25, 938–940, Nov. 1978.
32. S. Karamardian, The complementarity problem, *Mathemat. Program.*, 2, 107–129, 1972.
33. S. Karamardian and J. Katzenelson, An algorithm for solving nonlinear resistive networks, *Bell Syst. Tech. J.*, 44, 1605–1620, 1965.
34. R. J. Kaye and A. Sangiovanni-Vincentelli, Solution of piecewise-linear ordinary differential equations using waveform relaxation and Laplace transforms, *IEEE Trans. Circuits Syst.*, CAS-30, 353–357, June 1983.
35. T. A. M. Kevenaar and D. M. W. Leenaerts, A flexible hierarchical piecewise-linear simulator, *Integrat. VLSI J.*, 12, 211–235, 1991.
36. T. A. M. Kevenaar and D. M. W. Leenaerts, A comparison of piecewise-linear model descriptions, *IEEE Trans. Circuits Syst.*, 39, 996–1004, Dec. 1992.
37. M. Kojima and Y. Yamamoto, Variable dimension algorithms: Basic theory, interpretations and extensions of some existing methods, *Mathemat. Program.*, 24, 177–215, 1982.
38. S. Lee and K. Chao, Multiple solution of piecewise-linear resistive networks, *IEEE Trans. Circuits Syst.*, CAS-30, 84–89, Feb. 1983.
39. D. M. W. Leenaerts and J. A. Hegt, Finding all solutions of piecewise-linear functions and the application to circuits design, *Int. J. Circuit Theory Appl.*, 19, 107–123, 1991.
40. C. E. Lemke, On complementary pivot theory, in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritten, Eds., New York: Academic Press, 1968, pp. 349–384.
41. J. Lin and R. Unbehauen, Canonical piecewise-linear approximations, *IEEE Trans. Circuits Syst.*, 39, 697–699, Aug. 1992.
42. R. Lum and L. O. Chua, Generic properties of continuous piecewise-linear vector fields in 2-D space, *IEEE Trans. Circuits Syst.*, 38, 1043–1066, Sep. 1991.
43. R. Melville, L. Trajkovic, S.-C. Fang, and L. Watson, Artificial homotopy methods for the DC operating point problem, *IEEE Trans Comput.-Aided Design Integrat. Circuits Syst.*, 12, 861–877, June 1993.

44. T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall, The double description method, in *Contributions to the Theory of Games Ann. Mathemat. Studies*, H. W. Kuhn and A. W. Tucker, Eds., Princeton: Princeton University Press, 1953, pp. 51–73.
45. T. Ohtsuki, T. Fujisawa, and S. Kumagai, Existence theorem and a solution algorithm for piecewise-linear resistor circuits, *SIAM J. Math. Anal.*, 8(1), 69–99, 1977.
46. S. Pastore and A. Premoli, Polyhedral elements: A new algorithm for capturing all the equilibrium points of piecewise-linear circuits, *IEEE Trans. Circuits Syst. I.*, 40, 124–132, Feb. 1993.
47. V. C. Prasad and V. P. Prakash, Homeomorphic piecewise-linear resistive networks, *IEEE Trans. Circuits Syst.*, 35, 251–253, Feb. 1988.
48. T. Roska and J. Vandewalle, *Cellular Neural Networks*, New York: John Wiley & Sons, 1993.
49. I. W. Sandberg, A note on the operating-point equations of semiconductor-device networks, *IEEE Trans. Circuits Syst.*, 37, 966, July 1990.
50. A. S. Solodovnikov, *System of Linear Inequalities*, translated by L. M. Glasser and T. P. Branson, Chicago: University of Chicago, 1980.
51. T. E. Stern, *Theory of Nonlinear Networks and Systems: An Introduction*, Reading, MA: Addison-Wesley, 1965.
52. S. Stevens and P.-M. Lin, Analysis of piecewise-linear resistive networks using complementary pivot theory, *IEEE Trans Circuits Syst.*, CAS-28, 429–441, May 1981.
53. O. Tejayadi and I. N. Hajj, Dynamic partitioning method for piecewise-linear VLSI circuit simulation, *Int. J. Circuit Theory Appl.*, 16, 457–472, 1988.
54. S. N. Tschernikow, *Lineare Ungleichungen*. Berlin: VEB Deutscher Verlag der Wissenschaften, 1971; translation from H. Weinert and H. Hollatz, *Lineinye Neravenstva*, 1968, into German.
55. W. M. G. van Bokhoven, *Piecewise-Linear Modelling and Analysis*. Deventer: Kluwer Academic, 1980.
56. C. Van de Panne, A complementary variant of Lemke's method for the linear complementary problem, *Mathemat. Program.*, 7, 283–310, 1974.
57. L. Vandenberghe and J. Vandewalle, Variable dimension algorithms for solving resistive circuits, *Int. J. Circuit Theory Appl.*, 18, 443–474, 1990.
58. L. Vandenberghe and J. Vandewalle, A continuous deformation algorithm for DC-analysis of active nonlinear circuits, *J. Circuits Syst. Comput.*, 1, 327–351, 1991.
59. L. Vandenberghe, B. L. De Moor, and J. Vandewalle, The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits, *IEEE Trans. Circuits Syst.*, 36, 1382–1391, 1989.
60. K. Yamamura and K. Horiuchi, A globally and quadratically convergent algorithm for solving resistive nonlinear resistive networks, *IEEE Trans. Comput.-Aided Design Integrat. Circuits Syst.*, 9, 487–499, May 1990.
61. K. Yamamura and M. Ochiai, Efficient algorithm for finding all solutions of piecewise-linear resistive circuits, *IEEE Trans. Circuits Syst.*, 39, 213–221, Mar. 1992.
62. K. Yamamura, Piecewise-linear approximation of nonlinear mappings containing Gummel-Poon models or Shichman-Hodges models, *IEEE Trans. Circuits Syst.*, 39, 694–697, Aug. 1992.
63. M. E. Zaghloul and P. R. Bryant, Nonlinear network elements: Error bounds, *IEEE Trans. Circuits Syst.*, CAS-27, 20–29, Jan. 1980.

12

Simulation

12.1 Numerical Solution of Nonlinear Algebraic Equations **12**-2
12.2 Numerical Integration of Nonlinear Differential Equations **12**-3
12.3 Use of Simulation Programs **12**-4
SPICE • APLAC • NAP2 • ESACAP • DYNAST
References **12**-18

Erik Lindberg
Technical University of Denmark

This chapter deals with the simulation or analysis of a nonlinear electrical circuit by means of a computer program. The program creates and solves the differential-algebraic equations of a model of the circuit. The basic tools in the solution process are *linearization*, *difference approximation*, and *the solution of a set of linear equations*. The output of the analysis may consist of (1) all node and branch voltages and all branch currents of a bias point (dc analysis), (2) a linear small-signal model of a bias point that may be used for analysis in the frequency domain (ac analysis), or (3) all voltages and currents as functions of time in a certain time range for a certain excitation (transient analysis). A model is satisfactory if there is good agreement between measurements and simulation results. In this case, simulation may be used instead of measurement for obtaining a better understanding of the nature and abilities of the circuit. The crucial point is to set up a model that is as simple as possible, in order to obtain a fast and inexpensive simulation, but sufficiently detailed to give the proper answer to the questions concerning the behavior of the circuit under study. Modeling is the bottleneck of simulation.

The model is an equivalent scheme—"schematics-capture"—or a branch table—"net-list"—describing the basic components (n-terminal elements) of the circuit and their connection. It is always possible to model an n-terminal element by means of a number of 2-terminals (branches). These internal 2-terminals may be coupled. By pairing the terminals of an n-terminal element, a port description may be obtained. The branches are either admittance branches or impedance branches. All branches may be interpreted as controlled sources. An admittance branch is a current source primarily controlled by its own voltage or primarily controlled by the voltage or current of another branch (transadmittance). An impedance branch is a voltage source primarily controlled by its own current or primarily controlled by the current or voltage of another branch (transimpedance). Control by signal (voltage or current) and control by time-derivative of signal are allowed. Control by several variables is allowed. Examples of admittance branches are (1) the conductor is a current source controlled by its own voltage, (2) the capacitor is a current source controlled by the time-derivative of its own voltage, and (3) the open circuit is a zero-valued current source (a conductor with value zero). Examples of impedance branches are (1) the resistor is a voltage source controlled by its own current, (2) the inductor is a voltage source controlled by the time-derivative of its own current, and (3) the short circuit is a zero-valued voltage source (a resistor with value zero)

A component may often be modeled in different ways. A diode, for example, is normally modeled as a current source controlled by its own voltage such that the model can be linearized into a dynamic conductor in parallel with a current source during the iterative process of finding the bias point of the diode. The diode may also be modeled as (1) a voltage source controlled by its own current (a dynamic resistor in series with a voltage source), (2) a static conductor being a function of the voltage across the diode, or (3) a static resistor being a function of the current through the diode. Note that in the case where a small-signal model is wanted, for frequency analysis, only the dynamic model is appropriate.

The primary variables of the model are the currents of the impedance branches and the node potentials. The current law of Kirchhoff (the sum of all the currents leaving a node is zero) and the current–voltage relations of the impedance branches are used for the creation of the equations describing the relations between the primary variables of the model. The contributions to the equations from the branches are taken one branch at a time based on the question: Will this branch add new primary variables? If yes, then a new column (variables) and a new row (equations) must be created and updated, or else the columns and rows corresponding to the existing primary variables of the branch must be updated. This approach to equation formulation is called the extended nodal approach or the modified nodal approach (MNA).

In the following, some algorithms for solving a set of nonlinear algebraic equations and nonlinear differential equations are briefly described. Because we are dealing with physical systems and because we are responsible for the models, we assume that at least one solution is possible. The zero solution is, of course, always a solution. It might happen that our models become invalid if we, for example, increase the amplitudes of the exciting signals, diminish the risetime of the exciting signals, or by mistake create unstable models. It is important to define the range of validity for our models. What are the consequences of our assumptions? Can we believe in our models?

12.1 Numerical Solution of Nonlinear Algebraic Equations

Let the equation system to be solved be $f(x, u) = 0$, where x is the vector of primary variables and u is the excitation vector. Denote the solution by x_s. Then, if we define a new function $g(x) = \alpha(f(x, u)) + x$, where α may be some function of $f(x, u)$, which is zero for $f(x, u) = 0$, then we can define an iterative scheme where $g(x)$ converges to the solution x_s by means of the iteration: $x_{k+1} = g(x_k) = \alpha(f(x_k, u)) + x_k$ where k is the iteration counter.

If for all x in the interval $[x_a, x_b]$ the condition $\|g(x_a) - g(x_b)\| \leq L^* \|x_a - x_b\|$ for some $L < 1$ is satisfied, the iteration is called a contraction mapping. The condition is called a Lipschitz condition. Note that a function is a contraction if it has a derivative less than 1.

For $\alpha = -1$, the iterative formula becomes $x_{k+1} = g(x_k) = -f(x_k, u) + x_k$. This scheme is called the Picard method, the functional method, or the contraction mapping algorithm. At each step, each nonlinear component is replaced by a linear static component corresponding to the solution x_k. A nonlinear conductor, for example, is replaced by a linear conductor defined by the straight line through the origin and the solution point. Each iterative solution is calculated by solving a set of linear equations. All components are updated and the next iteration is made. When two consecutive solutions are within a prescribed tolerance, the solution point is accepted.

For $\alpha = -1/(\mathrm{d}f/\mathrm{d}x)$, the iterative formula becomes $x_{k+1} = g(x_k) = -f(x_k, u)/(\mathrm{d}f(x_k, u)/\mathrm{d}x) + x_k$. This scheme is called the Newton–Raphson method or the derivative method. At each step, each nonlinear component is replaced by a linear dynamic component plus an independent source corresponding to the solution x_k. A nonlinear conductor, for example, is replaced by a linear conductor defined by the derivative of the branch current with respect to the branch voltage (the slope of the nonlinearity) in parallel with a current source corresponding to the branch voltage of the previous solution. A new solution is then calculated by solving a set of linear equations. The components are updated and the next iteration is made. When the solutions converge within a prescribed tolerance, the solution point is accepted.

It may, of course, happen that the previously mentioned iterative schemes do not converge before the iteration limit k_{max} is reached. One reason may be that the nonlinearity $f(x)$ changes very rapidly for a small change in x. Another reason could be that $f(x)$ possess some kind of symmetry that causes cycles in the Newton–Raphson iteration scheme. If convergence problems are detected, the iteration scheme can be *modified* by introducing a limiting of the actual step size. Another approach may be to change the modeling of the nonlinear branches from voltage control to current control or vice versa. Often, the user of a circuit analysis program may be able to solve convergence problems by means of proper modeling and adjustment of the program options [1–5].

12.2 Numerical Integration of Nonlinear Differential Equations

The dynamics of a nonlinear electronic circuit may be described by a set of coupled first-order differential equations–algebraic equations of the form: $dx/dt = f(x, y, t)$ and $g(x, y, t) = 0$, where x is the vector of primary variables (node potentials and impedance branch currents), y is the vector of variables that cannot be explicitly eliminated, and f and g are nonlinear vector functions. It is always possible to express y as a function of x and t by inverting the function g and inserting it into the differential equations such that the general differential equation form $dx/dt = f(x, t)$ is obtained. The task is then to obtain a solution $x(t)$ when an initial value of x is given. The usual methods for solving differential equations reduce to the solution of difference equations, with either the derivatives or the integrals expressed *approximately* in terms of finite differences.

Assume, at a given time t_0, we have a known solution point $x_0 = x(t_0)$. At this point, the function f can be expanded in Taylor series: $dx/dt = f\ (x_0, t) + A\ (x_0)\ (x - x_0)\ + \cdots$ where $A(x_0)$ is the Jacobian of f evaluated at x_0. Truncating the series, we obtain a linearization of the equations such that the small-signal behavior of the circuit in the neighborhood of x_0 is described by $dx/dt = A * x + k$, where A is a constant matrix equal to the Jacobian and k is a constant vector.

The most simple scheme for the approximate solution of the differential equation $dx/dt = f(x, t) = Ax + k$ is the forward Euler formula $x(t) = x(t_0) + hA(t_0)$ where $h = t - t_0$ is the integration time step. From the actual solution point at time t_0, the next solution point at time t is found along the tangent of the solution curve. It is obvious that we will rapidly leave the vicinity of the exact solution curve if the integration step is too large. To guarantee stability of the computation, the time step h must be smaller than $2/|\lambda|$ where λ is the largest eigenvalue of the Jacobian A. Typically, h must not exceed $0.2/|\lambda|$.

The forward Euler formula is a linear *explicit* formula based on forward Taylor expansion from t_0. If we make backward Taylor expansion from t we arrive at the backward Euler formula: $x(t) = x(t_0) + hA(t)$. Because the unknown appears on both sides of the equation, it must in general be found by iteration so the formula is a linear *implicit* formula. From a stability point of view, the backward Euler formula has a much larger stability region than the forward Euler formula. The truncation error for the Euler formulas is of order h^2.

The two Euler formulas can be thought of as polynomials of degree 1 that approximate $x(t)$ in the interval $[t_0, t]$. If we compute $x(t)$ from a second-order polynomial $p(t)$ that matches the conditions that $p(t_0) = x(t_0)$, $dp/dt(t_0) = dx/dt(t_0)$ and $dp/dt(t) = dx/dt(t)$, we arrive at the trapezoidal rule: $x(t) = x(t_0) + 0.5hA(t_0) + 0.5hA(t)$. In this case, the truncation error is of order h^3.

At each integration step, the size of the local truncation error can be estimated. If it is too large, the step size must be reduced. An explicit formula such as the forward Euler may be used as a predictor giving a starting point for an implicit formula like the trapezoidal, which in turn is used as a corrector. The use of a predictor–corrector pair provides the base for the estimate of the local truncation error. The trapezoidal formula with varying integration step size is the main formula used in the Simulation Program with Integrated Circuit Emphasis (SPICE) program.

The two Euler formulas and the trapezoidal formula are special cases of a general linear multistep formula $\Sigma(a_i x_{n-i} + b_i h(dx/dt)_{n-i})$, where i goes from -1 to $m - 1$ and m is the degree of the polynomial

used for the approximation of the solution curve. The trapezoidal rule, for example, is obtained by setting $a_{-1} = -1$, $a_0 = +1$, and $b_{-1} = b_0 = 0.5$, all other coefficients being zero. The formula can be regarded as being derived from a polynomial of degree r which matches $r+1$ of the solution points x_{n-i} and their derivatives $(\mathrm{d}x/\mathrm{d}t)_{n-i}$.

Very fast transients often occur together with very slow transients in electronic circuits. We observe widely different time constants. The large spread in component values, for example, from large decoupling capacitors to small parasitic capacitors, implies a large spread in the modules of the eigenvalues. We say that the circuits are stiff. A family of implicit multistep methods suitable for stiff differential equations has been proposed by C.W. Gear. The methods are stable up to the polynomial of order 6. For example, the second-order Gear formula for fixed integration step size h may be stated as $x_{n+1} = -(1/3)x_{n-1} + (4/3)x_n + (2/3)h(\mathrm{d}x/\mathrm{d}t)_{n+1}$.

By changing both the *order* of the approximating polynomial and the integration *step size*, the methods adapt themselves dynamically to the performance of the solution curve. The family of Gear formulas is modified into a ''stiff-stable variable-order variable-step predictor–corrector'' method based on implicit approximation by means of backward difference formulas (BDFs). The resulting set of nonlinear equations is solved by modified Newton–Raphson iteration. Note that numerical integration, in a sense, is a kind of low-pass filtering defined by means of the minimum integration step [1–5].

12.3 Use of Simulation Programs

Since 1960, a large number of circuit-simulation programs have been developed by universities, industrial companies, and commercial software companies. In particular, the SPICE program has become a standard simulator both in the industry and in academia. Here, only a few programs, which together cover a very large number of simulation possibilities, are presented. Due to competition, there is a tendency to develop programs that are supposed to cover any kind of analysis so that only one program should be sufficient (the Swiss Army Knife Approach). Unfortunately this implies that the programs become very large and complex to use. Also, it may be difficult to judge the correctness and accuracy of the results of the simulation having only one program at your disposal. If you try to make the same analysis of the same model with different programs, you will frequently see that the results from the programs may not agree completely. By comparing the results, you may obtain a better feel for the correctness and accuracy of the simulation. The programs SPICE and Analysis Program for Linear Active Circuits (APLAC) supplemented with the programs Nonlinear Analysis Program version 2 (NAP2), Engineering System and Circuit Analysis (ESACAP), and DYNAmic Simulation Tool (DYNAST) have proven to be a good choice in the case where a large number of different kinds of circuits and systems are to be modeled and simulated (the Tool Box Approach). The programs are available in inexpensive evaluation versions running on IBM compatible personal computers. The ''net-list'' input languages are very close, making it possible to transfer input data easily between the programs. In order to make the programs more ''user-friendly'' graphics interphase language ''schematics-capture,'' where you draw the circuit on the screen, has been introduced. Unfortunately, this approach makes it a little more difficult for the user to transfer data between the programs. In the following, short descriptions of the programs are given and a small circuit is simulated in order to give the reader an idea of the capabilities of the programs.

12.3.1 SPICE

The first versions of SPICE (Simulation Program with Integrated Circuit Emphasis version 2), based on the MNA, were developed in 1975 at the Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA.

SPICE is a general-purpose circuit analysis program. Circuit models may contain resistors, capacitors, inductors, mutual inductors, independent sources, controlled sources, transmission lines, and the

most common semiconductor devices: diodes, bipolar junction transistors, and field effect transistors. SPICE has very detailed built-in models for the semiconductor devices, which may be described by about 50 parameters. Besides the normal dc, ac, and transient analyses, the program can make sensitivity, noise, and distortion analysis and analysis at different temperatures. In the various commercial versions of the program many other possibilities have been added; for example, analog behavior modeling (poles and zeros) and statistical analysis.

In order to give an impression of the ''net-list'' input language, the syntax of the statements describing controlled sources is the following:

```
Voltage Controlled Current Source: Gxxx N+ N− NC+ NC− VALUE
Voltage Controlled Voltage Source: Exxx N+ N− NC+ NC− VALUE
Current Controlled Current Source: Fxxx N+ N− VNAM VALUE
Current Controlled Voltage Source: Hxxx N+ N− VNAM VALUE
```

where the initial characters of the branch name G, E, F, and H indicate the type of the branch; $N+$ and $N-$ are integers (node numbers) indicating the placement and orientation of the branch, respectively; NC+, NC−, and VNAM indicate from where the control comes (VNAM is a dummy dc voltage source with value 0 inserted as an ammeter!); and VALUE specifies the numerical value of the control, which may be a constant or a polynomial expression in case of nonlinear dependent sources. Independent sources are specified with Ixxx for current and Vxxx for voltage sources.

The following input file describes an analysis of the Chua oscillator circuit. It is a simple harmonic oscillator with losses (C2, L2, and RL2) loaded with a linear resistor (R61) in series with a capacitor (C1) in parallel with a nonlinear resistor. The circuit is influenced by a sinusoidal voltage source VRS through a coil L1. Comments may be specified either as lines starting with an asterisk ''*'' or by means of a semicolon '';'' after the statement on a line. A statement may continue by means of a plus '' + '' as the first character on the following line.

```
PSpice input file CRC-CHUA.CIR, first line, title line                    :
* *:  The Chua Oscillator, sinusoidal excitation, F=150mV >                :
* :            RL2=1 ohm, RL1=0 ohm f=1286.336389 Hz                      :
* : ref. K. Murali and M. Lakshmanan,                                      :
* :      Effect of Sinusoidal Excitation on the Chua's Circuit,            :
* :      IEEE Transactions on Circuits and Systems-1:                      :
* :      Fundamental Theory and Applications,                              :
* :      vol.39, No.4, April 1992, pp. 264-270                             :
* : input source;  :-----------------------------------------:
    VRS    7   0  sin(0    150m   1.2863638893332e+3 0 0)                  :
* : choke                                                                  :
    L1     6  17  80e-3    ; mH                                            :
    VRL1  17   7  DC  0    ; ammeter for measure of IL1                    :
* : harmonic oscillator; :-----------------------------------:
    L2     6  16  13m                                                      :
    RL2   16   0  1                                                        :
    C2     6   0  1.250u                                                   :
* : load; :-------------------------------------------------:
    r61     6   10  1310                                                  :
    vrrC1  10   11  DC       0                                             :
    C1     11    0  0.017u                                                 :
*   i(vrr10)=current of nonlinear resistor                                 :
    vrr10  10    1  DC   0                                                 :
```

```
* : non-linear circuit; :-------------------------------------:
    .model n4148 d (is=0.1p rs=16 n=1); vt=n*k*T/q                :
    d13      1     3     n4148                                    :
    d21      2     1     n4148                                    :
    rm9      2    22     47k                                      :
    vrm9    22     0     DC    -9          ; negative power supply :
    rp9      3    33     47k                                      :
    vrp9    33     0     DC    +9                                 :
    r20      2     0     3.3k                                     :
    r30      3     0     3.3k                                     :
* : ideal op. amp.; :-----------------------------------------:
    evop   4 0     1     5     1e+20                              :
    r14    1 4   290                                             :
    r54    5 4   290                                             :
    r50    5 0  1.2k                                             :
* : ---------------------------------------------------------:
    .TRAN 0.05m 200m 0 0.018m UIC                                 :
    .plot tran v(11)                                              :
    .probe                                                        :
    .options acct nopage opts gmin=1e-15 reltol=1e-3              :
+   abstol=1e-12 vntol=1e-12 tnom=25 itl5=0                       :
+   limpts=15000                                                  :
    .end                                                          :
```

The analysis is controlled by means of the statements: .TRAN, where, for example, the maximum integration step is fixed to 18 μs, and .OPTIONS, where, for example, the relative truncation error is set to 1e-3. The result of the analysis is presented in Figure 12.1. It is seen that transition from chaotic behavior to a period 5 limit cycle takes place at about 100 ms. A very important observation is that the result of the analysis may depend on (1) the choice of the control parameters and (2) the order of the branches in the "net-list," for example, if the truncation error is set to 1e-6 instead of 1e-3 previously, the result becomes quite different. This observation is valid for all programs [5–11].

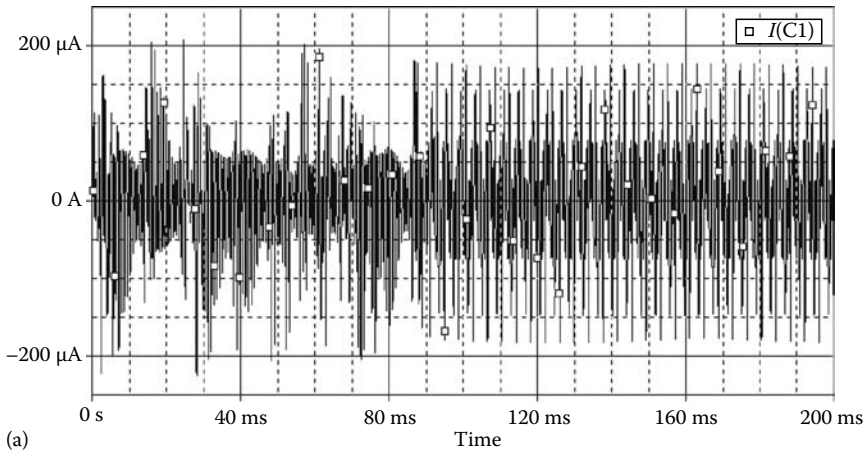
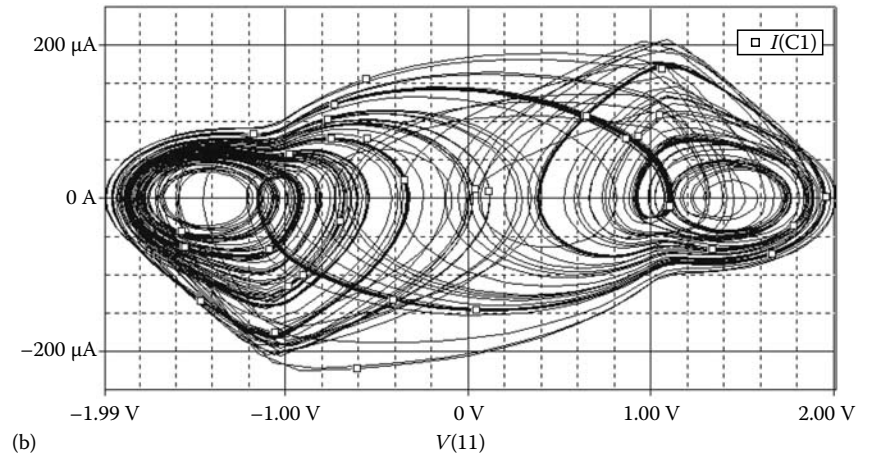
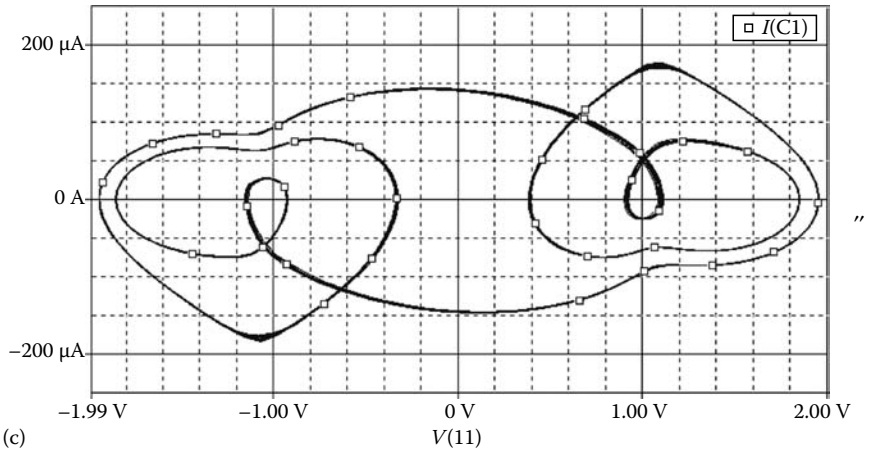


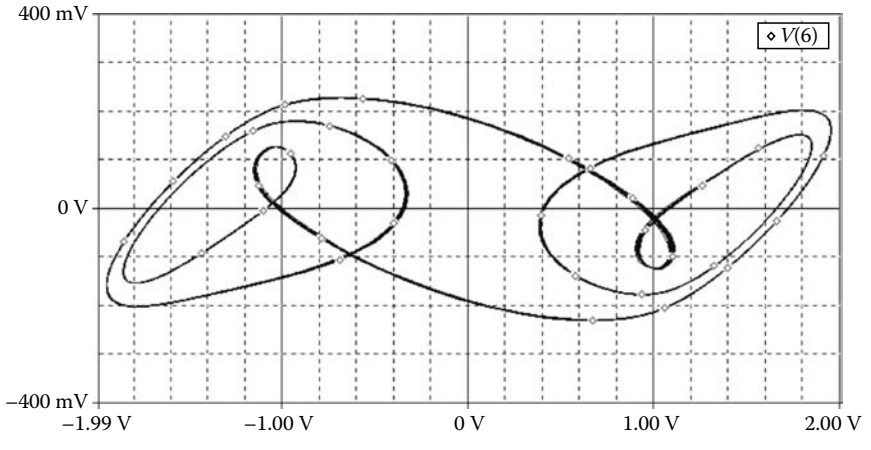
FIGURE 12.1 (a) PSPICE analysis. The current of C1: I(C1) as function of time in the interval 0–200 ms.



(b)



(c)



(d)

FIGURE 12.1 (continued) (b) The current of C1: I(C1) as function of the voltage of C1: V(11). (c) The current of C1: I(C1) as function of the voltage of C1: V(11) in the time interval 100–200 ms. (d) The voltage of C2: V(6) as function of the voltage of C1: V(11) in the time interval 100–200 ms.

12.3.2 APLAC

The program APLAC [5] has been under constant development at the Helsinki University of Technology, Finland, since 1972. Over time it has developed into an object-oriented analog circuits and systems simulation and design tool. Inclusion of a new model into APLAC requires only the labor of introducing the parameters and equations defining the model under the control of ''C-macros.'' The code of APLAC itself remains untouched. The APLAC *Interpreter* immediately understands the syntax of the new model. APLAC accepts SPICE ''net-lists'' by means of the program Spi2a (SPICE to APLAC net-list converter).

APLAC is capable of carrying out dc, ac, transient, noise, oscillator, and multitone harmonic steady-state analyses and measurements using IEEE-488 bus. Transient analysis correctly handles, through convolution, components defined by frequency-dependent characteristics. Monte Carlo analysis is available in all basic analysis modes and sensitivity analysis in dc and ac modes. *N*-port *z*, *y*, and *s* parameters, as well as two-port *h* parameters, are available in ac analysis. In addition, APLAC includes a versatile collection of system-level blocks for the simulation and design of analog and digital communication systems. APLAC includes seven different optimization methods. Any parameter in the design problem can be used as a variable, and any user-defined function may act as an objective. Combined time and frequency domain optimization is possible.

The file below is the APLAC ''net-list'' of the Chua oscillator circuit created by the Spi2a converter program with the PSpice file CRC-CHUA.CIR above as input. Comments are indicated by means of the dollar sign ''$'' or the asterisk ''*.'' Unfortunately, it is necessary to manually change the file. Comments semicolon '';'' and colon '':'' must be replaced with ''$;'' and ''$:''. Also, Spi2a indicates a few statements as ''$ not implemented.''

```
$$$
```

```
+  I=I_VRL1
   Ind L2 6 16 13m                                                    $:
   Res RL2 16 0 1                                                     $:
   Cap C2 6 0 1.250u                                                  $:
   Res r61 6 10 1310                                                  $:
$  Not implemented                                                    $:
$  vrrC1 10 11 DC 0                                                   $:
   Volt vrrC1 10 11 DC={vrrC1=0}
+  I=IC1
   Cap C1 11 0 0.017u
$  Not implemented                                                    $:
$  vrr10 10 1 DC 0                                                    $:
   Volt vrr10 10 1 DC={vrr10=0}                                       $:
+  I=IRNL
*  $: non-linear circuit$;  $:-------------------------------------$:
   Diode d13 1 3 MODEL="n4148"                                        $:
   Diode d21 2 1 MODEL="n4148"                                        $:
   Res rm9 2 22 47k                                                   $:
   Volt vrm9 22 0 DC={vrm9=-9} $ $; negative power supply             $:
+  I=I_vrm9
   Res rp9 3 33 47k                                                   $:
$  Not implemented                                                    $:
$  vrp9 33 0 DC +9                                                    $:
   Volt vrp9 33 0 DC={vrp9=9} $ +9 must be 9
   Res r20 2 0 3.3k                                                   $:
   Res r30 3 0 3.3k                                                   $:
   VCVS evop 4 0 1 1 5 [1e+20] LINEAR                                 $:
   Res r14 1 4 290                                                    $:
   Res r54 5 4 290                                                    $:
   Res r50 5 0 1.2k                                                   $:
$$ Analysis commands                                                  $:
$$ .TRAN 0.05m 200m 0 0.018m UIC                                      $:
$ Sweep "TRAN Analysis 1"
$+ LOOP (1+(200m-(0))/(0.05m)) TIME LIN 0 200m TMAX=0.018m
$+ NW=1 $ UIC                                                         $:
$$ .plot tran v(11)                                                   $:
$ Show Y Vtran(11) $                                                  $:
$ EndSweep
$ the following lines are added and the sweep above is commented
Sweep "TRAN Analysis 2"
+ LOOP (4001) TIME LIN 0 200m TMAX=0.018m
$+ NW=1 $ UIC                                                         $:
$ .plot  tran v(11)                                                   $:
Show Y  Itran(IC1) X Vtran(11) $                                      $:
EndSweep
$.probe                                                               $:
```

The result of the analysis is presented in Figure 12.2. It is observed that limit cycle behavior is not obtained in the APLAC analysis in the time interval from 0 to 200 ms.

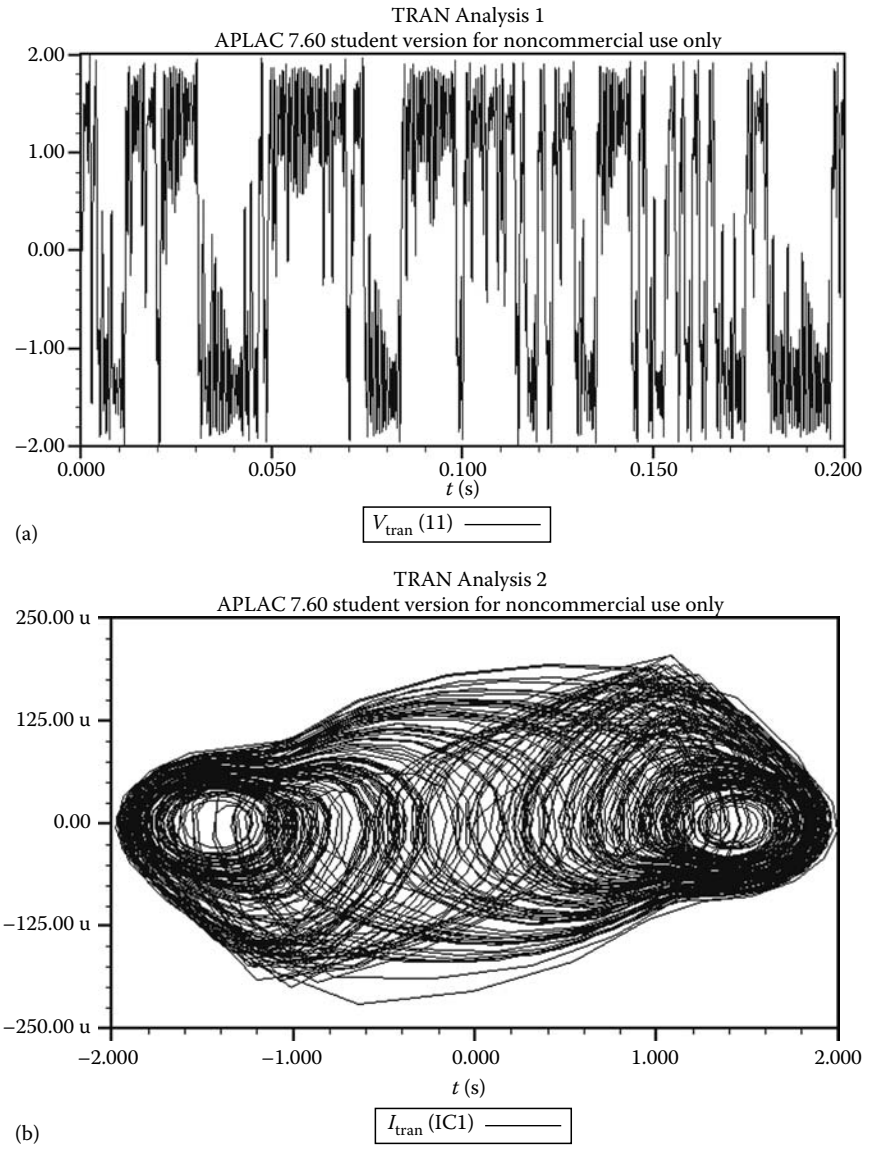


FIGURE 12.2 (a) The voltage of C1: $V(11)$ as function of time in the interval 0–200 ms. (b) The current of C1: $I(C1)$ as function of the voltage of C1: $V(11)$.

12.3.3 NAP2

The first versions of NAP2 (Nonlinear Analysis Program version 2) [10], based on the extended nodal equation formulation were developed in 1973 at the Institute of Circuit Theory and Telecommunication, Technical University of Denmark, Lyngby, Denmark.

NAP2 is a general-purpose circuit analysis program. Circuit models may contain resistors, conductors, capacitors, inductors, mutual inductors, ideal operational amplifiers, independent sources, controlled sources, and the most common semiconductor devices: diodes, bipolar junction transistors, and field effect transistors. NAP2 has only simple built-in models for the semiconductor devices, which require about 15 parameters. Besides the normal dc, ac, and transient analyses, the program can make parameter

variation analysis. Any parameter (e.g., component value or temperature) may be varied over a range in an arbitrary way and dc, ac, or transient analysis may be performed for each value of the parameter. Optimization of dc bias point (given: voltages, find: resistors) is possible. Event detection is included so that it is possible to interrupt the analysis when a certain signal, for example, goes from a positive to a negative value. The results may be combined into one output plot. It is also possible to calculate the poles and zeros of driving point and transfer functions for the linearized model in a certain bias point. Eigenvalue technique (based on the QR algorithm by J.G.F. Francis) is the method behind the calculation of poles and zeros. Group delay (i.e., the derivative of the phase with respect to the angular frequency) is calculated from the poles and zeros. This part of the program is available as an independent program named ANP3 (Analytical Network Program version 3).

In order to give an impression of the ''net-list'' input language, the syntax of the statements describing controlled sources is as follows:

```
Voltage Controlled Current Source:  Ixxx  N+  N−  VALUE VByyy
Voltage Controlled Voltage Source:  Vxxx  N+  N−  VALUE VByyy
Current Controlled Current Source:  Ixxx  N+  N−  VALUE IByyy
Current Controlled Voltage Source:  Vxxx  N+  N−  VALUE Ibyyy
```

where the initial characters of the branch name I and V indicate the type of the branch; *N*+ and *N*− are integers (node numbers) indicating the placement and orientation of the branch, respectively; and VALUE specifies the numerical value of the control, which may be a constant or an arbitrary functional expression in case of nonlinear control. IB and VB refer to the current or voltage of the branch, respectively, from where the control comes. If the control is the time derivative of the branch signal, SI or SV may be specified. Independent sources must be connected to a resistor R or a conductor G as follows: Rxxx N+ N− VALUE E = VALUE and Gxxx N+ N− VALUE J = VALUE, where VALUE may be any function of time, temperature, and components.

The input file ''net-list'' below describes the same analysis of the Chua oscillator circuit as performed by means of SPICE and APLAC. The circuit is a simple harmonic oscillator with losses (C2, L2, and RL2) loaded with a linear resistor (R61) in series with a capacitor (C1) in parallel with a nonlinear resistor. The circuit is excited by a sinusoidal voltage source through a coil L1. The frequency is specified as angular frequency in rps. It is possible to specify more than one statement on one line. Colon '':'' indicate start of a comment statement and semicolon '';'' indicates end of a statement. The greater than character ''>'' indicates continuation of a statement on the following line. It is observed that most of the lines are comment lines with the PSPICE input statements.

```
*circuit; *list 2, 9;  : file CRC-CHUA.NAP
*:  PSpice input file CRC-CHUA.CIR, first line, title line >                :
:   translated into NAP2 input file
: The Chua Oscillator, sinusoidal excitation, F = 150mV >                   :
:             RL2 = 1 ohm, RL1 = 0 ohm f = 1286.336389 Hz                   :
: ref. K. Murali and M. Lakshmanan,                                         :
:        Effect of Sinusoidal Excitation on the Chua's Circuit,             :
:        IEEE Transactions on Circuits and Systems − 1:                     :
:        Fundamental Theory and Applications,                               :
:        vol.39, No.4, April 1992, pp. 264-270                              :
: input source;  : ------------------------------------------               :
:  VRS   7  0    sin(0   150m    1.2863363889332e+3 0 0)                     :
  sin/sin/; Rs  7  0  0 e = 150m*sin(8.0822898994674e+3*time)                :
:choke   ;  L1   6    17     80mH;  RL1   17  7  0                          :
:  L1    6   17  80e-3       ;:mH                                           :
```

```
:  VRL1  17  7   DC  0        ;:ammeter for measure of IL1                  :
                              :-------------------------------------------:
:  harmonic oscillator;  L2  6    16   13mH;   RL2  16  0  1               :
                         C2  6    0    1.250uF                             :
:  L2    6   16  13m                                                        :
:  RL2   16  0   1                                                          :
:  C2    6   0   1.250u                                                     :
          :-------------------------------------------------------:
:  load;  r61   6  10  1310;  rrc1  10  11  0;    c1  11  0  0.017uF       :
       rr10  10   1  0:      irr10=current of nonlinear resistor           :
:  r61      6  10  1310                                                    :
:  vrrC1 10    11  DC  0                                                   :
:  C1      11   0  0.017u                                                  :
:  i(vrr10)=current of nonlinear resistor                                   :
:  vrr10   10   1  DC  0                                                   :
:  non-linear circuit;          :-------------------------------------:
:  .model  n4148  d     (is=0.1p rs=16 n=1);  vt=n*k*T/q                   :
   n4148   /diode/       is=0.1p gs=62.5m     vt=25mV;                     :
   td13  1 3 n4148;     td21  2  1  n4148;                                  :
:  d13   1  3  n4148                                                       :
:  d21   2  1  n4148                                                       :
   rm9   2  0   47k  e=-9; rp9 3 0 47k E=+9;                                :
:  rm9   2  22  47k                                                        :
:  vrm9  22 0   DC   -9; negative power supply                             :
:  rp9   3  33  47k                                                        :
:  vrp9  33 0   DC   +9                                                    :
   r20   2  0   3.3k; r30 3 0 3.3k;                                        :
:  r20   2  0   3.3k                                                       :
:  r30   3  0   3.3k                                                       :
:  ideal op. amp.;  :-----------------------------------------:
   gop   1  5   0;     vop 4 0 vgop: no  value means infinite value;:
:  evop  4  0   1      5     1e+20                                         :
   r14   1  4   290;  r54 5 4 290;  r50  5 0  1.2k;                        :
:  r14   1  4   290                                                        :
:  r54   5  4   290                                                        :
:  r50   5  0   1.2k                                                       :
:-------------------------------------------------------------------------:
   *time time 0 200m  :  variable order, variable step:
:  .TRAN 0.05m 200m  0  0.018m UIC:
   *tr vnall *plot(50  v6) v1 *probe:
:  .plot tran v(11)                                                        :
:  .probe                                                                  :
   *run cycle=15000 minstep=1e-20 >                                        :
        trunc=1e-3 step=50n:
:  .options acct nopage opts gmin=1e-15 reltol=1e-3                        :
:+ abstol=1e-12 vntol=1e-12 tnom=25 itl5=0                                 :
:+ limpts=15000                                                            :
:  .end                                                                    :
   *end                                                                    :
```

The program options are set by means of the statement *RUN, where, for example, the minimum integration step is set to 1e-20 s and the relative truncation error is set to 1e-6. The result of the analysis is presented in Figure 12.3. It can be observed that transition from chaotic behavior to a period 5 limit cycle takes place at about 50 ms. If we compare to the results obtained above by means of SPICE and APLAC,

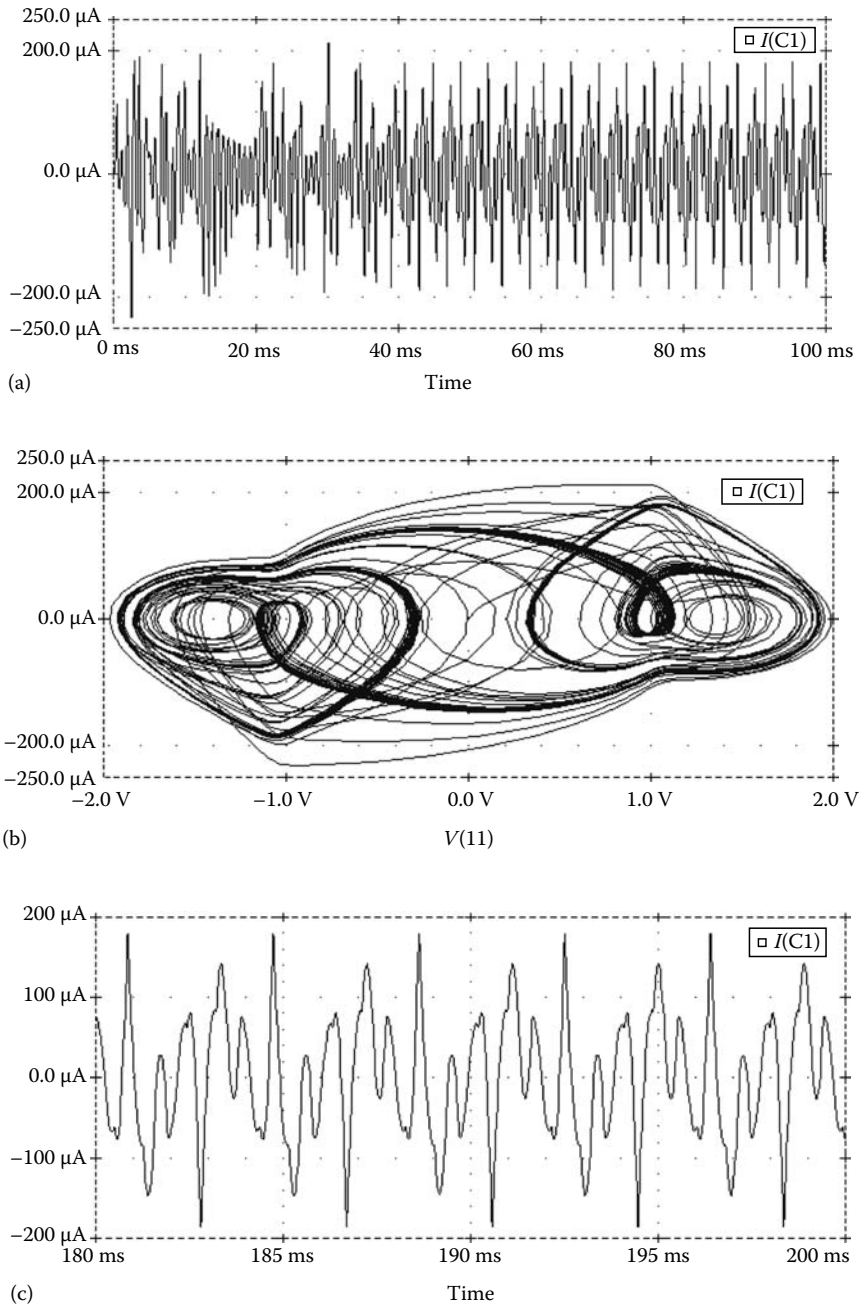


FIGURE 12.3 (a) NAP2 analysis. The current of C1: I(C1) as function of time in the interval 0–100 ms. (b) The current of C1: I(C1) as function of the voltage of C1: V(11) in the time interval 0–100 ms. (c) The current of C1: I(C1) as function of time in the interval 180–200 ms.

(*continued*)

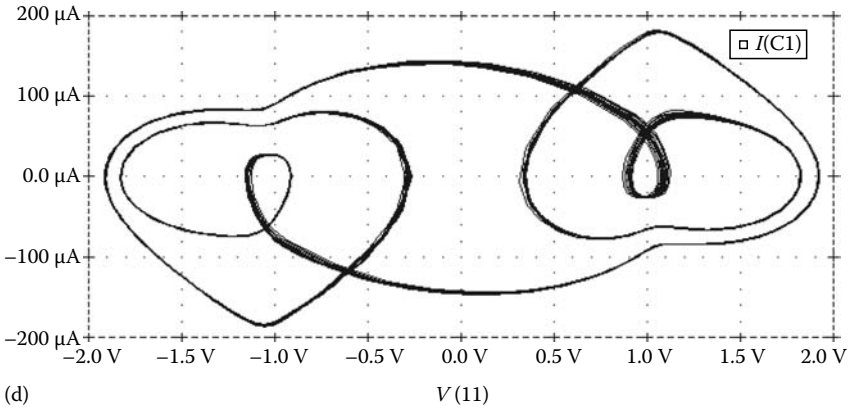


FIGURE 12.3 (continued) (d) The current of C1: I(C1) as function of the voltage of C1: V(11) in the time interval 100–200 ms.

we see that although the three programs are ''modeled and set'' the same way, for example, with the same relative tolerance 1e-3, the results are different due to the chaotic nature of the circuit and possibly also due to the different strategies of equation formulation and solution used in the three programs. For example, SPICE uses the trapezoidal integration method with variable step; APLAC and NAP2 use the Gear integration methods with variable order and variable step.

12.3.4 ESACAP

The first versions of ESACAP program based on the extended nodal equation formulation were developed in 1979 at Elektronik Centralen, Hoersholm, Denmark, for the European Space Agency as a result of a strong need for a simulation language capable of handling interdisciplinary problems (e.g., coupled electrical and thermal phenomena). ESACAP was therefore born with facilities that have only recently been implemented in other simulation programs (e.g., facilities referred to as behavioral or functional modeling).

ESACAP carries out analyses on nonlinear systems in dc and in the time domain. The nonlinear equations are solved by a hybrid method combining the robustness of the gradient method with the good convergence properties of the Newton–Raphson method. The derivatives required by the Jacobian matrix are symbolically evaluated from arbitrarily complex arithmetic expressions and are therefore exact. The symbolic evaluation of derivatives was available in the very first version of ESACAP. It has now become a general numerical discipline known as automatic differentiation. The time-domain solution is found by numerical integration implemented as BDFs of variable step and orders 1 through 6 (modified Gear method). An efficient extrapolation method (the epsilon algorithm) accelerates the asymptotic solution in the periodic steady-state case.

Frequency-domain analyses may be carried out on linear or linearized systems (e.g., after a dc analysis). Besides complex transfer functions, special outputs such as group delay and poles/zeros are available. The group delay is computed as the sum of the frequency sensitivities of all the reactive components in the system. Poles and zeros are found by a numerical interpolation of transfer functions evaluated on a circle in the complex frequency plane. ESACAP also includes a complex number postprocessor by means of which any function of the basic outputs can be generated (e.g., stability factor, s-parameters, complex ratios).

Sensitivities of all outputs with respect to all parameters are available in all analysis modes. The automatic differentiation combined with the adjoint network provides exact partial derivatives in the frequency domain. In the time domain, integration of a sensitivity network (using the already LUfactorized Jacobian) provides the partial derivatives as functions of time.

The ESACAP language combines procedural facilities, such as if-then-else, assignment statements, and do-loops, with the usual description by structure (nodes/branches). Arbitrary expressions containing system variables and their derivatives are allowed for specifying branch values thereby establishing any type of nonlinearity. System variables of nonpotential and noncurrent type may be defined and used everywhere in the description (e.g., for defining power, charge). The language also accepts the specification of nonlinear differential equations. Besides all the standard functions known from high-level computer languages, ESACAP provides a number of useful functions. One of the most important of these functions is the delay function. The delay function returns one of its arguments delayed by a specified value, which in turn may depend on system variables. Another important function is the threshold switch—the ZEROREF function—used in if-then-else constructs for triggering discontinuities. The ZEROREF function interacts with the integration algorithm that may be reinitialized at the exact threshold crossing. The ZEROREF function is an efficient means for separating cause and action in physical models thereby eliminating many types of causality problems. Causality problems are typical examples of bad modeling techniques and the most frequent reason for divergence in the simulation of dynamic systems.

Typical ESACAP applications include electronics as well as thermal and hydraulic systems. The frequency domain facilities have been a powerful tool for designing stable control systems including nonelectronics engineering disciplines.

In order to give an idea of the input language, the syntax of the statements describing sources is as follows:

```
Current  Source:  Jxxx(N+,  N−) = VALUE;
Voltage  Source:  Exxx(N+,  N−) = VALUE;
```

where the initial characters of the branch name: J and E indicate the type of the branch; $N+$ and $N-$ are node identifiers (character strings), which, as a special case, may be integer numbers (node numbers). The node identifiers indicate the placement and orientation of the branch. The VALUE specifies the numerical value of the source, which may be an arbitrary function of time, temperature, and parameters as well as system variables (including their time derivatives). Adding an apostrophe references the time derivative of a system variable. $V(\mathrm{N1,N2})'$, for example, is the time derivative of the voltage drop from node N1 to node N2.

The next input file—actually, a small program written in the ESACAP language—describes an analysis of a tapered transmission line. The example shows some of the powerful tools available in the ESACAP language such as (1) the delay function, (2) the do-loop, and (3) the sensitivity calculation. The description language of ESACAP is a genuine simulation and modeling language. However, for describing simple systems, the input language is just slightly more complicated than the languages of SPICE, APLAC, and NAP2. Data are specified in a number of blocks (“chapters” and “sections”) starting with $$ and $. Note how the line model is specified in a do-loop where ESACAP creates nodes and branches of a ladder network [11].

```
Example.155 Tapered line in the time domain.
# Calculation of sensitivities.
# This example shows the use of a do-loop for the discretization of a
# tapered transmission line into a chain of short line segments. The
# example also demonstrates how to specify partial derivatives of any
# parameter for sensitivity calculations.
$$DESCRIPTION # chapter - - - - - - - - - - - - - - - - - - - - - - - - -
$CON: n_sections=60; END; # section- - - - - - - - - - - - - - - - - - -
# n_sections is defined as a globally available constant.
# Only this constant needs to be modified in order to change
# the resolution of discretization
```

```
# Transmission line specified by characteristic impedance and length.
# Modelled by the ESACAP delay function (DEL).
$MODEL: LineCell(in,out): Z0,length;
 delay=length/3e8;
 J_reverse(0,in)=DEL(2*V(out)/Z0-I(J_forward), delay);
 J_forward(0,out)=DEL(2*V(in)/Z0-I(J_reverse), delay);
 G1(in,0)=1/Z0; G2(out,0)=1/Z0;
END; # end of section- - - - - - - - - - - - - - - - - -
# Tapered line specified by input and output impedance and length
# This model calls LineCell n_sections times in a do-loop.
$MODEL: TaperedLine(in,out): Z1,Z2,length;
 ALIAS_NODE(in,1);                  # Let node in and 1 be the same.
 ALIAS_NODE(out,[n_sections+1]);  # Let node out and n_sections+1 be
                                    # the same.
# Notice that values in square brackets become part of an identifier
FOR (i=1, n_sections) DO
X[i]([i],[i+1])=LineCell(Z1+i*(Z2-Z1)/n_sections, length/n_sections);
ENDDO;
END; # end of section - - - - - - - - - - - - - - - - - -
# Main network calls the model of the tapered line and terminates
# it by a 50 ohm source and 100 ohm load.
$NETWORK: # section- - - - - - - - - - - - - - - - - - - -
IF(TIME.LT.1n) THEN
 Esource(source,0)=0;
ELSE
 Esource(source,0)=1;
ENDIF;
# Esource(source,0)=TABLE(TIME,(0,0),(1n,0),(1.001n,1),(10n,1));
 Rsource(source,in)=50;
 Rload(out,0)=100;
 Z1=50; Z2=100; length=1;
 X1(in,out)=TaperedLine(Z1,Z2,length);
END;  #  end of section - - - - - - - - - - - - - - - - - - - - -
# Time-domain analysis
$$TRANSIENT  #  chapter- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# Analysis parameters
$PARAMETERS:  #  section- - - - - - - - - - - - - - - - - - - - - - - - - - - -
 TIME=0,20n;                        # Total time sweep
 HMAX=2p;                           # Max integration step
END;  #  end of section- - - - - - - - - - - - - - - - - - - - - -
# Specification of desired results. Adding an exclamation mark (!) to an
# output will show the value on the ESACAP real-time graphics display.
$DUMP:  #  section- - - - - - - - - - - - - - - - - - - - -
 FILE=<dump.155>; TIME=0,20n,20p;
 TIME; V(in)!; V(out)!;
 (V(in),DER(Z1))!;              # Partial derivatives with respect
 (V(out),DER(Z1))!;             # to Z1
END;  #  end of section - - - - - - - - - - - - - - - - - - -
$$STOP  #  chapter- - - - - - - - - - - - - - - - - - - - - - - - -
```

The result of the analysis is presented in Figure 12.4.

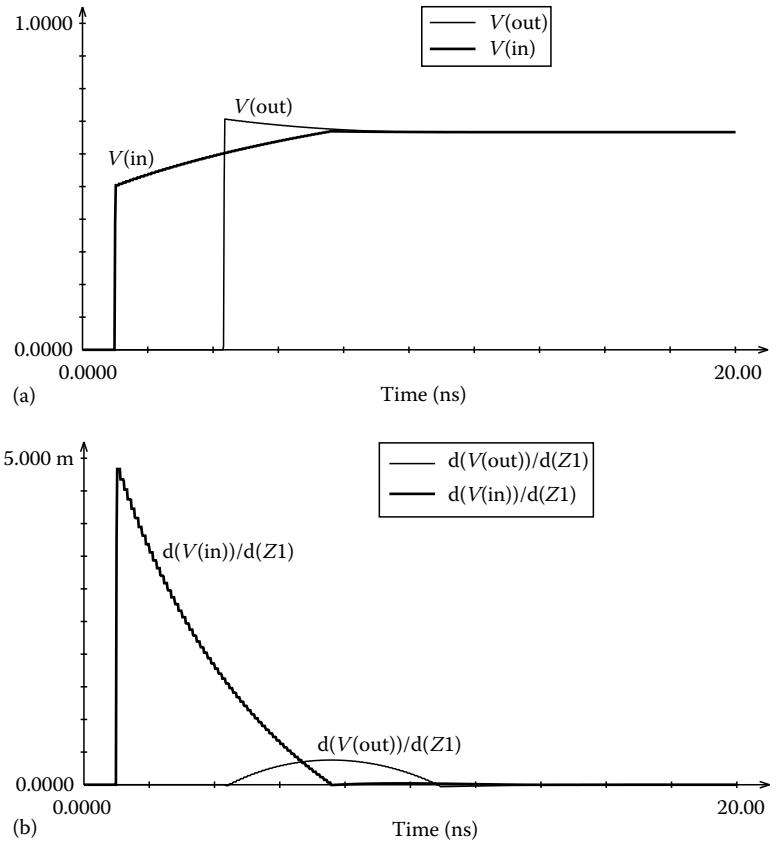


FIGURE 12.4 (a) ESACAP analysis. The input voltage of the tapered line: V(in) and the output voltage of the tapered line: V(out) as functions of time in the interval from 0 to 20 ns. (b) The sensitivities of V(in) and V(out) with respect to Z1.

12.3.5 DYNAST

DYNAST [7] was developed in 1992 in a joint venture between the Czech Technical University, Prague, the Czech Republic and Katholieke Universiteit Leuven, Heverlee, Belgium. The program was developed as an interdisciplinary simulation and design tool in the field of "mechatronics" (mixed mechanical/electrical systems).

The main purpose of DYNAST is to simulate dynamic systems decomposed into subsystems defined independently of the system structure. The structure can be hierarchical. DYNAST is a versatile software tool for modeling, simulation, and analysis of general linear as well as nonlinear dynamic systems, both in time and frequency domain. Semisymbolic analysis is possible (poles and zeros of network functions, inverse Laplace transformation using closed-form formulas).

Three types of subsystem models are available in DYNAST. The program admits systems descriptions in the form of (1) a multipole diagram respecting physical laws, (2) a causal or an acausal block diagram, (3) a set of equations, or (4) in a form combining the above approaches.

1. In DYNAST the physical-level modeling of dynamic systems is based on subsystem multipole models or multiterminal models. These models respect the continuity and compatibility postulates that apply to all physical energy-domains. (The former postulate corresponds to the laws of conservation of energy, mass, electrical charge, etc.; the latter is a consequence of the system

connectedness.) The multipole poles correspond directly to those subsystem locations in which the actual energetic interactions between the subsystems take place (such as shafts, electrical terminals, pipe inlets, etc.). The interactions are expressed in terms of products of complementary physical quantity pairs: the through variables flowing into the multipoles via the individual terminals, and the across variables identified between the terminals.

2. The causal blocks, specified by explicit functional expressions or transfer functions, are typical for any simulation program. But the variety of basic blocks is very poor in DYNAST, as its language permits definition of the block behavior in a very flexible way. Besides the built-in basic blocks, user specified multi-input multi-output macroblocks are available as well. The causal block interconnections are restricted by the rule that only one block output may be connected to one or several block inputs. In the DYNAST block variety, however, causal blocks are also available with no restrictions imposed on their interconnections, as they are defined by implicit-form expressions.
3. DYNAST can also be used as an equation solver for systems of nonlinear first-order algebro-differential and algebraic equations in the implicit form. The equations can be submitted in a natural way (without converting them into block diagrams) using a rich variety of functions including the Boolean, event-dependent, and tabular ones. The equations, as well as any other input data, are directly interpreted by the program without any compilation.

The equation formulation approach used for both multipoles and block diagrams evolved from the extended method of nodal voltages (MNA) developed for electrical systems. Because all the equations of the diagrams are formulated simultaneously, no problems occur with the *algebraic loops*. As the formulated equations are in the implicit form, it does not create any problems with the *causality* of the physical models.

The integration method used to solve the nonlinear algebro-differential and algebraic equations is based on a stiff-stable implicit backward-differentiation formula (a modified Gear method). During the integration, the step length as well as the order of the method is varied continuously to minimize the computational time while respecting the admissible computational error. Jacobians necessary for the integration are computed by symbolic differentiation. Their evaluation as well as their LU decomposition, however, is not performed at each iteration step if the convergence is fast enough. Considerable savings of computational time and memory are achieved by a consistent matrix sparsity exploitation.

To accelerate the computation of periodic responses of weakly damped dynamic systems, the iterative epsilon-algorithm is utilized. Also, fast-Fourier transformation is available for spectral analysis of the periodic steady-state responses.

DYNAST runs under DOS- or WINDOWS-control on IBM-compatible PCs. Because it is coded in FORTRAN 77 and C-languages, it is easily implemented on other platforms. It is accompanied by a menu-driven *graphical environment*. The block and multiport diagrams can be submitted in a graphical form by a schematic capture editor. DYNAST can be easily augmented by various pre- and postprocessors because all its input and output data are available in the ASCII code. Free ''net-list'' access to DYNAST is possible by means of e-mail or online over the Internet [7].

References

1. Calahan, D. A., *Computer-Aided Network Design*. New York: McGraw-Hill, 1972.
2. Chua, L. O. and P. M. Lin, *Computer-Aided Analysis of Electronic Circuits*. Englewood Cliffs, NJ: Prentice Hall, 1975.
3. Dertouzos, M. L. et al., *Systems, Networks, and Computation: Basic Concepts*. New York: McGraw-Hill, 1972.
4. Ostrowski, A. M., *Solution of Equations and Systems of Equations*. New York: Academic Press, 1966.

5. Valtonen, M. et al., *APLAC—An Object-Oriented Analog Circuit Simulator and Design Tool*. Espoo, Finland: Circuit Theory Lab., Helsinki University of Technology and Nokia Corporation Research Center, 1992, http://www.aplac.hut.fi/aplac/general.html, http://www.aplac.com/
6. Intusoft, *IsSpice3—ICAPS System Packages*. San Pedro, CA: Intusoft, 1994, http://www.intusoft.com/
7. Mann, H., *DYNAST—A Multipurpose Engineering Simulation Tool*. Prague, Czech Republic: The Czech Technical University, 1994, http://www.it.dtu.dk/ecs/teacher.htm, http://icosym.cvut.cz/cacsd/msa/onlinetools.html, http://icosym.cvut.cz/dyn/download/public/
8. Meta-Software, *HSPICE User's Manual H9001*. Campbell, CA: Meta-Software, 1990.
9. MicroSim, *PSpice—The Design Center*. Irvine, CA: MicroSim, 1994, http://www.cadencepcb.com and http://www.pspice.com/
10. Rübner-Petersen, T., *NAP2—A Nonlinear Analysis Program for Electronic Circuits*, Version 2, Users Manual 16/5-73, Report IT-63, ISSN-0105-8541. Lyngby, Denmark: Institute of Circuit Theory and Telecommunication, Technical University of Denmark, 1973, http://www.it.dtu.dk/ecs/programs htm#nnn, http://www.it.dtu.dk/ecs/napanp.htm
11. Stangerup, P., *ESACAP User's Manual*. Nivaa, Denmark: StanSim Research Aps., 1990, http://www.it.dtu.dk/ecs/esacap.htm
12. Vlach, J. and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand Reinhold, 1983.
13. Funk, D. G. and D. Christiansen (Eds.), *Electronic Engineers' Handbook*, 3rd ed. New York: McGraw-Hill, 1989.

13

Cellular Neural Networks and Cellular Wave Computers

Tamás Roska
Hungarian Academy of Sciences and Pázmány Péter Catholic University

Ákos Zarándy
Hungarian Academy of Sciences and Pázmány Péter Catholic University

Csaba Rekeczky
Eutecus Inc. and Pázmány Péter Catholic University

13.1 Introduction: Definition and Classification **13**-1
Typical CNN Models
13.2 Simple CNN Circuit Structure **13**-3
13.3 Stored Program CNN Universal Machine and the Analogic Supercomputer Chip **13**-6
13.4 Applications **13**-8
Image Processing—Form, Motion, Color, and Depth • Partial Differential Equations • Relation to Biology
13.5 Template Library: Analogical CNN Algorithms **13**-11
13.6 Recent Advances **13**-16
13.7 Recent Developments and Outlook **13**-17
References **13**-18

13.1 Introduction: Definition and Classification

Current very-large-scale integration (VLSI) technologies provide for the fabrication of chips with several million transistors. With these technologies a single chip may contain one powerful digital processor, a huge memory containing millions of very simple units placed in a regular structure, and other complex functions. A powerful combination of a simple logic processor placed in a regular structure is the cellular automaton invented by John von Neumann. The cellular automaton is a highly parallel computer architecture. Although many living neural circuits resemble this architecture, the neurons do not function in a simple logical mode: they are analog ''devices.'' The cellular neural network architecture, invented by Chua and his graduate student Yang [1], has both the properties: the cell units are nonlinear continuous-time dynamic elements placed in a cellular array. Of course, the resulting nonlinear dynamics in space could be extremely complex. The inventors, however, showed that these networks can be designed and used for a variety of engineering purposes, while maintaining stability and keeping the dynamic range within well-designed limits. Subsequent developments have uncovered the many inherent capabilities of this architecture (IEEE conferences: CNNA-90, CNNA-92, CNNA-94, 96, 98, 00, 02; Special issues: *International Journal of Circuit Theory and Applications*, 1993, 1996, 1998, 2002; and *IEEE Transactions on Circuits and Systems*, *I and II*, 1993, 1999, etc.). In the circuit implementation, unlike analog computers or general neural networks, the cellular neural/nonlinear network (CNN) cells are not the ubiquitous high-gain operational amplifiers. In most practical cases, they are either simple unity-gain amplifiers or simple second- or third-order simple dynamic circuits with one to two simple nonlinear components. Tractability

in the design and the possibility for exploiting the complex nonlinear dynamic phenomena in space, as well as the trillion operations per second (TeraOPS) computing speed in a single chip are but some of the many attractive properties of cellular neural networks. The trade-off is in the accuracy; however, in many cases, the accuracy achieved with current technologies is enough to solve a lot of real-life problems.

The CNN is a new paradigm for multidimensional, nonlinear, dynamic processor arrays [1,2]. The mainly uniform processing elements, called cells or artificial neurons, are placed on a regular geometric grid (with a square, hexagonal, or other pattern). This grid may consist of several two-dimensional (2-D) layers packed upon each other (Figure 13.1). Each processing element or cell is an analog dynamical system, the state (x), the input (u), and the output (y) signals are analog (real-valued) functions of time (both continuous-time and discrete-time signals are allowed). The interconnection and interaction pattern assumed at each cell is mainly local within a neighborhood N_r, where N_r denotes the first "r" circular layers of surrounding cells. Figure 13.2 shows a 2-D layer with a square grid of interconnection

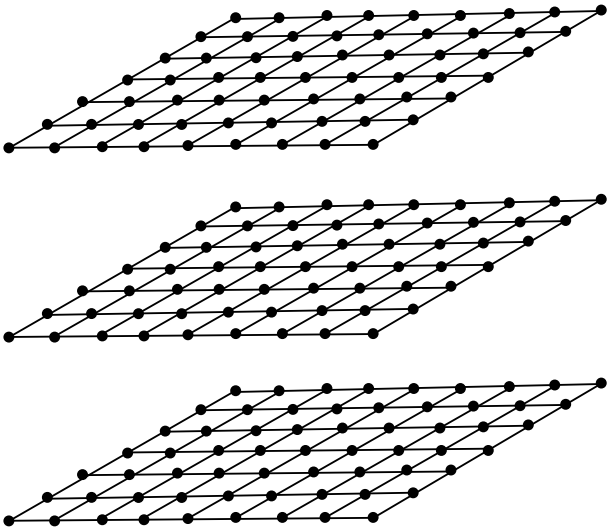


FIGURE 13.1 CNN grid structure with the processing elements (cells) located at the vertices.

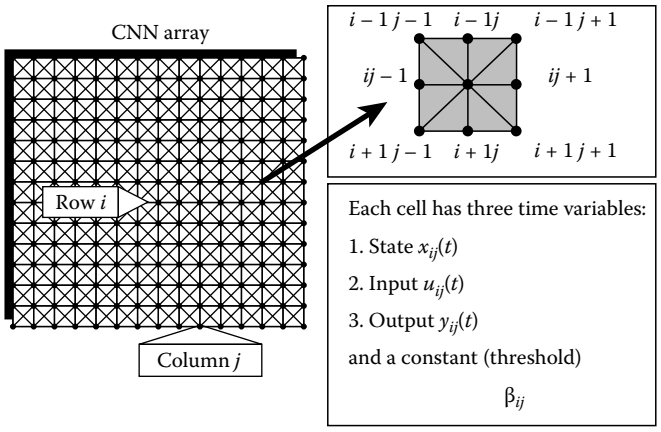


FIGURE 13.2 A single, 2-D CNN layer and a magnified cell with its neighbor cells with the normal neighborhood radius $r = 1$.

radius of 1 (nearest neighborhood). Each vertex contains a cell and the edges represent the interconnections between the cells. The pattern of interaction strengths between each cell and its neighbors is the ''program'' of the CNN array. It is called a cloning template (or just template).

Depending on the types of grids, processors (cells), interactions, and modes of operation, several classes of CNN architectures and models have been introduced. Although the summary below is not complete, it gives an impression of vast diversities.

13.1.1 Typical CNN Models

1. Grid type
   - Square
   - Hexagonal
   - Planar
   - Circular
   - Equidistant
   - Logarithmic
2. Processor type
   - Linear
   - Sigmoid
   - First-order
   - Second-order
   - Third-order
3. Interaction type
   - Linear memoryless
   - Nonlinear
   - Dynamic
   - Delay-type
4. Mode of operation
   - Continuous-time
   - Discrete-time
   - Equilibrium
   - Oscillating
   - Chaotic

13.2 Simple CNN Circuit Structure

The simplest first-order dynamic CNN cell used in the seminal paper [1] is illustrated in Figure 13.3. It is placed on the grid in the position *ij* (row *i* and column *j*). It consists of a single state capacitor with a parallel resistor and an amplifier $[f(x_{ij})]$. This amplifier is a voltage-controlled current source (VCCS), where the controlling voltage is the state capacitor voltage. To make the amplifier model self-contained, a parallel resistor of unit value is assumed to be connected across the output port. Hence, the voltage transfer characteristic of this amplifier is also equal to $f(\cdot)$. In its simplest form this amplifier has a unity-gain saturation characteristic (see Figure 13.7 for more details).

The aggregate feedforward and feedback interactions are represented by the current sources i_{input} and i_{output}, respectively. Figure 13.4 shows these interactions in more detail. In fact, the feedforward interaction term i_{input} is a weighted sum of the input voltages (u_{kl}) of all cells in the neighborhood (N_r). Hence, the feedforward template, the so-called $\boldsymbol{B}$ template, is a small matrix of size $(2r+1) \times (2r+1)$ containing the template elements b_{kl}, which can be implemented by an array of linear VCCSs. The controlling voltages of these controlled sources are the input voltages of the cells within the neighborhood of radius r. This means, for example, that b_{12} is the VCCS controlled by the input voltage of the cell lying

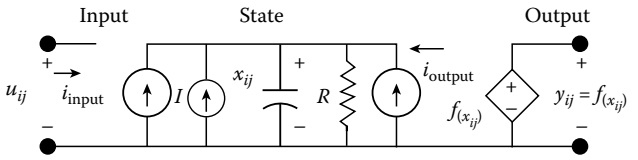


FIGURE 13.3 Simple first-order CNN cell.

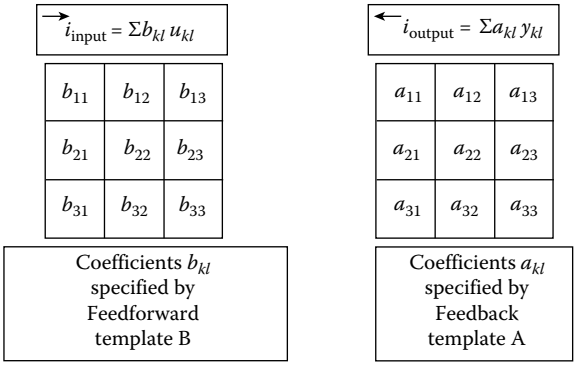


FIGURE 13.4 The 19 numbers (a program) that govern the CNN array (the 19th number is the constant bias term I, but it is not shown in the figure) define the cloning template (***A***, ***B***, and ***I***).

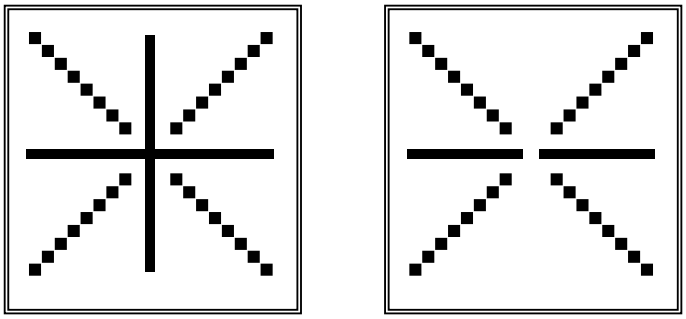
north from the cell ij. In most practical cases the ***B*** template is translation invariant, i.e., the interaction pattern (the ***B*** template) is the same for all cells. Hence, the chip layout will be very regular (as in memories or PLAs). The feedback interaction term i_{output} is a weighted sum of the output voltages (y_{kl}) of all cells in the neighborhood (N_r). The weights are the elements of a small matrix ***A*** called the ***A*** template (or feedback template). Similar arguments apply for the ***A*** template as for the ***B*** template discussed previously. If the constant threshold term is translation invariant as denoted by the constant current source ***I***, then in the case of $r = 1$, the complete cloning template contains only 19 numbers (***A*** and ***B*** and ***I***, i.e., $9 + 9 + 1$ terms), irrespective of the size of the CNN array. These 19 numbers define the task which the CNN array can solve.

What kind of tasks are we talking about? The simplest, and perhaps the most important, are image-processing tasks. In the CNN array computer, the input and output images are coded as follows. For each picture element (called pixel) in the image, a single cell is assigned in the CNN. This means that a one-to-one correspondence exists between the pixels and the CNN cells. Voltages in the CNN cells code the grayscale values of the pixels. Black is coded by $+1$ V, white is -1 V, and the grayscale values are in between. Two independent input images can be defined pixel-by-pixel: the input voltages u_{ij} and the initial voltage values of the capacitors x_{ij} (0) (cell-by-cell). Placing these input images onto the cell array and starting the transient, the steady-state outputs y_{ij} will encode the output image. The computing time is equal to the settling time of the CNN array. This time is below 1 μs using a CNN chip made with a 1.0–1.5 μm technology containing thousands of CNN processing elements, i.e., pixels, in an area of about 2 cm^2. This translates to a computing power of several 100 billion operations per second (GXPS). The first tested CNN chip [3] was followed by several others implementing a discrete-time CNN model [4] and chips with on-chip photosensors in each cell [5].

For example, if we place the array of voltage values defined by the image shown in Figure 13.5b as the input voltage and the initial state capacitor voltage values in the CNN array with the cloning template shown in Figure 13.5a, then after the transients have settled down, the output voltages will encode the output image of Figure 13.5c. Observe that the vertical line has been deleted. Since the image

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & -0.25 & 0 \\ 0 & 0 & 0 \\ 0 & -0.25 & 0 \end{bmatrix}, \quad \boldsymbol{I} = -1.5$$

(a)



(b) (c)

FIGURE 13.5 An input and output image where the vertical line was deleted.

contains 40×40 pixels, the CNN array contains 40×40 cells. It is quite interesting that if we had more than one vertical line, the computing time would be the same. Moreover, if we had an array of 100×100 cells on the chip, the computing time would remain the same as well. This remarkable result is due to the fully parallel nonlinear dynamics of the CNN computer. Some propagating-type templates induce wavelike phenomena. Their settling times increase with the size of the array.

For other image-processing tasks, processing form, motion, color, and depth, more than 100 cloning templates have been developed to date and the library of new templates is growing rapidly. Using the Cellular Neural Network Workstation Tool Kit [6], they can be called in from a CNN template library (CTL). New templates are being developed and published continually.

The dynamics of the CNN array is described by the following set of differential equations:

$$\begin{aligned} \mathrm{d}x_{ij}/\mathrm{d}t &= -x_{ij} + I + i_{\text{output}} + i_{\text{input}} \\ y_{ij} &= f(x_{ij}) \\ i &= 1, 2, \ldots, N \quad \text{and} \quad j = 1, 2, \ldots, M \text{ (the array has } N \times M \text{ cells)} \end{aligned}$$

where the last two terms in the state equation are given by the sums shown in Figure 13.4.

We can generalize the domain covered by the original CNN defined via linear and time-invariant templates by introducing the "nonlinear" templates (denoted by "$\wedge$") and the "delay" templates (indicated by τ in the superscript) as well, to obtain the generalized state equation shown below. The unity-gain nonlinear sigmoid characteristics f are depicted in Figure 13.6.

$$\begin{aligned} \frac{\mathrm{d}vx_{ij}}{\mathrm{d}t} = -vx_{ij} + I_{ij} &+ \sum_{kl \in N_r(ij)} \hat{A}_{ij;kl}\,(vy_{kl}(t), vy_{ij}(t)) + \sum_{kl \in N_r(ij)} \hat{B}_{ij;kl}(vu_{kl}(t), vu_{ij}(t)) \\ &+ \sum_{kl \in N_r(ij)} A^{\tau}_{ij;kl} vy_{kl}(t - \tau) + \sum_{kl \in N_r(ij)} B^{\tau}_{ij;kl} vu_{kl}(t - \tau) \end{aligned}$$

Several strong results have been proved that assure stable and reliable operations. If the $\boldsymbol{A}$ template is symmetric, then the CNN is stable. Several other results have extended this condition [4,7]. The sum of the absolute values of all the 19 template elements plus 1 defines the dynamic range within which the

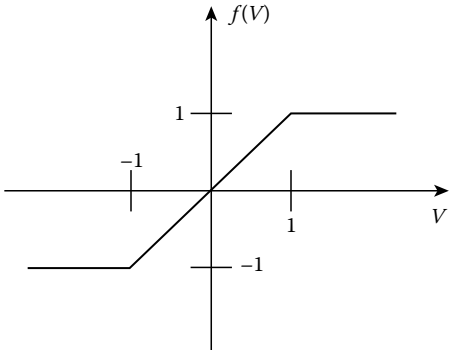


FIGURE 13.6 The simple unity-gain sigmoid characteristics.

- 1-, 2-, 3-, or *n*-dimensional *array* of mainly identical *dynamical systems*, called cells or processor units, which satisfies two properties:
  - Most *interactions are local* within a finite radius *r*
  - All *state variables are continuous valued* signals

FIGURE 13.7 CNN definition.

state voltage remains bounded during the entire transient, if the input and initial state signals are <1 V in absolute value [1].

In a broader sense, the CNN is defined [8] as shown in Figure 13.7, see also Ref. [32].

13.3 Stored Program CNN Universal Machine and the Analogic Supercomputer Chip

For different tasks, say image-processing, we need different cloning templates. If we want to implement them in hardware, we would need different chips. This is inefficient except for dedicated, mass-production applications.

The invention of the CNN universal machine [9] has overcome the problem above. It is the first stored-program array computer with analog nonlinear array dynamics. One CNN operation, for example, solving thousands of nonlinear differential equations in a microsecond, is just one single instruction. In addition, a single instruction is represented by just a few analog (real) values (numbers). In the case when the nearest neighborhood is used, only 19 numbers are generated. When combining several CNN templates, for example, extracting first contours in a grayscale image, then detecting those areas where the contour has holes, etc., we have to design a flowchart logic that satisfies the correct sequence of the different templates. The simple flowchart for the previous example is shown in Figure 13.8. One key point is that, in order to exploit the high speed of the CNN chips, we have to store the intermediate results cell-by-cell (pixel-by-pixel). Therefore, we need a local analog memory (LAM). By combining several template actions we can write more complex flowcharts for implementing almost any analogic algorithms. The name analogic is an acronym for ''analog and logic.'' It is important to realize that analogic computation is completely different from hybrid computing. To cite just one point, among others, no A/D or D/A conversions occur during the computation of an analogic program. As with digital microprocessors, to control the execution of an analogic algorithm, we need a global programming unit. The global architecture of the CNN universal machine is shown in Figure 13.9.

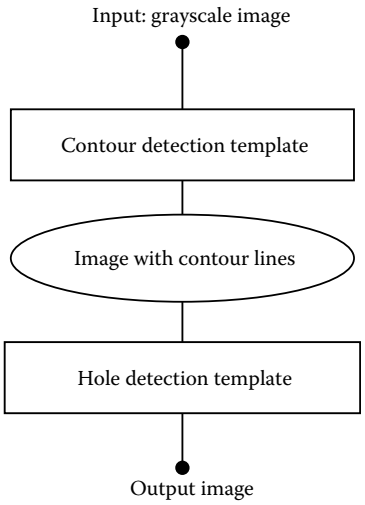


FIGURE 13.8 Flowchart representing the logic sequence of two templates.

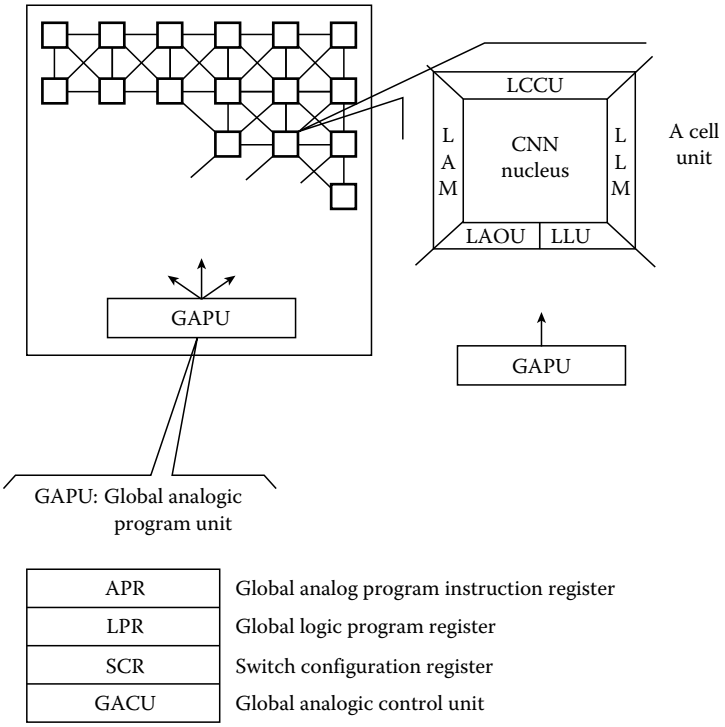


FIGURE 13.9 Global architecture of the CNN universal machine.

As we can see from this figure, the CNN nucleus described in Section 13.2 has been generalized to include several crucial functions depicted in the periphery. We have already discussed the role of the LAM that provides the local (on-chip) storage of intermediate analog results. Because the results of many detection tasks in applications involve only black-and-white logic values, adding a local logic memory (LLM) in each cell is crucial. After applying several templates in a sequence, it is often necessary to combine their results. For example, to analyze motion, consecutive snapshots processed by CNN templates are compared. The local analog output unit (LAOU) and the local logic unit (LLU) perform these tasks, both on the local analog (gray scale) and the logical (black-and-white) values. The local communication and control unit (LCCU) of each cell decodes the various instructions coming from the global analogic program unit (GAPU).

The global control of each cell is provided by the GAPU. It consists of four parts:

1. Analog program (instruction) register (APR) stores the CNN template values (19 values for each CNN template instruction in the case of nearest interconnection). The templates stored here will be used during the run of the prescribed analogical algorithm.
2. Global logic program register (LPR) stores the code for the LLUs.
3. Flexibility of the extended CNN cells is provided by embedding controllable switches in each cell. By changing the switch configurations of each cell simultaneously, we can execute many tasks using the same cell. For example, the CNN program starts by loading a given template, storing the results of this template action in the LAM, placing this intermediate result back on the input to prepare the cell, starting the action with another template, etc. The switch configurations of the cells are coded in the switch configuration register (SCR).
4. Finally, the heart of the GAPU is the global analogic control unit (GACU), which contains the physical machine code of the logic sequence of analogical algorithm. It is important to emphasize that here the control code is digital; hence, although its internal operation is analog and logical, a CNN universal chip can be programmed with the same flexibility and ease as a digital microprocessor—except the language is much simpler. Indeed, a high-level language, a compiler, an operating system, and an algorithm development system are available for CNN universal chip architectures. Moreover, by fabricating optical sensors cell-by-cell on the chip [5], the image input is directly interfaced.

The CNN universal chip is called supercomputer chip because the execution speed of an analogic algorithm falls in the same range as the computing power of today's average digital supercomputers (a TeraOPS). Another reason for this enormous computing power is that the reprogramming time of a new analog instruction (template) is of the same order, or less, than the analog array execution time (less than a microsecond). This is about 1 million times faster than some fully interconnected analog chips.

Based on the previously mentioned novel characteristics, the CNN universal chip can be considered to be an analogic microprocessor.

13.4 Applications

In view of its flexibility and its very high speed in image-processing tasks, the CNN universal machine is ideal for many applications. In the following, we briefly describe three areas. For more applications, the reader should consult the references at the end of this chapter.

13.4.1 Image Processing—Form, Motion, Color, and Depth

Image processing is currently the most popular application of CNN. Of the more than 100 different templates currently available, the vast majority are for image-processing tasks. Eventually, we will have templates for almost all conceivable local image-processing operations. Form (shape), motion, color, and depth can all be ideally processed via CNN. The interested reader can find many examples and

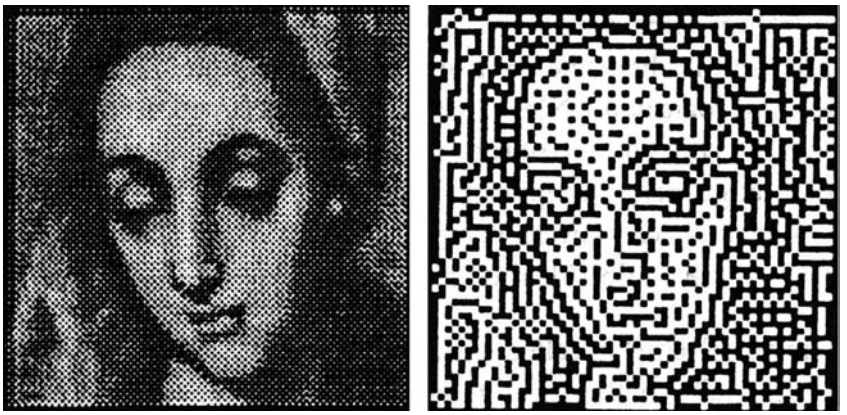


FIGURE 13.10 Halftoning: an original grayscale image (LHS) and its halftoned version (RHS). A low resolution is deliberately chosen in (b) in order to reveal the differing dot densities at various regions of the image.

applications in the references. CNN handles analog pixel values, so grayscale images are processed directly.

Many templates detect simple features like different types of edges, convex or concave corners, lines with a prescribed orientation, etc. Other templates detect semiglobal features like holes, groups of objects within a given size of area, or delete objects smaller than a given size. There are also many CNN global operations like calculating the shadow, histogram, etc. Halftoning is commonly used in fax machines, laser printers, and newspapers. In this case, the local gray level is represented by black dots of identical size, whose density varies in accordance with the gray level. CNN templates can do this job as well. A simple example is shown in Figure 13.10. The original grayscale image is shown on the left-hand side, the halftoned image is shown on the right-hand side. The ''smoothing'' function of our eye completes the image-processing task.

More complex templates detect patterns defined within the neighborhood of interaction. In this case, the patterns of the $\boldsymbol{A}$ and $\boldsymbol{B}$ templates somehow reflect the pattern of the object to be detected.

Because the simplest templates are translation invariant, the detection or pattern recognition is translation invariant as well. By clever design, however, some rotationally invariant detection procedures have been developed as well.

Combining several templates according to some prescribed logic sequence, more complex pattern detection tasks can be performed, e.g., halftoning.

Color-processing CNN arrays represent the three basis colors by single layers via a multilayer CNN. For example, using the red-green-blue (RGB) representation in a three-layer CNN, simple color-processing operations can be performed. Combining them with logic, conversions between various color representations are possible.

One of the most complex tasks that has been undertaken by an analogic CNN algorithm is the recognition of bank notes. Recognition of bank notes in a few milliseconds is becoming more and more important. Recent advances in the copy machine industry have made currency counterfeiting easier. Therefore, automatic bank note detection is a pressing need. Figure 13.11 shows a part of this process (which involves color processing as well). The dollar bill shown in the foreground is analyzed and the circles of a given size are detected (colors are not shown). The ''color cube'' means that each color intensity is within prescribed lower and upper limit values.

Motion detection can be achieved by CNN in many ways. One approach to process motion is to apply two consecutive snapshots to the input and the initial state of the CNN cell. The CNN array calculates the various combinations between the two snapshots. The simplest case is just taking the difference to detect

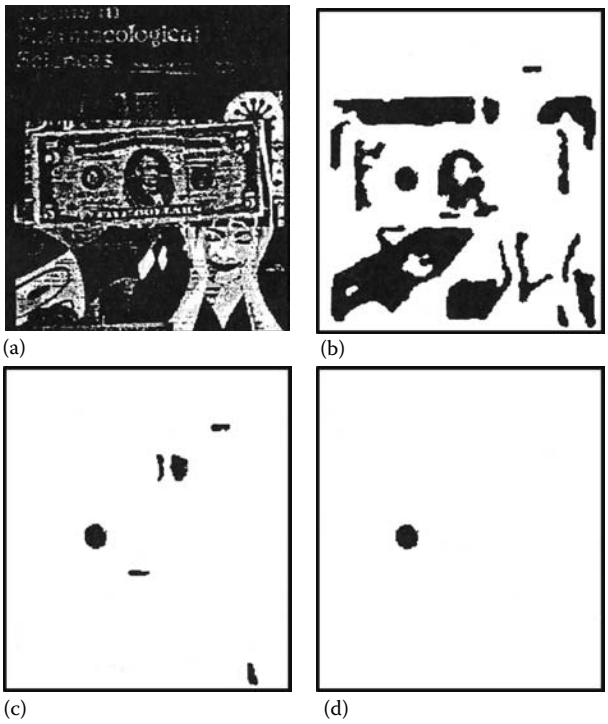


FIGURE 13.11 Some intermediate steps in the dollar bill recognition process. An input image (a) shown here in single color results in the ''color cube'' (b), the convex objects (c), and the size classification (d).* (From Zarándy, A., Werblin, F., Roska, T., Chua, L.O., and Novel type of analogical CNN algorithms for recognizing bank notes, Memorandum UCB/ERL, M94/29 1994, Electronics Research Laboratory, University of California, Berkeley, CA, 1994. With permission.)

motion. Detecting direction, shape, etc. of moving objects are only the simplest problems that can be solved via CNN. In fact, even depth detection can be included as well.

13.4.2 Partial Differential Equations

As noted in the original paper [1], even the simple-cell CNN with the linear template

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -3 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} = 0, \quad \boldsymbol{I} = 0$$

can approximate the solution of a diffusion-type partial differential equation (PDE) on a discrete spatial grid. This solution maintains continuity in time, a nice property not possible in digital computers.

By adding just a simple capacitor to the output, i.e., by placing a parallel RC circuit across the output port of the cell of Figure 13.3, the following wave equation will be represented in a discrete space grid:

$$\mathrm{d}^2 p(t)/\mathrm{d}t^2 = \Delta p$$

* Figure 13.11 shows a part of this process [31] (which involves color processing as well). The dollar bill shown in the foreground is analyzed and the circles of a given size are detected (colors are not shown).

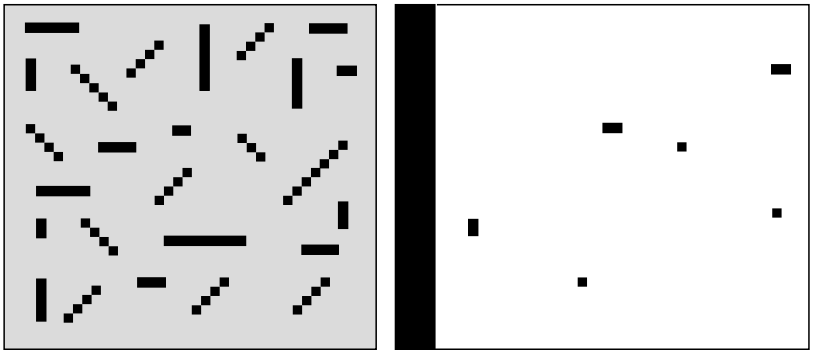


FIGURE 13.12 The length tuning effect. The input image on the LHS contains bars of different lengths. The out image on the RHS contains only those that are smaller than a given length. (From Roska, T., Hámori, J., Lábos, E., Lotz, K., Takács, J., Venetianer, P., Vidnyánszki, Z., and Zarándy, A., *IEEE Trans. Circuits Syst. I*, 40, 182, 1993. With permission.)

where

$p(t) = P(x, y, t)$ is the state (intensity) variable on a 2-D plane (x, y)

Δ is the Laplacian operator (the sum of the second derivatives related to x and y)

In some cases, it is useful to use a cell circuit that is chaotic. Using the canonical Chua's circuit, other types of PDEs can be modeled, generating effects like autowaves, spiral waves, Turing patterns, and so on (e.g., Perez-Munuzuri et al. in Ref. [7]).

13.4.3 Relation to Biology

Many topographical sensory organs have processing neural-cell structures very similar to the CNN model. Local connectivity in a few sheets of regularly situated neurons is very typical. Vision, especially the retina, reflects these properties strikingly. It is not surprising that, based on standard neurobiological models, CNN models have been applied to the modeling of the subcortical visual pathway [10]. Moreover, a new method has been devised to use the CNN universal machine for combining retina models of different species in a programmed way. Modalities from other sensory organs can be modeled similarly and combined with the retina models [11]. This has been called bionic eye.

Many of these models are neuromorphic. This means that there is a one-to-one correspondence between the neuroanatomy and the CNN structure. Moreover, the CNN template reflects the interconnection pattern of the neurons (called receptive field organizations). Length tuning is such an example. A corresponding input and output picture of the neuromorphic length tuning model is shown in Figure 13.12. Those bars that have lengths smaller than or equal to three pixels are detected.

13.5 Template Library: Analogical CNN Algorithms

During the last few years, after the invention of the cellular neural network paradigm and the CNN universal machine, many new cloning templates have been discovered. In addition, the number of innovative analogical CNN algorithms, combining both analog cloning templates and local as well as global logic, is presently steadily increasing at a rapid rate.

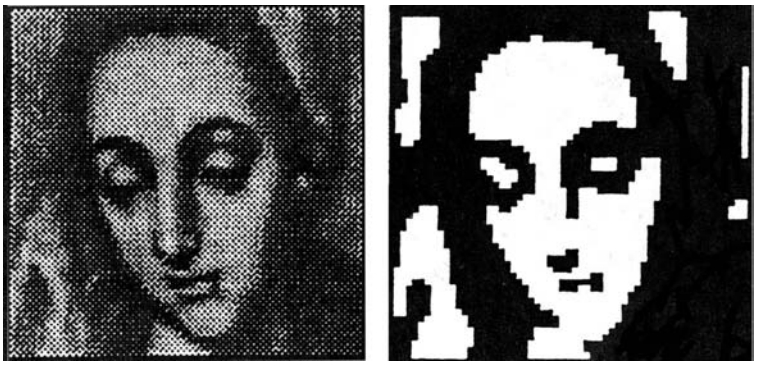
As an illustration, let us choose a couple of cloning templates from the CNN library [1,12]. In each case, a name, a short description of the function, the cloning templates, and a representative input–output image pair are shown. With regard to the inputs, the default case means that the input and initial state are the same. If $\boldsymbol{B} = 0$, then the input picture is chosen as the initial state.

Name: AVERAGE

Function. Spatial averaging of pixel intensities over the $r = 1$ convolutional window.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = 0$$

Example. Input and output picture.

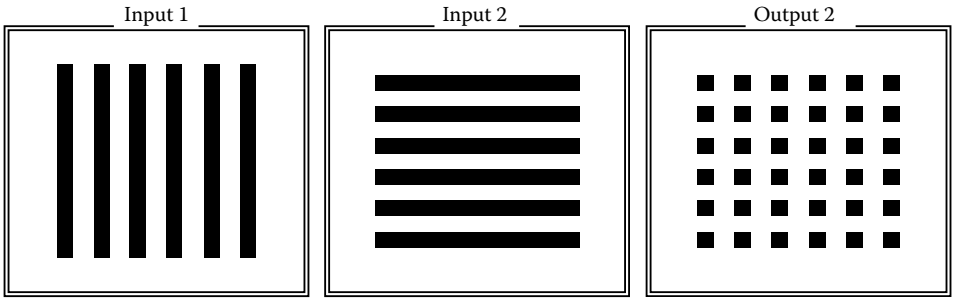


Name: AND

Function. Logical "AND" function of the input and the initial state pictures.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = -1$$

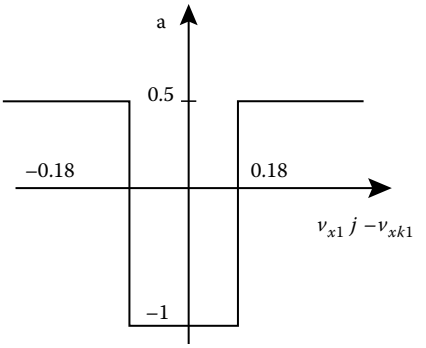
Example. Input, initial state, and output picture.



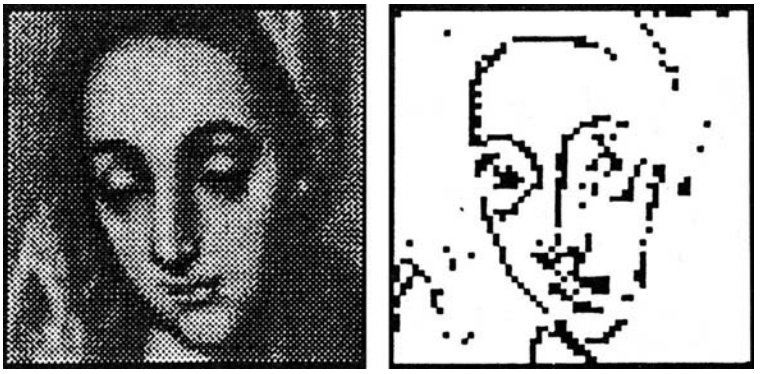
Name: CONTOUR

Function. Grayscale contour detector.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} a & a & a \\ a & a & a \\ a & a & a \end{bmatrix}, \quad \boldsymbol{I} = 0.7$$



Example. Input and output picture.

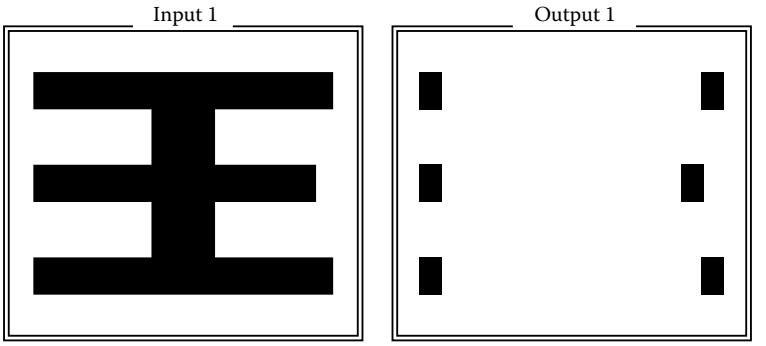


Name: CORNER

Function. Convex corner detector.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}, \quad \boldsymbol{I} = -3$$

Example. Input and output picture.

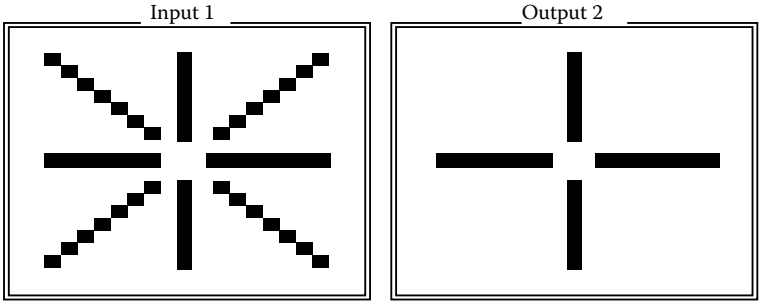


Name: DELDIAG1

Function. Deletes one pixel wide diagonal lines (5).

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -0.25 & 0 & -0.25 \\ 0 & 0 & 0 \\ -0.25 & 0 & -0.25 \end{bmatrix}, \quad \boldsymbol{I} = -2$$

Example. Input and output picture.

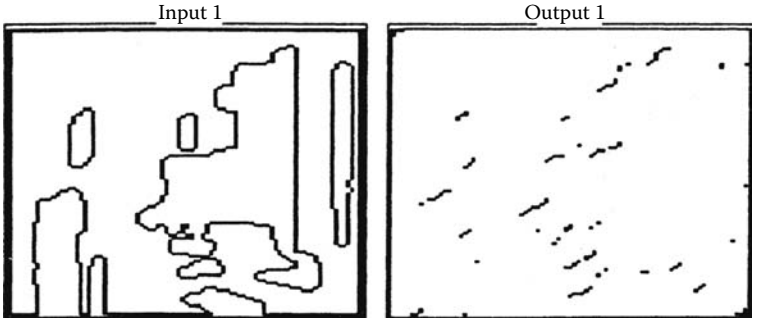


Name: DIAG

Function. Deletes the diagonal lines.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -1 & -1 & -0.5 & 0.5 & 1 \\ -1 & -0.5 & 1 & 1 & 0.5 \\ -0.5 & 1 & 5 & 1 & -0.5 \\ 0.5 & 1 & 1 & -0.5 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 \end{bmatrix}, \quad \boldsymbol{I} = -9$$

Example. Input and output picture.

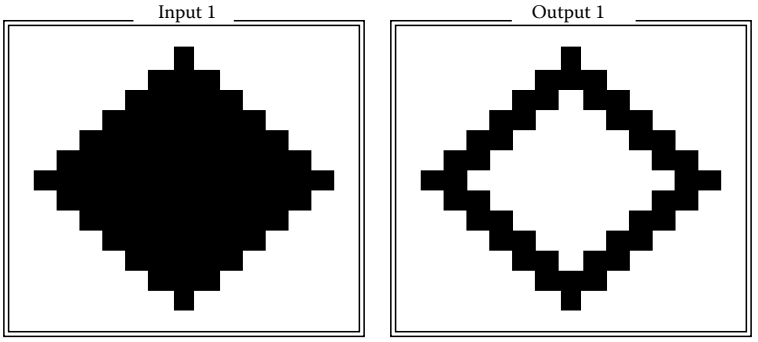


Name: EDGE

Function. Black-and-white edge detector.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -0.25 & -0.25 & -0.25 \\ -0.25 & 2 & -0.25 \\ -0.25 & -0.25 & -0.25 \end{bmatrix}, \quad \boldsymbol{I} = -1.5$$

Example. Input and output picture.



Name: MATCH

Function. Detects 3×3 patterns matching exactly the one prescribed by the template $\boldsymbol{B}$, namely, having a black/white pixel where the template value is ± 1, respectively.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} v & v & v \\ v & v & v \\ v & v & v \end{bmatrix}, \quad \boldsymbol{I} = -N + 0.5$$

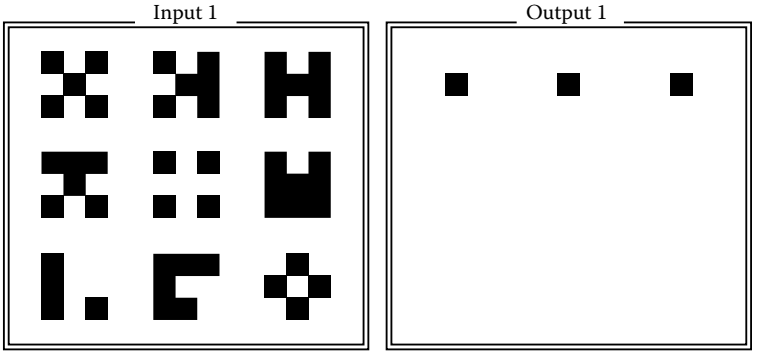
where

$v = +1$, if corresponding pixel is required to be black
$v = 0$, if corresponding pixel is don't care
$v = -1$, if corresponding pixel is required to be white

$N =$ number of pixels required to be either black or white, i.e., the number of nonzero values in the $\boldsymbol{B}$ template

Example. Input and output picture, using the following values:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{I} = -6.5$$

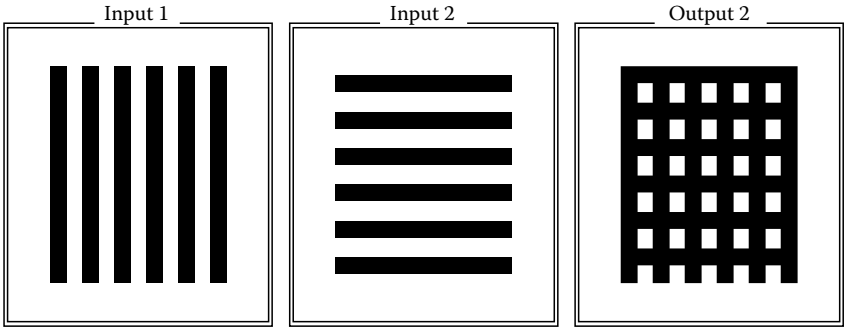


Name: OR

Function. Logical "OR" function of the input and the initial state.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{I} = 2$$

Example. Input, initial state, and output picture.

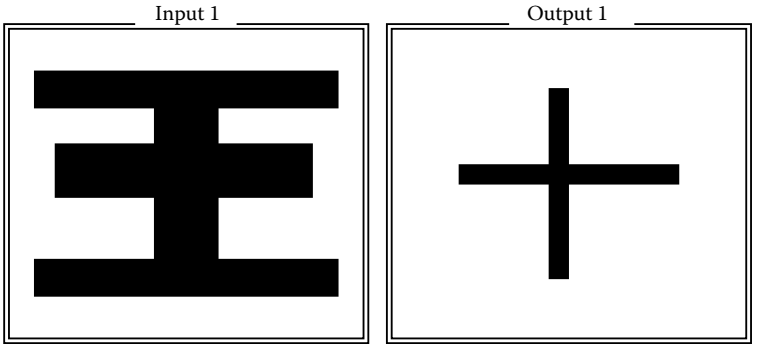


Name: PEELIPIX

Function. Peels one pixel from all directions.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0.4 & 0 \\ 0.4 & 1.4 & 0.4 \\ 0 & 0.4 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 4.6 & -2.8 & 4.6 \\ -2.8 & 1 & -2.8 \\ 4.6 & -2.8 & 4.6 \end{bmatrix}, \quad \boldsymbol{I} = -7.2$$

Example. Input and output picture.



13.6 Recent Advances

After the first few integrated circuit implementations of the basic CNN circuits, stored programmable analogic CNN universal machine chips have been fabricated. Indeed, a full-fledged version of them [13] is the first visual microprocessor, all the 4096 cell processors of it contain an optical sensor right on the surface of the chip (a focal plane). This implementation represents, at the same time, the most complex operational, stored programmable analog CMOS integrated circuit ever reported, in terms of the number of transistors operating in analog mode (about 1 million). The equivalent digital computing power of this

visual microprocessor is about a few TeraOPS. It processes grayscale input images and has a grayscale output. A 128×128 processor version has recently been fabricated. A binary input/output CNN universal machine chip with 48×48 cell processors has a higher cell density [14], and another circuit design strategy [15] is aiming to implement 5×5 or even higher neighborhood templates.

These chips are the first examples of a new, analogic, topographic (spatial-temporal) computing technology. Its computational infrastructure (high-level language, called Alpha, compiler, operating system, etc.) has also been developed [16], and the industrial applications have been started in a couple of companies worldwide. Moreover, a key application area of this technology is sensor computing [17]. Integrating 2-D topographic sensor arrays with the CNN universal machine on a single chip, providing a direct, dynamic interaction with tuning of the sensors, this is a capability no other technology offers with comparable computational power.

Recently, it has been shown that PDE-based techniques, the most advanced methods for complex image-processing problems, could solve tasks intractable with other methods. Their only drawback is the excessive digital computing power they need. In our cellular computing technology, however, the elementary instruction could be a solution of a PDE. It has been shown that, in addition to the simple diffusion PDE implementation described previously, almost all PDEs can be implemented by CNN [18]. Indeed, active waves [2] have been successfully applied using operational analogic CNN universal machine chips with 4096 cell processors, manifesting at least three orders of magnitude speed advantage compared to fully digital chips of comparable IC technology feature size.

Following the first steps in modeling living sensory modalities, especially vision, motivated especially by a breakthrough in understanding the neurobiological constructs of the mammalian retina [19], new models and a modeling framework [20] have been developed based on CNNs. Their implementation in complex cell CNN universal machines [21] is under construction.

Studies on complexity related to CNN models and implementations have been emerging recently. Following the groundbreaking theoretical studies of Turing on the morphogenesis of CNN-like coupled nonlinear units [22] and a few experimental case studies of the well-publicized ''complex systems,'' as well as many exotic waves generated by coupled $\boldsymbol{A}$ template CNNs, the root of complexity in pattern formation at the edge of chaos has been discovered [23]. As far as the computational complexity is concerned, the study of a new quality of computational complexity has been explored [24], showing qualitatively different properties compared to the classical digital complexity theory as well as the complexity on reals [25].

To further explore the vast amount of literature on CNN technology and analogic cellular computing, the interested reader could consult the bibliography at the website of the technical committee on Cellular Neural Networks and Array Computing of the IEEE Circuits and Systems Society (http://www.ieee-cas.org/~cnnactc), some recent monographs [26–28], and an undergraduate textbook [29].

13.7 Recent Developments and Outlook

During the last few years, CNN technology, as well as its offsprings and generalizations, has generated several significant results and the first few companies have entered into the market. The main events are as follows.

The appearance of the first commercially available, stored programmable, visual microprocessor, the Q-Eye (with 176×144 cell processors and on-chip optical sensors, 10,000 frame per second, and $<$300 mW), embedded in the self-contained Eye-RIS system, both are from AnaFocus Ltd., Seville (www.anafocus.com). A new version of the Bi-i camera-computer, using emulated digital cellular array architecture, has also been developed, and applied in some mission critical tasks (Eutecus Inc., Berkeley, California, www.eutecus.com).

Several algorithmic and theoretical results as well as application case studies have been published. A new type of algorithmic thinking is emerging that is based on spatial-temporal waves as elementary instructions.

The CNN universal machine, with some extensions and diverse physical implementations became a general computational platform: the cellular wave computer. This has been happening in parallel with a dramatic new trend in microprocessor architecture development, namely the multi- and many core architectures on a single chip. The other side of the story is the cellular supercomputer development.

The essence of this new trend is that processor arrays on a single chip are becoming the next main direction in CMOS technology, when saturating the clock frequency due to the power dissipation constraints. Today, products are already on the market with about 100 processors and the trend is continuing, including the latest FPGAs. Hence, the cellular many-core processor arrays are becoming the major trend in computing in general, and in sensory computing, in particular. Since the wire delay is becoming greater than the gate delay, the communication speed is reduced dramatically far from a given processor. Hence, the mainly (not exclusively) locally connected cellular processor array architectures on a single chip become a physical necessity. Interestingly, the high-end supercomputers are also following this lead (the highest speed Blue Gene of IBM has a 3-D toroidal cellular architecture).

This new development has made the cellular wave computer architecture a must in many applications. As an example, a very efficient PDE implementation was made by using CNN models on an FPGA and on a CELL Multiprocessor (IBM, Sony, Toshiba) developed for a game console, as well.

Interestingly, the circuit hardware implementation issues (now at a 45 nm CMOS technology) are forcing the conversion of the computing architectures. The details can be found in a recent review article [30]. It seems that the cellular wave computer might become a prototype architecture, although the cell processors are of different forms and implementations (digital, analog, mixed mode, optical, etc.), and the spatial-temporal, mainly locally and sparsely globally connected communication framework is necessary. Interestingly, many different sensory organs, uncovered recently by neurobiologists, are following this architecture.

References

1. L. O. Chua and L. Yang, Cellular neural networks: Theory, *IEEE Trans. Circuits Syst*., 35, 1257–1272, 1988.
2. Cs. Rekeczky and L. O. Chua, Computing with front propagation-active contours and skeleton models in continuous-time CNN, *J. VLSI Signal Process. Syst*., 23, 373–402, 1999.
3. J. Cruz and L. O. Chua, A CNN chip for connected component detection, *IEEE Trans. Circuits Syst*., 38, 812–817, 1991.
4. H. Harrer, J. A. Nossek, and R. Stelzl, An analog implementation of discrete-time cellular neural networks, *IEEE Trans. Neural Networks*, 3, 466–476, 1992.
5. R. Dominguez-Castro, S. Espejo, A. Rodriguez-Vazquez, and R. Carmona, A CNN universal chip in CMOS technology, *Proceedings of the IEEE Third International Workshop on CNN and Applications* (CNNA-94), Rome, Italy, pp. 91–96, 1994.
6. T. Roska and J. Vandewalle, Eds., *Cellular Neural Networks*, Chichester, Wiley, 1993.
7. J. A. Nossek and T. Roska, Eds., Special issues on cellular neural networks, *IEEE Trans. Circuits Syst. I*, 40, Mar. 1993; Special issue on cellular neural networks, *IEEE Trans. Circuits Syst. II*, 40, Mar. 1993.
8. L. O. Chua and L. Yang, Cellular neural networks: Applications, *IEEE Trans. Circuits Syst*., 35, 1273–1290, 1988.
9. T. Roska and L. O. Chua, The CNN universal machine: An analogic array computer, *IEEE Trans. Circuits Syst. II*, 40, 163–173, 1993.
10. T. Roska, J. Hamori, E. Labos, K. Lotz, K. Takacs, P. Venetianer, Z. Vidnyanszki, and A. Zarandy, The use of CNN models in the subcortical visual pathway, *IEEE Trans. Circuits Syst. I*, 40, 182–195, 1993.
11. F. Werblin, T. Roska, and L. O. Chua, The analogic cellular neural network as a bionic eye, *Int. J. Circuit Theory Appl.* (*CTA*), 23(6), 541–569, 1995.

12. CNN Analogic Nonlinear Dynamics (CANDY) Simulator, guide and program (student version), Available at http://lab.analogic.sztaki.hu
13. G. Linán, S. Espejo, R. Dominguez-Castro, E. Roca, and A. Rodriguez-Vázquez, CNNUC3: A mixed signal 64 × 64 CNN Universal Chip, *Proceedings of IEEE MicroNeuro*, Granada, Spain, pp. 61–68, 1999.
14. A. Paasio, A. Davidzuk, K. Halonen, and V. Porra, Minimum-size 0.5 micron CMOS programmable 48 by 48 test chip, *Proceedings of the European Conference on Circuit Theory and Design ECCTD'97*, Vol. I, Budapest, Hungary, pp. 154–156, 1997.
15. W. C. Yen and C. Y. Wu, The design of neuron-bipolar junction transistor (vBJT) cellular neural network (CNN) structure with multi-neighborhood-layer templates, *Proceedings of 6th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA'2000)*, 23–25 May, Catania Uniiversity, Catania, Italy, pp. 195–200, 2000.
16. T. Roska, Á. Zarándy, S. Zöld, P. Földesy, and P. Szolgay, The computational infrastructure of analogic CNN computing—Part I: The CNN-UM chip prototyping system, *IEEE Trans. Circuits Syst. I*: Special issue on bio-inspired processors and cellular neural networks for vision (CAS-I Special Issue), 46(2), 261–268, 1999.
17. T. Roska, Computer-sensors: Spatial-temporal computers for analog array signals, dynamically integrated with sensors, *J. VLSI Signal Process. Syst.*, 23, 221–238, 1999.
18. T. Roska, L. O. Chua, D. Wolf, T. Kozek, R. Tetzlaff, and F. Puffer, Simulating nonlinear waves and partial differential equations via CNN, *IEEE Trans. Circuits Syst. I*, 42, 807–815, 1995.
19. B. Roska and F. S. Werblin, Vertical interactions across ten parallel, stacked representations in the mammalian retina, *Nature*, 410, 583–587, Mar. 29, 2001.
20. F. Werblin, B. Roska, D. Bálya, Cs. Rekeczky, and T. Roska, Implementing a retinal visual language in CNN: A neuromorphic case study, *Proceedings of the IEEE ISCAS* 2001, Vol. III, Sydney, pp. 333–336, 2001.
21. Cs. Rekeczky, T. Serrano, T. Roska, and Á. Rodríguez-Vázquez, A stored program 2nd order/3-layer complex cell CNN-UM, *Proceedings of 6th IEEE International Workshop on Cellular Neural Networks and Their Applications* (CNNA-2000), 23–25 May, Catania University, Catania, Italy, pp. 15–20, 2000.
22. A. M. Turing, The chemical basis of morphogenesis, *Philos. Trans. R. Soc. London*, B237, 37–72, 1952.
23. L. O. Chua, *CNN: A Paradigm for Complexity*, World Scientific, Singapore, 1998.
24. T. Roska, AnaLogic wave computers—wave-type algorithms: Canonical description, computer classes, and computational complexity, *Proceedings of IEEE ISCAS* 2001, Sydney, Australia, Vol. III, pp. 41–44, 2001.
25. L. Blum, F. Cucker, M. Shub, and S. Smale, *Complexity and Real Computation*, Springer, New York, 1998.
26. G. Manganaro, P. Arena, and L. Fortuna, *Cellular Neural Networks—Chaos, Complexity and VLSI Processing*, Springer, Berlin, 1999.
27. M. Hänggi and G. Moschitz, *Cellular Neural Networks—Analysis, Design and Optimization*, Kluwer Academic, Boston, MA, 2000.
28. T. Roska and Á. Rodríguez-Vázquez, Eds., *Towards the Visual Microprocessor—VLSI Design and the Use of Cellular Neural Network Universal Machine*, Wiley, New York, 2001.
29. L. O. Chua and T. Roska, *Cellular Neural Network and Visual Computing—Foundations and Applications*, Cambridge University Press, New York, 2002.
30. T. Roska, Circuits, computers, and beyond Boolean logic, *Int. J. Circuit Theory Appl.*, Sept.–Oct., 2007.
31. Zarándy, A., Werblin, F., Roska, T., Chua, L. O., and Novel type of analogical CNN algorithms for recognizing bank notes, Memorandum UCB/ERL, M94/29 1994, Electronics Research Laboratory, University of California, Berkeley, CA, 1994.
32. L. O. Chua and T. Roska, The CNN paradigm, *IEEE Trans. Circuits Syst. I*, 40, 147–156, 1993.

14

Bifurcation and Chaos

14.1 Introduction to Chaos **14**-1
Electrical and Electronic Circuits as Dynamical Systems • Classification and Uniqueness of Steady-State Behaviors • Stability of Steady-State Trajectories • Horseshoes and Chaos • Structural Stability and Bifurcations

14.2 Chua's Circuit: A Paradigm for Chaos **14**-27
Dynamics of Chua's Circuit • Chaos in Chua's Circuit • Steady-States and Bifurcations in Chua's Circuit • Manifestations of Chaos • Practical Realization of Chua's Circuit • Experimental Steady-State Solutions • Simulation of Chua's Circuit • Dimensionless Coordinates and the α–β Parameter-Space Diagram

14.3 Chua's Oscillator **14**-43
State Equations • Topological Conjugacy • Eigenvalues-to-Parameters Mapping Algorithm for Chua's Oscillator

14.4 van der Pol Neon Bulb Oscillator **14**-48
Winding Numbers • Circle Map • Experimental Observations of Mode-Locking and Chaos in van der Pol's Neon Bulb Circuit • Circuit Model

14.5 Synchronization of Chaotic Circuits **14**-56
Linear Mutual Coupling • Pecora–Carroll Drive-Response Concept

14.6 Applications of Chaos **14**-64
Pseudorandom Sequence Generation • Communicating with Chaos

References **14**-71

Further Information **14**-72

Michael Peter Kennedy
University College Cork

14.1 Introduction to Chaos

14.1.1 Electrical and Electronic Circuits as Dynamical Systems

A system is something having parts that may be perceived as a single entity. A dynamical system is one that changes with time; what changes is the state of the system. Mathematically, a dynamical system consists of a space of states (called the state space or phase space) and a rule, called the dynamic, for determining which state corresponds at a given future time to a given present state [8]. A deterministic dynamical system is one where the state, at any time, is completely determined by its initial state and dynamic. In this section, we consider only deterministic dynamical systems.

A deterministic dynamical system may have a continuous or discrete state space and a continuous-time or discrete-time dynamic.

A lumped* circuit containing resistive elements (resistors, voltage and current sources) and energy-storage elements (capacitors and inductors) may be modeled as a continuous-time deterministic dynamical system in $\mathbb{R}^n$. The evolution of the state of the circuit is described by a system of ordinary differential equations (ODEs) called state equations.

Discrete-time deterministic dynamical systems occur in electrical engineering as models of switched-capacitor (SC) and digital filters, sampled phase-locked loops, and sigma–delta modulators. Discrete-time dynamical systems also arise when analyzing the stability of steady-state solutions of continuous-time systems. The evolution of a discrete-time dynamical system is described by a system of *difference equations*.

14.1.1.1 Continuous-Time Dynamical Systems

THEOREM 14.1 (Existence and Uniqueness of Solution for a Differential Equation)

Consider a continuous-time deterministic dynamical system defined by a system of ordinary differential equations of the form

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), t) \tag{14.1}$$

where $\mathbf{X}(t) \in \mathbb{R}^n$ *is called the state,* $\dot{\mathbf{X}}(t)$ *denotes the derivative of* $\mathbf{X}(t)$ *with respect to time,* $\mathbf{X}(t_0) = \mathbf{X}_0$ *is called the initial condition, and the map* $\mathbf{F}(\cdot, \cdot): \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$ *is* (1) *continuous almost everywhere*† *on* $\mathbb{R}^n \times \mathbb{R}_+$ *and* (2) *globally Lipschitz*‡ *in* $\mathbf{X}$. *Then, for each* $(\mathbf{X}_0, t_0) \in \mathbb{R}^n \times \mathbb{R}_+$, *there exists a continuous function* $\phi(\cdot; \mathbf{X}_0, t_0): \mathbb{R}_+ \to \mathbb{R}^n$ *such that*

$$\phi(t_0; \mathbf{X}_0, t_0) = \mathbf{X}_0$$

and

$$\dot{\phi}(t; \mathbf{X}_0, t_0) = \mathbf{F}(\phi(t; \mathbf{X}_0, t_0), t) \tag{14.2}$$

Furthermore, this function is unique.

The function $\phi(\cdot; \mathbf{X}_0, t_0)$ *is called the solution or trajectory* $(\mathbf{X}_0, t_0)$ *of the differential Equation* 14.1.

The image $\{\phi(t; \mathbf{X}_0, t_0) \in \mathbb{R}^n | t \in \mathbb{R}_+\}$ of the trajectory through $(\mathbf{X}_0, t_0)$ is a continuous curve in $\mathbb{R}^n$ called the orbit through $(\mathbf{X}_0, t_0)$.

$\mathbf{F}(\cdot, \cdot)$ is called the vector field of Equation 14.1 because its image $\mathbf{F}(\mathbf{X}, t)$ is a vector that defines the direction and speed of the trajectory through $\mathbf{X}$ at time t.

The vector field $\mathbf{F}$ generates the *flow* ϕ, where $\phi(\cdot; \cdot, \cdot): \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$ is a collection of continuous maps $\{\phi(t; \cdot, \cdot): \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n | t \in \mathbb{R}_+\}$.

In particular, a point $\mathbf{X}_0 \in \mathbb{R}^n$ at t_0 is mapped by the flow into $\mathbf{X}(t) = \phi_t(t; \mathbf{X}_0, t_0)$ at time t.

---

* A *lumped* circuit is one with physical dimensions that are small compared with the wavelengths of its voltage and current waveforms [2].

† By *continuous almost everywhere*, we mean the following: let D be a set in $\mathbb{R}_+$ that contains a countable number of discontinuities and for each $\mathbf{X} \in \mathbb{R}^n$, assume that the function $t \in \mathbb{R}_+ \backslash D \to \mathbf{F}(\mathbf{X}, t) \in \mathbb{R}^n$ *is continuous* and for any $\tau \in D$ the left-hand and right-hand limits $\mathbf{F}(\mathbf{X}, \tau)$ and $\mathbf{F}(\mathbf{X}, \tau_+)$, respectively, are *finite* in $\mathbb{R}^n$ [1]. This condition includes circuits that contain switches and/or squarewave voltage and current sources.

‡ There is a piecewise continuous function $k(\cdot): \mathbb{R}_+ \to \mathbb{R}_+$ such that $\|\mathbf{F}(\mathbf{X}, t) - \mathbf{F}(\mathbf{X}', t)\| \leq k(t) \|\mathbf{X} - \mathbf{X}'\|$, $\forall t \in \mathbb{R}_+$, $\forall \mathbf{X}, \mathbf{X}' \in \mathbb{R}^n$.

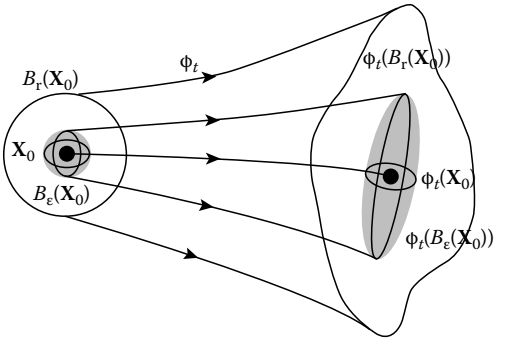


FIGURE 14.1 Vector field **F** of an autonomous continuous-time dynamical system generates a flow ϕ that maps a point $\mathbf{X}_0$ in the state space to its image $\phi_t(\mathbf{X}_0)$ t seconds later. A volume of state space $B_r(\mathbf{X}_0)$ evolves under the flow into a region $\phi_t[B_r(\mathbf{X}_0)]$. Sufficiently close to the trajectory $\phi_t(\mathbf{X}_0)$, the linearized flow maps a sphere of radius ε into an ellipsoid.

14.1.1.2 Autonomous Continuous-Time Dynamical Systems

If the vector field of a continuous-time deterministic dynamical system depends only on the state and is *independent* of time t, then the system is said to be *autonomous* and may be written as

$$\dot{\mathbf{X}}(t) = \mathbf{F}[\mathbf{X}(t)]$$

or simply

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}) \tag{14.3}$$

If, in addition, the vector field $\mathbf{F}(\cdot)$: $\mathbb{R}^n \to \mathbb{R}^n$ is *Lipschitz*,* then there is a *unique* continuous function $\phi(\cdot, \mathbf{X}_0)$: $\mathbb{R}_+ \to \mathbb{R}^n$ (called the trajectory through $\mathbf{X}_0$), which satisfies,

$$\dot{\phi}(t, \mathbf{X}_0) = \mathbf{F}[\phi(t, \mathbf{X}_0)], \quad \phi(t_0, \mathbf{X}_0) = \mathbf{X}_0 \tag{14.4}$$

Because the vector field is independent of time, we choose $t_0 \equiv 0$. For shorthand, we denote the flow by ϕ and the map $\phi(t, \cdot)$: $\mathbb{R}^n \to \mathbb{R}^n$ by ϕ_t.

The t-advance map ϕ_t takes a state $\mathbf{X}_0 \in \mathbb{R}^n$ to state $\mathbf{X}(t) = \phi_t(\mathbf{X}_0)$ t seconds later. In particular, ϕ_0 is the identity mapping. Furthermore, $\phi_{t+s} = \phi_t\phi_s$, because the state $\mathbf{Y} = \phi_s(\mathbf{X})$ to which $\mathbf{X}$ evolves after time s evolves after an additional time t into the same state $\mathbf{Z}$ as that to which $\mathbf{X}$ evolves after time $t + s$:

$$\mathbf{Z} = \phi_t(\mathbf{Y}) = \phi_t[\phi_s(\mathbf{X})] = \phi_{t+s}(\mathbf{X})$$

A bundle of trajectories emanating from a ball $B_r(\mathbf{X}_0)$ of radius r centered at $\mathbf{X}_0$ is mapped by the flow into some region $\phi_t[B_r(\mathbf{X}_0)]$ after t seconds (see Figure 14.1). Consider a short segment of the trajectory ϕ_t $(\mathbf{X}_0)$ along which the flow is differentiable with respect to $\mathbf{X}$: in a sufficiently small neighborhood of this trajectory, the flow is almost linear, so the ball $B_\varepsilon(\mathbf{X}_0)$ of radius ε about $\mathbf{X}_0$ evolves into an ellipsoid $\phi_t[B_\varepsilon(\mathbf{X}_0)]$, as shown.

An important consequence of Lipschitz continuity in an autonomous vector field and the resulting uniqueness of solution of Equation 14.3 is that a trajectory of the dynamical system cannot go through the same point twice in two different directions. In particular, no two trajectories may cross each other; this is called the noncrossing property [18].

14.1.1.3 Nonautonomous Dynamical Systems

A nonautonomous, n-dimensional, continuous-time dynamical system may be transformed to an $(n+1)$-dimensional autonomous system by appending time as an additional state variable and writing

$$\begin{aligned} \dot{\mathbf{X}}(t) &= \mathbf{F}[\mathbf{X}(t), X_{n+1}(t)] \\ \dot{X}_{n+1}(t) &= 1 \end{aligned} \tag{14.5}$$

* There exists a finite $k \in \mathbb{R}^n$ such that $\|\mathbf{F}(\mathbf{X}) - \mathbf{F}(\mathbf{X}')\| \leq k \|\mathbf{X} - \mathbf{X}'\|$, $\forall \mathbf{X}, \mathbf{X}' \in \mathbb{R}^n$.

In the special case where the vector field is periodic with period T, as for example in the case of an oscillator with sinusoidal forcing, the periodically forced system Equation 14.5 is equivalent to the $(n+1)$ th-order autonomous system

$$\begin{aligned}\dot{\mathbf{X}}(t) &= \mathbf{F}(\mathbf{X}(t), \theta(t)T)\\ \dot{\theta}(t) &= \frac{1}{T}\end{aligned} \tag{14.6}$$

where $\theta(t) = X_{n+1}/T$.

By identifying the n-dimensional hyperplanes corresponding to $\theta = 0$ and $\theta = 1$, the state space may be transformed from $\mathbb{R}^n \times \mathbb{R}_+$ into an equivalent cylindrical state space $\mathbb{R}^n \times S^1$, where S^1 denotes the circle. In the new coordinate system, the solution through $(\mathbf{X}_0, t_0)$ of Equation 14.6 is

$$\begin{pmatrix} \mathbf{X}(t) \\ \theta_{S^1}(t) \end{pmatrix} = \begin{pmatrix} \phi_t(\mathbf{X}_0, t_0) \\ t/T \bmod 1 \end{pmatrix}$$

where $\theta(t) \in \mathbb{R}_+$ is identified with a point on S^1 (which has normalized angular coordinate $\theta_{S^1}(t) \in [0, 1)$) via the transformation $\theta_s^1(t) = \theta(t) \bmod 1$. Using this technique, periodically forced nonautonomous systems can be treated like autonomous systems.

14.1.1.4 Discrete-Time Dynamical Systems

Consider a discrete-time deterministic dynamical system defined by a system of difference equations of the form

$$\mathbf{X}(k+1) = \mathbf{G}(\mathbf{X}(k), k) \tag{14.7}$$

where

$\mathbf{X}(k) \in \mathbb{R}^n$ is called the state

$\mathbf{X}(k_0) = \mathbf{X}_0$ is the initial condition

$\mathbf{G}(\cdot, \cdot)$: $\mathbb{R}^n \times \mathbb{Z}_+ \to \mathbb{R}^n$ maps the current state $\mathbf{X}(k)$ into the next state $\mathbf{X}(k+1)$, where $k_0 \in \mathbb{Z}_+$

By analogy with the continuous-time case, there exists a function $\phi(\cdot, \mathbf{X}_0, k_0)$: $\mathbb{Z}_+ \to \mathbb{R}^n$ such that

$$\phi(k_0; \mathbf{X}_0, k_0) = \mathbf{X}_0$$

and

$$\phi(k+1; \mathbf{X}_0, k_0) = \mathbf{G}(\phi(k; \mathbf{X}_0, k_0), k)$$

The function $\phi(\cdot; \mathbf{X}_0, k_0)$: $\mathbb{Z}_+ \to \mathbb{R}^n$ is called the solution or trajectory through $(\mathbf{X}_0, k_0)$ of the difference Equation 14.7.

The image $\{\phi(k; \mathbf{X}_0, k_0) \in \mathbb{R}^n \mid k \in \mathbb{Z}_+\}$ in $\mathbb{R}^n$ of the trajectory through $(\mathbf{X}_0, k_0)$ is called an orbit through $(\mathbf{X}_0, k_0)$.

If the map $\mathbf{G}(\cdot, \cdot)$ of a discrete-time dynamical system depends only on the state $\mathbf{X}(k)$ and is independent of k then the system is said to be autonomous and may be written more simply as

$$\mathbf{X}_{k+1} = \mathbf{G}(\mathbf{X}_k) \tag{14.8}$$

where $\mathbf{X}_k$ is shorthand for $\mathbf{X}(k)$ and the initial iterate k_0 is chosen, without loss of generality, to be zero. Using this notation, $\mathbf{X}_k$ is the image $\mathbf{X}_0$ after k iterations of the map $\mathbf{G}(\cdot)$: $\mathbb{R}^n \to \mathbb{R}^n$.

Example: Nonlinear Parallel *RLC* Circuit

Consider the parallel *RLC* circuit in Figure 14.2. This circuit contains a linear inductor L, a linear capacitor C_2, and a nonlinear resistor N'_R, where the continuous piecewise-linear driving-point (DP) characteristic (see Figure 14.3) has slope G'_a for $|V'_R| \leq E$ and slope G'_b for $|V'_R| > E$. The DP characteristic of N'_R may be written explicitly

$$I'_R(V'_R) = G'_b V'_R + \frac{1}{2}(G'_a - G'_b)\left(|V'_R + E| - |V'_R - E|\right)$$

This circuit may be described by a pair of ODEs and is therefore a second-order, continuous-time dynamical system. Choosing I_3 and V_2 as state variables, we write

$$\frac{dI_3}{dt} = -\frac{1}{L}V_2$$
$$\frac{dV_2}{dt} = \frac{1}{C_2}I_3 - \frac{1}{C_2}I'_R(V_2)$$

with $I_3(0) = I_{3_0}$ and $V_2(0) = V_{2_0}$.

We illustrate the vector field by drawing vectors at uniformly spaced points in the two-dimensional state space defined by (I_3, V_2). Starting from a given initial condition (I_{3_0}, V_{2_0}), a solution curve in state space is the locus of points plotted out by the state as it moves through the vector field, following the direction of the arrow at every point. Figure 14.4 illustrates typical vector fields and trajectories of the circuit.

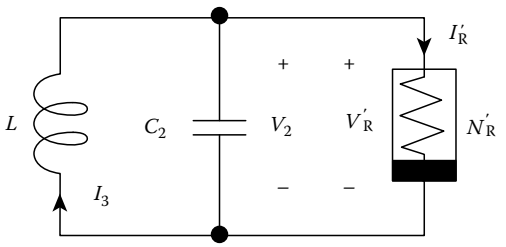


FIGURE 14.2 Parallel *RLC* circuit where the nonlinear resistor N'_R has a DP characteristic as illustrated in Figure 14.3. By Kirchhoff's voltage law, $V'_R = V_2$.

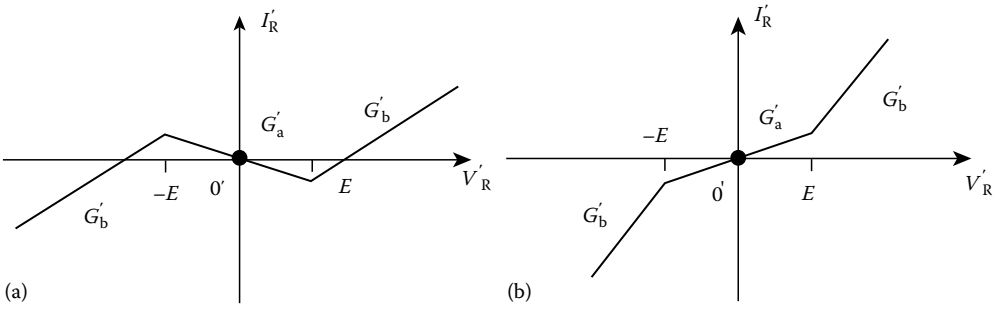


FIGURE 14.3 DP characteristic of N'_R in Figure 14.2 when (a) $G'_a < 0$ and (b) $G'_a > 0$.

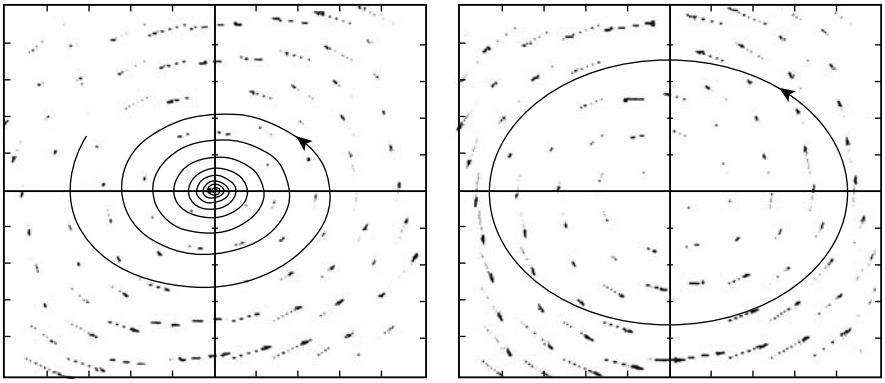


FIGURE 14.4 Vector fields for the nonlinear *RLC* circuit in Figure 14.2. $L = 18$ mH, $C_2 = 100$ nF, $E = 0.47$ V. (a) $G'_a = 242.424$ μS, $G'_b = 1045.455$ μS: all trajectories converge to the origin. (b) $G'_a = -257.576$ μS, $G'_b = 545.455$ μS: the unique steady-state solution is a limit cycle. Horizontal axis: I_3, 400 μA/div; vertical axis: V_2, 200 mV/div. (From Kennedy, M. P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 647, Oct. 1993. With permission.)

If L, C_2, and G'_b are positive, the steady-state behavior of the circuit depends on the sign of G'_a. When $G'_a > 0$, the circuit is dissipative everywhere and all trajectories collapse toward the origin. The unique steady-state solution of the circuit is the stable dc equilibrium condition $I_3 = V_2 = 0$.

If $G'_a > 0$, N'_R looks like a negative resistor close to the origin and injects energy into the circuit, pushing trajectories away. Further out, where the characteristic has positive slope, trajectories are pulled in by the dissipative vector field. The resulting balance of forces produces a steady-state orbit called a *limit cycle*, which is approached asymptotically by all initial conditions of this circuit.

This limit cycle is said to be *attracting* because nearby trajectories move toward it and it is structurally stable in the sense that, for almost all values of G'_a, a small change in the parameters of the circuit has little effect on it. In the special case when $G'_a \equiv 0$, a perturbation of G'_a causes the steady-state behavior to change from an equilibrium point to a limit cycle; this is called a *bifurcation*.

In the following sections, we consider in detail steady-state behaviors, stability, structural stability, and bifurcations.

14.1.2 Classification and Uniqueness of Steady-State Behaviors

A trajectory of a dynamical system from an initial state $\mathbf{X}_0$ settles, possibly after some transient, onto a set of points called a *limit set*. The **ω**-limit set corresponds to the asymptotic behavior of the system as $t \to +\infty$ and is called the steady-state response. We use the idea of recurrent states to determine when the system has reached steady state.

A state $\mathbf{X}$ of a dynamical system is called recurrent under the flow ϕ if, for every neighborhood $B_\varepsilon(\mathbf{X})$ of $\mathbf{X}$ and for every $T > 0$, there is a time $t > T$ such that $\phi_t(\mathbf{X}) \cap B_\varepsilon(\mathbf{X}) \neq \emptyset$. Thus, a state $\mathbf{X}$ is recurrent if, by waiting long enough, the trajectory through $\mathbf{X}$ repeatedly returns arbitrarily close to $\mathbf{X}$ [7].

Wandering points correspond to transient behavior, while steady-state or asymptotic behavior corresponds to orbits of recurrent states.

A point $\mathbf{X}_\omega$ is an ω-limit point of $\mathbf{X}_0$ if and only if $\lim_{k\to+\infty} \phi_{t_k}(\mathbf{X}_0) = (\mathbf{X}_\omega)$ for some sequence $\{t_k | k \in \mathbb{Z}_+$ such that $t_k \to +\infty$. The set $L(\mathbf{X}_0)$ of ω-limit points of $\mathbf{X}_0$ is called ω-limit set of $\mathbf{X}_0$.*

A limit set L is called attracting if there exists a neighborhood U of L such that $L(\mathbf{X})_0 = L$ for all $\mathbf{X}_0 \in U$. Thus, nearby trajectories converge toward an attracting limit set as $t \to \infty$.

* The set of points to which trajectories converge from $\mathbf{X}_0$ as $t \to -\infty$ is called the α-limit set of $\mathbf{X}_0$. We consider only positive time, therefore, by *limit set*, we mean the ω-limit set.

An attracting set $\mathcal{A}$ that contains at least one orbit that comes arbitrarily close to every point in $\mathcal{A}$ is called an attractor [7].

In an asymptotically stable linear system the limit set is independent of the initial condition and unique so it makes sense to talk of the steady-state behavior. By contrast, a nonlinear system may possess several different limit sets and therefore may exhibit a variety of steady-state behaviors, depending on the initial condition.

The set of all points in the state space that converge to a particular limit set L is called the basin of attraction of L.

Because nonattracting limit sets cannot be observed in physical systems, the asymptotic or steady-state behavior of a real electronic circuit corresponds to motion on an attracting limit set.

14.1.2.1 Equilibrium Point

The simplest steady-state behavior of a dynamical system is an equilibrium point. An equilibrium point or stationary point of Equation 14.3 is a state $\mathbf{X_Q}$ at which the vector field is zero. Thus, $\mathbf{F}(\mathbf{X_Q}) = 0$ and $\phi_t(\mathbf{X_Q}) = \mathbf{X_Q}$; a trajectory starting from an equilibrium point remains indefinitely at that point.

In state space, the limit set consists of a single nonwandering point $\mathbf{X_Q}$. A point is a zero-dimensional object. Thus, an equilibrium point is said to have dimension zero.

In the time domain, an equilibrium point of an equilibrium circuit is simply a dc solution or operating point.

An equilibrium point or fixed point of a discrete-time dynamical system is a point $\mathbf{X_Q}$ that satisfies

$$\mathbf{G}(\mathbf{X_Q}) = \mathbf{X_Q}$$

Example: Nonlinear Parallel *RLC* Circuit

The nonlinear *RLC* circuit shown in Figure 14.2 has just one equilibrium point $(I_{3_Q}, V_{2_Q}) = (0,0)$. When G'_a is positive, a trajectory originating at any point in the state converges to this attracting dc steady-state (as shown in Figure 14.4a). The basin of attraction of the origin is the entire state space.

All trajectories, and not just those that start close to it, converge to the origin, so this equilibrium point is said to be a global attractor.

When $G'_a < 0$, the circuit possesses two steady-state solutions: the equilibrium point at the origin, and the limit cycle Γ. The equilibrium point is unstable in this case. All trajectories, except that which starts at the origin, are attracted to Γ.

14.1.2.2 Periodic Steady-State

A state $\mathbf{X}$ is called periodic if there exists $T > 0$ such that $\phi_T(\mathbf{X}) = \mathbf{X}$. A periodic orbit which is not a stationary point is called a cycle.

A limit cycle Γ is an isolated periodic orbit of a dynamical system (see Figure 14.5b). The limit cycle trajectory visits every point on the simple closed curve Γ with period T. Indeed, $\phi_t(\mathbf{X}) = \phi_{t+T}(\mathbf{X})\ \forall$ $\mathbf{X} \in \Gamma$. Thus, every point on the limit cycle Γ is a nonwandering point.

A limit cycle is said to have dimension one because a small piece of it looks like a one-dimensional object: a line. Then, n components $X_i(t)$ of a limit cycle trajectory $\mathbf{X}(t) = [X_1(t), X_2(t), \ldots, X_n(t)]^T$ in $\mathbb{R}^n$ are periodic time waveforms with period T.

Every periodic signal $X(t)$ may be decomposed into a Fourier series—a weighted sum of sinusoids at integer multiples of a fundamental frequency. Thus, a periodic signal appears in the frequency domain as a set of spikes at integer multiples (harmonics) of the fundamental frequency. The amplitudes of these spikes correspond to the coefficient in the Fourier series expansion of $X(t)$. The Fourier transform is an extension of these ideas to aperiodic signals; one considers the distribution of the signal's power over a continuum of frequencies rather than on a discrete set of harmonics.

The distribution of power in a signal $X(t)$ is most commonly quantified by means of the power density spectrum, often simply called the power spectrum. The simplest estimator of the power spectrum is the periodogram [17] which, given N uniformly spaced samples $X(k/f_s)$, $k = 0, 1, \ldots, N-1$ of $X(t)$, yields $N/2+1$ numbers $P(nf_s/N)$, $n = 0, 1, \ldots, N/2$, where f_s is the sampling frequency.

If one considers the signal $X(t)$ as being composed of sinusoidal components at discrete frequencies, then $P(nf_s/N)$ is an estimate of the power in the component at frequency nf_s/N. By Parseval's theorem, the sum of the power in each of these components equals the mean-squared amplitude of the N samples of $X(t)$ [17].

If $X(t)$ is periodic with period T, then its power will be concentrated in a dc component, a fundamental frequency component $1/T$, and harmonics. In practice, the discrete nature of the sampling process causes power to "leak" between adjacent frequency components; this leakage may be reduced by "windowing" the measured data before calculating the periodogram [17].

Example: Periodic Steady-State Solution

Figure 14.5b depicts a state-space orbit, time waveform, and power spectrum of a periodic steady-state solution of a third-order, autonomous, continuous-time dynamical system.

The orbit in state space is an asymmetric closed curve consisting of four loops. In the time domain, the waveform has four crests per period and a dc offset. In the power spectrum, the dc offset manifests itself as a spike at zero frequency. The period of approximately 270 Hz produces a fundamental component at that frequency. Notice that the fourth harmonic (arising from "four crests per period") has the largest magnitude. This power spectrum is reminiscent of subharmonic mode locking in a forced oscillator.

14.1.2.3 Subharmonic Periodic Steady-State

A subharmonic periodic solution or period-K orbit of a discrete-time dynamical system is a set of K points $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k\}$ that satisfy

$$\mathbf{X}_2 = \mathbf{G}(\mathbf{X}_1), \quad \mathbf{X}_3 = \mathbf{G}(\mathbf{X}_2) \cdots \mathbf{X}_K = \mathbf{G}(\mathbf{X}_{K-1}), \quad \mathbf{X}_1 = \mathbf{G}(\mathbf{X}_K)$$

More succinctly, we may write $\mathbf{X}_i = \mathbf{G}^{(K)}(\mathbf{X}_i)$, where $\mathbf{G}^{(K)} = \mathbf{G}[\mathbf{G}(\cdots[\mathbf{G}(\cdot)]\cdots)]$ denotes $\mathbf{G}$ applied K times to the argument of the map; this is called the Kth iterate of $\mathbf{G}$.

Subharmonic periodic solutions occur in systems that contain two or more competing frequencies, such as forced oscillators or sampled-data circuits. Subharmonic solutions also arise following period-doubling bifurcations (see Section 14.1.5).

14.1.2.4 Quasiperiodic Steady-State

The next most complicated form of steady-state behavior is called quasiperiodicity. In state space, this corresponds to a *torus* (see Figure 14.5a). Although a small piece of a limit cycle in $\mathbb{R}^3$ looks like a line, a small section of two-torus looks like a plane; a two-torus has dimension two.

A quasiperiodic function is one that may be expressed as a countable sum of periodic functions with incommensurate frequencies, i.e., frequencies that are not rationally related. For example, $X(t) = \sin(t) + \sin(2\pi t)$ is a quasiperiodic signal. In the time domain, a quasiperiodic signal may look like an amplitude-or phase-modulated waveform.

Although the Fourier spectrum of a periodic signal consists of a discrete set of spikes at integer multiples of a fundamental frequency, that of a quasiperiodic solution comprises a discrete set of spikes at incommensurate frequencies, as presented in Figure 14.5a.

In principle, a quasiperiodic signal may be distinguished from a periodic one by determining whether the frequency spikes in the Fourier spectrum are harmonically related. In practice, it is impossible to

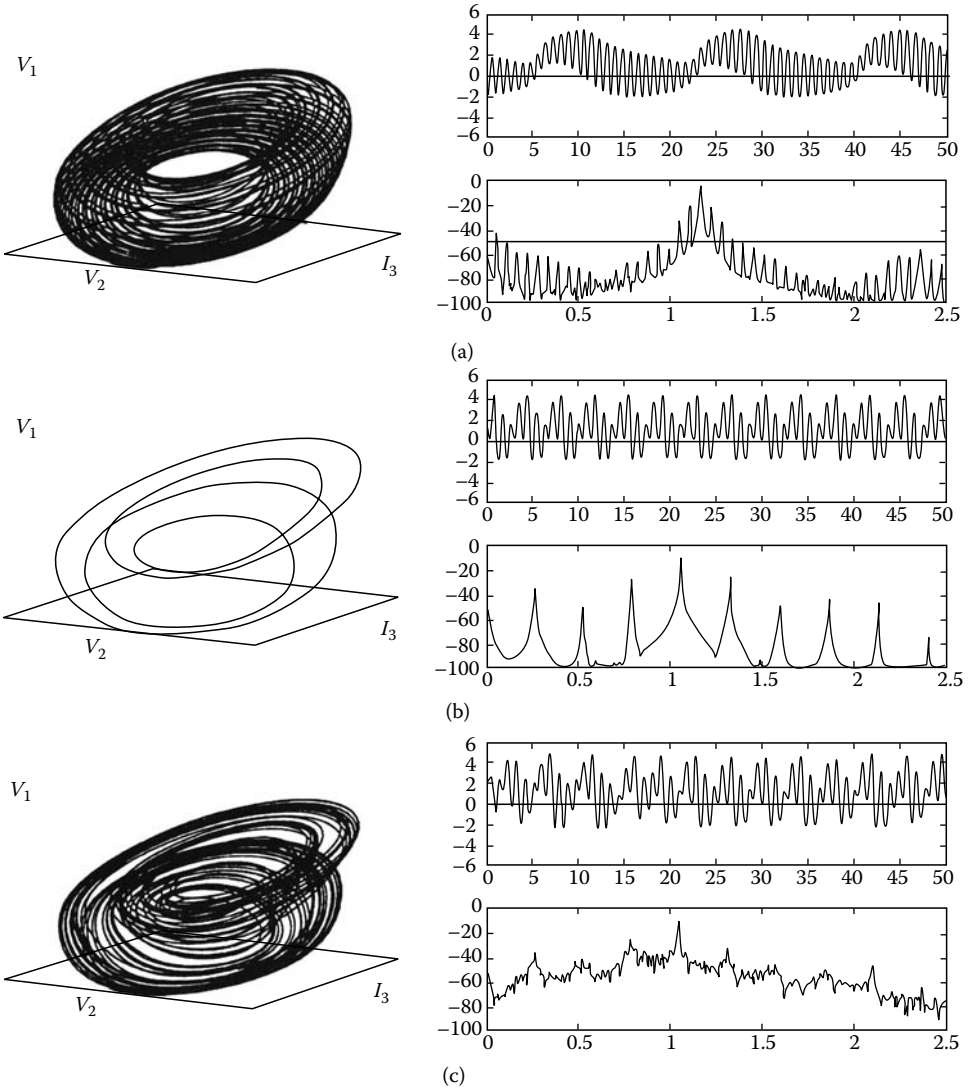


FIGURE 14.5 Quasiperiodicity (torus breakdown) route to chaos in Chua's oscillator. Simulated state-space trajectories, time waveforms $V_1(t)$, and power spectra of $V_2(t)$. (a) Quasiperiodic steady-state—the signal is characterized by a discrete power spectrum with incommensurate frequency components; (b) periodic window—all spikes in the power spectrum are harmonically related to the fundamental frequency; (c) chaotic steady-state following breakdown of the torus—the waveform has a broadband power spectrum. Time plots; horizontal axis—t (ms); vertical axis—V_1 (V). Power spectra: horizontal axis—frequency (kHz); vertical axis—power (mean-squared amplitude) of $V_2(t)$ (dB).

determine whether a measured number is rational or irrational; therefore, any spectrum that appears to be quasiperiodic may simply be periodic with an extremely long period.

A two-torus in a three-dimensional state space looks like a doughnut. Quasiperiodic behavior on a higher dimensional torus is more difficult to visualize in state space but appears in the power spectrum as a set of discrete components at incommensurate frequencies. A K-torus has dimension K.

Quasiperiodic behavior occurs in discrete-time systems where two incommensurate frequencies are present. A periodically forced or discrete-time dynamical system has a frequency associated with

the period of the forcing or sampling interval of the system; if a second frequency is introduced that is not rationally related to the period of the forcing or the sampling interval, then quasiperiodicity may occur.

Example: Discrete Torus

Consider a map from the circle S^1 onto itself. In polar coordinates, a point on the circle is parameterized by an angle $\boldsymbol{\theta}$. Assume that $\boldsymbol{\theta}$ has been normalized so that one complete revolution of the circle corresponds to a change in $\boldsymbol{\theta}$ of 1. The state of this system is determined by the normalized angle $\boldsymbol{\theta}$ and the dynamics by

$$\boldsymbol{\theta}_{k+1} = (\boldsymbol{\theta}_k + \boldsymbol{\Omega}) \bmod 1$$

If $\boldsymbol{\Omega}$ is a rational number (of the form J/K where $J, K \in \mathbb{Z}_+$), then the steady-state solution is a period-**K** (subharmonic) orbit. If $\boldsymbol{\Omega}$ is irrational, we obtain quasiperiodic behavior.

14.1.2.5 Chaotic Steady-State

DC equilibrium periodic as well as quasiperiodic steady-state behaviors have been correctly identified and classified since the pioneering days of electronics in the 1920s. By contrast, the existence of more exotic steady-state behaviors in electronic circuits has been acknowledged only in the past 30 years. Although the notion of chaotic behavior in dynamical systems has existed in the mathematics literature since the turn of the century, unusual behaviors in the physical sciences as recently as the 1960s were described as ''strange.'' Today, we classify as *chaos* the recurrent* motion in a deterministic dynamical system, which is characterized by a positive Lyapunov exponent (LE).

From an experimentalist's point of view, chaos may be defined as bounded steady-state behavior in a deterministic dynamical system that is not an equilibrium point, nor periodic, and not quasiperiodic [15].

Chaos is characterized by repeated stretching and folding of bundles of trajectories in state space. Two trajectories started from almost identical initial conditions diverge and soon become uncorrelated; this is called sensitive dependence on initial conditions and gives rise to long-term unpredictability.

In the time domain, a chaotic trajectory is neither periodic nor quasiperiodic, but looks ''random.'' This ''randomness'' manifests itself in the frequency domain as a broad ''noise-like'' Fourier spectrum, as presented in Figure 14.5c.

Although an equilibrium point, a limit cycle, and a K-torus each have integer dimension, the repeated stretching and folding of trajectories in a chaotic steady-state gives the limit set a more complicated structure that, for three-dimensional continuous-time circuits, is something more than a surface but not quite a volume.

14.1.2.6 Dimension

The structure of a limit set $L \subset \mathbb{R}^n$ of a dynamical system may be quantified using a generalized notion of dimension that considers not just the geometrical structure of the set, but also the time evolution of trajectories on L.

Capacity (D_0 *Dimension*)

The simplest notion of dimension, called capacity (or D_0 dimension), considers a limit set simply as set of points, without reference to the dynamical system that produced it.

* Because a chaotic steady-state does not settle down onto a single well-defined trajectory, the definition of recurrent states must be used to identify posttransient behavior.

To estimate the capacity of L, cover the set with n-dimensional cubes having side length ε. If L is a D_0-dimensional object, then the minimum number $N(\varepsilon)$ of cubes required to cover L is proportional to ε^{-D_0}. Thus, $N(\varepsilon) \propto \varepsilon^{-D_0}$.

The D_0 dimension is given by

$$D_0 = \lim_{\varepsilon \to 0} - \frac{\ln N(\varepsilon)}{\ln \varepsilon}$$

When this definition is applied to a point, a limit cycle (or line), or a two-torus (or surface) $\mathbb{R}^3$, the calculated dimensions are 0, 1, and 2, respectively, as expected. When applied to the set of nonwandering points that comprise a chaotic steady-state, the D_0 dimension is typically noninteger. An object that has noninteger dimension is called a fractal.

Example: The Middle-Third Cantor Set

Consider the set of points that is obtained by repeatedly deleting the middle third of an interval, as indicated in Figure 14.6a. At the first iteration, the unit interval is divided into 2^1 pieces of length 1/3 each; after k iterations, the set is covered by 2^k pieces of length $1/3^k$. By contrast, the set that is obtained by dividing the intervals into thirds but not throwing away the middle third each time (Figure 14.6b) is covered at the kth step by 3^k pieces of length $1/3^k$.

Applying the definition of capacity, the dimension of the unit interval is

$$\lim_{k \to \infty} \frac{k \ln 3}{k \ln 3} = 1.00$$

By contrast, the middle-third Cantor set has dimension

$$\lim_{k \to \infty} \frac{k \ln 2}{k \ln 3} \approx 0.63$$

The set is something more than a zero-dimensional object (a point) but not quite one-dimensional (like a line segment); it is a fractal.

Correlation (D_2) Dimension

The D_2 dimension considers not just the geometry of a limit set, but also the time evolution of trajectories on the set.

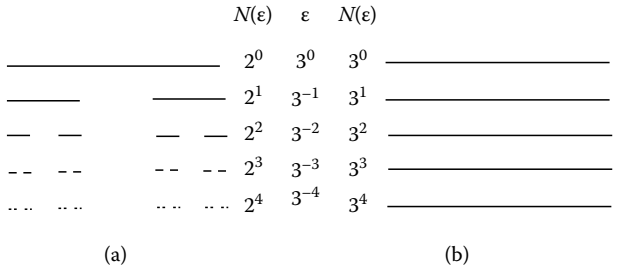


FIGURE 14.6 (a) The Middle-third Cantor set is obtained by recursively removing the central portion of an interval. At the kth step, the set consists of $N(\varepsilon) = 2^k$ pieces of length $\varepsilon = 3^{-k}$. The limit set has capacity 0.63. (b) By contrast, the unit interval is covered by 3^k pieces of length 3^{-k}. The unit interval has dimension 1.00.

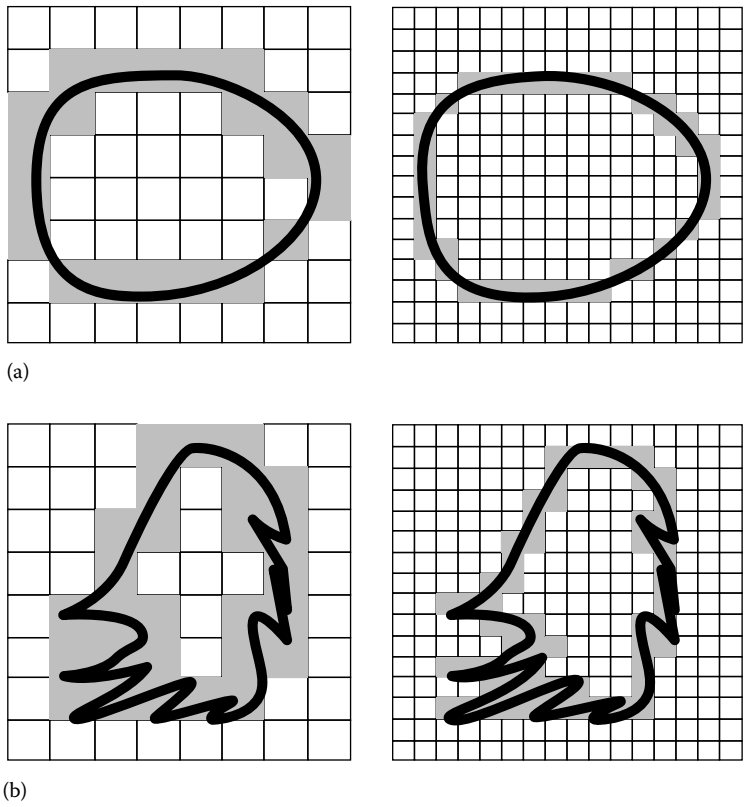


FIGURE 14.7 Coverings of two limit sets L_a (a) and L_b (b) with squares of side length ε_0 and $\varepsilon_0/2$, respectively.

Consider the two limit sets L_a and L_b in $\mathbb{R}^2$ shown in Figure 14.7a and b, respectively. The D_0 dimension of these sets may be determined by iteratively covering them with squares (two-dimensional "cubes") of side length $\varepsilon = \varepsilon_0/2^k$, $k = 0, 1, 2, \ldots$, counting the required number of squares $N(\varepsilon)$ for each ε, and evaluating the limit

$$D_0 = \lim_{k \to \infty} -\frac{\ln N(\varepsilon)}{\ln(\varepsilon)}$$

For the smooth curve L_a, the number of squares required to cover the set grows linearly with $1/\varepsilon$; hence $D_0 = 1.0$. By contrast, if the kinks and folds in set L_b are present at all scales, then the growth of $N(\varepsilon)$ versus $1/\varepsilon$ is superlinear and the object has a noninteger D_0 dimension between 1.0 and 2.0.

Imagine now that L_a and L_b are not simply static geometrical objects but are orbits of discrete-time dynamical systems. In this case, a steady-state trajectory corresponds to a sequence of points moving around the limit set.

Cover the limit set with the minimum number $N(\varepsilon)$ of "cubes" with side length ε, and label the boxes 1, 2, ..., i, ..., $N(\varepsilon)$. Count the number of times $n_i(N, \varepsilon)$ that a typical steady-state trajectory of length N visits box i and define

$$p_i = \lim_{N \to \infty} \frac{n_i(N, \varepsilon)}{N}$$

where p_i is the relative frequency with which a trajectory visits the ith cube. The D_2 dimension is defined as

$$D_2 = \lim_{\varepsilon \to 0} \frac{\ln \sum_{i=1}^{N(\varepsilon)} p_i^2}{\ln \varepsilon}$$

In general, $D_2 \leq D_0$ with equality when a typical trajectory visits all $N(\varepsilon)$ cubes with the same relative frequency $p = 1/N(\varepsilon)$. In this special case,

$$\begin{aligned} D_2 &= \lim_{\varepsilon \to 0} \frac{\ln \sum_{i=1}^{N(\varepsilon)} \frac{1}{N(\varepsilon)^2}}{\ln \varepsilon} \\ &= \lim_{\varepsilon \to 0} -\frac{\ln N(\varepsilon)}{\ln \varepsilon} \\ &= D_0 \end{aligned}$$

An efficient algorithm (developed by Grassberger and Procaccia) for estimating D_2 is based on the approximation $\sum_{i=1}^{N(e)} p_i^2 \approx C(\varepsilon)$ [15], where $C(\varepsilon) = \lim_{N \to 0} \frac{1}{N^2}$ (the number of pairs of points (X_i, X_j) such that $\|X_i - X_j\| < \varepsilon$) is called the correlation. The D_2 or correlation dimension is given by

$$D_2 = \lim_{\varepsilon \to 0} \frac{\ln C(\varepsilon)}{\ln \varepsilon}$$

Example: Correlation (D_2) Dimension

The correlation dimension of the chaotic attractor in Figure 14.5c, estimated using INSITE, is 2.1, while D_2 for the uniformly covered torus in Figure 14.5a is 2.0.

14.1.3 Stability of Steady-State Trajectories

Consider once more the nonlinear *RLC* circuit in Figure 14.2. If G'_a is negative, this circuit settles to a periodic steady-state from almost every initial condition. However, a trajectory started from the origin will, in principle, remain indefinitely at the origin since this is an equilibrium point. The circuit has two possible steady-state solutions. Experimentally, only the limit cycle will be observed. Why?

If trajectories starting from states close to a limit set converge to that steady state, the limit set is called an attracting limit set. If, in addition, the attracting limit set contains at least one trajectory that comes arbitrarily close to every point in the set, then it is an attractor. If nearby points diverge from the limit set, it is called a repellor.

In the nonlinear *RLC* circuit with $G'_a < 0$, the equilibrium point is a repellor and the limit cycle is an attractor.

14.1.3.1 Stability of Equilibrium Points

Qualitatively, an equilibrium point is said to be stable if trajectories starting close to it remain nearby for all future time and unstable otherwise. Stability is a local concept, dealing with trajectories in a small neighborhood of the equilibrium point.

To analyze the behavior of the vector field in the vicinity of an equilibrium point $\mathbf{X_Q}$, we write $\mathbf{X} = \mathbf{X_Q} + \mathbf{x}$ and substitute into Equation 14.3 to obtain

$$\begin{aligned} \dot{\mathbf{X}}_\mathbf{Q} + \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{X}_\mathbf{Q} + \mathbf{x}) \\ \mathbf{F}(\mathbf{X}_\mathbf{Q}) + \dot{\mathbf{x}} &\approx \mathbf{F}(\mathbf{X}_\mathbf{Q}) + \mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}_\mathbf{Q})\mathbf{x} \end{aligned} \tag{14.9}$$

where we have kept just the first two terms of the Taylor series expansion of $\mathbf{F}(\mathbf{X})$ about $\mathbf{X}_\mathbf{Q}$. The Jacobian matrix $\mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X})$ is the matrix of partial derivatives of $\mathbf{F}(\mathbf{X})$ with respect to $\mathbf{X}$:

$$\mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \frac{\partial F_1(\mathbf{X})}{\partial X_1} & \frac{\partial F_1(\mathbf{X})}{\partial X_2} & \cdots & \frac{\partial F_1(\mathbf{X})}{\partial X_n} \\ \frac{\partial F_2(\mathbf{X})}{\partial X_1} & \frac{\partial F_2(\mathbf{X})}{\partial X_2} & \cdots & \frac{\partial F_2(\mathbf{X})}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n(\mathbf{X})}{\partial X_1} & \frac{\partial F_n(\mathbf{X})}{\partial X_2} & \cdots & \frac{\partial F_n(\mathbf{X})}{\partial X_n} \end{bmatrix}$$

Subtracting $\mathbf{F}(\mathbf{X}_\mathbf{Q})$ from both sides of Equation 14.9 we obtain the linear system

$$\dot{\mathbf{x}} = \mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}_\mathbf{Q})\mathbf{x}$$

where the Jacobian matrix is evaluated at $\mathbf{X}_\mathbf{Q}$. This linearization describes the behavior of the circuit in the vicinity of $\mathbf{X}_\mathbf{Q}$; we call this the local behavior.

Note that the linearization is simply the small-signal equivalent circuit at the operating point $\mathbf{X}_\mathbf{Q}$. In general, the local behavior of a circuit depends explicitly on the operating point $\mathbf{X}_\mathbf{Q}$. For example, a *pn*-junction diode exhibits a small incremental resistance under forward bias, but a large small-signal resistance under reverse bias.

14.1.3.2 Eigenvalues

If $\mathbf{X}_\mathbf{Q}$ is an equilibrium point of Equation 14.3, a complete description of its stability is contained in the eigenvalues of the linearization of Equation 14.3 about $\mathbf{X}_\mathbf{Q}$. These are defined as the roots λ of the characteristic equation

$$\det[\lambda\mathbf{I} - \mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}_\mathbf{Q})] = 0 \tag{14.10}$$

where $\mathbf{I}$ is the identity matrix.

If the real parts of all of the eigenvalues $\mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}_\mathbf{Q})$ are strictly negative, then the equilibrium point $\mathbf{X}_\mathbf{Q}$ is asymptotically stable and is called a sink because all nearby trajectories converge toward it.

If any of the eigenvalues has a positive real part, the equilibrium point is unstable; if all of the eigenvalues have positive real parts, the equilibrium point is called a source. An equilibrium point that has eigenvalues with both negative and positive real parts is called a saddle. A saddle is unstable.

An equilibrium point is said to be hyperbolic if all the eigenvalues of $\mathbf{D}_\mathbf{x}\mathbf{F}(\mathbf{X}_\mathbf{Q})$ have nonzero real parts. All hyperbolic equilibrium points are either unstable or asymptotically stable.

14.1.3.3 Discrete-Time Systems

The stability of a fixed point $\mathbf{X}_\mathbf{Q}$ of a discrete-time dynamical system

$$\mathbf{X}_{k+1} = \mathbf{G}(\mathbf{X}_k)$$

is determined by the eigenvalues of the linearization $\mathbf{D}_\mathbf{X}\mathbf{G}(\mathbf{X}_\mathbf{Q})$ of the vector field $\mathbf{G}$, evaluated at $\mathbf{X}_\mathbf{Q}$.

The equilibrium point is classified as stable if all of the eigenvalues of $\mathbf{D}_\mathbf{X}\mathbf{G}(\mathbf{X}_\mathbf{Q})$ are strictly less than unity in modulus, and unstable if any has modulus greater than unity.

14.1.3.4 Eigenvectors, Eigenspaces, Stable and Unstable Manifolds

Associated with each distinct eigenvalue λ of the Jacobian matrix $\mathbf{D_X F(X_Q)}$ is an eigenvector $\vec{\mathbf{v}}$ defined by

$$\mathbf{D_X F(X_Q)}\vec{v} = \lambda\vec{v}$$

A real eigenvalue γ has a real eigenvector $\vec{\eta}$. Complex eigenvalues of a real matrix occur in pairs of the form $\sigma \pm j\omega$. The real and imaginary parts of the associated eigenvectors $\vec{\eta}_r \pm j\vec{\eta}_c$ span a plane called a complex eigenplane.

The n_s-dimensional subspace of $\mathbb{R}^n$ associated with the stable eigenvalues of the Jacobian matrix is called the stable eigenspace, denoted $E^s(\mathbf{X_Q})$. The n_u-dimensional subspace corresponding to the unstable eigenvalues is called the unstable eigenspace, denoted $E^u(\mathbf{X_Q})$.

The analogs of the stable and unstable eigenspaces for a general nonlinear system are called the local stable and unstable *manifolds** $W^s(\mathbf{X_Q})$ and $W^u(\mathbf{X_Q})$.

The stable manifold $W^s(\mathbf{X_Q})$ is defined as the set of all states from which trajectories remain in the manifold and converge under the flow to $\mathbf{X_Q}$. The unstable manifold $W^u(\mathbf{X_Q})$ is defined as the set of all states from which trajectories remain in the manifold and diverge under the flow from $\mathbf{X_Q}$.

By definition, the stable and unstable manifolds are invariant under the flow (if $\mathbf{X} \in W^s$, then $\phi_t(\mathbf{X}) \in W^s$). Furthermore, the n_s- and n_u-dimensional tangent spaces to W^s and W^u at $\mathbf{X_Q}$ are E^s and E^u (as shown in Figure 14.8). In the special case of a linear or affine vector field **F**, the stable and unstable manifolds are simply the eigenspaces E^s and E^u.

Chaos is associated with two characteristic connections of the stable and unstable manifolds. A homoclinic orbit (see Figure 14.9a) joins an isolated equilibrium point $\mathbf{X_Q}$ to itself along its stable and unstable manifolds. A heteroclinic orbit (Figure 14.9b) joins two distinct equilibrium points, $\mathbf{X_{Q1}}$ and $\mathbf{X_{Q2}}$, along the unstable manifold of one and the stable manifold of the other.

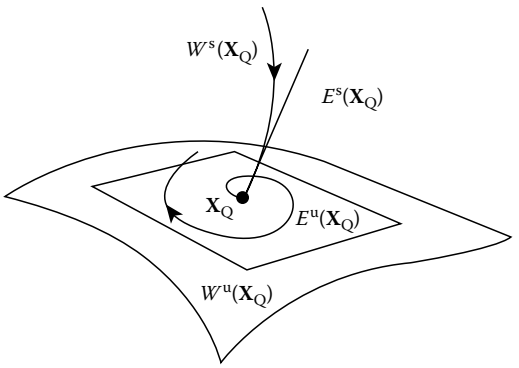


FIGURE 14.8 Stable and unstable manifolds $W^s(\mathbf{X_Q})$ and $W^u(\mathbf{X_Q})$ of an equilibrium point $\mathbf{X_Q}$. The stable and unstable eigenspaces $E^s(\mathbf{X_Q})$ and $E^u(\mathbf{X_Q})$ derived from the linearization of the vector field at $\mathbf{X_Q}$ are tangent to the corresponding manifolds W^s and W^u at $\mathbf{X_Q}$. A trajectory approaching the equilibrium point along the stable manifold is tangential to $E^s(\mathbf{X_Q})$ at $\mathbf{X_Q}$; a trajectory leaving $\mathbf{X_Q}$ along the unstable manifold is tangential to $E^u(\mathbf{X_Q})$ at $\mathbf{X_Q}$.

14.1.3.5 Stability of Limit Cycles

Although the stability of an equilibrium point may be determined by considering the eigenvalues of the linearization of the vector field near the point, how does one study the stability of a limit cycle, torus, or chaotic steady-state trajectory?

The idea introduced by Poincaré is to convert a continuous-time dynamical system into an equivalent discrete-time dynamical system by taking a transverse slice through the flow. Intersections of trajectories with this so-called Poincaré section define a Poincaré map from the section to themselves. Since the limit cycle is a fixed point $\mathbf{X_Q}$ of the associated discrete-time dynamical system, its stability may be determined by examining the eigenvalues of the linearization of the Poincaré map at $\mathbf{X_Q}$.

* An m-dimensional manifold is a geometrical object every small section of which looks like $\mathbb{R}^m$. More precisely, M is an m-dimensional manifold if, for every $x \in M$, there exists an open neighborhood U of x and a smooth invertible map which takes U to some open neighborhood of $\mathbb{R}^m$. For example, a limit cycle of a continuous-time dynamical system is a one-dimensional manifold.

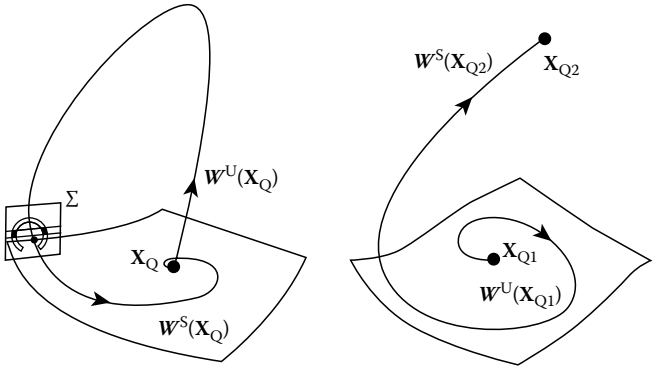


FIGURE 14.9 (a) Homoclinic orbit joins an isolated equilibrium point $\mathbf{X_Q}$ to itself along its stable and unstable manifolds. (b) A heteroclinic orbit joins two distinct equilibrium points, $\mathbf{X_{Q1}}$ and $\mathbf{X_{Q2}}$, along the unstable manifolds of one and the stable manifold of the other.

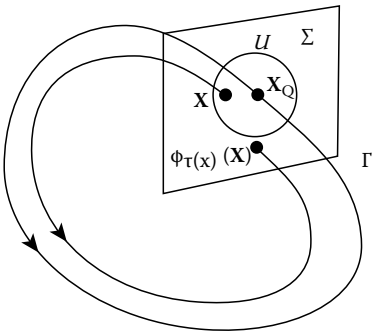


FIGURE 14.10 Transverse Poincaré section Σ through the flow of a dynamical system induces a discrete Poincaré map from a neighborhood U of the point of intersection X_Q to Σ.

14.1.3.6 Poincaré Sections

A Poincaré section of an n-dimensional autonomous continuous-time dynamical system is an $(n-1)$-dimensional hyperplane Σ in the state space that is intersected transversally* by the flow.

Let Γ be a closed orbit of the flow of a smooth vector field $\mathbf{F}$, and let $\mathbf{X_Q}$ be a point of intersection of Γ with Σ. If T is the period of Γ and $\mathbf{X} \in \Sigma$ is sufficiently close to $\mathbf{X_Q}$, then the trajectory $\phi_t(\mathbf{X})$ through $\mathbf{X}$ will return to Σ after a time $\tau(\mathbf{X}) \approx T$ and intersect the hyperplane at a point $\phi_{\tau(\mathbf{X})}(\mathbf{X})$, as illustrated in Figure 14.10.

This construction implicitly defines a function (called a Poincaré map or first return map) $\mathbf{G}$: $U \rightarrow \Sigma$

$$\mathbf{G}(\mathbf{X}) = \phi_{\tau(\mathbf{X})}(\mathbf{X})$$

where U is a small region of Σ close to $\mathbf{X_Q}$. The corresponding discrete-time dynamical system

$$\mathbf{X}_{k+1} = \mathbf{G}(\mathbf{X}_k)$$

has a fixed point at $\mathbf{X_Q}$.

The stability of the limit cycle is determined by the eigenvalues of the linearization $\mathbf{D_X G}(\mathbf{X_Q})$ of $\mathbf{G}$ at $\mathbf{X_Q}$. If all of the eigenvalues of $\mathbf{D_X G}(\mathbf{X_Q})$ have modulus less than unity, the limit cycle is asymptotically stable; if any has modulus greater than unity, the limit cycle is unstable.

Note that the stability of the limit cycle is independent of the position and orientation of the Poincaré plane, provided that the intersection is chosen *transverse* to the flow. For a nonautonomous system with periodic forcing, a natural choice for the hyperplane is at a fixed phase θ_o of the forcing.

* A *transverse* intersection of manifolds in $\mathbb{R}^n$ is an intersection of manifolds such that, from any point in the intersection, all directions in $\mathbb{R}^n$ can be generated by linear combinations of vectors tangent to the manifolds.



(a) (b) (c)

FIGURE 14.11 Experimental Poincaré sections corresponding to a torus breakdown sequence in Chua's oscillator. (a) Torus, (b) period-four orbit, (c) chaotic attractor resulting from torus breakdown. (From Chua, L.O., Wu, C.W., Hung, A., and Zhong, G.-Q., *IEEE Trans. Circuits Syst*., 40, 738, Oct. 1993. With permission.)

In the Poincaré section, a limit cycle looks like a fixed point. A period-K subharmonic of a non-autonomous system with periodic forcing appears as a period-K orbit of the corresponding map (see Figure 14.11b).

The Poincaré section of a quasiperiodic attractor consisting of two incommensurate frequencies looks like a closed curve—a transverse cut through a two-torus (Figure 14.11a).

The Poincaré section of chaotic attractor has fractal structure, as depicted in Figure 14.11c.

14.1.4 Horseshoes and Chaos

Chaotic behavior is characterized by sensitive dependence on initial conditions. This phrase emphasizes the fact that small differences in initial conditions are persistently magnified by the dynamics of the system so that trajectories starting from nearby initial conditions reach totally different states in a finite time.

Trajectories of the nonlinear *RLC* circuit in Figure 14.2 that originate near the equilibrium point are initially stretched apart exponentially by the locally negative resistance in the case $G'_a < 0$. Eventually, however, they are squeezed together onto a limit cycle, so the stretching is not persistent. This is a consequence of the noncrossing property and eventual passivity.

Although perhaps locally active, every physical resistor is eventually passive meaning that, for a large enough voltage across its terminals, it dissipates power. This in turn limits the maximum values of the voltages and currents in the circuit giving a bounded steady-state solution. All physical systems are bounded, so how can small differences be magnified persistently in a real circuit?

14.1.4.1 Chaos in the Sense of Shil'nikov

Consider a flow ϕ in $\mathbb{R}^3$ that has an equilibrium point at the origin with a real eigenvalue $\gamma > 0$ and a pair of complex conjugate eigenvalues $\sigma \pm j\omega$ with $\sigma < 0$ and $\omega \neq 0$. Assume that the flow has a homoclinic orbit Γ through the origin.

One may define a Poincaré map for this system by taking a transverse section through the homoclinic orbit, as illustrated in Figure 14.9a.

THEOREM 14.2 (Shil'nikov)

If $|\sigma/\gamma| < 1$, *the flow* ϕ *can be perturbed to* ϕ' *such that* ϕ' *has a homoclinic orbit* Γ' *near* Γ *and the Poincaré map of* ϕ' *defined in a neighborhood of* Γ' *has a countable number of horseshoes in its discrete dynamics.*

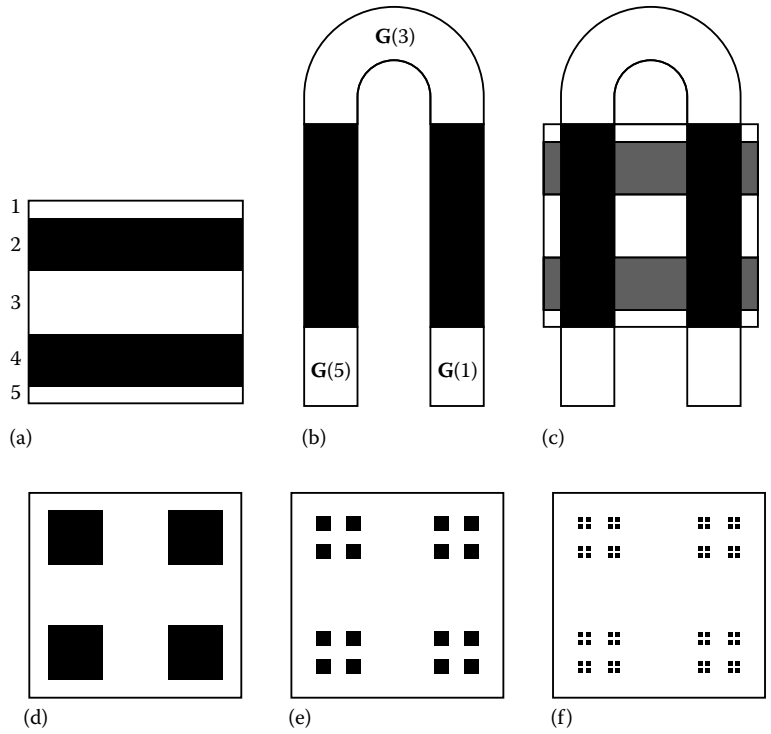


FIGURE 14.12 The Smale horseshoe map stretches the unit square (a), folds it into a horseshoe (b), and lays it back on itself (c), so that only points lying in bands 2 and 4 of (a) are mapped into the square. At the next iteration, only those points in $(\mathbf{G}(2) \cup \mathbf{G}(4)) \cap (2 \cup 4)$ (d) are mapped back to the square. Repeated iterations of the map (d)–(f) remove all points from the square except an invariant (fractal) set of fixed points.

The characteristic horseshoe shape in the Poincaré map stretches and folds trajectories repeatedly (see Figure 14.12). The resulting dynamics exhibit extreme sensitivity to initial conditions [7].

The presence of horseshoes in the flow of a continuous-time system that satisfies the assumptions of Shil'nikov's theorem implies the existence of countable numbers of unstable periodic orbits of arbitrarily long period as well as an uncountable number of complicated bounded nonperiodic chaotic solutions [7].

Horseshoes

The action of the Smale horseshoe map is to take the unit square (Figure 14.12a), stretch it, fold it into a horseshoe shape (Figure 14.12b), and lay it down on itself (Figure 14.12c). Under the action of this map, only four regions of the unit square are returned to the square.

Successive iterations of the horseshoe map return smaller and smaller regions of square to itself, as shown in Figure 14.12d through f. If the map is iterated *ad infinitum*, the unit square is ultimately mapped onto a set of points. These points form an invariant (fractal) limit set L that contains a countable set of periodic orbits of arbitrarily long periods, an uncountable set of bounded nonperiodic orbits, and at least one orbit that comes arbitrarily close to every point in L.

The properties of the map still hold if the horseshoe is distorted by a perturbation of small size but arbitrary shape. Thus, the dynamical behavior of the horseshoe map is structurally stable.*

Although the invariant limit set of a horseshoe map consists of nonwandering points, it is not attracting. Therefore, the existence of a horseshoe in the flow of a third-order system does not imply

* *Structural stability* is discussed in more detail in Section 14.1.5.

that the system will exhibit chaotic steady-state behavior. However if a typical trajectory in the Poincaré map remains in a neighborhood of the invariant set, then the system may exhibit chaos. Thus, although Shil'nikov's theorem is a strong indicator of choas, it does not provide definitive proof that a system is chaotic.

Example: Chaos in a Piecewise-Linear System

Although we have stated it for the case $\sigma < 0$, $\gamma > 0$, Shil'nikov's theorem also applies when the equilibrium point at the origin has an unstable pair of complex conjugate eigenvalues and a stable real eigenvalue. In that case, it is somewhat easier to visualize the stretching and folding of bundles of trajectories close to a homoclinic orbit.

Consider the trajectory in a three-region piecewise-linear vector field in Figure 14.13. We assume that the equilibrium point P_- has a stable real eigenvalue γ_1 (where the eigenvector is $E^r(P_-)$) and an unstable complex conjugate pair of eigenvalues $\sigma_1 \pm j\omega_1$, the real and imaginary parts of whose eigenvectors span the plane $E^c(P_-)$ [2], as illustrated. A trajectory originating from a point $\mathbf{X_0}$ on $E^c(P_-)$ spirals away from the equilibrium point along $E^c(P_-)$ until it enters the D_0 region, where it is folded back into D_{-1}. Upon reentering D_{-1}, the trajectory is pulled toward P_- roughly in the direction of the real eigenvector $E^r(P_-)$, as illustrated.

Now imagine what would happen if the trajectory entering D_{-1} from D_0 were in precisely the direction $E^r(P_-)$. Such a trajectory would follow $E^r(P_-)$ toward P_-, reaching the equilibrium point asymptotically as $t \to \infty$. Similarly, if we were to follow this trajectory backward in time through D_0 and back onto $E^c(P_-)$ in D_{-1}, it would then spiral toward P_-, reaching it asymptotically as $t \to -\infty$. The closed curve thus formed would be a homoclinic orbit, reaching the same equilibrium point P_- asymptotically in forward and reverse time.

Although the homoclinic orbit itself is not structurally stable, and therefore cannot be observed experimentally, horseshoes are structurally stable. A flow ϕ that satisfies the assumptions of Shil'nikov's theorem contains a countable infinity of horseshoes; for sufficiently small perturbations ϕ' of the flow, finitely many of the horseshoes will persist. Thus, both the original flow and the perturbed flow exhibit chaos in the sense of Shil'nikov.

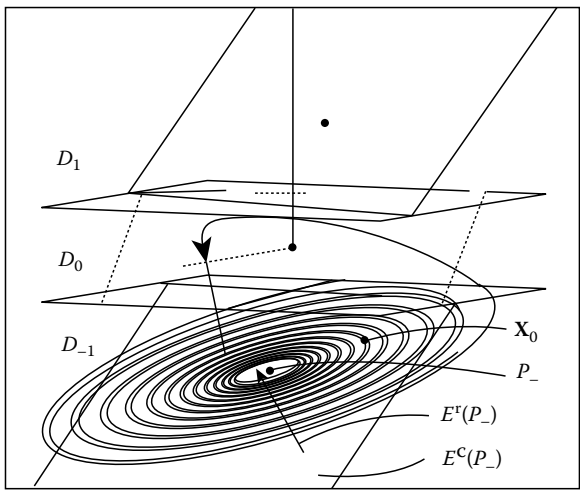


FIGURE 14.13 Stretching and folding mechanism of chaos generation in Chua's circult. A trajectory spirals away from the equilibrium point P_- along the eigenplane $E^c(P_-)$ until it enters the D_0 region, where it is folded back into D_{-1} and returns to the unstable eigenplane $E^c(P_-)$ close to P_-. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 657, Oct. 1993. With permission.)

In Figure 14.13, we see that a trajectory lying close to a homoclinic orbit exhibits similar qualitative behavior: it spirals away from P_- along the unstable complex plane $E^c(P_-)$, is folded in D_0, reenters D_{-1} above $E^c(P_-)$, and is pulled back toward $E^c(P_-)$, only to be spun away from P_- once more.

Thus, two trajectories starting from distinct initial states close to P_- on $E^c(P_-)$ are stretched apart exponentially along the unstable eigenplane before being folded in D_1 and reinjected close to P_-; this gives rise to sensitive dependence on initial conditions. The recurrent stretching and folding continues *ad infinitum*, producing a chaotic steady-state solution.

14.1.4.2 Lyapunov Exponents

The notion of sensitive dependence on initial conditions may be made more precise through the concept of LEs. LEs quantify the average exponential rates of separation of trajectories along the flow.

The flow in a neighborhood of asympototically stable trajectory is contracting so the LEs are zero or negative.* Sensitive dependence on initial conditions results from a positive LE.

To determine the stability of an equilibrium point, we consider the eigenvalues of the linearization of the vector field in the vicinity of equilibrium trajectory. This idea can be generalized to any trajectory of the flow.

The local behavior of the vector field along a trajectory $\phi_t(\mathbf{X}_0)$ of an autonomous continuous-time dynamical system (Equation 14.3) is governed by the linearized dynamics

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{D}_X\mathbf{F}(\mathbf{X})\mathbf{x}, \quad \mathbf{x}(0) = \mathrm{x}_0 \\ &= \mathbf{D}_X\mathbf{F}[\phi_t(\mathbf{X}_0)]\mathbf{x}\end{aligned}$$

This is a linear time-varying system where the state transition matrix, $\Phi_t(\mathbf{X}_0)$, maps a point $\mathbf{x}_0$ into $\mathbf{x}(t)$. Thus

$$\mathbf{x}(t) = \mathbf{\Phi}_t(\mathbf{X}_0)\mathbf{x}_0$$

Note that Φ_t is a linear operator. Therefore, a ball $B_\varepsilon(\mathbf{X}_0)$ of radius ε about $\mathbf{X}_0$ is mapped into an ellipsoid, as presented in Figure 14.1. The principal axes of the ellipsoid are determined by the singular values of Φ_t.

The singular values $\sigma_1(t), \sigma_2(t), \ldots, \sigma_n(t)$ of Φ_t are defined as the square roots of the eigenvalues of $\mathbf{\Phi}_t^H\mathbf{\Phi}_t$, where $\mathbf{\Phi}_t^H$ is the complex conjugate transpose of Φ_t. The singular values are ordered so that $\sigma_1(t) > \sigma_2(t) > \cdots > \sigma_n(t)$.

In particular, a ball of radius ε is mapped by the linearized flow into an ellipsoid (see Figure 14.1), the maximum and minimum radii of which are bounded by $\sigma_1(t)\varepsilon$ and $\sigma_n(t)\varepsilon$, respectively.

The stability of a steady-state orbit is governed by the average local rates of expansion and contraction of volumes of state space close to the orbit. The LEs λ_i are defined by

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \ln \sigma_i(t)$$

whenever this limit exists. The LEs quantify the average exponential rates of separation of trajectories along the flow.

The LEs are a property of a steady-state trajectory. Any transient effect is averaged out by taking the limit as $t \to \infty$. Furthermore, the LEs are global quantities of an attracting set that depend on the local stability properties of a trajectory within the set.

* A continuous flow that has a bounded trajectory not tending to an equlibrium point has a zero LE (in the direction of flow).

The set $\{\lambda_i, i = 1, 2, \ldots, n\}$ is called the Lyapunov spectrum. An attractor has the property that the sum of its LEs is negative.

Lyapunov Exponents of Dicrete-Time Systems

The local behavior along an orbit of the autonomous discrete-time dynamical system (Equation 14.8) is governed by the linearized dynamics

$$\mathbf{x}_{k+1} = \mathbf{D}_x\mathbf{G}(\mathbf{x}_k)\mathbf{x}_k, \quad k = 0, 1, 2, \ldots$$

where the state transition matrix, $\Phi_k(\mathbf{X}_0)$, maps a point $\mathbf{x}_0$ into $\mathbf{x}_k$. Thus,

$$\mathbf{x}_k = \mathbf{\Phi}_k(\mathbf{X}_0)\mathbf{x}_0$$

The LEs λ_i for the discrete-time dynamical system (Equation 14.8) are defined by

$$\lambda_i = \lim_{t\to\infty} \frac{1}{k} \ln \sigma_i(k)$$

whenever this limit exists. $\sigma_i(k)$ denotes the ith singular value of $\Phi_k^H\Phi_k$.

Lyapunov Exponents of Steady-State Solutions

Consider once more the continuous-time dynamical system (Equation 14.3). If $\mathbf{D_xF}$ were constant along the flow, with n distinct eigenvalues $\tilde{\lambda}_i$,

$i = 1, 2, \ldots, n$, then

$$\Phi_t = \begin{pmatrix} \exp(\tilde{\lambda}_1 t) & 0 & \cdots & 0 \\ 0 & \exp(\tilde{\lambda}_2 t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \exp(\tilde{\lambda} a_n t) \end{pmatrix}$$

and

$$\Phi_t^H\Phi_t = \begin{pmatrix} \exp\big(2\mathrm{Re}(\tilde{\lambda}_1)t\big) & 0 & \cdots & 0 \\ 0 & \exp\big(2\mathrm{Re}(\tilde{\lambda}_2)t\big) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \exp\big(2\mathrm{Re}(\tilde{\lambda}_n)t\big) \end{pmatrix}$$

giving $\sigma_i(t) = \exp(\mathrm{Re}(\tilde{\lambda}_i)t)$ and

$$\begin{aligned} \lambda_i &= \lim_{t\to\infty} \frac{1}{t} \ln\big(\exp\big[\mathrm{Re}(\tilde{\lambda}_i)t\big]\big) \\ &= \mathrm{Re}(\tilde{\lambda}_i) \end{aligned}$$

In this case, the LEs are simply the real parts of the eigenvalues of $\mathbf{D_XF}$.

All the eigenvalues of a stable equilibrium point have negative real parts and therefore the largest LE of an attracting equilibrium point is negative.

Trajectories close to a stable limit cycle converge onto the limit cycle. Therefore, the largest LE of a periodic steady-state is zero (corresponding to motion along the limit cycle [15]), and all its other LEs are negative.

TABLE 14.1 Classification of Steady-State Behaviors according to Their Limit Sets, Power Spectra, LEs, and Dimension

Steady State	Limit Set	Spectrum	LEs	Dimension
DC	Fixed point	Spike at DC	$0 > \lambda_1 \geq \cdots \geq \lambda_n$	0
Periodic	Closed curve	Fundamental plus	$\lambda_1 = 0$	1
		Integer harmonics	$0 > \lambda_2 \geq \cdots \geq \lambda_n$	
Quasiperiodic	K-torus	Incommensurate	$\lambda_1 = \cdots = \lambda_K = 0$	K
		Frequencies	$0 > \lambda_{K+1} \geq \cdots \geq \lambda_n$	
Chaotic	Fractal	Broad spectrum	$\lambda_1 > 0$	Noninteger
			$\sum_{i=1}^{n} \lambda_i < 0$	

A quasiperiodic K-torus has K zero LEs because the flow is locally neither contracting nor expanding along the surface of the K-torus.

A chaotic trajectory is locally unstable and therefore has a positive LE; this produces sensitive dependence on initial conditions. Nevertheless, in the case of a chaotic attractor, this locally unstable chaotic trajectory belongs to an attracting limit set to which nearby trajectories converge.

The steady-state behavior of a four-dimensional continuous-time dynamical system which has two positive, one zero, and one negative LE is called hyperchaos.

The Lyapunov spectrum may be used to identify attractors, as summarized in Table 14.1.

14.1.5 Structural Stability and Bifurcations

Structural stability refers to the sensitivity of a phenomenon to small changes in the parameter of a system. A structurally stable vector field $\mathbf{F}$ is one for which sufficiently close vector fields $\mathbf{F}'$ have equivalent* dynamics [18].

The behavior of a typical circuit depends on a set of parameters one or more of which may be varied in order to optimize some performance criteria. In particular, one may think of a one-parameter family of systems

$$\dot{\mathbf{X}} = \mathbf{F}_\mu(\mathbf{X}) \tag{14.11}$$

where the vector field is parametrized by a control parameter μ. A value μ_0 of Equation 14.11 for which the flow of Equation 14.11 is not structurally stable is a bifurcation value of μ [7].

The dynamics in the state space may be qualitatively very different from one value of μ to another. In the nonlinear *RLC* circuit example, the steady-state solution is a limit cycle if the control parameter G'_a is negative and an equilibrium point if G'_a is positive. If G'_a is identically equal to zero, trajectories starting from $I_{30} = 0$, $V_{20} < E$ yield sinusoidal solutions. These sinusoidal solutions are not structurally stable because the slightest perturbation of G'_a will cause the oscillation to decay to zero or converge to the limit cycle, depending on whether G'_a is made slightly larger or smaller than zero.

If we think of this circuit as being parametrized by G'_a, then its vector field is not structurally stable at $G'_a \equiv 0$. We say that the equilibrium point undergoes a bifurcation (from stability to instability) as the value of the bifurcation parameter G'_a is reduced through the bifurcation point $G'_a = 0$.

* Equivalent means that there exists a continuous invertible function h that transforms $\mathbf{F}$ into $\mathbf{F}'$.

14.1.5.1 Bifurcation Types

In this section, we consider three types of local bifurcation: the Hopf bifurcation, the saddle-node bifurcation, and the period-doubling bifurcation [18]. These bifurcations are called local because they may be understood by linearizing the system close to an equilibrium point or limit cycle.

Hopf Bifurcation

A Hopf bifurcation occurs in a continuous-time dynamical system (Equation 14.3) when a simple pair of complex conjugate eigenvalues of the linearization $\mathbf{D_x F}(\mathbf{X_Q})$ of the vector field at an equilibrium point $\mathbf{X_Q}$ crosses the imaginary axis.

Typically, the equilibrium point changes stability from stable to unstable and a stable limit cycle is born. The bifurcation at $G_a' \equiv 0$ in the nonlinear *RLC* circuit is Hopf-like.*

When an equilibrium point undergoes a Hopf bifurcation, a limit cycle is born. When a limit cycle undergoes a Hopf bifurcation, motion on a two-torus results.

Saddle-Node Bifurcation

A saddle-node bifurcation occurs when a stable and an unstable equilibrium point merge and disappear; this typically manifests itself as the abrupt disappearance of an attractor.

A common example of a saddle-node bifurcation in electronic circuits is switching between equilibrium states in a Schmitt trigger. At the threshold for switching, a stable equilibrium point corresponding to the ''high'' saturated state merges with the high-gain region's unstable saddle-type equilibrium point and disappears. After a switching transient, the trajectory settles to the other stable equilibrium point, which corresponds to the ''low'' state.

A saddle-node bifurcation may also manifest itself as a switch between periodic attractors of different sizes between a periodic attractor and a chaotic attractor, or between a limit cycle at one frequency and a limit cycle at another frequency.

Period-Doubling Bifurcation

A period-doubling bifurcation occurs in a discrete-time dynamical system (Equation 14.8) when a real eigenvalue of the linearization $\mathbf{D_X G}(\mathbf{X_Q})$ of the map $\mathbf{G}$ at an equilibrium point crosses the unit circle at -1 [7].

In a continuous-time system, a period-doubling bifurcation occurs only from a periodic solution (an equilibrium point of the Poincaré map). At the bifurcation point, a periodic orbit with period T changes smoothly into one with period $2T$, as illustrated in Figure 14.14a and b.

Blue Sky Catastrophe

A blue sky catastrophe is a global bifurcation that occurs when an attractor disappears ''into the blue,'' usually because of a collision with a saddle-type limit set. Hysteresis involving a chaotic attractor is often caused by a blue sky catastrophe [18].

14.1.5.2 Routes to Chaos

Each of the three local bifurcations may give rise to a distinct route to chaos, and all three have been reported in electronic circuits. These routes are important because it is often difficult to conclude from experimental data alone whether irregular behavior is due to measurement noise or to underlying chaotic dynamics. If, upon adjusting a control parameter, one of the three prototype routes is observed, this indicates that the dynamics might be chaotic.

---

* Note that the Hopf bifurcation theorem is proven for sufficiently smooth systems and does not strictly apply to piecewise-linear systems. However, a physical implementation of a piecewise-linear characteristic, such as that of N_R, is always smooth.

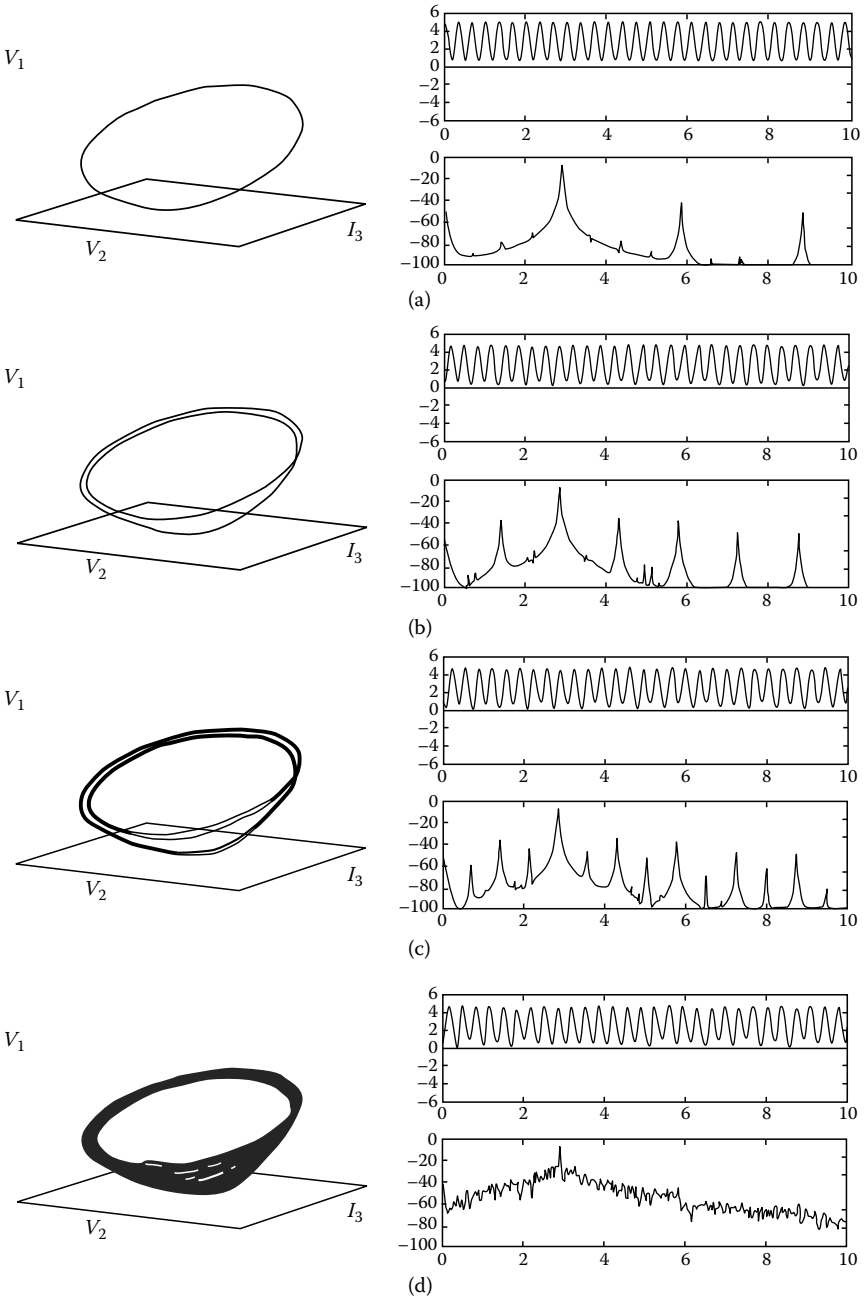


FIGURE 14.14 Period-doubling route to chaos in Chua's oscillator. Simulated state-space trajectories, time waveforms $V_1(t)$, and power spectra of $V_2(t)$ (a) $G = 530$ μS: periodic steady-state—the signal is characterized by a discrete power spectrum with energy at integer multiples of the fundamental frequency f_0; (b) $G = 537$ μS; period-two—after a period-doubling bifurcation, the period of the signal is approximately twice that of (a). In the power spectrum, a spike appears at the new fundamental frequency $\approx f_0/2$. (c) $G = 539$ μS: period-four—a second period-doubling bifurcation gives rise to a fundamental frequency of $\approx f_0/4$; (d) $G = 541$ μS: spiral Chua's attractor—a cascade of period doublings results in a chaotic attractor that has a broadband power spectrum. Time plots: horizontal axis—t (ms); vertical axis—V_1 (V). Power spectra: horizontal axis—frequency (kHz); vertical axis—power (mean-squared amplitude of $V_2(t)$) (dB).

Periodic-Doubling Route to Chaos

The period-doubling route to chaos is characterized by a cascade of period-doubling bifurcations. Each period-doubling transforms a limit cycle into one at half the frequency, spreading the energy of the system over a wider range of frequencies. An infinite cascade of such doublings results in a chaotic trajectory of infinite period and a broad frequency spectrum that contains energy at all frequencies. Figure 14.14 is a set of snapshots of the period-doubling route to chaos in Chua's oscillator.

An infinite number of period-doubling bifurcations to chaos can occur over a finite range of the bifurcation parameter because of a geometric relationship between the intervals over which the control parameter must be moved to cause successive bifurcations. Period-doubling is governed by a universal scaling law that holds in the vicinity of the bifurcation point to chaos μ_∞.

Define the ratio δ_k of successive intervals μ, in each of which there is a constant period of oscillation, as follows,

$$\delta_k = \frac{\mu_{2^k} - \mu_{2^{k-1}}}{\mu_{2^{k+1}} - \mu_{2^k}}$$

where μ_{2^k} is the bifurcation point for the period from $2^k T$ to $2^{k+1}T$. In the limit as $k \to \infty$, a universal constant called the Feigenbaum number δ is obtained:

$$\lim_{k\to\infty} \delta_k = \delta = 4.6692\ldots$$

The period-doubling route to chaos is readily identified from a state-space plot, time series, power spectrum, or a Poincaré map.

Intermittency Route to Chaos

The route to chaos caused by saddle-node bifurcations comes in different forms, the common feature of which is direct transition from regular motion to chaos. The most common type is the intermittency route and results from a single saddle-node bifurcation. This is a route and not just a jump because, straight after the bifurcation, the trajectory is characterized by long intervals of almost regular motion (called laminar phases) and short bursts of irregular motion. The period of the oscillations is approximately equal to that of the system just before the bifurcation. This is illustrated in Figure 14.15.

As the parameter passes through the critical value μ_c at the bifurcation point into the chaotic region, the laminar phases become shorter and the bursts become more frequent, until the regular intervals disappear altogether. The scaling law for the average interval of the laminar phases depends on $|\mu - \mu_c|$, so chaos is not fully developed until some distance from the bifurcation point [13].

Intermittency is best characterized in the time domain because its scaling law depends on the length of the laminar phases.

Another type of bifurcation to chaos associated with saddle-nodes is the direct transition from a regular attractor (fixed point or limit cycle) to a coexisting chaotic one, without the phenomenon of intermittency.

Quasiperiodic (Torus Breakdown) Route to Chaos

The quasiperiodic route to chaos results from a sequence of Hopf bifurcations. Starting from a fixed point, the three-torus generated after three Hopf bifurcations is not stable in the sense that there exists an arbitrarily small perturbation of the system (in terms of parameters) for which the three-torus gives way to chaos.

A quasiperiodic–periodic–chaotic sequence corresponding to torus breakdown in Chua's oscillator is given in Figure 14.5.

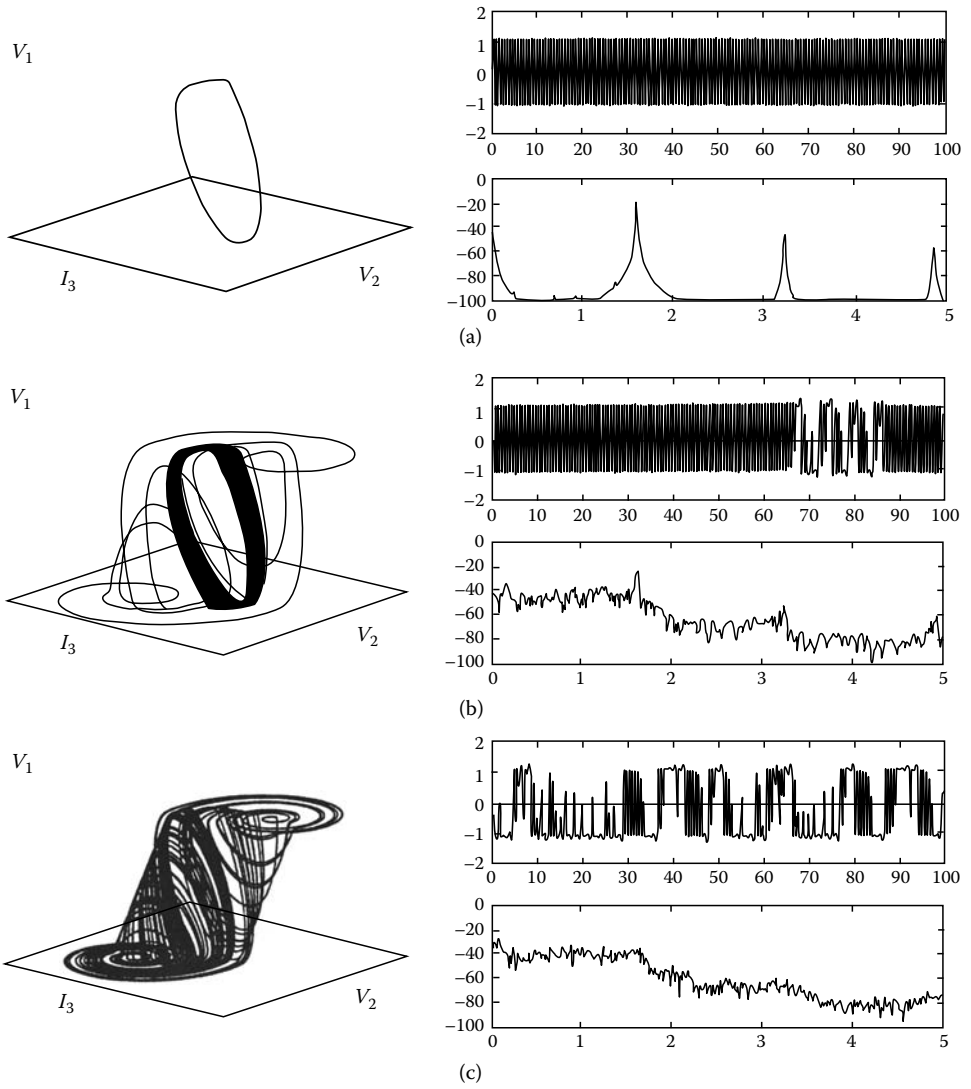


FIGURE 14.15 Intermittency route to chaos in Chua's oscillator. Simulated state-space trajectories, time waveforms $V_1(t)$ and power spectra of $V_2(t)$ (a) Periodic steady-state—the signal is characterized by a discrete power spectrum with energy at integer multiples of the fundamental frequency; (b) onset of intermittency—the time signal contains long regular laminar phases and occasional bursts of irregular motion—in the frequency domain, intermittency manifests itself as a raising of the noise floor; (c) fully developed chaos-laminar phases are infrequent and the power spectrum is broad. Time plots: horizontal axis—t (ms); vertical axis—V_1 (V). Power spectra: horizontal axis—frequency (kHz); vertical axis—power (mean-squared amplitude of $V_2(t)$) (dB).

Quasiperiodicity is difficult to detect from a time series; it is more readily identified by means of a power spectrum or Poincaré map (see Figure 14.11).

14.1.5.3 Bifurcation Diagrams and Parameter Space Diagrams

Although state-space, time- and frequency-domain measurements are useful for characterizing steady-state behaviors, nonlinear dynamics offers several other tools for summarizing qualitative information concerning bifurcations.

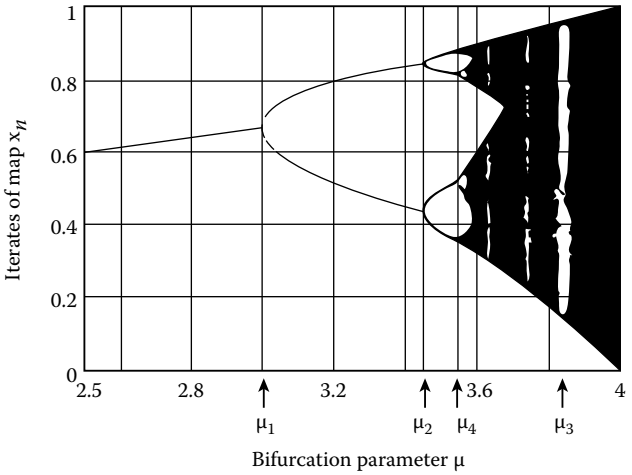


FIGURE 14.16 Bifurcation diagram for the logistic map: $X_{k+1} = \mu X_k(1-X_k)$. The first period-doubling bifurcation occurs at $\mu = \mu_1$, the second at μ_2, and the third at μ_4. μ_3 corresponds to a period-three window. When $\mu = 4$, the entire interval (0, 1) is visited by a chaotic orbit $\{X_k, k = 0, 1, \ldots\}$. (From Wu, C.W. and Rul'kov, N.R., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 708, Oct. 1993. With permission.)

A bifurcation diagram is a plot of the attracting sets of a system versus a control parameter. Typically, one chooses a state variable and plots this against a single control parameter. In discrete systems, one simply plots successive values of a state variable. In the continuous-time case, some type of discretization is needed, typically by means of a Poincaré section.

Figure 14.16 is a bifurcation diagram of the logistic map $X_{k+1} = \mu X_k(1 - X_k)$ for $\mu \in [2.5, 4]$ and $X_k \in [0, 1]$. Period doubling from period-one to period-two occurs at μ_2; the next two doublings in the period-doubling cascade occur at μ_2 and μ_4, respectively. A periodic window in the chaotic region is indicated by μ_3. The map becomes chaotic by the period-doubling route if μ is increased from μ_3 and by the intermittency route if μ is reduced out of the window.

When more than one control parameter is present in a system, the steady-state behavior may be summarized in a series of bifurcation diagrams, where one parameter is chosen as the control parameter, with the others held fixed, and only changed from one diagram to the next. This provides a complete but cumbersome representation of the dynamics [13].

A clearer picture of the global behavior is obtained by partitioning the parameter space by means of bifurcation curves, and labeling the regions according to the observed steady-state behaviors within these regions. Such a picture is called a parameter space diagram.

14.2 Chua's Circuit: A Paradigm for Chaos

Chaos is characterized by a stretching and folding mechanism; nearby trajectories of a dynamical system are repeatedly pulled apart exponentially and folded back together.

In order to exhibit chaos, as autonomous circuit consisting of resistors, capacitors, and inductors must contain (1) at least one locally active resistor, (2) at least one nonlinear element, and (3) at least three energy-storage elements. The active resistor supplies energy to separate trajectories, the nonlinearity provides folding, and the three-dimensional state space permits persistent stretching and folding in a bounded region without violating the noncrossing property of trajectories.

Chua's circuit (see Figure 14.17) is the simplest electronic circuit that satisfies these criteria. It consists of a linear inductor, a linear resistor, two linear capacitors, and a single nonlinear resistor N_R. The circuit is readily constructed at low cost using standard electronic components and exhibits a rich variety of bifurcations and chaos [10].

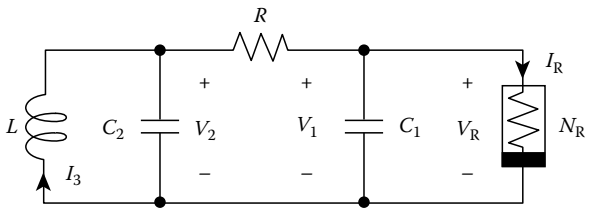


FIGURE 14.17 Chua's circuit consists of a linear inductor L, two linear capacitors (C_2, C_1), a linear resistor R, and a voltage-controlled nonlinear resistor N_R.

14.2.1 Dynamics of Chua's Circuit

14.2.1.1 State Equations

Chua's circuit may be described by three ODEs. Choosing V_1, V_2, and I_3 as state variables, we write

$$\begin{aligned}\frac{dV_1}{dt} &= \frac{G}{C_1}(V_2 - V_1) - \frac{1}{C_1}f(V_1)\\ \frac{dV_2}{dt} &= \frac{G}{C_2}(V_1 - V_2) + \frac{1}{C_2}I_3\\ \frac{dI_3}{dt} &= -\frac{1}{L}V_2\end{aligned} \tag{14.12}$$

where $G = 1/R$ and $f(V_R) = G_b V_R + 1/2(G_a + G_b)\,(|V_R + E| - |V_R - E|)$, as depicted in Figure 14.18.

Because of the piecewise-linear nature of N_R, the vector field of Chua's circuit may be decomposed into three distinct affine regions: $V_1 < -E$, $|V_1| \le E$, and $V_1 > E$. We call these the D_{-1}, D_0, and D_1 regions, respectively. The global dynamics may be determined by considering separately the behavior in each of the three regions (D_{-1}, D_0, and D_1) and then gluing the pieces together along the boundary planes U_{-1} and U_1.

14.2.1.2 Piecewise-Linear Dynamics

In each region, the circuit is governed by a three-dimensional autonomous affine dynamical system of the form

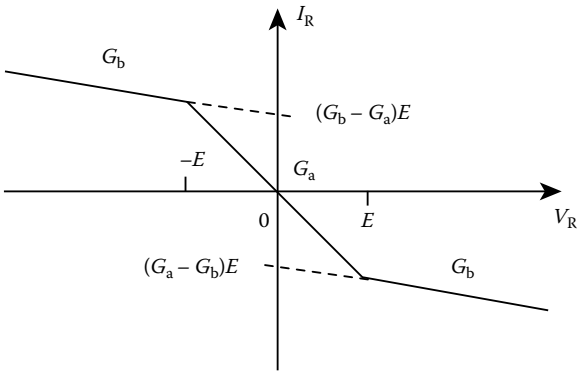


FIGURE 14.18 The DP characteristic of the nonlinear resistor N_R in Chua's circuit has breakpoints at $\pm E$ and slopes G_a and G_b in the inner and outer regions, respectively.

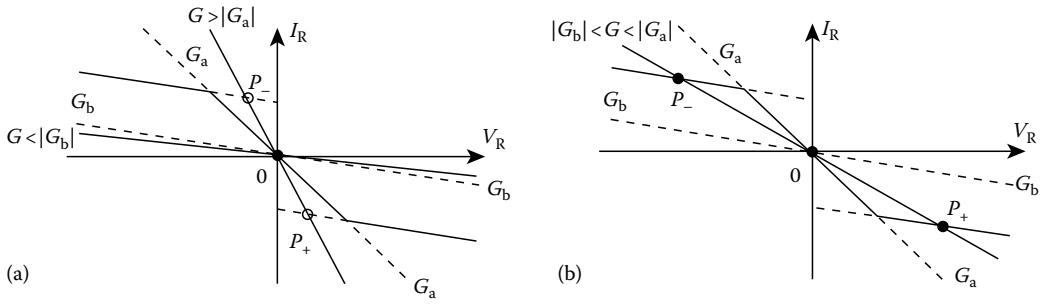


FIGURE 14.19 DC equilibrium points of Figure 14.17 may be determined graphically by intersecting the load line $I_R = -GV_R$ with the DP characteristic of N_R. (a) If $G > |G_a|$ or $G < |g_b|$, the circuit has a unique equilibrium point at the origin (P_- and P_+ are virtual equilibria in this case). (b) When $|G_b| < G < |G_a|$, the circuit has three equilibrium points at P_-, 0 and P_+.

$$\dot{\mathbf{X}} = \mathbf{AX} + \mathbf{b} \tag{14.13}$$

where

A is the (constant) system matrix

b is a constant vector

The equilibrium points of the circuit may be determined graphically by intersecting the load line $I_R = -GV_R$ with the DP characteristic $I_R = f(V_R)$ of the nonlinear resistor N_R, as presented in Figure 14.19 [2]. When $G > |G_a|$ or $G > |G_b|$, the circuit has a unique equilibrium point at the origin (and two virtual equilibria P_- and P_+); otherwise, it has three equilibrium points at P_-, 0, and P_+.

The dynamics close to an equilibrium point $\mathbf{X_Q}$ are governed locally by the linear system

$$\dot{\mathbf{x}} = \mathbf{Ax} \tag{14.14}$$

If the eigenvalues λ_1, λ_2, and λ_3 of **A** are distinct, then every solution $\mathbf{x}(t)$ of Equation 14.14 may be expressed in the form

$$\mathbf{x}(t) = c_1 \exp(\lambda_1 t)\vec{\xi}_1 + c_2 \exp(\lambda_2 t)\vec{\xi}_2 + c_3 \exp(\lambda_3 t)\vec{\xi}_3$$

where

$\vec{\xi}_1$, $\vec{\xi}_2$, and $\vec{\xi}_3$ are the (possibly complex) eigenvectors associated with the eigenvalues λ_1, λ_2, and λ_3, respectively

c_k's are (possibly complex) constants that depend on the initial state $\mathbf{X}_0$

In the special case when **A** has one real eigenvalue γ and a complex conjugate pair of eigenvalues $\sigma \pm j\omega$, the solution of Equation 14.14 has the form

$$\mathbf{x}(t) = c_r \exp(\gamma t)\vec{\xi}_\gamma + 2c_c \exp(\sigma t)[\cos(\omega t + \phi_c)\vec{\eta}_r - \sin(\omega t + \phi_c)\vec{\eta}_i]$$

where

$\vec{\eta}_r$ and $\vec{\eta}_i$ are the real and imaginary parts of the eigenvectors associated with the complex conjugate pair of eigenvalues

$\vec{\xi}_\gamma$ is the eigenvector defined by $\mathbf{A}\vec{\xi}_\gamma = \gamma\ \vec{\xi}_\gamma$

c_r, c_c, and ϕ_c are real constants that are determined by the initial conditions

Let us relabel the real eigenvector E^r, and define E^c as the complex eigenplane spanned by $\vec{\eta}_r$ and $\vec{\eta}_i$.

We can think of the solution $\mathbf{x}(t)$ of Equation 14.14 as being the sum of two distinct components $\mathbf{x}_r(t) \in E^r$ and $\mathbf{x}_c(t) \in E^c$:

$$\mathbf{x}_r(t) = c_r \exp(\gamma t)\vec{\xi}_\gamma$$
$$\mathbf{x}_c(t) = 2c_c \exp(\sigma t)[\cos(\omega t + \phi_c)\vec{\eta}_r - \sin(\omega t + \phi_c)\vec{\eta}_i]$$

The complete solution $\mathbf{X}(t)$ of Equation 14.13 may be found by translating the origin of the linearized coordinate system to the equilibrium point $\mathbf{X_Q}$. Thus,

$$\begin{aligned}\mathbf{X}(t) &= \mathbf{X_Q} + \mathbf{x}(t)\\ &= \mathbf{X_Q} + \mathbf{x}_r(t) + \mathbf{x}_c(t)\end{aligned}$$

We can determine the qualitative behavior of the complete solution $\mathbf{X}(t)$ by considering separately the components $\mathbf{x}_r(t)$ and $\mathbf{x}_c(t)$ along E^r and E^c, respectively.

If $\gamma > 0$, $\mathbf{x}_r(t)$ grows exponentially in the direction of E^r; if $\gamma < 0$ the component $\mathbf{x}_r(t)$ tends asymptotically to zero. When $\sigma > 0$ and $\omega \neq 0$, $\mathbf{x}_c(t)$ spirals away from $\mathbf{X_Q}$ along the complex eigenplane E^c, and if $\sigma < 0$, $\mathbf{x}_c(t)$ spirals toward $\mathbf{X_Q}$ and E^c.

We remark that the vector E^r and plane E^c are invariant under the flow of Equation 14.13: if $\mathbf{X}(0) \in E^r$, then $\mathbf{X}(t) \in E^r$ for all t; if $\mathbf{X}(0) \in E^c$ then $\mathbf{X}(t) \in E^c$ for all t. An important consequence of this is that a trajectory $\mathbf{X}(t)$ cannot cross through the complex eigenspace E^c; suppose $\mathbf{X}(t_0) \in E^c$ at some time t_0, then $\mathbf{X}(t) \in E^c$ for all $t > t_0$.

14.2.2 Chaos in Chua's Circuit

In the following discussion, we consider a fixed set of component values: $L = 18$ mH, $C_2 = 100$ nF, $C_1 = 10$ nF, $G_a = -50/66$ mS $= -757.576$ μS, $G_b = -9/22$ mS $= -409.091$ μS, and $E = 1$ V. When $G = 550$ μS, three equilibrium points occur at P_+, 0, and P_-. The equilibrium point at the origin (0) has one unstable real eigenvalue γ_0 and a stable complex pair $\sigma_0 \pm j\omega_0$. The outer equilibria (P_- and P_+) each have a stable real eigenvalue γ_1 and an unstable complex pair $\sigma_0 \pm j\omega_1$.

14.2.2.1 Dynamics of D_0

A trajectory starting from some initial state $\mathbf{X}_0$ in the D_0 region may be decomposed into its components along the complex eigenplane $E^c(0)$ and along the eigenvector $E^r(0)$. When $\gamma_0 > 0$ and $\sigma_0 < 0$, the component along $E^c(0)$ spirals toward the origin along this plane while the component in the direction $E^r(0)$ grows exponentially. Adding the two components, we see that a trajectory starting slightly above the stable complex eigenplane $E^c(0)$ spirals toward the origin along the $E^c(0)$ direction, all the while being pushing away from $E^c(0)$ along the unstable direction $E^r(0)$. As the (stable) component along $E^c(0)$ shrinks in magnitude, the (unstable) component grows exponentially, and the trajectory follows a helix of exponentially decreasing radius whose axis lies in the direction of $E^r(0)$; this is illustrated in Figure 14.20.

14.2.2.2 Dynamics of D_{-1} and D_1

The eigenvector $E^r(P_+)$ is associated with the stable real eigenvalue γ_1 in the D_1 region. The real and imaginary parts of the complex eigenvectors associated with $\sigma_1 \pm j\omega_1$ define a complex eigenplane $E^c(P_+)$.

A trajectory starting from some initial state $\mathbf{X}_0$ in the D_1 region may be decomposed into its components along the complex eigenplane $E^c(P_+)$ and the eigenvector $E^r(P_+)$. When $\gamma_1 < 0$ and $\sigma_1 > 0$, the component on $E^c(P_+)$ spirals away from P_+ along this plane while the component in the direction of $E^r(0)$ tends asymptotically toward P_+. Adding the two components, we see that a trajectory starting close to the stable real eigenvector $E^r(P_+)$ above the complex eigenplane moves toward $E^c(P_+)$

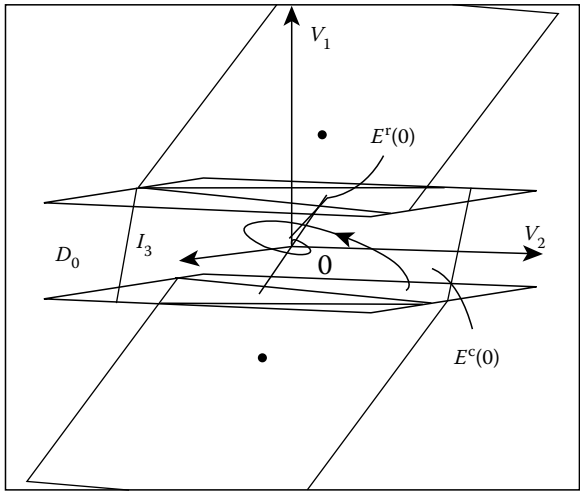


FIGURE 14.20 Dynamics of the D_0 region. A trajectory starting slightly above the stable complex eigenplane $E^c(0)$ spirals toward the origin along this plane and is repelled close to 0 in the direction of the unstabe eigenvector $E^r(0)$. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 660, Oct. 1993. With permission.)

along a helix of exponentially increasing radius. Because the component along $E^r(P_+)$ shrinks exponentially in magnitude and the component on $E^c(P_+)$ grows exponentially, the trajectory is quickly flattened onto $E^c(P_+)$, where it spirals away from P_+ along the complex eigenplane; this is illustrated in Figure 14.21.

By symmetry, the equilibrium point P_- in the D_{-1} region has three eigenvalues: γ_1 and $\sigma_1 \pm j\omega_1$. The eigenvector $E^r(P_-)$ is associated with the stable real eigenvalue γ_1; the real and imaginary parts of the eigenvectors associated with the unstable complex pair $\sigma_1 \pm j\omega_1$ define an eigenplane $E^c(P_-)$, along which trajectories spiral away from P_-.

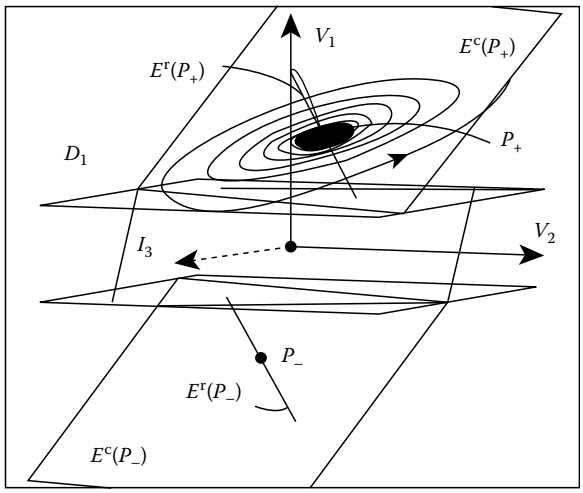


FIGURE 14.21 Dynamics of the D_1 region. A trajectory starting above the unstable complex eigenplane $E^c(P_+)$ close to the eigenvector $E^r(P_+)$ moves toward the plane and spirals away from P_+ along $E^c(P_+)$. By symmetry, the D_{-1} region has equivalent dynamics. (From Kennedy, M.P., *IEEE Trans. Circuits Syst.*, 40, 662, Oct. 1993. With permission.)

14.2.2.3 Global Dynamics

With the given set of parameter values, the equilibrium point at the origin has an unstable real eigenvalue and a stable pair of complex conjugate eigenvalues; the outer equilibrium point P_- has a stable real eigenvalue and an unstable complex pair.

In particular, P_- has a pair of unstable complex conjugate eigenvalues $\sigma_1 \pm \omega_1$ ($\sigma_1 > 0$, $\omega_1 \neq 0$) and a stable real eigenvalue γ_1, where $|\sigma_1| < |\omega_1|$. In order to prove that the circuit is chaotic in the sense of Shil'nikov, it is necessary to show that it possesses a homoclinic orbit for this set of parameter values.

A trajectory starting on the eigenvector $E^r(0)$ close to 0 moves away from the equilibrium point until it crosses the boundary U_1 and enters D_1, as illustrated in Figure 14.20. If this trajectory is folded back into D_0 by the dynamics of the outer region, and reinjected toward 0 along the stable complex eigenplane $E^c(0)$ then a homoclinic orbit is produced.

That Chua's circuit is chaotic in the sense of Shil'nikov was first proven by Chua et al. [21] in 1985. Since then, there has been an intensive effort to understand every aspect of the dynamics of this circuit with a view to developing it as a paradigm for learning, understanding, and teaching about nonlinear dynamics and chaos [3].

14.2.3 Steady-States and Bifurcations in Chua's Circuit

In the following discussion, we consider the global behavior of the circuit using our chosen set of parameters with R in the range $0 \leq R \leq 2000\ \Omega$ ($500\ \mu S \leq G < \infty S$).

Figure 14.14 is a series of simulations of the equivalent circuit in Figure 14.26 with the following parameter values: $L = 18$ mH, $C_2 = 100$ nF, $C_1 = 10$ nF, $G_a = -50/66$ mS $= -757.576$ μS, $G_b = -9/22$ mS $= -409.091$ μS, and $E = 1$ V. $R_0 = 12.5\ \Omega$, the parasitic series resistance of a real inductor. R is the bifurcation parameter.

14.2.3.1 Equilibrium Point and Hopf Bifurcation

When R is large (2000 Ω), the outer equilibrium points P_- and P_+ are stable ($\gamma_1 < 0$ and $\sigma_1 < 0$, $\omega_1 \neq 0$); the inner equilibrium point 0 is unstable ($\gamma_0 > 0$ and $\sigma_0 < 0$, $\omega_0 \neq 0$).

Depending on the initial state of the circuit, the system remains at one outer equilibrium point or the other. Let us assume that we start at P_+ in the D_1 region. This equilibrium point has one negative real eigenvalue and a complex pair with negative real parts. The action of the negative real eigenvalue γ_1 is to squeeze trajectories down onto the complex eigenplane $E^c(P_+)$, where they spiral toward the equilibrium point P_+.

As the resistance R is decreased, the real part of the complex pair of eigenvalues changes sign and becomes positive. Correspondingly, the outer equilibrium points become unstable as σ_1 passes through 0; this is a Hopf-like bifurcation.* The real eigenvalue of P_+ remains negative so trajectories in the D_1 region converge toward the complex eigenplane $E^c(P_+)$. However, they spiral away from the equilibrium point P_+ along $E^c(P_+)$ until they reach the dividing plane U_1 (defined by $V_1 + E$) and enter the D_0 region.

The equilibrium point at the origin in the D_0 region has a stable complex pair of eigenvalues and an unstable real eigenvalue. Trajectories that enter the D_0 region on the complex eigenplane $E^c(0)$ are attracted to the origin along this plane. Trajectories that enter D_0 from D_1 below or above the eigenplane either cross-over to D_{-1} or are turned back toward D_1, respectively. For R sufficiently large, trajectories that spiral away from P_+ along $E^c(P_+)$ and enter D_0 above $E^c(0)$ are returned to D_1, producing a stable period-one limit cycle. This is illustrated in Figure 14.14a.

---

* Recall that the Hopf bifurcation theorem strictly applies only for sufficiently smooth systems, but that physical implementations of piecewise-linear characteristics are typically smooth.

14.2.3.2 Period-Doubling Cascade

As the resistance R is decreased further, a period-doubling bifurcation occurs. The limit cycle now closes on itself after encircling P_+ twice; this is called a period-two cycle because a trajectory takes approximately twice the time to complete this closed orbit as to complete the preceding period-one orbit (see Figure 14.14b).

Decreasing the resistance R still further produces a cascade of period-doubling bifurcations to period-four (Figure 14.14c), period-eight, period-sixteen, and so on until an orbit of infinite period is reached, beyond which we have chaos (see Figure 14.14d). This is a spiral Chua's chaotic attractor.

The spiral Chua's attractor in Figure 14.14d looks like a ribbon or band that is smoothly folded on itself; this folded band is the simplest type of chaotic attractor [18]. A trajectory from an initial condition $\mathbf{X}_0$ winds around the strip repeatedly, returning close to $\mathbf{X}_0$, but never closing on itself.

14.2.3.3 Periodic Windows

Between the chaotic regions in the parameter space of Chua's circuit, there exist ranges of the bifurcation parameter R over which stable periodic motion occurs. These regions of periodicity are called periodic windows and are similar to those that exist in the bifurcation diagram of the logistic map (see Figure 14.16).

Periodic windows of periods three and five are readily found in Chua's circuit. These limit cycles undergo period-doubling bifurcations to chaos as the resistance R is decreased.

For certain sets of parameters, Chua's circuit follows the intermittency route to chaos as R is increased out of the period-three window.

14.2.3.4 Spiral Chua's Attractor

Figure 14.22 outlines three views of another simulated spiral Chua's chaotic attractor. Figure 14.22b is a view along the edge of the outer complex eigenplanes $E^c(P_+)$ and $E^c(P_-)$; notice how trajectories in the D_1 region are compressed toward the complex eigenplane $E^c(P_+)$ along the direction of the stable real eigenvector $E^c(P_+)$ and they spiral away from the equilibrium point P_+ along $E^c(P_+)$.

When a trajectory enters the D_0 region through U_1 from D_1, it is twisted around the unstable real eigenvector $E^r(0)$ and returned to D_1.

Figure 14.22c illustrates clearly that when the trajectory enters D_0 from D_1, it crosses U_1 above the eigenplace $E^c(0)$. The trajectory cannot cross through this eigenplane and therefore it must return to the D_1 region.

14.2.3.5 Double-Scroll Chua's Attractor

Because we chose a nonlinear resistor with a symmetric nonlinearity, every attractor that exists in the D_1 and D_0 regions has a counterpart (mirror image) in the D_{-1} and D_0 regions. As the coupling resistance R is decreased further, the spiral Chua's attractor "collides" with its mirror image and the two merge to form a single compound attractor called a double-scroll Chua's chaotic attractor [10], as presented in Figure 14.23.

Once more, we show three views of this attractor in order to illustrate its geometrical structure. Figure 14.23b is a view of the attractor along the edge of the outer complex eigenplanes $E^c(P_+)$ and $E^c(P_-)$. Upon entering the D_1 region form D_0, the trajectory collapses onto $E^c(P_+)$ and spirals away from P_+ along this plane.

Figure 14.23c is a view of the attractor along the edge of the complex eigenplane $E^c(0)$ in the inner region. Notice once more that when the trajectory crosses U_1 into D_0 above $E^c(0)$, it must remain above $E^c(0)$ and so returns to D_1. Similarly, if the trajectory crosses U_1 below $E^c(0)$, it must remain below $E^c(0)$ and therefore crosses over to the D_{-1} region. Thus, $E^c(0)$ presents a knife-edge to the trajectory as it crosses U_1 into the D_0 region, forcing it back toward D_1 or across D_0 to D_{-1}.

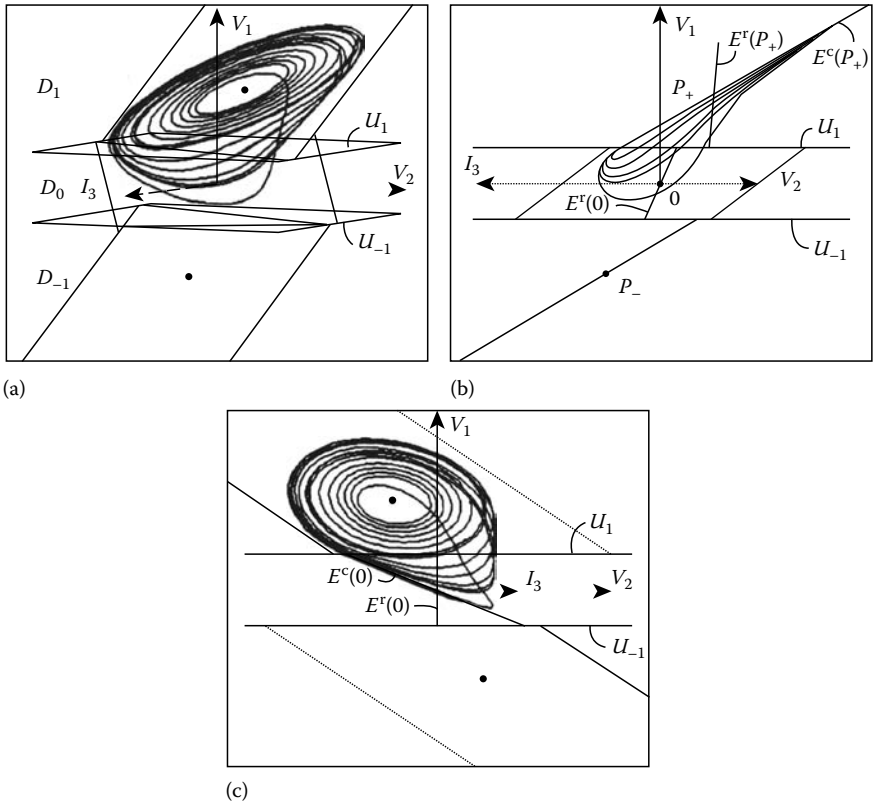


FIGURE 14.22 Three views of a simulated spiral Chua's attractor in Chua's oscillator with $G = 550$ μS. (a) Reference view (compare with Figure 14.14d). (b) View of the edge of the outer complex eigenplanes $E^c(P_+)$ and $E^c(P_-)$; note how trajectory in D_1 is flattened onto $E^c(P_+)$. (c) View along the edge of the complex eigenplane $E^c(0)$; trajectories cannot cross this plane. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 664, Oct. 1993. With permission.)

14.2.3.6 Boundary Crisis

Reducing the resistance R still further produces more regions of chaos, interspersed with periodic windows. Eventually, for a sufficiently small value of R, the unstable saddle trajectory that normally resides outside the stable steady-state solution collides with the double-scroll Chua's attractor and a blue sky catastrophe called a boundary crisis [10] occurs. After this, all trajectories become unbounded.

14.2.4 Manifestations of Chaos

14.2.4.1 Sensitive Dependence on Initial Conditions

Consider once more the double-scroll Chua's attractor shown in Figure 14.23. Two trajectories starting from distinct but almost identical initial states in D_1 will remain "close together" until they reach the separating plane U_1. Imagine that the trajectories are still "close" at the knife-edge, but that one trajectory crosses into D_0 slightly above $E^c(0)$ and the other slightly below $E^c(0)$. The former trajectory returns to D_1 and the latter crosses over to D_1: their "closeness" is lost.

The time-domain waveforms $V_1(t)$ for two such trajectories are shown in Figure 14.24. These are solutions of Chua's oscillator with the same parameters as in Figure 14.23; the initial conditions are $(I_3, V_2, V_1) = (1.810$ mA, 222.014 mV, -2.286 V) [solid curve] and $(I_3, V_2, V_1) = (1.810$ mA,

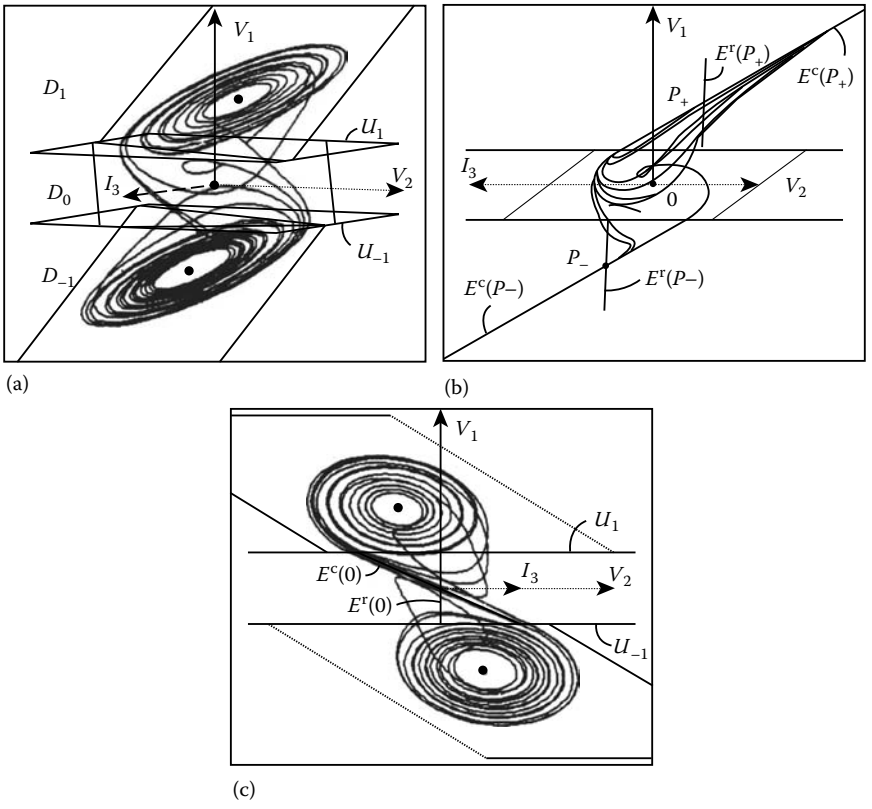


FIGURE 14.23 Three views of a simulated double-scroll Chua's attractor in Chua's oscillator with $G = 565$ μS. (a) Reference view (compare with Figure 14.14d). (b) View along the edge of the outer complex eigenplanes $E^c(P_+)$ and $E^c(P_-)$; note how the trajectory in D_1 is flattened onto $E^c(P_+)$ and onto $E^c(P_-)$ in D_{-1}. (c) View along the edge of the complex eigenplane $E^c(0)$; a trajectory entering D_0 from D_1 above this plane returns to D_1 while one entering D_0 below $E^c(0)$ crosses to D_{-1}. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 665, Oct. 1993. With permission.)

222.000 mV, −2.286 V) [dashed curve]. Although the initial conditions differ by less than 0.01% in just one component (V_2), the trajectories diverge and become uncorrelated within 5 ms because one crosses the knife-edge before the other.

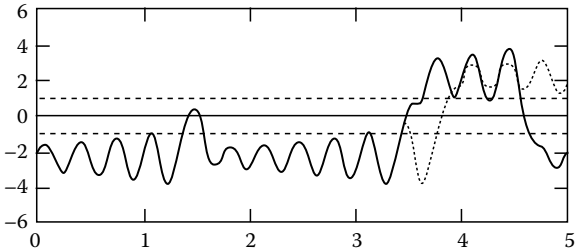


FIGURE 14.24 Sensitive dependence on initial conditions. Two time waveforms $V_1(t)$ from Chua's oscillator with $G = 550$ μS, starting from $(I_3, V_2, V_1) = (1.810$ mA, 222.01 mV, −2.286 V) [solid line] and $(I_3, V_2, V_1) = (1.810$ mA, 222.000 mV, −2.286 V) [dashed line]. Note that the trajectories diverge within 5 ms. Horizontal axis: t (ms); vertical axis: V_1 (V). Compare with Figure 14.23.

This rapid decorrelation of trajectories that originate in nearby initial states, commonly called sensitive dependence on initial conditions, is a generic property of chaotic systems. It gives rise to an apparent randomness in the output of the system and long-term unpredictability of the state.

14.2.4.2 ''Randomness'' in the Time Domian

Figures 14.14a through d show the state-space trajectories of period-one, period-two, and period-four periodic attractors, a spiral Chua's chaotic attractor, respectively, and the corresponding voltage waveforms $V_1(t)$.

The ''period-one'' waveform is periodic; it looks like a slightly distorted sinusoid. The ''period-two'' waveform is also periodic. It differs qualitatively from the ''period-one'' in that the pattern of a large peak followed by a small peak repeats approximately once every two cycles of the period-one signal; that is why it is called period-two.

In contrast with these periodic time waveforms, $V_1(t)$ for the spiral Chua's attractor is quite irregular and does not appear to repeat itself in any observation period of finite length. Although it is produced by a third-order deterministic differential equation, the solution looks ''random.''

14.2.4.3 Broadband ''Noise-Like'' Power Spectrum

In the following discussion, we consider 8192 samples of $V_2(t)$ recorded at 200 kHz; leakage in the power spectrum is controlled by applying a Welch window [17] to the data.

We remarked earlier that the period-one time waveform corresponding to the attractor in Figure 14.14a, is almost sinusoidal; we expect, therefore, that most of its power should be concentrated at the fundamental frequency. The power spectrum of the period-one waveform $V_2(t)$ shown in Figure 14.14a consists of a sharp spike at approximately 3 kHz and higher harmonic components that are over 30 dB below the fundamental.

Because the period-two waveform repeats roughly once every 0.67 ms, this periodic signal has a fundamental frequency component at approximately 1.5 kHz (see Figure 14.14b). Notice, however, that most of the power in the signal is concentrated close to 3 kHz.

The period-four waveform repeats roughly once every 1.34 ms, corresponding to a fundamental frequency component at approximately 750 Hz (see Figure 14.14c). Note once more that most of the power in the signal is still concentrated close to 3 kHz.

The spiral Chua's attractor is qualitatively different from these periodic signals. The aperiodic nature of its time-domain waveforms is reflected in the broadband noise-like power spectrum (Figure 14.14d). No longer is the power of the signal concentrated in a small number of frequency components; rather, it is distributed over a broad range of frequencies. This broadband structure of the power spectrum persists even if the spectral resolution is increased by sampling at a higher frequency f_s. Notice that the spectrum still contains a peak at approximately 3 kHz that corresponds to the average frequency of rotation of the trajectory about the fixed point.

14.2.5 Practical Realization of Chua's Circuit

Chua's circuit can be realized in a variety of ways using standard or custom-made electronic components. All the linear elements (capacitor, resistor, and inductor) are readily available as two-terminal devices. A nonlinear resistor N_R with the prescribed DP characteristic (called a Chua diode [10]) may be implemented by connecting two negative resistance converters in parallel as outlined in Figure 14.25. A complete list of components is given in Table 14.2.

The op-amp subcircuit consisting of A_1, A_2, and R_1–R_6 functions as a negative resistance converter N_R with DP characteristic as shown in Figure 14.28b. Using two 9 V batteries to power the op-amps gives $V^+ = 9$ V and $V^- = -9$ V. From measurements of the saturation levels of the AD712 outputs, $E_{sat} \approx 8.3$ V, giving $E \approx 1$ V. With $R_2 = R_3$ and $R_5 = R_6$, the nonlinear characteristic is defined by $G_a = -1/R_1 - 1/R_4 = -50/66$ mS, $G_b = 1/R_3 - 1/R_4 = -9/22$ mS, and $E = R_1 E_{sat}/(R_1 + R_2) \approx 1$ V [10].

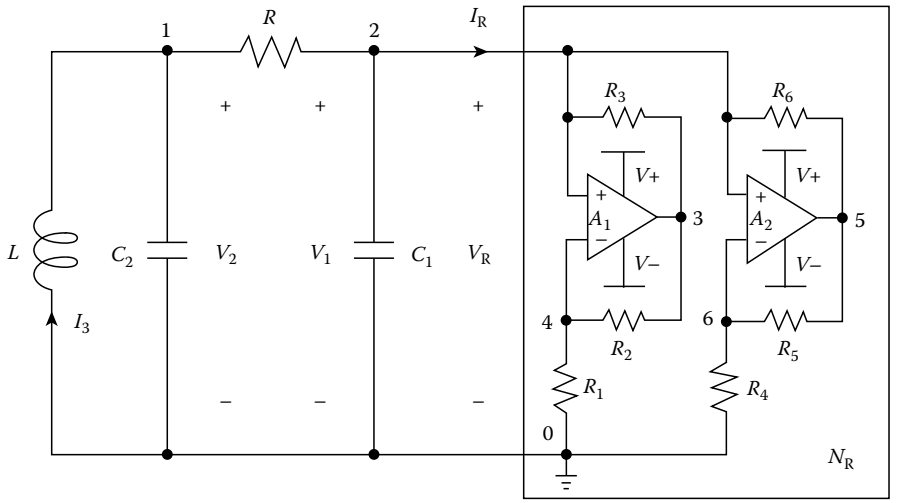


FIGURE 14.25 Practical implementation of Chua's circuit using two op-amps and six resistors to realize the Chua diode. Component values are listed in Table 14.2. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 640–656, 657–674, Oct. 1993. With permission.)

TABLE 14.2 Component List for the Practical Implementation of Chua's Circuit, Depicted in Figure 14.25

Element	Description	Value	Tolerance (%)
A_1	Op-amp ($\frac{1}{2}$ AD712, TL082, or equivalent)	—	—
A_2	Op-amp ($\frac{1}{2}$ AD712, TL082, or equivalent)	—	—
C_1	Capacitor	10 nF	±5
C_2	Capacitor	100 nF	±5
R	Potentiometer	2 kΩ	—
R_1	$\frac{1}{4}$ W resistor	3.3 kΩ	±5
R_2	$\frac{1}{4}$ W resistor	22 kΩ	±5
R_3	$\frac{1}{4}$ W resistor	22 kΩ	±5
R_4	$\frac{1}{4}$ W resistor	2.2 kΩ	±5
R_5	$\frac{1}{4}$ W resistor	220 Ω	±5
R_6	$\frac{1}{4}$ W resistor	220 Ω	±5
L	Inductor (TOKO-type 10 RB, or equivalent)	118 mH	±10

The equivalent circuit of Figure 14.25 is presented in Figure 14.26, where the real inductor is modeled as a series connection of an ideal linear inductor L and a linear resistor R_0. When the inductor's resistance is modeled explicitly in this way, the circuit is called Chua's oscillator [5].

14.2.6 Experimental Steady-State Solutions

A two-dimensional projection of the steady-state attractor in Chua's circuit may be obtained by connecting V_2 and V_1 to the X and Y channels, respectively, of an oscilloscope in X–Y mode.

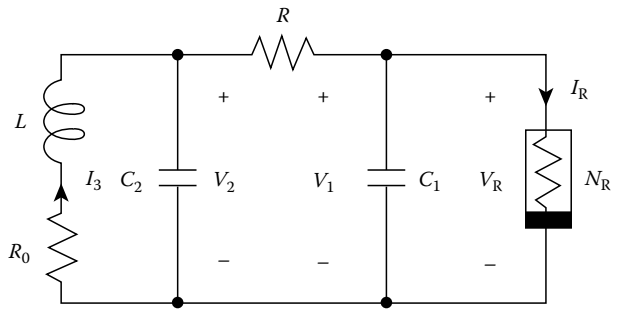


FIGURE 14.26 Chua's oscillator.

14.2.6.1 Bifurcation Sequence with *R* as Control Parameter

By reducing the variable resistor R in Figure 14.25 from 2000 Ω to 0, Chua's circuit exhibits a Hopf bifurcation from dc equilibrium, a sequence of period-doubling bifurcations to a spiral Chua's attractor, periodic windows, a double-scroll Chua's chaotic attractor, and a boundary crisis, as illustrated in Figure 14.27.

Notice that varying R in this way causes the size of the attractors to change: the period-one orbit is large, period-two is smaller, the spiral Chua's attractor is smaller again, and the double-scroll Chua's attractor shrinks considerably before it dies. This shrinking is due to the equilibrium points P_+ and P_- moving closer toward the origin as R is decreased. Consider the load line in Figure 14.19b: as R is decreased, the slope G increases, and the equilibrium points P_- and P_+ move toward the origin. Compare also the positions of P_+ in Figures 14.22a and 14.23a.

14.2.6.2 Outer Limit Cycle

No physical system can have unbounded trajectories. In particular, any physical realization of a Chua diode is eventually passive, meaning simply that for a large enough voltage across its terminals, the instantaneous power $P_R(t)$ $[= V_R(t)I_R(t)]$ consumed by the device is positive.

Hence, the DP characteristic of a real Chua diode must include at least two outer segments with positive slopes which return the characteristic to the first and third quadrants (see Figure 14.28b). From a practical point of view, as long as the voltages and currents on the attractor are restricted to the negative resistance region of the characteristic, these outer segments will not affect the circuit's behavior.

The DP characteristic of the op-amp-based Chua diode differs from the desired piecewise-linear characteristic depicted in Figure 14.28a in that it has five segments, the outer two of which have positive slopes $G_c = 1/R_5 = 1/220$ S.

The "unbounded" trajectories that follow the boundary crisis in the ideal three-region system are limited in amplitude by these dissipative outer segments and a large limit cycle results, as illustrated in Figure 14.27i. This effect could, of course, be simulated by using a five-segment DP characteristic for N_R as illustrated in Figure 14.28b.

The parameter value at which the double-scroll Chua's attractor disappears and the outer limit cycle appears is different from that at which the outer limit cycle disappears and the chaotic attractor reappears. This "hysteresis" in parameter space is the characteristic of a blue sky catastrophe.

14.2.7 Simulation of Chua's Circuit

Our experimental observations and qualitative descriptive description of the global dynamics of Chua's circuit may be confirmed by simulation using a general purpose ODE solver such as MATLAB® [22] or by employing a customized simulator such as "ABC++" [23].

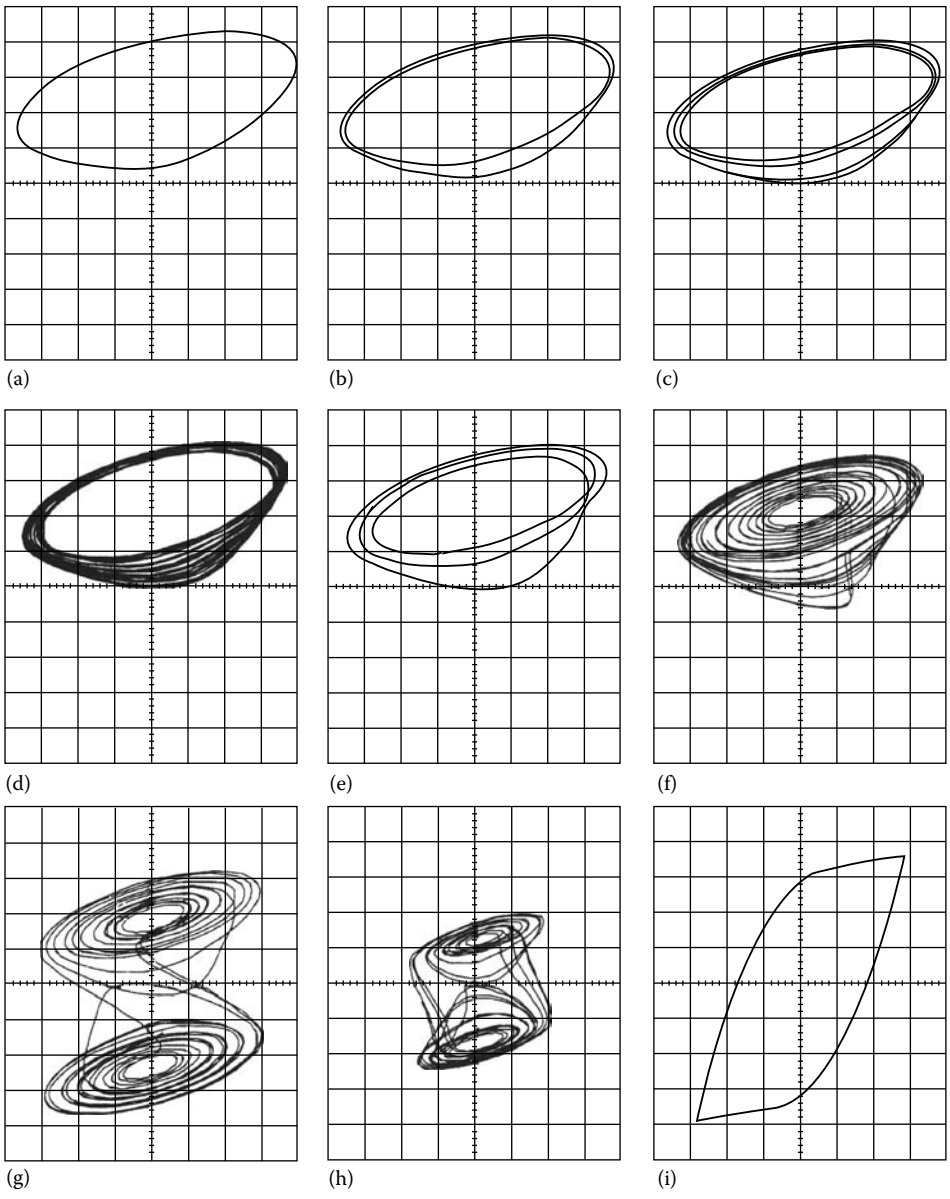


FIGURE 14.27 Typical experimental bifurcation sequence in Chua's circuit (component values as in Table 14.2) recorded using a digital storage oscilloscope. Horizontal axis V_2 (a)–(h) 200 mV/div, (i) 2 V/div; vertical axis V_1 (a)–(h) 1 V/div, (i) 2 V/div. (a) $R = 1.83$ kΩ, period–one; (b) $R = 1.82$ kΩ, period-two; (c) $R = 1.81$ kΩ, period-four; (d) $R = 1.80$ kΩ, spiral Chua's attractor; (e) $R = 1.797$ kΩ, period-three window; (f) $R = 1.76$ kΩ, spiral Chua's attractor; (g) $R = 1.73$ kΩ, double-scroll Chua's attractor; (h) $R = 1.52$ kΩ, double-scroll Chua's attractor; (i) $R = 1.42$ kΩ, large limit cycle corresponding to the outer segments of the Chua diode's DP characteristic. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 669, Oct. 1993. With permission.)

For electrical engineers who are familiar with the SPICE circuit simulator but perhaps not with chaos, we present a net-list and simulation results for a robust op-amp-based implementation of Chua's circuit. The AD712 op-amps in this realization of the circuit are modeled using Analog Devices' AD712 macromodel. The TOKO 10RB inductor has a nonzero series resistance that we have included in the

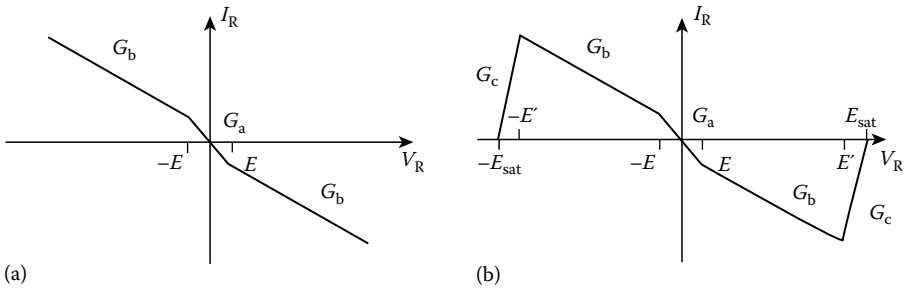


FIGURE 14.28 (a) Required three-segment piecewise-linear DP characteristic for the Chua diode in Figure 14.17. (b) Every physically realizable nonlinear resistor N_R is eventually passive—the outermost segments (while not necessarily linear as presented here) must lie completely within the first and third quadrants of the $V_R - I_R$ place for sufficiently large $|V_R|$ and $|I_R|$.

SPICE net-list; a typical value of RO for this inductor is 12.5 Ω. Node numbers are as Figure 14.25: the power rails are 111 and 222; 10 is the ''internal'' node of the physical inductor, where its series inductance is connected to its series resistance.

A double-scroll Chua's attractor results from a PSPICE simulation using the input deck shown in Figure 14.29; this attractor is plotted in Figure 14.30.

14.2.8 Dimensionless Coordinates and the α–β Parameter-Space Diagram

Thus far, we have discussed Chua's circuit equations in terms of seven parameters: L, C_2, G, C_1, E, G_a, and G_b. We can reduce the number of parameters by normalizing the nonlinear resistor such that its breakpoints are at ± 1 V instead of $\pm E$ V. Furthermore, we may write Chua's circuit Equation 14.12 in normalized dimensionless form by making the following change of variables: $X_1 = V_1/E$, $X_2 = V_2/E$, $X_3 = I_3/(EG)$, and $\tau = tG/C_2$. The resulting state equations are

$$\begin{aligned} \frac{dX_1}{d\tau} &= \alpha[X_2 - X_1 - f(X_1)] \\ \frac{dX_2}{d\tau} &= X_1 - X_2 + X_3 \\ \frac{dX_3}{d\tau} &= -\beta X_2 \end{aligned} \tag{14.15}$$

where $\alpha = C_2/C_1$, $\beta = C_2/(LG^2)$, and $f(X) = bX + 1/2(a-b)(|X+1| - |X-1|)$; $a = G_a/G$ and $b = G_b/G$. Thus, each set of seven circuit parameters has an equivalent set of four normalized dimensionless parameters $\{\alpha, \beta, a, b\}$. If we fix the values of a and b (which correspond to the slopes G_a and G_b of the Chua diode), we can summarize the steady-state dynamical behavior of Chua's circuit by means of a two-dimensional parameter-space diagram.

Figure 14.31 presents the (α, β) parameter-space diagram with $a = -8/7$ and $b = -5/7$. In this diagram, each region denotes a particular type of steady-state behavior: for example, an equilibrium point, period-one orbit, period-two, spiral Chua's attractor, double-scroll Chua's attractor. Typical state-space behaviors are shown in the insets. For clarity, we show chaotic regions in a single shade; it should be noted that these chaotic regions are further partitioned by periodic windows and ''islands'' of periodic behavior.

To interpret the α–β diagram, imagine fixing the value of $\beta = C_2/(LG_2)$ and increasing $\alpha = C_2/C_1$ from a positive value to the left of the curve labeled ''Hopf at $P^{\pm}$''; experimentally, this corresponds to fixing the parameters L, C_2, G, E, G_a, and G_b, and reducing the value of C_1—this is called a ''C_1 bifurcation sequence.''

```
ROBUST OP-AMP REALIZATION OF CHUA'S CIRCUIT

V+   111  0    DC  9
V−  0   222   DC 9
L     1  10    0.018
R0    10 0     12.5
R     1  2     1770
C2    1  0      100.0N
C1    2  0      10.0N
XA1   2 4 111  222   3 AD712
R1    2 3   220
R2    3 4   220
R3    4 0   2200
XA2   2 6  111  222  5 AD712
R4    2 5   22000
R5    5 6   22000
R6    6 0   3300

* AD712 SPICE Macro-model                    1/91, Rev. A
* Copyright 1991 by Analog Devices, Inc. (reproduced with permission)
*
.SUBCKT AD 712 13  15  21 16  14
*
VOS 15   8   DC  0
EC  9  0 14   0   1
C1 6 7  .5P
RP  16 12  12K
GB 11  0 3  0  1.67K
RD1 6  16  16k
RD2 7  16  16k
ISS 12 1  DC  100U
CCI  3 11  150P
GCM 0  3  0  1   1.76N
GA 3  0  7  6   2.3M
RE 1  0  2.5MEG
RGM 3  0  1.69K
VC  12 2  DC  2.8
VE  10  16 DC  2.8
RO1  11  14  25
CE  1  0  2P
RO2  0  11 30
RS1  1 4  5.77K
RS2  1 5  5.77K
J1  6 13  4  FET
J2  7  8 5  FET
DC  14  2 DIODE
DE  10 14  DIODE
DP  16  12  DIODE
D1 9  11  DIODE
D2  11  9   DIODE
IOS  15  13  5E-12
.MODEL DIODE D
.MODEL FET PJF(VTO = −1 BETA = 1M IS = 25E−12)
.ENDS

.IC V(2) = 0.1 V(1) = 0
.TRAN 0.01MS 100MS 50MS
.OPTIONS RELTOL = 1.0E−4 ABSTOL = 1.0E−4
.PRINT TRAN V(2) V(1)
.END
```

FIGURE 14.29 SPICE deck to simulate the transient response of the dual op-amp implementation of Chua's circuit. Node numbers are as in Figure 14.25. The op-amps are modeled using the Analog Devices AD712 macromodel. R0 models the series resistance of the real inductor L.

Initially, the steady-state solution is an equilibrium point. As the value of C_1 is reduced, the circuit undergoes a Hopf bifurcation when α crosses the "Hopf at $P^{\pm}$" curve. Decreasing C_1 still further, the steady-state behavior bifurcates from period-one to period-two to period-four and so on to chaos, periodic windows, and a double-scroll Chua's attractor. The right-hand side edge of the chaotic region is delimited by a curve corresponding to the boundary crisis and "death" of the attractor.

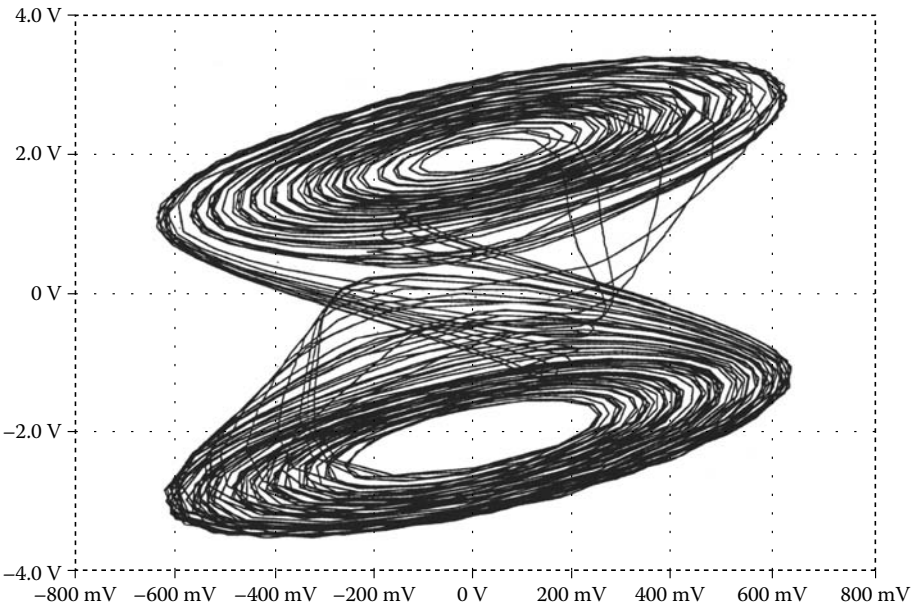


FIGURE 14.30 PSPICE (evaluation version 5.4, July 1993) simulation of Figure 14.25 using the input deck from Figure 14.29 yields this double-scroll Chua's attractor. Horizontal axis V_2 (V) and vertical axis V_1 (V).

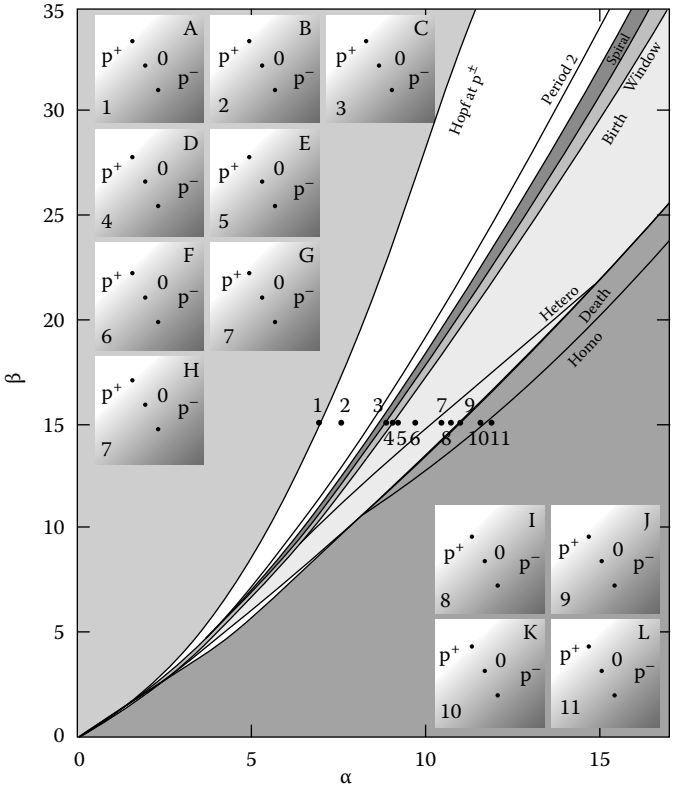


FIGURE 14.31 α–β parameter space diagram for the normalized dimensionless Chua's circuit Equation 14.31 with $a = -8/7$ and $b = -5/7$. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 673, Oct. 1993. With permission.)

Beyond this curve, trajectories diverge toward infinity. Because of eventual passivity in a real circuit, these divergent trajectories will of course converge to a limit cycle in any physical implementation of Chua's circuit.

14.3 Chua's Oscillator

Chua's oscillator [5] (see Figure 14.26) is derived from Chua's circuit by adding a resistor R_0 in series with the inductor L. The oscillator contains a linear inductor, two linear resistors, two linear capacitors, and a single Chua diode N_R. N_R is a voltage-controlled piecewise-linear resistor whose continuous odd-symmetric three-segment DP characteristic (see Figure 14.18) is described explicitly by the relationship

$$I_R = G_b V_R + \frac{1}{2}(G_a - G_b)(|V_R + E| - |V_R - E|)$$

The primary motivation for studying this circuit is that the vector field of Chua's oscillator is topologically conjugate to the vector field of a large class of three-dimensional, piecewise-linear vector fields. In particular, the oscillator can exhibit every dynamical behavior known to be possible in an autonomous three-dimensional, continuous-time dynamical system described by a continuous odd-symmetric three-region piecewise-linear vector field. With appropriate choices of component values, the circuit follows the period-doubling, intermittency, and quasiperiodic routes to chaos.

14.3.1 State Equations

Choosing V_1, V_2, V, and I_3 as state variables, Chua's oscillator may be described by three ODEs:

$$\frac{dV_1}{dt} = \frac{G}{C_1}(V_2 - V_1) - \frac{1}{C_1}f(V_1)$$
$$\frac{dV_2}{dt} = \frac{G}{C_2}(V_1 - V_2) + \frac{1}{C_2}I_3$$
$$\frac{dI_3}{dt} = -\frac{1}{L}V_2 - \frac{R_0}{L}I_3$$

where

$G = 1/R$

$f(V_R) = G_b V_R + 1/2\ (G_a - G_b)\ (|V_R + E| - |V_R - E|)$

The vector field is parameterized by eight constants: L, C_2, G, C_1, R_0, E, G_a, and G_b. We can reduce the number of parameters by normalizing the nonlinear resistor such that its breakpoints are at ± 1 V instead of $\pm EV$, scaling the state variables, and scaling time.

By making the following change of variables: $X_1 = V_1/E$, $X_2 = V_2/E$, $X_3 = I_3/(EG)$, $\tau = t|G/C_2|$, and $k = \text{sgn}(G/C_2)$,* we can rewrite the state Equations 14.16 in normalized dimensionless form:

$$\frac{dX_1}{d\tau} = k\alpha[X_2 - X_1 - f(X_1)]$$
$$\frac{dX_2}{d\tau} = k(X_1 - X_2 + X_3)$$
$$\frac{dX_3}{d\tau} = -k(\beta X_2 + \gamma X_3)$$

* The signum function is defined by $\text{sgn}(x) = x$ if $x > 0$, $\text{sgn}(x) = -x$ if $x < 0$, and $\text{sgn}(0) = 0$.

where $\alpha = C_2/C_1$, $\beta = C_2/(LG^2)$, $\gamma = R_0C_2/(LG)$, and $f(x) = bX + 1/2(a-b)(|X+1| - |X-1|)$ with $a = G_a/G$ and $b = G_b/G$. Thus, each set of eight circuit parameters has an equivalent set of six normalized dimensionless parameters $\{\alpha, \beta, \gamma, a, b, k\}$.

14.3.2 Topological Conjugacy

Two vector fields **F** and **F′** are topologically conjugate if there exists a continuous map h (which has a continuous inverse) such that h maps trajectories of **F** into trajectories of **F′**, preserving time orientation and parametrization of time. If ϕ_t and ϕ'_t are the flows of **F** and **F′**, respectively, then $\phi_t \circ h = h \circ \phi'_t$ for all t. This means that the dynamics of **F** and **F′** are qualitatively the same. If h is linear, then **F** and **F′** are said to be linearly conjugate.

14.3.2.1 Class $\mathscr{C}$

The three-dimensional, autonomous, continuous-time dynamical system defined by the state equation

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}),\ \mathbf{X} \in \mathbb{R}^3$$

is said to belong to class $\mathscr{C}$ iff

1. $\mathbf{F}$: $\mathbb{R}^3 \to \mathbb{R}^3$ is continuous
2. $\mathbf{F}$ is odd-symmetric, i.e., $\mathbf{F}(-X) = -\mathbf{F}(X)$
3. $\mathbb{R}^3$ is partitioned by two parallel boundary planes U_1 and U_{-1} into an inner region D_0, which contains the origin, and two outer regions D_1 and D_{-1}, and $\mathbf{F}$ is affine in each region.

Without loss of generality, the boundary planes and the regions they separate can be chosen as follows:

$$\begin{aligned} D_{-1} &= \{\mathbf{X}: \mathbf{X}_1 \le -1\} \\ U_{-1} &= \{\mathbf{X}: \mathbf{X}_1 = -1\} \\ D_0 &= \{\mathbf{X}: |\mathbf{X}_1| \le 1\} \\ U_1 &= \{\mathbf{X}: \mathbf{X}_1 = 1\} \\ D_1 &= \{\mathbf{X}: \mathbf{X}_1 \ge 1\} \end{aligned}$$

Any vector field in the family $\mathscr{C}$ can then be written in the form

$$\dot{\mathbf{X}} = \begin{cases} \mathbf{A}_{-1}\mathbf{X} - \mathbf{b} & \mathbf{X}_1 \le -1 \\ \mathbf{A}_0\mathbf{X} & -1 \le \mathbf{X}_1 \le 1 \\ \mathbf{A}_1\mathbf{X} + \mathbf{b} & \mathbf{X}_1 \ge 1 \end{cases}$$

where

$$\mathbf{A}_{-1} = \mathbf{A}_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

By continuity of the vector field across the boundary planes,

$$\mathbf{A}_0 = \begin{bmatrix} (a_{11} + b_1) & a_{12} & a_{13} \\ (a_{21} + b_2) & a_{22} & a_{23} \\ (a_{31} + b_3) & a_{32} & a_{33} \end{bmatrix}$$

14.3.2.2 Equivalent Eigenvalue Parameters

Let (μ_1, μ_2, μ_3) denote the eigenvalues associated with the linear vector field in the D_0 region and let (ν_1, ν_2, ν_3) denote the eigenvalues associated with the affine vector fields in the outer regions D_1 and D_{-1}. Define

$$\left.\begin{aligned} p_1 &= \mu_1 + \mu_2 + \mu_3 \\ p_2 &= \mu_1\mu_2 + \mu_2\mu_3 + \mu_3\mu_1 \\ p_3 &= \mu_1\mu_2\mu_3 \\ \\ q_1 &= \nu_1 + \nu_2 + \nu_3 \\ q_2 &= \nu_1\nu_2 + \nu_2\nu_3 + \nu_3\nu_1 \\ q_3 &= \nu_1\nu_2\nu_3 \end{aligned}\right\} \tag{14.16}$$

Because the six parameters $\{p_1, p_2, p_3, q_1, q_2, q_3\}$ are uniquely determined by the eigenvalues $\{\mu_1, \mu_2, \mu_3, \nu_1, \nu_2, \nu_3\}$ and vice versa, the former are called the equivalent eigenvalue parameters. Note that the equivalent eigenvalues are real; they are simply the coefficients of the characteristic polynomials:

$$(s-\mu_1)(s-\mu_2)(s-\mu_3) = s^3 - p_1 s^2 + p_2 s - p_3$$
$$(s-\nu_1)(s-\nu_2)(s-\nu_3) = s^3 - q_1 s^2 + q_2 s - q_3$$

THEOREM 14.3 (Chua et al.) [5]

Let $\{\mu_1, \mu_2, \mu_3, \nu_1, \nu_2, \nu_3\}$ be the eigenvalues associated with a vector field $\mathbf{F(X)} \in \mathscr{C}/\mathscr{E}_0$, *where $\mathscr{E}_0$ is the set of measure zero in the space of equivalent eigenvalue parameters where one of the five conditions in Equation* 14.17 *is satisfied. Then, Chua's oscillator with parameters defined by Equations* 14.18 *and* 14.19 *is linearly conjugate to this vector field.*

$$\left.\begin{aligned} p_1 - q_1 &= 0 \\ p_2 - \left(\frac{p_3 - q_3}{p_1 - q_1}\right) - \left(\frac{p_2 - q_2}{p_1 - q_1}\right)\left(p_1 - \frac{p_2 - q_2}{p_1 - q_1}\right) &= 0 \\ -\left(\frac{p_2 - q_2}{p_1 - q_1}\right) - \frac{k_1}{k_2} &= 0 \\ -k_1 k_3 + k_2\left(\frac{p_3 - q_3}{p_1 - q_1}\right) &= 0 \\ \det \mathbf{K} = \det\begin{bmatrix} 1 & 0 & 0 \\ a_{11} & a_{12} & a_{13} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} &= a_{12}k_{33} - a_{13}k_{32} = 0 \end{aligned}\right\} \tag{14.17}$$

where

$$K_{3i} = \sum_{j=i}^{3} a_{1j}a_{ji}, \quad i = 1, 2, 3$$

We denote by $\tilde{\mathscr{C}}$ the set of vector fields $\mathscr{C}/\mathscr{E}_0$. Two vector fields in $\tilde{\mathscr{C}}$ are linearly conjugate if they have the same eigenvalues in each region.

14.3.3 Eigenvalues-to-Parameters Mapping Algorithm for Chua's Oscillator

Every continuous, third-order, odd-symmetric, three-region, piecewise-linear vector field $\mathbf{F}'$ in $\mathcal{C}$ may be mapped onto a Chua's oscillator (where the vector field $\mathbf{F}$ is topologically conjugate to $\mathbf{F}'$) by means of the following algorithm [5]:

1. Calculate the eigenvalues (μ_1', μ_2', μ_3') and (ν_1', ν_2', ν_3') associated with the linear and affine regions, respectively, of the vector field $\mathbf{F}'$ of the circuit or system whose attractor is to be reproduced (up to linear conjugacy) by Chua's oscillator.
2. Find a set of circuit parameters $\{C_1, C_2, L, R, R_0, G_a, G_b, E\}$ (or dimensionless parameters $\{\alpha, \beta, \gamma, a, b, k\}$) so that the resulting eigenvalues μ_j and ν_j for Chua's oscillator satisfy $\mu_j = \mu_j'$ and $\nu = \nu_j'$, $j = 1, 2, 3$.

Let $\{p_1, p_2, p_3, q_1, q_2, q_3\}$ be the equivalent eigenvalue parameters defined by Equation 14.16. Furthermore, let

$$\left.\begin{aligned} k_1 &= -p_3 + \left(\frac{p_3 - q_3}{p_1 - q_1}\right)\left(p_1 - \frac{p_2 - q_2}{p_1 - q_1}\right) \\ k_2 &= p_2 - \left(\frac{p_3 - q_3}{p_1 - q_1}\right) - \left(\frac{p_2 - q_2}{p_1 - q_1}\right)\left(p_1 - \frac{p_2 - q_2}{p_1 - q_1}\right) \\ k_3 &= -\left(\frac{p_2 - q_2}{p_1 - q_1}\right) - \frac{k_1}{k_2} \\ k_4 &= -k_1 k_3 + k_2\left(\frac{p_3 - q_3}{p_1 - q_1}\right) \end{aligned}\right\} \tag{14.18}$$

The corresponding circuit parameters are given by

$$\left.\begin{aligned} C_1 &= 1 \\ C_2 &= -\frac{k_2}{k_3^2} \\ L &= -\frac{k_3^2}{k_4} \\ R &= -\frac{k_3}{k_2} \\ R_0 &= -\frac{k_1 k_3^2}{k_2 k_4} \\ G_a &= -p_1 + \left(\frac{p_2 - q_2}{p_1 - q_1}\right) + \frac{k_2}{k_3} \\ G_b &= -q_1 + \left(\frac{p_2 - q_2}{p_1 - q_1}\right) + \frac{k_2}{k_3} \end{aligned}\right\} \tag{14.19}$$

The breakpoint E of the piecewise-linear Chua diode can be chosen arbitrarily because the choice of E does not affect either the eigenvalues or the dynamics; it simply scales the circuit variables. In a practical

realization of the circuit, one should scale the voltages and currents so that they lie within the inner three segments of the nonlinear resistor N_R.

The dimensionless parameters can be calculated as follows:

$$\left.\begin{aligned}
\alpha &= -\frac{k_2}{k_3^2} \\
\beta &= \frac{k_4}{k_2 k_3^2} \\
\gamma &= \frac{k_1}{k_2 k_3} \\
a &= -1 + \frac{k_3}{k_2}\left(p_1 - \frac{p_2 - q_2}{p_1 - q_1}\right) \\
b &= -1 + \frac{k_3}{k_2}\left(q_1 - \frac{p_2 - q_2}{p_1 - q_1}\right) \\
k &= \mathrm{sgn}(k_3)
\end{aligned}\right\} \tag{14.20}$$

Example: Torus

Figure 14.32 shows a practical implementation of Chua's oscillator that exhibits a transition to chaos by torus breakdown. A complete list of components is given in Table 14.3.

A SPICE simulation of this circuit produces a quasiperiodic voltage $\nu(2)$ $(= -V_1)$, as expected (see Figure 14.33). The resistor R0 is not explicitly added to the circuit, but models the dc resistance of the inductor.

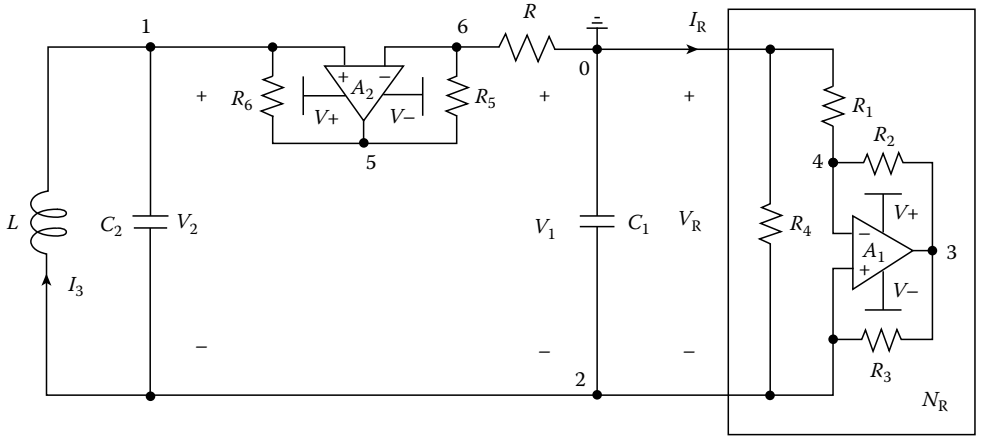


FIGURE 14.32 Practical implementation of Chua's oscillator using an op-amp and four resistors to realize the Chua diode. (From Kennedy, M.P., *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 640, 657, Oct. 1993. With permission.) The negative resistor G is realized by means of a negative resistance converter (A_2, R_5, R_6, and positive resistor R). If $R_2 = R_3$ and $R_5 = R_6$, and $G_a = 1/R_4 - 1/R_1$, $G_b = 1/R_4 + 1/R_2$, and $G = -1/R$. Component values are listed in Table 14.3.

TABLE 14.3 Component List for the Chua Oscillator in Figure 14.32

Element	Description	Value	Tolerance (%)
A_1	Op-amp ($\frac{1}{2}$ AD712, TL082, or equivalent)	—	—
A_2	Op-amp ($\frac{1}{2}$ AD712, TL082, or equivalent)	—	—
C_1	Capacitor	47 nF	±5
C_2	Capacitor	820 nF	±5
R_1	$\frac{1}{4}$ W resistor	6.8 kΩ	±5
R_2	$\frac{1}{4}$ W resistor	47 kΩ	±5
R_3	$\frac{1}{4}$ W resistor	47 kΩ	±5
R_4	Potentiometer	2 kΩ	—
R_5	W resistor	220 Ω	±5
R_6	$\frac{1}{4}$ W resistor	220 Ω	±5
R	Potentiometer	2 kΩ	—
L	Inductor(TOKO-type 10 RB, or equivalent)	18 mH	±10

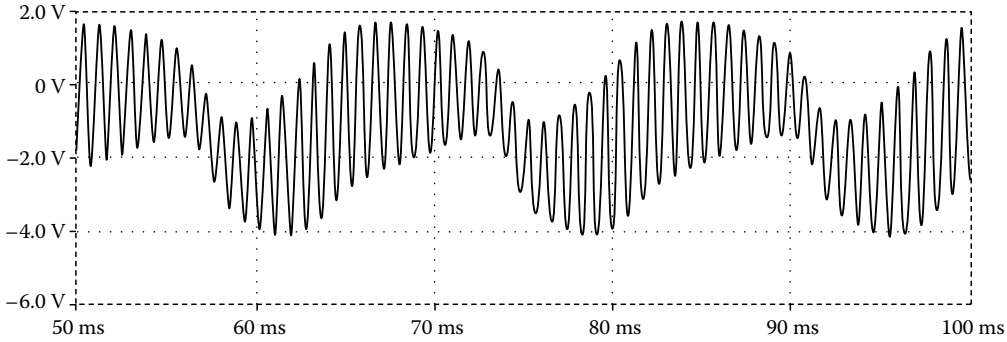


FIGURE 14.33 PSPICE simulation (.TRAN 0.01 ms 100 ms 50 ms) of Figure 14.32 with initial conditions .IC $v(2) = -0.1$ $v(1) = -0.1$ and tolerances .OPTIONS RELTOL = 1 E − 4 ABSTOL = 1 E − 4 yields this quasiperiodic voltage waveform at node 2.

14.4 van der Pol Neon Bulb Oscillator

In a paper titled "Frequency demultiplication," the eminent Dutch electrical engineer Balthazar van der Pol described an experiment in which, by tuning the capacitor in a neon bulb *RC* relaxation oscillator driven by a sinusoidal voltage source (see Figure 14.34), "currents and voltages appear in the system which are whole submultiples of the driving frequency" [11].

The circuit consists of a high-voltage dc source E attached via a large series resistance R to a neon bulb and capacitor C that are connected in parallel; this forms the basic relaxation oscillator. Initially, the capacitor is discharged and the neon bulb is nonconducting. The dc source charges C with time constant RC until the voltage across the neon bulb is

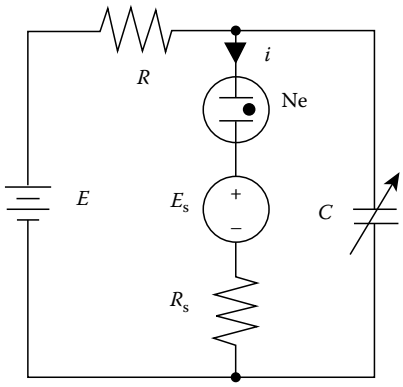


FIGURE 14.34 Sinusoidally driven neon bulb relaxation oscillator. Ne is the neon bulb.

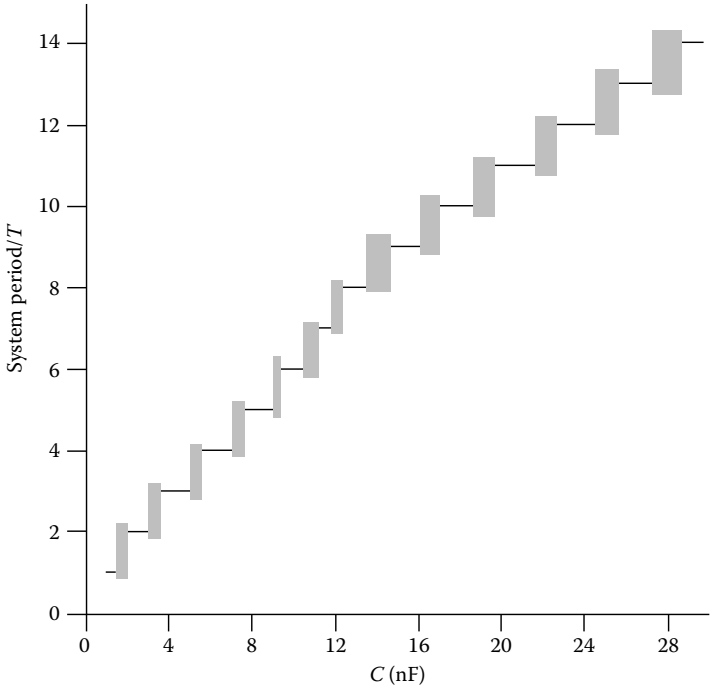


FIGURE 14.35 Normalized current pulse pattern repetition rate versus C for the sinusoidally driven neon relaxation oscillator in Figure 14.34, showing a coarse staircase structure of mode-lockings. (From Kennedy, M.P. and Chua, L.O., *IEEE Trans. Circuits Syst.*, CAS-33, 975, Oct. 1986. With permission.)

sufficient to turn it on. Once lit, the bulb presents a shunt low resistance path to the capacitor. The voltage across the capacitor falls exponentially until the neon arc is quenched, the bulb is returned to its "off" state, and the cycle repeats.

A sinusoidal voltage source $E_s = E_0 \sin(2\pi f_s t)$ is inserted in series with the neon bulb. Its effect is to perturb the "on" and "off" switching thresholds of the capacitor voltage.

Experimental results for this circuit are summarized in Figure 14.35, where the ratio of the system period (time interval before the pattern of current pulses repeats itself) to the period T of the forcing is plotted versus the capacitance C.

van der Pol noted that as the capacitance was increased from that value (C_0) for which the natural frequency f_0 of the undriven relaxation oscillator equaled that of the sinusoidal source (system period/$T = 1$), the system frequency made "discrete jumps from one whole submultiple of the driving frequency to the next" (detected by means of "a telephone coupled loosely in some way to the system"). van der Pol noted that "often an irregular noise is heard in the telephone receiver before the frequency jumps to the next lower value"; van der Pol had observed chaos. Interested primarily in frequency demultiplication, he dismissed the "noise" as "a subsidiary phenomenon."

Typical current waveforms, detected by means of a small current-sensing resistor R_s placed in series with the bulb are shown in Figure 14.36. These consist of a series of sharp spikes, corresponding to the periodic firing of the bulb. Figure 14.36c shows a nonperiodic "noisy" signal of the type noticed by van der Pol.

The frequency locking behavior of the driven neon bulb oscillator circuit is characteristic of forced oscillators that contain two competing frequencies: the natural frequency f_0 of the undriven oscillator and the driving frequency f_s. If the amplitude of the forcing is small, either quasiperiodicity or mode-locking occurs. For a sufficiently large amplitude of the forcing, the system may exhibit chaos.

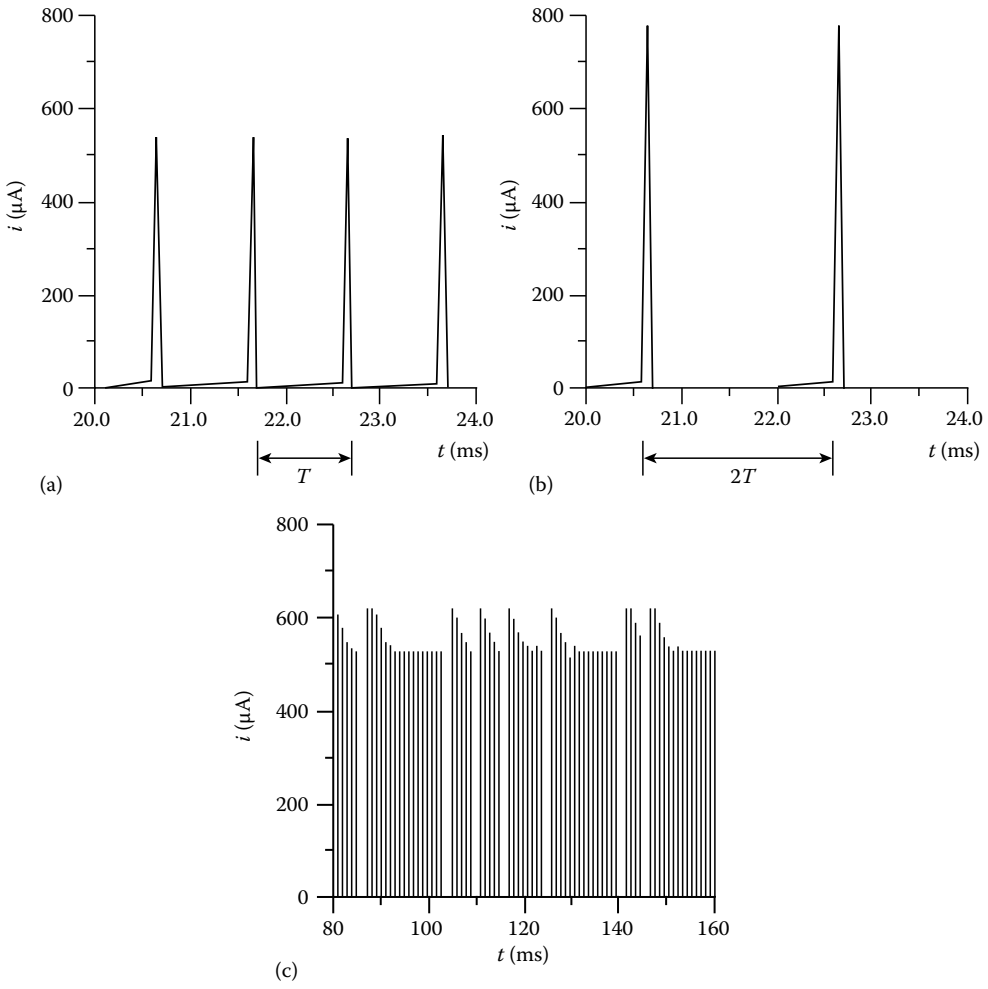


FIGURE 14.36 Periodic and chaotic neon bulb current waveforms. (a) One current pulse per cycle of E_s: $f_s/f_d = 1/1$; (b) one current pulse every two cycles of E_s: $f_s/f_d = 2/1$; (c) "noisy" current waveform.

14.4.1 Winding Numbers

Subharmonic frequency locking in a forced oscillator containing two competing frequencies f_1 and f_2 may be understood in terms of a winding number. The concept of a winding number was introduced by Poincaré to describe periodic and quasiperiodic trajectories on a torus.

A trajectory on a torus that winds around the minor axis of the torus with frequency f_1 revolutions per second, and completes revolutions of the major axis with frequency f_2, may be parametrized by two angular coordinates $\theta_1 \equiv f_1 t$ and $\theta_2 \equiv f_2 t$, as illustrated in Figure 14.37 [6]. The angles of rotation θ_1 and θ_2 about the major and minor axes of the torus are normalized so that one revolution corresponds to a change in θ of 1.

A Poincaré map for this system can be defined by sampling the state θ_1 with period $\tau = 1/f_2$. Let $\theta_k = \theta_1(k\tau)$. The Poincaré map has the form

$$\theta_{k+1} = G(\theta_k), \quad k = 0, 1, 2, \ldots$$

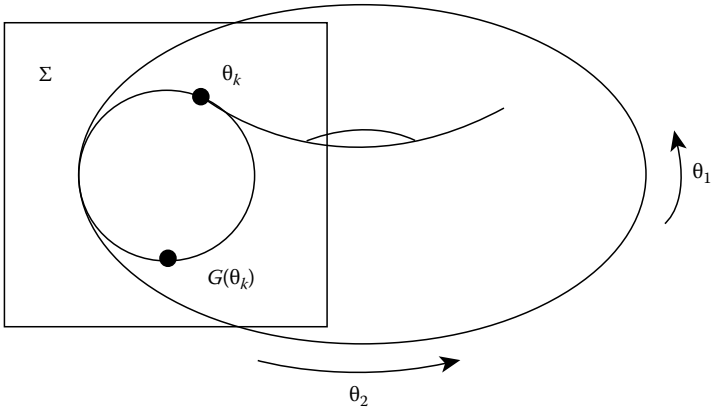


FIGURE 14.37 A trajectory on a torus is characterized by two normalized angular coordinates. $\theta_1 = f_1 t$ is the angle of rotation about the minor axis of the torus, while $\theta_2 = f_2 t$ is the angle of rotation along the major axis, where f_1 and f_2 are the frequencies of rotation about the corresponding axes. A Poincaré map $\theta_{k+1} = G(\theta_k)$ is defined by sampling the trajectory with frequency $1/f_2$. The winding number w counts the average number of revolutions in the Poincaré section per iteration of the map.

If $f_1/f_2 = p/q$ is rational, then the trajectory is periodic, closing on itself after completing q revolutions about the major axis of the torus. In this case, we say that the system is periodic with period q and completes p cycles per period. If the ratio p/q is irrational then the system is quasiperiodic; a trajectory covers the surface of the torus, coming arbitrarily close to every point on it, but does not close on itself.

The winding number w is defined by

$$w = \lim_{k \to \infty} \frac{G^{(k)}(\theta_0)}{k}$$

where $G^{(k)}$ denotes the k-fold iterate of G and θ_0 is the initial state.

The winding number counts the average number of revolutions in the Poincaré section per iteration. Equivalently, w equals the average number of turns about the minor axis per revolution about the major axis of the torus.* Periodic orbits possess rational winding numbers and are called resonant; quasiperiodic trajectories have irrational winding numbers.

14.4.2 Circle Map

A popular paradigm for explaining the behavior of coupled nonlinear oscillators with two competing frequencies is the circle map:

$$\theta_{k+1} = \left(\theta_k + \frac{K}{2\pi} \sin(2\pi\theta_k) + \Omega \right) \bmod 1, \quad k = 0, 1, 2, \ldots \tag{14.21}$$

so-called because it maps the circle into itself. The sinusoidal term represents the amplitude of the forcing, and Ω is the ratio of the natural frequency of the unperturbed system and the forcing frequency [18].

When $K \equiv 0$, the steady state of the discrete-time dynamical system (Equation 14.22) is either periodic or quasiperiodic, depending on whether Ω is rational or irrational.

* Either frequency may be chosen to correspond to the major axis of the torus, so the winding number and its reciprocal are equivalent.

If the amplitude K of the forcing is nonzero but less than unity, the steady-state is q-periodic when $\Omega = p/q$ is rational. In this case, a nonzero mode-locked window $[\Omega_{\min}(w), \Omega_{\max}(w)]$ occurs, over which $w = p/q$. A mode-locked region is delimited by saddle-node bifurcations at $\Omega_{\min}(w)$ and $\Omega_{\max}(w)$ [18].

The function $w(\Omega)$ in Figure 14.38 is monotone increasing and forms a Devil's staircase with plateaus at every rational value of w—for example, the step with winding number 1/2 is centered at $\Omega = 0.5$. An experimental Devil's staircase for the driven neon bulb circuit, with low-amplitude forcing, is shown in Figure 14.39.

As the amplitude k is increased, the width of each locked interval in the circle map increases so that mode-locking becomes more common and quasiperiodicity occurs over smaller ranges of driving frequencies. The corresponding (K, Ω) parameter space diagram (see Figure 14.40) consists of a series of distorted triangles, known as Arnold Tongues, with apexes that converge to rational values of Ω at $K = 0$.

Within a tongue, the winding number is constant, yielding one step of the Devil's staircase. The winding numbers of adjacent tongues are related by a *Farey tree* structure. Given two periodic windows with winding number $w_1 = p/q$ and $w_2 = r/s$, another periodic window with winding number $w = (\alpha p + \beta r)/(\alpha q + \beta s)$ can always be found, where p, q, r, and s are relatively prime and α and β are strictly positive integers. Furthermore, the widest mode-locked window between w_1 and w_2 has winding number $(p + r)/(q + s)$. For example, the widest step between those with winding numbers 1/2 and 2/3 in Figure 14.38 has $w = 3/5$.

The sum of the widths of the mode-locked states increases monotonically from zero at $K = 1$ to unity at $K = 1$. Below the critical line $K = 1$, the tongues bend away from each other and do not overlap. At $K = 1$, tongues begin to overlap, a kink appears in the Poincaré section and the Poincaré map develops a horseshoe; this produces coexisting attractors and chaos.

The transition to chaos as K is increased through $K = 1$ may be by a period-doubling cascade within a tongue, intermittency, or directly from a quasiperiodic trajectory by the abrupt disappearance of that trajectory (a blue sky catastrophe). This qualitative behavior is observed in van der Pol's neon bulb circuit.

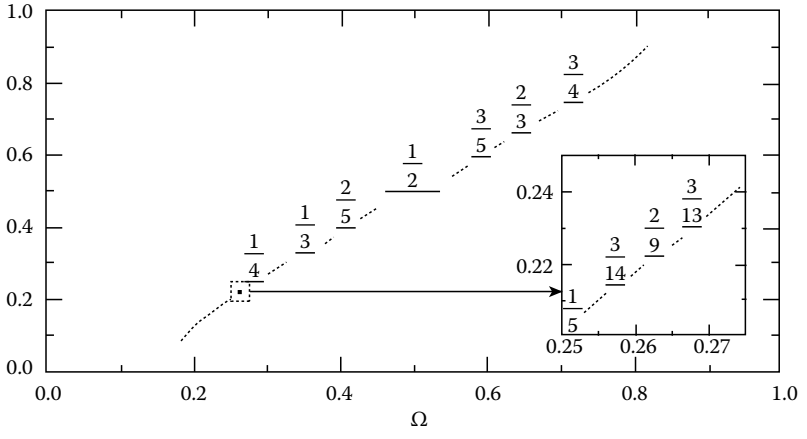


FIGURE 14.38 Devil's staircase for the circle map with $K = 1$. The steps indicate the regions in which w is constant. The staircase is self-similar in the sense that its structure is reproduced qualitatively at smaller scales (see inset). (From Glazier, J.A. and Libchaber, A., *IEEE Trans. Circuits Syst.*, 35, 793, July 1988. With permission.)

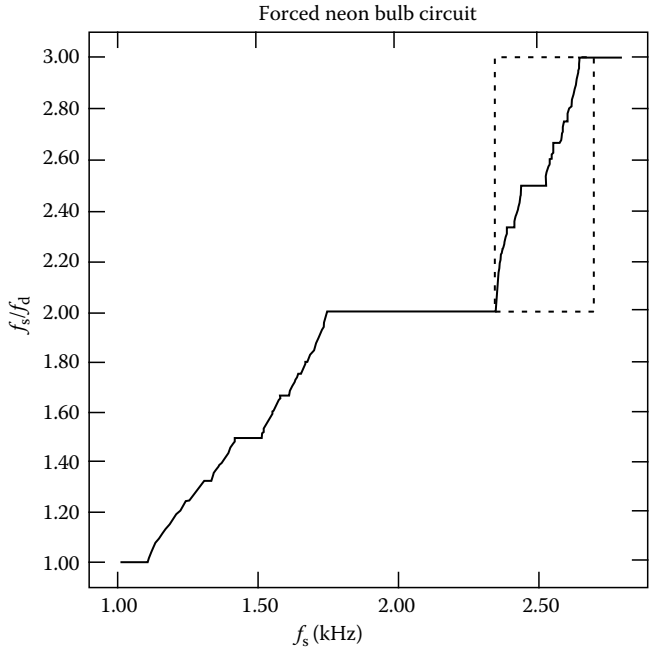


FIGURE 14.39 Experimentally measured staircase structure of lockings for a forced neon bulb relaxation oscillator. The winding number is given by f_s/f_d, the ratio of the frequency of the sinusoidal driving signal to the average frequency of current pulses through the bulb. (From Kennedy, M.P., Krieg, K.R., and Chua, L.O., *IEEE Trans. Circuits Syst*., 36, 1137, Aug. 1989. With permission.)

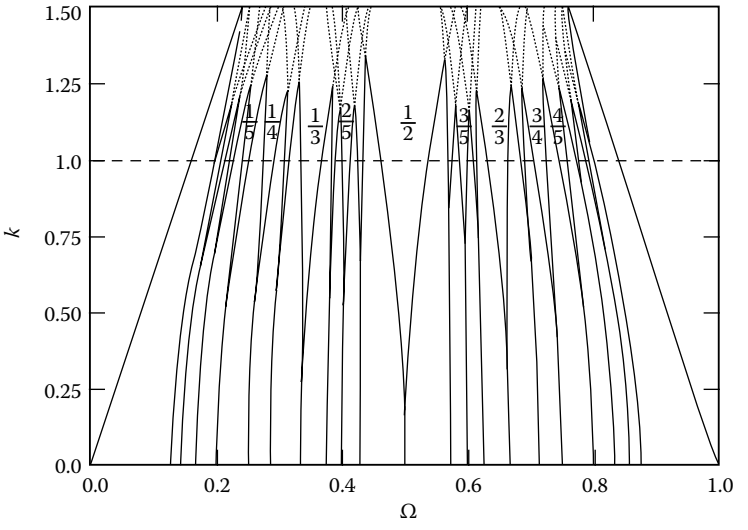


FIGURE 14.40 Parameter space diagram for the circle map showing Arnold tongue structure of lockings in the K–Ω plane. The relative widths of the tongues decrease as the denominator of the winding number increases. Below the critical line $K = 1$, the tongues bend away from each other and do not overlap; for $K > 1$, the Poincaré map develops a fold and chaos can occur. (From Glazier, J.A. and Libchaber, A., *IEEE Trans. Circuits Syst*., 35, 793, July 1988. With permission.)

14.4.3 Experimental Observations of Mode-Locking and Chaos in van der Pol's Neon Bulb Circuit

With the signal source E_s zeroed, the natural frequency of the undriven relaxation oscillator is set to 1 kHz by tuning capacitance C to C_0. A sinusoidal signal with frequency 1 kHz and amplitude E_0 is applied as shown in Figure 14.34. The resulting frequency of the current pulses (detected by measuring the voltage across R_s) is recorded with C as the bifurcation parameter.

14.4.3.1 C Bifurcation Sequence

If a fixed large-amplitude forcing E_s is applied and C is increased slowly, the system at first continues to oscillate at 1 kHz (Figure 14.36a) over a range of C, until the frequency "suddenly" drops to 1000/2 Hz (Figure 14.36b), stays at that value over an additional range of capacitance, drops to 1000/3 Hz, then 1000/4, 1000/5, and so on as far as 1000/20 Hz. These results are summarized in Figure 14.35.

Between each two submultiples of the oscillator driving frequency, a further rich structure of submultiples is found. At the macroscopic level (the coarse structure examined by van der Pol) increasing the value of C causes the system periods to step from T (1 ms) to $2T$, $3T$, $4T, \ldots$, where the range of C for which the period is fixed is much greater than that over which the transitions occur (Figure 14.35).

Examining the shaded transition regions more closely, one finds that between any two "macroscopic" regions where the period is fixed at $(n-1)T$ and nT $(n>1)$, respectively, there lies a narrower region over which the system oscillates with stable period $(2n-1)T$. Further, between $(n-1)T$ and $(2n-1)T$, one finds a region of C for which the period is $(3n-2)T$, and between $(2n-1)T$ and nT, a region with period $(3n-1)T$. Indeed, between any two stable regions with periods $(n-1)T$ and nT, respectively, a region with period $(2n-1)T$ can be found. Figure 14.41 depicts an enlargement of the C axis in the region of the T to $2T$ macrotransition, showing the finer period-adding structure.

Between T and $2T$ is a region with stable period $3T$. Between this and $2T$, regions of periods $5T$, $7T$, $9T, \ldots$, $25T$ are detected. Current waveforms corresponding to period-three and period-five steps, with winding numbers 3/2 and 5/3, respectively, are shown in Figure 14.42.

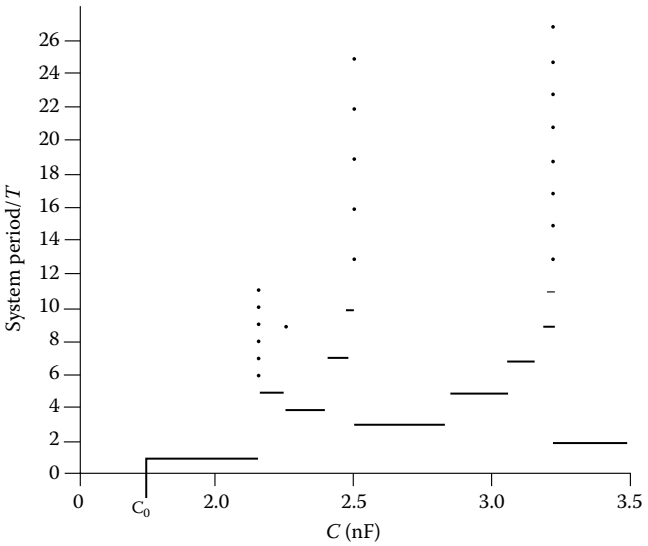


FIGURE 14.41 Experimental pulse pattern repetition rate versus C for van der Pol's forced neon bulb oscillator, showing fine period-adding structure. (From Kennedy, M.P. and Chua, L.O., *IEEE Trans. Circuits Syst.*, CAS-33, 975, Oct. 1986. With permission.)

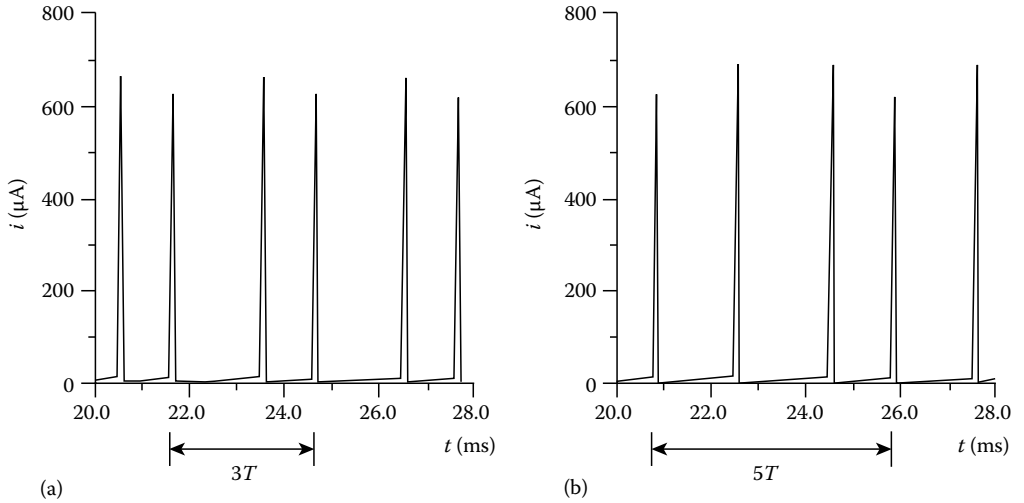


FIGURE 14.42 Neon bulb current waveforms. (a) Two pulses every three cycles of f_s: $f_s/f_d = 3/2$; (b) three pulses every five cycles $f_s/f_d = 5/3$.

A region of period $4T$ lies between T and $3T$, with steps $7T$, $10T$, $13T, \ldots,$ $25T$ between that and $3T$.

In practice, it becomes difficult to observe cycles with longer periods because stochastic noise in the experimental circuit can throw the solution out of the narrow window of existence of a high period orbit.

14.4.3.2 Experimental f_s Bifurcation Sequence with Low-Amplitude Forcing

An experimental Devil's staircase may be plotted for this circuit by fixing the parameters of the relaxation oscillator and the amplitude of the sinusoidal forcing signal, and choosing the forcing frequency f_s as the bifurcation parameter. The quantity f_s/f_d is the equivalent winding number in this case, where f_d is the average frequency of the current pulses through the neon bulb.

Experimental results for the neon bulb circuit with low-amplitude forcing are presented in Figures 14.39 and 14.43. The monotone staircase of lockings is consistent with a forcing signal of small amplitude. Note that the staircase is self-similar in the sense that its structure is reproduced qualitatively at smaller scales of the bifurcation parameter. If the amplitude of the forcing is increased, the onset of chaos is indicated by a nonmonotonicity in the staircase.

14.4.4 Circuit Model

The experimental behavior of van der Pol's sinusoidally driven neon bulb circuit may be reproduced in simulation by an equivalent circuit (see Figure 14.44) in which the only nonlinear element (the neon bulb) is modeled by a nonmonotone current-controlled resistor with a series parasitic inductor L_p.

The corresponding state equations are

$$\frac{dV_C}{dt} = -\frac{1}{RC}V_C - \frac{1}{C}I_L + \frac{E}{RC}$$

$$\frac{dI_L}{dt} = \frac{1}{L_p}V_C - \frac{R_s}{L_p}I_L - \frac{f(I_L)}{L_p} - \frac{E_0 \sin(2\pi f_s t)}{L_p}$$

where $V = f(I)$ is the DP characteristic of the current-controlled resistor (see Figure 14.45).

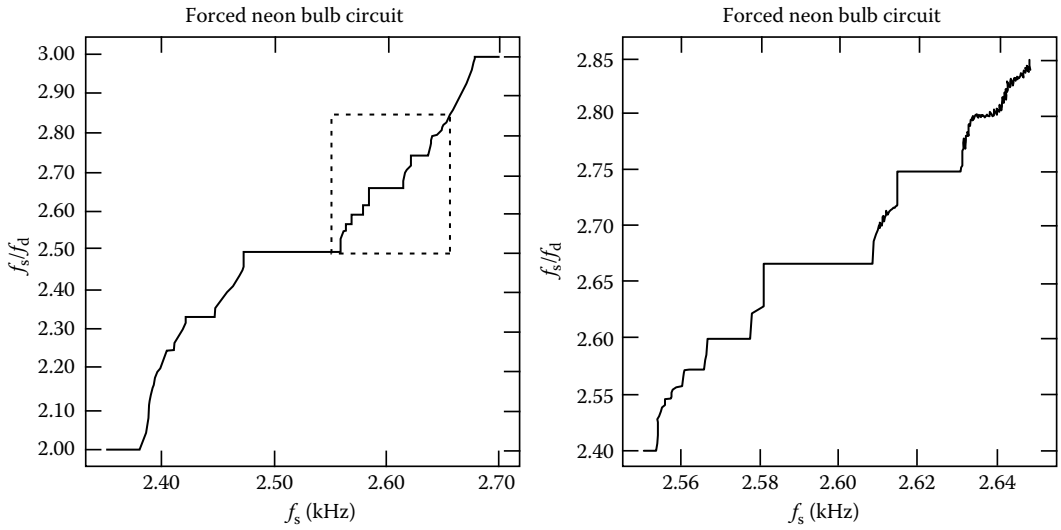


FIGURE 14.43 Magnification of Figure 14.39 showing self-similarity. (From Kennedy, M.P., Krieg, K.R., and Chua, L.O., *IEEE Trans. Circuits Syst.*, 36, 1137, Aug. 1989. With permission.)

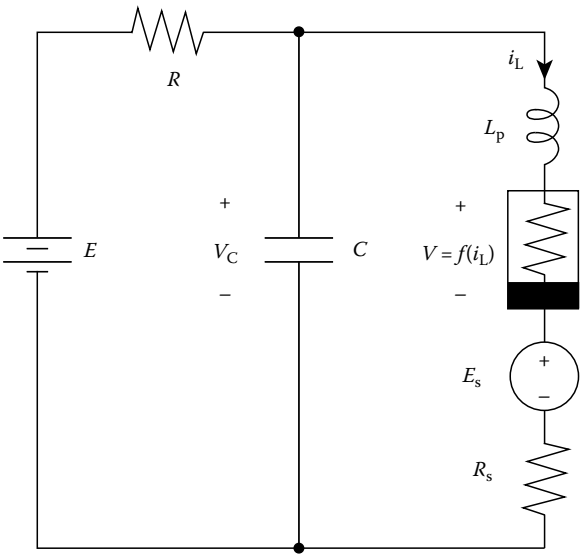


FIGURE 14.44 van der Pol's neon bulb circuit—computer model. The bulb is modeled by a nonmonotonic current-controlled nonlinear resistor with parasitic transit inductance L_p.

14.5 Synchronization of Chaotic Circuits

Chaotic steady-state solutions are characterized by sensitive dependence on initial conditions; trajectories of two identical autonomous continuous-time dynamical systems started from slightly different initial conditions quickly become uncorrelated. Surprisingly perhaps, it is nevertheless possible to synchronize these systems in the sense that a trajectory of one asymptotically approaches that of the other. Two trajectories $X_1(t)$ and $X_2(t)$ are said to *synchronize* if

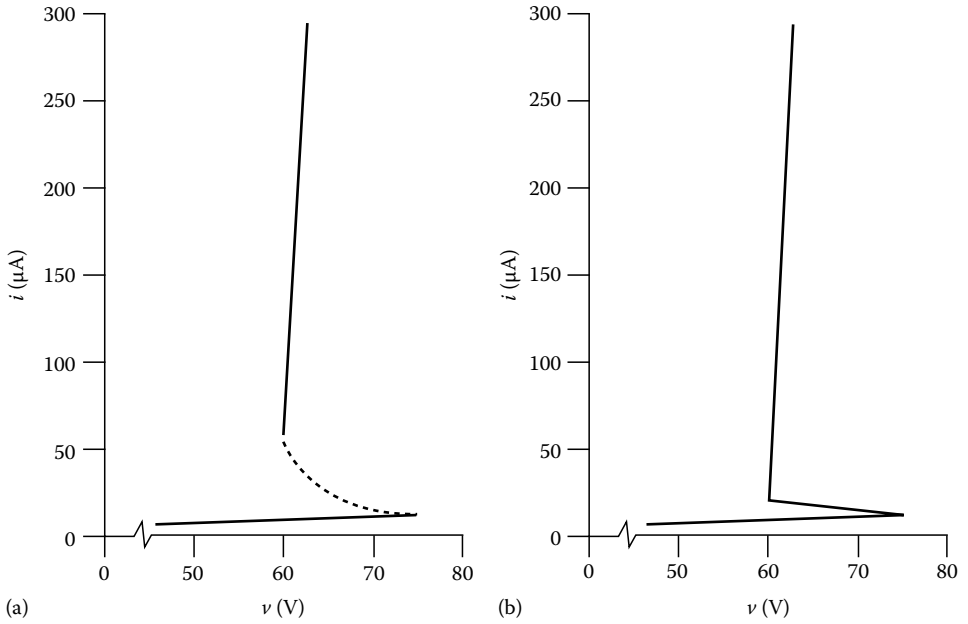


FIGURE 14.45 Neon bulb DP characteristics: (a) measured and (b) simulated. (From Kennedy, M.P. and Chua, L. O., *IEEE Trans. Circuits Syst*., CAS-33, 976, Oct. 1986. With permission.)

$$\lim_{t\to\infty} \|\mathbf{X}_1(t) - \mathbf{X}_2(t)\| = 0$$

In this section, we describe two techniques for synchronizing chaotic trajectories.

14.5.1 Linear Mutual Coupling

The simplest technique for synchronizing two dynamical systems

$$\begin{aligned}\dot{\mathbf{X}}_1 &= \mathbf{F}_1(\mathbf{X}_1), \quad \mathbf{X}_1(0) = \mathbf{X}_{1_0}\\ \dot{\mathbf{X}}_2 &= \mathbf{F}_2(\mathbf{X}_2), \quad \mathbf{X}_2(0) = \mathbf{X}_{2_0}\end{aligned}$$

is by linear mutual coupling of the form

$$\begin{aligned}\dot{\mathbf{X}}_1 &= \mathbf{F}_1(\mathbf{X}_1) + \mathbf{K}(\mathbf{X}_2 - \mathbf{X}_1), \quad \mathbf{X}_1(0) = \mathbf{X}_{1_0}\\ \dot{\mathbf{X}}_2 &= \mathbf{F}_2(\mathbf{X}_2) + \mathbf{K}(\mathbf{X}_2 - \mathbf{X}_1), \quad \mathbf{X}_2(0) = \mathbf{X}_{2_0}\end{aligned} \tag{14.22}$$

where $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^n$ and $K = \text{diag}\,(K_{11}, K_{22}, \ldots, K_{nn})^{\mathrm{T}}$.

Here, $\mathbf{X}_1(t)$ is called the goal dynamics. The synchronization problem may be stated as follows: find $\mathbf{K}$ such that

$$\lim_{t\to\infty} \|\mathbf{X}_1(t) - \mathbf{X}_2(t)\| = 0$$

that is, that the solution $\mathbf{X}_2(t)$ synchronizes with the goal trajectory $\mathbf{X}_1(t)$.

In general, it is difficult to prove that synchronization occurs, unless an appropriate Lyapunov function* of the error system $\mathbf{E}(t) = \mathbf{X}_1(t) - \mathbf{X}_2(t)$ can be found. However, several examples exist in the literature where mutually coupled chaotic systems synchronize over particular ranges of parameters.

Example: Mutually Coupled Chua's Circuits

Consider a linear mutual coupling of two Chua's circuits. In dimensionless coordinates, the system under consideration is

$$\frac{dX_1}{dt} = \alpha[X_2 - X_1 - f(X_1)] + K_{11}(X_4 - X_1)$$
$$\frac{dX_2}{dt} = X_1 - X_2 - X_3 + K_{22}(X_5 - X_2)$$
$$\frac{dX_3}{dt} = -\beta y + K_{33}(X_6 - X_3)$$
$$\frac{dX_4}{dt} = \alpha(X_5 - X_4 - f(X_4)) + K_{11}(X_1 - X_4)$$
$$\frac{dX_5}{dt} = X_4 - X_5 - X_6 + K_{22}(X_2 - X_5)$$
$$\frac{dX_6}{dt} = -\beta X_5 + K_{33}(X_3 - X_6)$$

Two mutually coupled Chua's circuits characterized by $\alpha = 10.0$, $\beta = 14.87$, $a = -1.27$, and $b = -0.68$ will synchronize (the solutions of the two systems will approach each other asymptotically) for the following matrices K:

X_1 – coupling $K_{11} > 0.5$, $K_{22} = K_{33} = 0$
X_2 – coupling $K_{22} > 5.5$, $K_{11} = K_{33} = 0$
X_3 – coupling $0.7 < K_{33} < 2$, $K_{11} = K_{22} = 0$

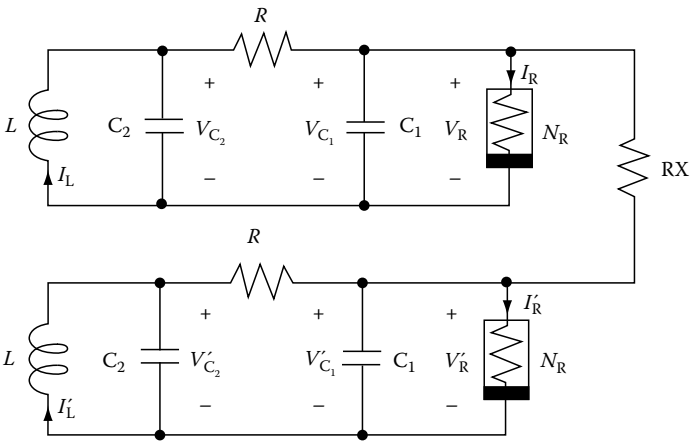


FIGURE 14.46 Synchronization of two Chua's circuits by means of resistive coupling between V_{C_1} and V'_{C_1}.

* For a comprehensive exposition of Lyapunov stability theory, see Ref. [19].

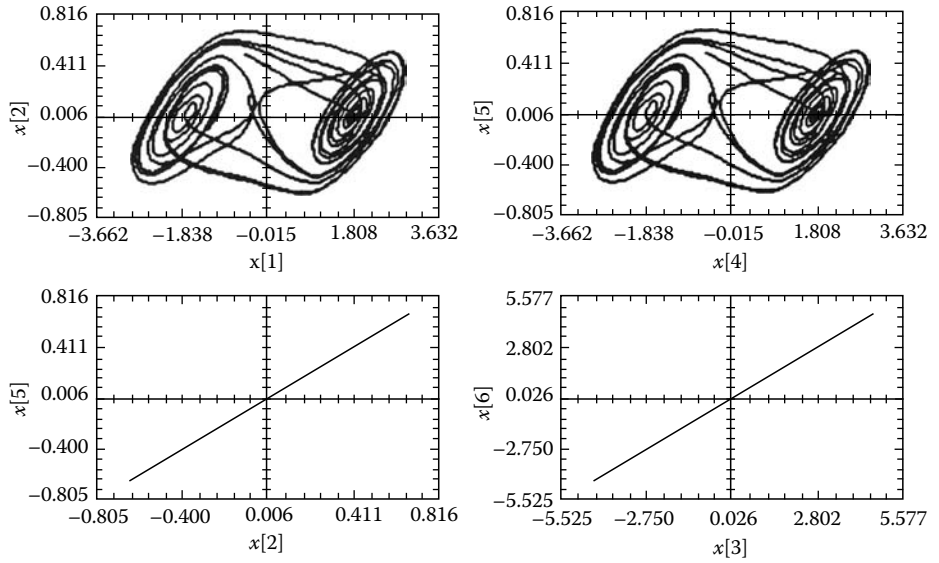


FIGURE 14.47 Simulation of the normalized dimensionless form of Figure 14.46, illustrating synchronization by mutual coupling of state variables. Identify $\{x[1], x[2], x[3]\}$ with $\{V_{C_1}\ V_{C_2}\ I_L\}$ and $\{x[4], x[5], x[6]\}$ with $\{V'_{C_1}\ V'_{C_2}\ I'_L\}$ synchronizes with $V_{C_2}(t)$ and $I'_L(t)$ synchronizes with $I_L(t)$. (From Chua, L.O., Itoh, M., Kočarev, L., and Eckert, K., *J. Circuits Syst. Comput.*, 3, 1, 99, Mar. 1993. With permission.)

Coupling between states X_1 and X_4 may be realized experimentally by connecting a resistor between the tops of the nonlinear resistors, as shown in Figure 14.46. States X_2 and X_5 may be coupled by connecting the tops of capacitors C_2 by means of a resistor.

A simulation of the system, which confirms synchronization of the two chaotic Chua's circuits in the case of linear mutual coupling between state V_{C_1} and V'_{C_1} is presented in Figure 14.47.

14.5.2 Pecora–Carroll Drive-Response Concept

The drive-response synchronization scheme proposed by Pecora and Carroll applies to systems that are drive-decomposable [16]. A dynamical system is called drive-decomposable if it can be partitioned into two subsystems that are coupled so that the behavior of the second (called the response subsystem) depends on that of the first, but the behavior of the first (called the drive subsystem) is independent of that of the second.

To construct a drive-decomposable system, an n-dimensional autonomous continuous-time dynamical system

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}), \quad \mathbf{X}(0) = \mathbf{X}_0 \tag{14.23}$$

where $\mathbf{X} = (X_1,\ X_2, \ldots,\ X_n)^T$ and $\mathbf{F}(\mathbf{X}) = [F_1(\mathbf{X}),\ F_2(\mathbf{X}), \ldots,\ F_n(\mathbf{X})]^T$, is first partitioned into two subsystems

$$\dot{\mathbf{X}}_1 = \mathbf{F}_1(\mathbf{X}_1, \mathbf{X}_2), \quad \mathbf{X}_1(0) = \mathbf{X}_{1_0} \tag{14.24}$$

$$\dot{\mathbf{X}}_2 = \mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_2), \quad \mathbf{X}_2(0) = \mathbf{X}_{2_0} \tag{14.25}$$

where $\mathbf{X}_1 = (X_1, X_2, \ldots, X_m)^T$, $\mathbf{X}_2 = (X_{m+1}, X_{m+2}, \ldots, X_n)^T$,

$$\mathbf{F}_1(\mathbf{X}_1, \mathbf{X}_2) = \begin{pmatrix} F_1(\mathbf{X}_1, \mathbf{X}_2) \\ F_2(\mathbf{X}_1, \mathbf{X}_2) \\ \vdots \\ F_m(\mathbf{X}_1, \mathbf{X}_2) \end{pmatrix}$$

and

$$\mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_2) = \begin{pmatrix} F_{m+1}(\mathbf{X}_1, \mathbf{X}_2) \\ F_{m+2}(\mathbf{X}_1, \mathbf{X}_2) \\ \vdots \\ F_n(\mathbf{X}_1, \mathbf{X}_2) \end{pmatrix}$$

An identical $(n - m)$-dimensional copy of the second subsystem, with $\mathbf{X}_3$ as state variable and $\mathbf{X}_1$ as input, is appended to form the following $(2n - m)$-dimensional coupled drive-response system:

$$\dot{\mathbf{X}}_1 = \mathbf{F}_1(\mathbf{X}_1, \mathbf{X}_2), \quad \mathbf{X}_1(0) = \mathbf{X}_{1_0} \tag{14.26}$$

$$\dot{\mathbf{X}}_2 = \mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_2), \quad \mathbf{X}_2(0) = \mathbf{X}_{2_0} \tag{14.27}$$

$$\dot{\mathbf{X}}_3 = \mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_3), \quad \mathbf{X}_3(0) = \mathbf{X}_{3_0} \tag{14.28}$$

The n-dimensional dynamical system defined by Equations 14.26 and 14.27 is called the drive system and Equation 14.28 is called the response subsystem.

Note that the second drive subsystem (Equation 14.27) and the response subsystem (Equation 14.28) lie in state spaces of dimension $\mathbb{R}^{(n-m)}$ and have identical vector fields $\mathbf{F}_2$ and inputs $\mathbf{X}_1$.

Consider a trajectory $\mathbf{X}_3(t)$ of Equation 14.29 that originates from an initial state $\mathbf{X}_{3_0}$ "close" to $\mathbf{X}_{2_0}$. We may think of $\mathbf{X}_2(t)$ as a perturbation of $\mathbf{X}_3(t)$. In particular, define the error $\mathbf{X}_3(t) = \mathbf{X}_2(t) - \mathbf{X}_3(t)$. The trajectory $\mathbf{X}_2(t)$ approaches $\mathbf{X}_3(t)$ asymptotically (synchronizes) if $\|\mathbf{X}_2\| \to 0$ as $t \to \infty$. Equivalently, the response subsystem (Equation 14.29) is asymptotically stable when driven with $\mathbf{X}_1(t)$.

The stability of an orbit of a dynamical system may be determined by examining the linearization of the vector field along the orbit. The linearized response subsystem is governed by

$$\dot{\mathbf{x}}_3 = \mathbf{D}_{\mathbf{X}_3}\mathbf{F}_2(\mathbf{X}_1(t), \mathbf{X}_3)\mathbf{x}_3, \quad \mathbf{x}_3(0) = \mathbf{x}_{3_0}$$

where $\mathbf{D}_{\mathbf{x}_3}\mathbf{F}_2(\mathbf{X}_1(t), \mathbf{X}_3)$ denotes the partial derivatives of the vector field $\mathbf{F}_2$ of the response subsystem with respect to $\mathbf{X}_3$. This is a linear time-varying system whose state transition matrix $\Phi_t(\mathbf{X}_{1_0}, \mathbf{X}_{3_0})$ maps a point $\mathbf{x}_{3_0}$ into $\mathbf{X}_3(t)$. Thus,

$$\mathbf{x}_3(t) = \Phi_t(\mathbf{X}_{1_0}, \mathbf{X}_{3_0})\mathbf{x}_{3_0}$$

Note that Φ_t is a linear operator. Therefore, an $(n - m)$-dimensional ball $B_\varepsilon(\mathbf{X}_{3_0})$ of radius ε about $\mathbf{X}_{3_0}$ is mapped into an ellipsoid whose principal axes are determined by the singular values of Φ_t. In particular, a ball of radius ε is mapped by Φ_t into an ellipsoid, the maximum and minimum radii of which are bounded by the largest and smallest singular values, respectively, of Φ_t.

The conditional LEs $\lambda_i(\mathbf{X}_{1_0}, \mathbf{X}_{2_0})$ (hereafter denoted CLE) are defined by

$$\lambda_i(\mathbf{X}_{1_0}, \mathbf{X}_{2_0}) = \lim_{t\to\infty} \frac{1}{t} \ln \sigma_i[\phi_t(\mathbf{X}_{1_0}, \mathbf{X}_{2_0})], \quad i = 1, 2, \ldots, (n - m)$$

whenever the limit exists.

The term conditional refers to the fact that the exponents depend explicitly on the trajectory $\phi_t(\mathbf{X}_{1_0}, \mathbf{X}_{2_0})$ of the drive system.

Given that ε remains infinitesimally small, one needs only to consider the local linearized dynamics along the flow determined by $\phi_t(\mathbf{X}_{1_0}, \mathbf{X}_{2_0})$ and to determine the average local exponential rates of expansion and contraction along the principal axes of an ellipsoid. If all CLEs are negative, the response subsystem is asymptotically stable. A subsystem, where all the CLEs are negative, is called a stable subsystem.

A stable subsystem does not necessarily exhibit dc steady-state behavior. For example, while an asymptotically stable linear parallel *RLC* circuit has all negative LEs, the system settles to a periodic steady-state solution when driven with a sinusoidal current. Although the *RLC* subcircuit has negative CLEs in this case, the complete forced circuit has one nonnegative LE corresponding to motion along the direction of the flow.

THEOREM 14.4 (Pecora and Carroll)

The trajectories $\mathbf{X}_2(t)$ *and* $\mathbf{X}_3(t)$ *will synchronize only if the CLEs of the response system* (*Equation* 14.28) *are all negative.*

Note that this is a necessary but not sufficient condition for synchronization. If the response and second drive subsystems are identical and the initial conditions $\mathbf{X}_{2_0}$ and $\mathbf{X}_{3_0}$ are sufficiently close, and the CLEs of Equation 14.28 are all negative, synchronization will occur. However, if the systems are not identical or the initial conditions are not sufficiently close, synchronization might not occur, even if all of the CLEs are negative.

Although we have described it only for an autonomous continuous-time system, the drive-response technique may also be applied for synchronizing nonautonomous and discrete-time circuits.

14.5.2.1 Cascaded Drive-Response Systems

The drive-response concept may be extended to the case where a dynamical system can be partitioned into more than two parts. A simple two-level drive-response cascade is constructed as follows. Divide the dynamical system

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}), \quad \mathbf{X}(0) = \mathbf{X}_0 \tag{14.29}$$

into three parts:

$$\dot{\mathbf{X}}_1 = \mathbf{F}_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \quad \mathbf{X}_1(0) = \mathbf{X}_{1_0} \tag{14.30}$$

$$\dot{\mathbf{X}}_2 = \mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \quad \mathbf{X}_2(0) = \mathbf{X}_{2_0} \tag{14.31}$$

$$\dot{\mathbf{X}}_3 = \mathbf{F}_3(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \quad \mathbf{X}_3(0) = \mathbf{X}_{3_0} \tag{14.32}$$

Now, construct an identical copy of the subsystems corresponding to Equations 14.31 and 14.32 with $\mathbf{X}_1(t)$ as input:

$$\dot{\mathbf{X}}_4 = \mathbf{F}_2(\mathbf{X}_1, \mathbf{X}_4, \mathbf{X}_5), \quad \mathbf{X}_4(0) = \mathbf{X}_{4_0} \tag{14.33}$$

$$\dot{\mathbf{X}}_5 = \mathbf{F}_3(\mathbf{X}_1, \mathbf{X}_4, \mathbf{X}_5), \quad \mathbf{X}_5(0) = \mathbf{X}_{5_0} \tag{14.34}$$

If all the CLEs of the driven subsystem composed of Equations 14.33 and 14.34 are negative then, after the transient decays, $\mathbf{X}_4(t) = \mathbf{X}_2(t)$ and $\mathbf{X}_5(t) = \mathbf{X}_3(t)$.

Note that Equations 14.30 through 14.34 together define one large coupled dynamical system. Hence, the response subsystem can exhibit chaos even if all of its CLEs are negative.

Proceeding one step further, we reproduce subsystem (Equation 14.30):

$$\dot{\mathbf{X}}_6 = \mathbf{F}_1(\mathbf{X}_6, \mathbf{X}_4, \mathbf{X}_5), \quad \mathbf{X}_6(0) = \mathbf{X}_{6_0} \tag{14.35}$$

As before, if all of the conditional LEs of Equation 14.35 are negative, then $\|\mathbf{X}_6(t) - \mathbf{X}_1(t) \to 0\|$.

If the original system could be partitioned so that Equation 14.30 is one-dimensional, then using Equations 14.33 through 14.35 as a driven response system, *all* of the variables in the drive system could be reproduced by driving with just *one* variable $X_1(t)$. It has been suggested that chaotic synchronization might be useful in chaotic communications.

Example: Synchronization of Chua's Circuits Using the Drive-Response Concept

Chua's circuit may be partitioned in three distinct ways to form five-dimensional, drive-decomposable systems:

X_1-drive configuration:

$$\frac{dX_1}{dt} = \alpha[X_2 - X_1 - f(X_1)]$$
$$\frac{dX_2}{dt} = X_1 - X_2 - X_3$$
$$\frac{dX_3}{dt} = -\beta X_2$$
$$\frac{dX_4}{dt} = X_1 - X_4 - X_5$$
$$\frac{dX_5}{dt} = -\beta X_5$$

With $\alpha = 10.0$, $\beta = 14.87$, $a = -1.27$, and $b = -0.68$, the CLEs for the (X_2, X_3) subsystem are $(-0.5, -0.5)$.

X_2-drive configuration:

$$\frac{dX_2}{dt} = X_1 - X_2 - X_3$$
$$\frac{dX_1}{dt} = \alpha[X_2 - X_1 - f(X_1)]$$
$$\frac{dX_2}{dt} = -\beta X_2$$
$$\frac{dX_4}{dt} = \alpha[X_2 - X_4 - f(X_4)]$$
$$\frac{dX_5}{dt} = -\beta X_5$$

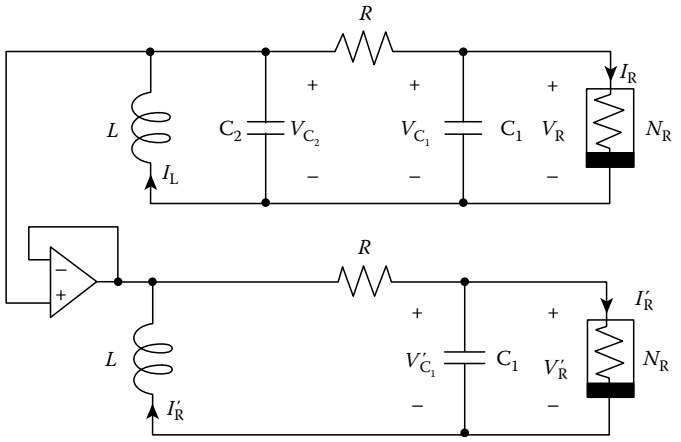


FIGURE 14.48 Synchronization of two Chua's circuits using the Pecora–Carroll drive-response method with V_{C_2} as the drive variable.

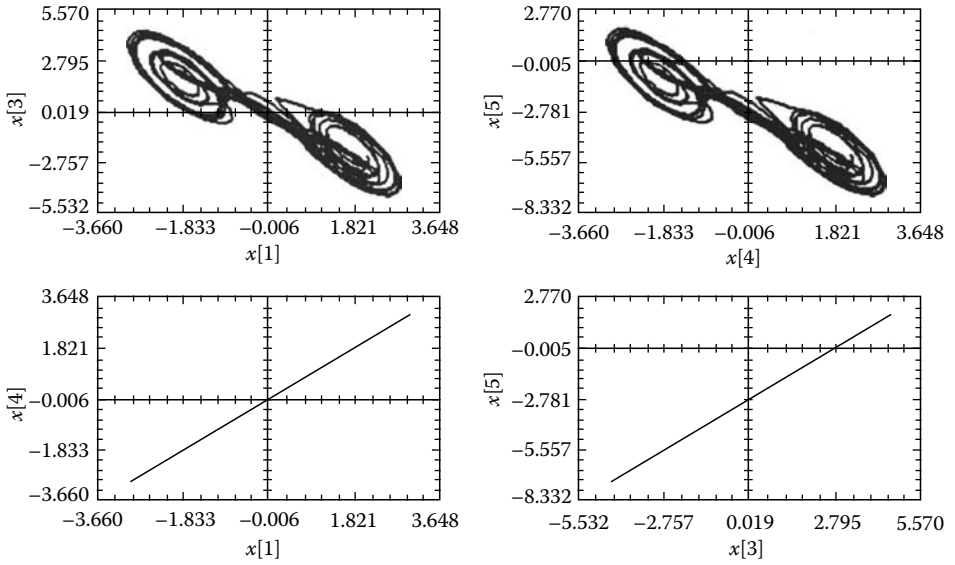


FIGURE 14.49 Simulation of the normalized dimensionless form of Figure 14.48, illustrating synchronization of state variables. Identify $\{x[1], x[2], x[3]\}$ with $\{V_{C_1}\ V_{C_2}\ I_L\}$, and $\{x[4], x[5]\}$ with $\left\{V'_{C_1}\ I'_L\right\} \cdot V'_{C_1}(t)$ synchronizes with $V'_{C_1}(t)$, and $I'_L(t)$ synchronizes with $I_L(t)$. Because one of the CLEs of the response subsystem is zero, the difference in the initial conditions $I_{L_0} - I_{L_0}$ does not decay to zero. (From Chua, L.O., Itoh, M., Kočarev, L., and Eckert, K., *J. Circuits Syst. Comput.*, 3, 1, 106, Mar. 1993. With permission.)

This case is illustrated in Figure 14.48. The CLEs of the (X_1, X_3) subsystem are 0 and -2.5 ± 0.05. Because of the zero CLE, states $X_5(t)$ and $X_3(t)$ remain a constant distance $|X_{3_0} - X_{5_0}|$ apart, as depicted in Figure 14.49.

X_3-drive configuration:

$$\frac{dX_3}{dt} = -\beta X_2$$
$$\frac{dX_1}{dt} = \alpha[X_2 - X_1 - f(X_1)]$$
$$\frac{dX_2}{dt} = X_1 - X_2 - X_3$$
$$\frac{dX_4}{dt} = \alpha[X_5 - X_4 - f(X_4)]$$
$$\frac{dX_5}{dt} = X_4 - X_5 - X_3$$

The CLEs in this case are 1.23 ± 0.03 and -5.42 ± 0.02. Because the (X_1, X_2) subsystem has a positive CLE, the response subsystem does not synchronize.

14.6 Applications of Chaos

14.6.1 Pseudorandom Sequence Generation

One of the most widely used deterministic "random" number generators is the linear congruential generator, which is a discrete-time dynamical system of the form

$$X_{k+1} = (AX_k + B) \bmod M, \quad k = 0, 1, \ldots \tag{14.36}$$

where A, B, and M are called the multiplier, increment, and modulus, respectively.

If $A > 1$, then all equilibrium points of Equation 14.36 are unstable. With the appropriate choice of constants, this system exhibits a chaotic solution with a positive LE equal to ln A. However, if the state space is discrete, for example in the case of digital implementations of Equation 14.37, then every steady-state orbit is periodic with a maximum period equal to the number of distinct states in the state space; such orbits are termed pseudorandom.

By using an analog state space, a truly "random" chaotic sequence can be generated. A discrete-time chaotic circuit with an analog state space may be realized in SC technology. Figure 14.50 is an SC realization of the parabolic map

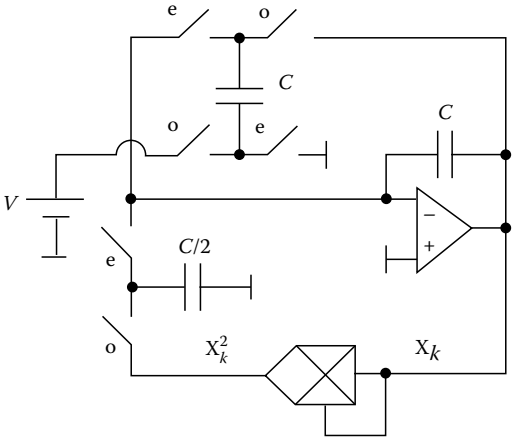


FIGURE 14.50 SC realization of the parabolic map $x_{k+1} = V - 0.5X_k^2$. The switches labeled o and e are driven by the odd and even phases, respectively, of a nonoverlapping two-phase clock.

$$x_{k+1} = V - 0.5x_k^2 \tag{14.37}$$

which, by the change of variables

$$X_k = Ax_k + B$$

with $\mu = 1/(2A)$, $B = 0.5$, and $A = (-1 \pm \sqrt{1 + 2V})/(4V)$, is equivalent to the logistic map

$$X_{k+1} = \mu X_k(1 - X_k) \tag{14.38}$$

The logistic map is chaotic for $\mu = 4$ with LE ln 2 [18]. Figure 14.16 is a bifurcation diagram of Equation 14.38 with $0 \leq \mu \leq 4$.

For $V < 1.5$, the steady-state solution of the SC parabolic map described by Equation 14.37 is a fixed point. As the bifurcation parameter V is increased from 1.5 to 3 V, the circuit undergoes a series of period-doubling bifurcations to chaos. $V = 4$ corresponds to fully developed chaos on the open interval $(0 < X_k < 1)$ in the logistic map with $\mu = 4$.

14.6.2 Communicating with Chaos

Modulation and coding techniques for mobile communication systems are driven by two fundamental requirements: that the communication channel should be secure and the modulation scheme should be tolerant of multipath effects. Security is ensured by coding and immunity from multipath degradation may be achieved by using a spread-spectrum transmission.

With appropriate modulation and demodulation techniques, the ''random'' nature and ''noise-like'' spectral properties of chaotic circuits can be exploited to provide simultaneous coding and spreading of a transmission.

14.6.2.1 Chaotic Masking

Chaotic masking is a method of hiding an information signal by adding it to a chaotic carrier at the transmitter. The drive-response synchronization technique is used to recover the carrier at the receiver.

Figure 14.51 is a block diagram of a communication system using matched Chua's circuits. The receiver has the same two-layer structure as in the previous example. The first subcircuit, which has very negative CLEs, synchronizes with the incoming signal, despite the perturbation $s(t)$, and recovers V_{C_2}. The second subcircuit, when driven by V_{C_2}, produces the receiver's copy of V_{C_1}. The information signal $r(t)$ is recovered by subtracting the local copy of V_{C_1} from the incoming signal $V_{C_1} + s(t)$.

Chaotic masking is not suitable for practical communications applications for two reasons:

- The signal is indistinguishable from additive noise in the channel; this results in a poor signal to noise ratio in the received signal and therefore poor overall noise performance
- The received signal is corrupted not only by noise in the channel but by the message itself; this makes synchronization of the receiver, and therefore coherent demodulation using synchronized chaos, difficult

Another possibility is to use chaotic signals as the basis functions in a digital communications system.

14.6.2.2 Chaos Shift Keying

The simplest idea for data transmission (digital communications) using a chaotic circuit is use the data to modulate some parameter(s) of the transmitter. This technique is called parameter modulation, chaotic switching, or chaos shift keying (CSK).

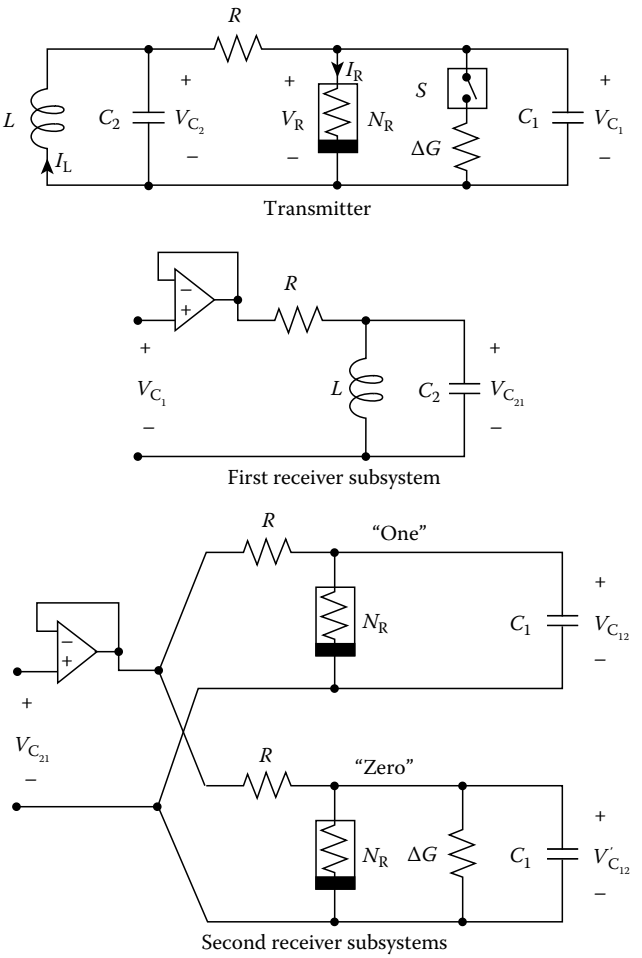


FIGURE 14.51 CSK communication system using Chua's circuit. When a ''one'' is transmitted, switch S remains open, $V_{C_{21}}(t)$ synchronizes with $V_{C_2}(t)$, $V_{C_{12}}(t)$ synchronizes with $V_{C_1}(t)$, and $V'_{C_{12}}(t)$ falls out of synchronization with $V_{C_1}(t)$. When a ''zero'' is transmitted, switch S is closed, $V_{C_{21}}(t)$ synchronizes with $V_{C_2}(t)$, $V_{C_{12}}(t)$ falls out of synchronization with $V_{C_1}(t)$, and $V'_{C_{12}}(t)$ synchronizes with $V_{C_1}(t)$.

In the case of binary data, the information signal is encoded as a pair of circuit parameter sets which produce distinct attractors in a dynamical system (the transmitter). In particular, a single control parameter μ may be switched between two values μ_0, corresponding to attractor $\mathcal{A}_0$ and μ_1, corresponding to $\mathcal{A}_1$. By analogy with frequency shift keying (FSK) and phase shift keying (PSK), this technique is known as CSK.

The binary sequence to be transmitted is mapped into the appropriate control signal $\mu(t)$ and the corresponding trajectory switches, as required, between $\mathcal{A}_1$ and $\mathcal{A}_0$.

One of the state variables of the transmitter is conveyed to the receiver, where the remaining state variables are recovered by drive-response synchronization. These states are then applied to the second stage of a drive-response cascade.

At the second level, two matched receiver subsystems are constructed, one of which synchronizes with the incoming signal if a ''zero'' was transmitted, the other of which synchronizes only if a ''one'' was transmitted. The use of two receiver circuits with mutually exclusive synchronization properties improves the reliability of the communication system.

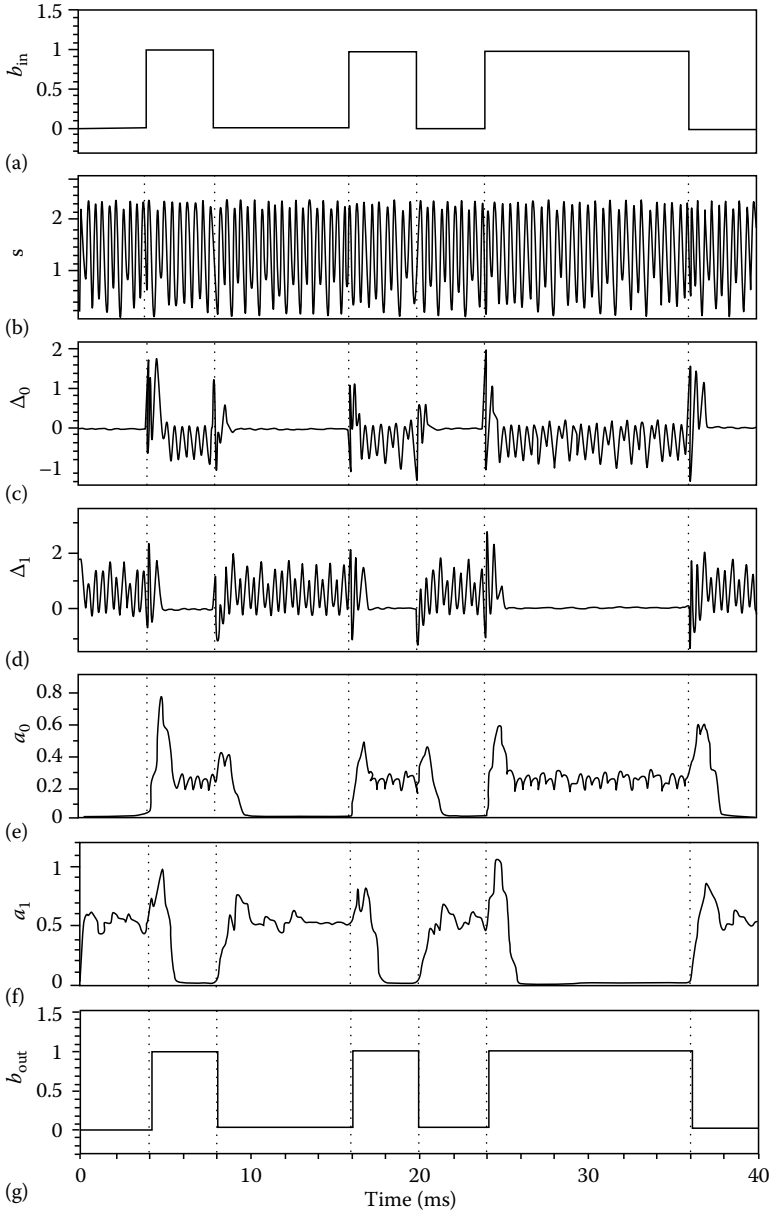


FIGURE 14.52 CSK waveforms. (a) Binary input signal b_{in}; (b) transmitted signal $s(t)$; (c) response $\Delta_0 = V_{C_{12}} - V_{C_1}$; (d) response $\Delta_0 = V'_{C_{12}} - V_{C_1}$; (e) 40-point moving average of Δ_0; (f) 40-point moving average of Δ_1; (g) output binary signal b_{out} when $\varepsilon = 0.1$. (From Ogorzalek, M., *IEEE Trans. Circuits Syst. I*, 40, 696, Oct. 1993. With permission.)

CSK has been demonstrated both theoretically and experimentally. Figure 14.52 depicts a CSK transmitter and receiver based on Chua's circuit. The control parameter is a resistor with conductance ΔG whose effect is to modulate the slopes G_a and G_b of the Chua diode. Switch S is opened and closed by the binary data sequence and V_{C_1} is transmitted. At the receiver, the first subsystem (a copy of the (V_{C_2}, I_L) subsystem of the transmitter) synchronizes with the incoming signal, recovering $V_{C_2}(t)$. Thus, $V_{C_{21}}(t) \rightarrow V_{C_2}(t)$.

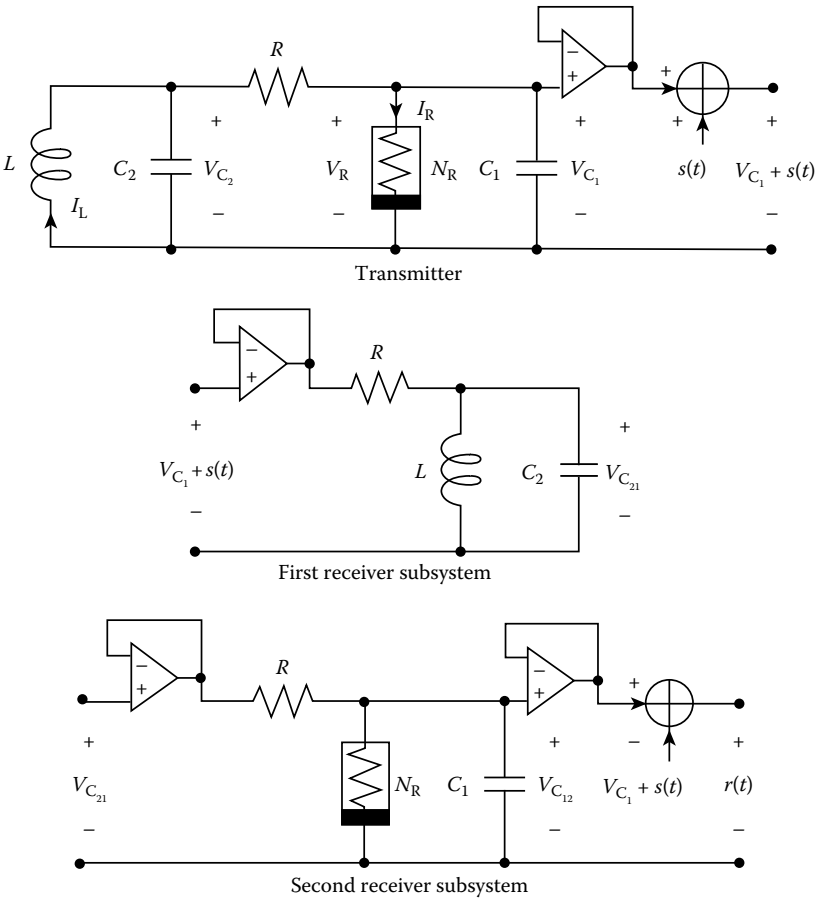


FIGURE 14.53 Chaos masking using Chua's circuits. At the transmitter, the information signal $s(t)$ is added to the chaotic carrier signal $V_{C_1}(t)$. Provided $s(t)$ is sufficiently small and the first receiver subsystem is sufficiently stable, $V_{C_{12}}(t)$ synchronizes with $V_{C_2}(t)$. This signal is applied to a second receiver subsystem, from which an estimate $V_{C_{12}}(t)$ of the unmodulated carrier $V_{C_1}(t)$ is derived. $V_{C_{12}}$ is subtracted from the incoming signal $V_{C_1} + s(t)$ to yield the received signal $r(t)$. The method works well only if $s(t)$ is much smaller than $V_{C_1}(t)$.

The synchronized local copy of $V_{C_1}(t)$ is then used to synchronize two further subsystems corresponding to the V_{C_1} subsystem of the transmitter with and without the resistor ΔG.

If the switch is closed at the transmitter, $V'_{C_{12}}$ (but not $V_{C_{12}}$) synchronizes with V_{C_1} and if the switch is open, $V_{C_{12}}$ (but not $V'_{C_{12}}$) synchronizes with V_{C_1}.

Figure 14.53 presents simulated results for a similar system consisting of two Chua's circuits. At the receiver, a decision must be made as to which bit has been transmitted. In this case, b_{out} was derived using the rule

$$b_{\text{out}} = \begin{pmatrix} 0, b_{\text{old}} = 0 & \text{for} & a_0 < \varepsilon,\ a_1 > \varepsilon \\ 1, b_{\text{old}} = 1 & \text{for} & a_0 > \varepsilon,\ a_1 < \varepsilon \\ b_{\text{old}} & \text{for} & a_0 < \varepsilon,\ a_1 < \varepsilon \\ 1 - b_{\text{old}} & \text{for} & a_0 > \varepsilon,\ a_1 > \varepsilon \end{pmatrix}$$

where b_{old} is the last bit received and b_{out} is the current bit [14].

As with chaos masking, noise in the channel perturbs the transmitted signal, making coherent demodulation using synchronized chaos difficult; this degrades the overall system performance.

An alternative to regenerating the chaotic basis functions at the receiver by means of synchronization is to transmit a chaotic reference signal as part of the message and to recover the information by correlating the received signal with the transmitted reference. An example of this is differential chaos shift keying (DCSK).

14.6.2.3 Differential Chaos Shift Keying

In chaotic modulation schemes, the transmitted signal is a wideband signal. The objectives of using a wideband signal as the carrier are twofold:

- To overcome the multipath propagation problem
- To reduce the transmitted power spectral density in order to reduce the interference caused in adjacent radio communications channels

Of the chaotic modulation schemes published to date, DCSK with orthonormal basis functions, also referred in the literature as to FM-DCSK [24] offers the best robustness against multipath and channel imperfections. In an FM-DCSK signal, symbol "0" is represented by a piece of FM-modulated chaotic waveform for time $T/2$, followed by a copy of itself; symbol "1" comprises a piece of FM-modulated chaotic waveform for time $T/2$, followed by an inverted copy of itself, where T is the symbol duration. Correlating the two halves of the waveform at the receiver yields $+1$ in the case of "0" and -1 in the case of "1."

It is frequently asked whether chaotic modulation schemes can offer any advantages over conventional narrow band systems. If the propagation conditions are so good that the basis function(s) can be regenerated at the receiver, then digital modulation schemes using conventional orthonormal (typically periodic) basis functions and orthonormal chaotic basis functions can achieve similar levels of noise performance.

The main question from an implementation perspective is the ease with which the basis functions can be regenerated. It is fundamentally easier to regenerate a periodic basis function than a chaotic one. Therefore, the noise performance of digital chaotic modulation with coherent correlation receivers will always lag behind that of equivalent modulation schemes using periodic basis functions.

If the propagation conditions are such that coherent detection is impossible, then chaotic switching with orthonormal DCSK basis functions and a differentially coherent receiver (DCSK, for short), offers the best possible performance for a chaotic digital modulation scheme. In the limit, the noise performance of DCSK lags only 3 dB behind that of DPSK with autocorrelation demodulation [24].

In this case, the choice of periodic or chaotic basis functions is determined by the propagation conditions. In particular, the multipath performance of a DCSK system can be improved by increasing the transmission bandwidth.

Figure 14.54 shows the simulated best case noise performance for CSK, chaotic on–off keying (COOK), and DCSK. The noise performance of binary phase shift keying (BPSK) and frequency shift keying (FSK) are also shown, for comparison.

Another important issue is interference. If the operating environment is such that interference with other narrow band radio systems working in the same frequency band should be minimized, then a spread spectrum system must be used; this can be implemented using chaotic basis functions, where the chaotic waveform is the spreading signal. For a fixed transmitted power, the wider the bandwidth of the chaotic signal, the lower the interference caused.

While signal spreading in a chaotic communications scheme reduces the interfering signal level for narrow band systems sharing the same frequency band, the converse is not necessarily true. Interference to a chaotic communications system caused by other users in the same band can be suppressed only by using coherent demodulation techniques.

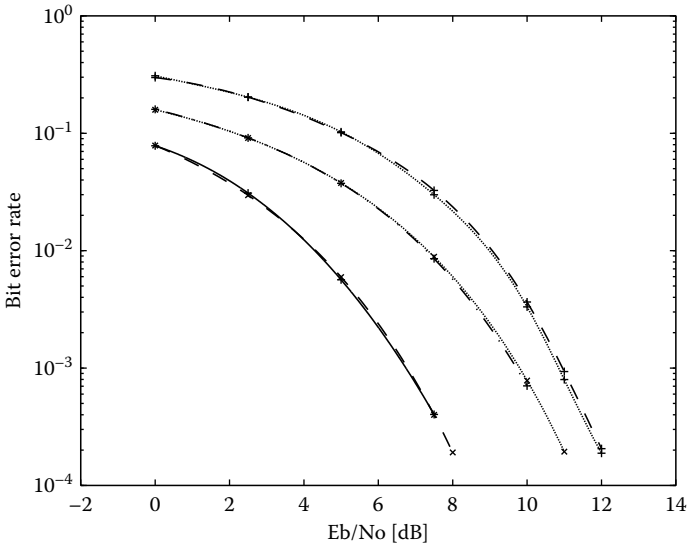


FIGURE 14.54 Simulated optimum noise performance of antipodal CSK modulation with coherent demodulation (solid curve with "+" marks [left]), COOK with noncoherent demodulation (dashed curve with "+" marks [right]), chaotic switching with orthonormal basis functions and coherent demodulation (dash-dot curve with "+" marks [center]), and chaotic switching with DCSK basis functions and a differentially coherent receiver (dotted curve with "o" marks [right]). The noise performance curves for BPSK (dashed curve with "o" marks [left]) and coherent FSK (dotted curve with "×" marks [center]) are also shown, for comparison. (From Kolumban, G., Kennedy, M.P., Jako, Z., and Kis, G., *Proc. IEEE*, 90(5), 711, May 2002. With permission.)

Although it performs worse than conventional narrowband modulation schemes in a simple additive white Gaussian noise (AWGN) channel, the advantage of DCSK is that the fall off in its performance in a multipath channel is more gradual than that of an equivalent narrow band modulation scheme.

This result is highlighted dramatically in Figure 14.55, where the performance degradation in a narrow band DPSK system; (classical DPSK with the optimum receiver configuration, referred as to "optimum DPSK") is compared with that of a wideband differentially coherent FM-DCSK system. The bit duration T was set to 2 μs in both cases and the RF bandwidth of the DCSK signal was 17 MHz. The solid and dashed curves give the noise performance of optimum DPSK and FM-DCSK, respectively, over a single-ray AWGN channel.

The noise performance of these modulation schemes was evaluated over a two-ray multipath channel with an excess delay of 100 ns that is typical of WLAN applications in large warehouses.

Although the single-ray performance of differentially coherent FM-DCSK is worse than that of DPSK, its multipath performance is significantly better, despite the fact that the carrier recovery problem which would accentuate the problem further does not appear in optimum DPSK.

In summary, FM-DCSK offers a performance advantage over conventional communications schemes in multipath environments when the propagation conditions are so poor that coherent detection or detection of DPSK with an optimum receiver configuration is not possible [24].

14.6.2.4 Miscellaneous

Chaotic circuits may also be used for suppressing spurious tones in ΣΔ modulators, for modeling musical instruments, fractal pattern generation, image processing, and pattern recognition [3].

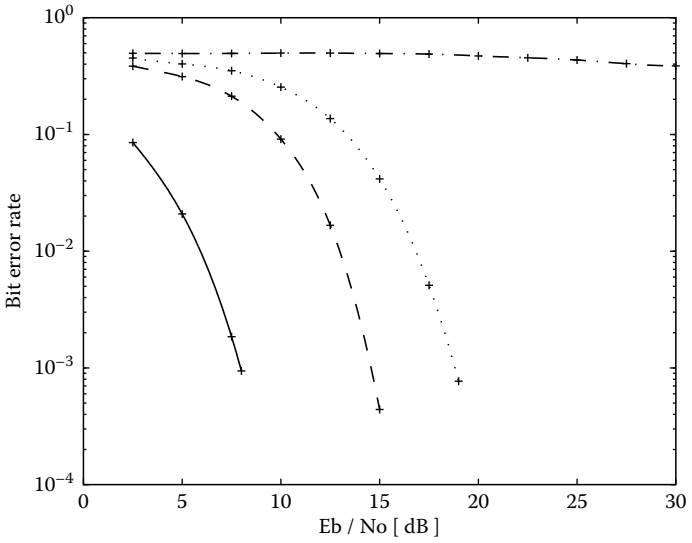


FIGURE 14.55 Simulated noise performance curves for optimum DPSK and wideband FM-DCSK in a single-ray channel (solid and dashed, respectively) and in a two-ray multipath channel (dash-dot and dotted, respectively). While FM-DCSK disimproves by about 4 dB, DPSK fails completely. (From Kolumban, G., Kennedy, M.P., Jako, Z., and Kis, G., *Proc. IEEE*, 90, 711, 2002. With permission.)

A chaotic attractor contains an infinite number of unstable periodic trajectories of different periods. Various control schemes for stabilizing particular orbits in chaotic circuits have been successfully demonstrated [14].

There has also been significant interest in using chaos for cryptographic purposes. An analog chaotic system with a continous state space has potentially greater complexity (in terms of crpytographic keys) than a digital finite state machine. However, many practical difficulties aris in trying to implement identical analog systems. An overview of the subject can be found in Ref. [25].

References

1. F. M. Callier and C. A. Desoer, *Linear System Theory*, New York: Springer-Verlag, 1991.
2. L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*, New York: McGraw-Hill, 1987.
3. L. O. Chua and M. Hasler, Eds. (Special issue on *Chaos in Nonlinear Electronic Circuits*) Part A: Tutorials and Reviews, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, Oct. 1993; Part B: Bifurcation and Chaos, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, Nov. 1993; Part C: Applications, *IEEE Trans. Circuits Syst. II Analog Digital Signal Process.*, 40, Oct. 1993.
4. L. O. Chua, M. Itoh, L.Kočarev, and K. Eckert, Chaos synchronization in Chua's circuit, *J. Circuits Syst. Comput.*, 3, 1, 93–108, Mar. 1993.
5. L. O. Chua, C. W. Wu, A. Huang, and G. -Q. Zhong, A universal circuit for studying and generating chaos—Part I: Routes to chaos, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 732–744, Oct. 1933.
6. J. A. Glazier and A. Libchaber, Quasi-periodically and dynamical systems: An experimentalist's view, *IEEE Trans. Circuits Syst.*, 35, 790–809, July 1988.
7. J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, New York: Springer-Verlag, 1983.
8. M. W. Hirsch, The dynamical systems approach to differential equations, *Bull. Am. Math. Soc.*, 11(1), 1–64, July 1984.

9. M. Itoh, H. Murakami, K. S. Halle, and L. O. Chua, Transmission of signals by chaos synchronization, IEICE Technical Report, CAS93–39, NLP93–27, pp. 89–96, 1993.
10. M. P. Kennedy, Three steps to chaos—Part I: Evolution, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 640–656, Oct. 1993; Three steps to chaos—Part II: A Chua's circuit primer, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 657–674, Oct. 1993.
11. M. P. Kennedy and L. O. Chua, Van der Pol and chaos, *IEEE Trans. Circuits Syst.*, CAS-33, 974–980, Oct. 1986.
12. M. P. Kennedy, K. R. Krieg, and L. O. Chua, The devil's staircase: The electrical engineer's fractal, *IEEE Trans. Circuits Syst.*, 36, 1133–1139, 1989.
13. W. Lauterborn and U. Parlitz, Methods of chaos physics and their application to acoustics, *J. Acoust. Soc. Am.*, 84(6), 1975–1993, Dec. 1988.
14. M. Ogarzalek, Taming chaos—Part I: Synchronization, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 693–699, Oct. 1993; Taming chaos—Part II: Control, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.,* 40, 700–706, Oct. 1993.
15. T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*, New York: Springer-Verlag, 1989.
16. L. M. Pecora and T. Carroll, Driving systems with chaotic signals, *Phys. Rev.,* 44, 4, 2374–2383, Aug. 15, 1991.
17. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge: Cambridge University, 1988.
18. J. M. T. Thompson and H. B. Stewart, *Nonlinear Dynamics and Chaos*, New York: Wiley, 1986.
19. M. Vidyasagar, *Nonlinear Systems Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1978.
20. C. W. Wu and N. F. Rul'kov, Studying chaos via 1-D maps—A tutorial, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, 40, 707–721, Oct. 1993.
21. L. O. Chua, M. Komuro, and T. Matsumoto, The double scroll family, Parts I and II, *IEEE Trans. Circuits Syst.*, 33, 1073–1118, Nov. 1986.
22. Matlab: http://www.mathworks.com.
23. ABC++: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do? objectId = 3541& objectType = file.
24. G. Kolumban, M. P. Kennedy, Z. Jako, and G. Kis, Chaotic communications with correlation receivers: theory and performance limits, *Proc. IEEE*, 90(5), 711–732, May 2002.
25. F. Dachselt and W. Schwarz, Chaos and cryptography, *IEEE Trans. Circuits Syst.*, 48(12), 1498–1509, Dec. 2001.

Further Information

Current Research in Chaotic Circuits. The August 1987 issue of the *Proceedings of the IEEE* is devoted to ''Chaotic Systems.'' The *IEEE Transactions on Circuits and Systems*, July 1988, focuses on ''Chaos and Bifurcations of Circuits and Systems.''

A three-part special issue of the *IEEE Transactions on Circuits and Systems* on ''Chaos in Electronic Circuits'' appeared in October (Parts I and II) and November 1993 (Part I). This special issue contains 42 papers on various aspects of bifurcations and chaos.

Two special issues (March and June 1993) of the *Journal of Circuits, Systems and Computers* are devoted to ''Chua's Circuit: A Paradigm for Chaos.'' These works, along with several additional papers, a pictorial guide to 45 attractors in Chua's oscillator, and the Adventures in Bifurcations and Chaos (ABC) simulator, have been compiled into a book of the same name—*Chua's Circuit: A Paradigm for Chaos*, R.N. Madan, Ed. Singapore: World Scientific, Singapore, 1993.

Developments in the field of bifurcations and chaos, with particular emphasis on the applied sciences and engineering, are reported in *International Journal of Bifurcation and Chaos*, which is published quarterly by World Scientific, Singapore.

Research in chaos in electronic circuits appears regularly in the *IEEE Transactions on Circuits and Systems*.

Simulation of Chaotic Circuits. A variety of general-purpose and custom software tools has been developed for studying bifurcations and chaos in nonlinear circuits systems.

ABC++ is a graphical simulator of Chua's oscillator. ABC++ contains a database of component values for all known attractors in Chua's oscillator, initial conditions, and parameter sets corresponding to homoclinic and heteroclinic trajectories, and bifurcation sequences for the period-doubling, intermittency, and quasiperiodic routes to chaos.

In "Learning about Chaotic Circuits with SPICE," *IEEE Transactions on Education*, Vol. 36, pp. 28–35, Jan. 1993, David Hamill describes how to simulate a variety of smooth chaotic circuits using the general-purpose circuit simulator SPICE. A commercial variant of SPICE, called PSpice, is available from http://www.cadence.com/orcad/index.html.

The free student evaluation version of this program is sufficiently powerful for studying simple chaotic circuits. PSpice runs on both workstations and PCs.

III

Distributed Circuits

Thomas Koryu Ishii
Marquette University

15 **Transmission Lines** *Thomas Koryu Ishii* **15**-1
Generic Relations • Two-Wire Lines • Coaxial Lines • Waveguides • Microstrip Lines • Coplanar Waveguide • References

16 **Multiconductor Transmission Lines** *Daniël De Zutter and Luc Martens* **16**-1
Introduction: Frequency versus Time Domain Analysis • Telegrapher's Equations for Uniform Multiconductor Transmission Lines • References

17 **Time and Frequency Domain Responses** *Luc Martens and Daniël De Zutter* **17**-1
Time-Domain Reflectometry • Frequency-Domain Network Analysis • References

18 **Distributed RC Networks** *Vladimír Székely* **18**-1
Uniform Distributed RC Lines • Nonuniform Distributed RC Lines • Infinite-Length RC Lines • Inverse Problem for Distributed RC Circuits • References

19 **Synthesis of Distributed Circuits** *Thomas Koryu Ishii* **19**-1
Generic Relations • Synthesis of a Capacitance • Synthesis of an Inductance • Synthesis of a Resistance • Synthesis of Transformers • Synthesis Examples • Synthesis of Couplers • References

15

Transmission Lines

15.1 Generic Relations **15**-1
Equivalent Circuit • Transmission Line Equations • General Solutions and Propagation Constant • Characteristic Impedance • Wavelength • Phase Velocity • Voltage Reflection Coefficient at the Load • Voltage Reflection Coefficient at the Input • Input Impedance • Input Admittance

15.2 Two-Wire Lines **15**-6
Geometric Structure • Transmission Line Parameters • Wavelength and Phase Velocity

15.3 Coaxial Lines **15**-8
Geometric Structure • Transmission Line Parameters • Wavelength and Phase Velocity

15.4 Waveguides **15**-10
Rectangular Waveguides • Waveguide Parameters • Circular Waveguides

15.5 Microstrip Lines **15**-15
Geometric Structure • Transmission Line Parameters • Wavelength and Phase Velocity

15.6 Coplanar Waveguide **15**-17
Geometric Structure • Transmission Line Parameters • Wavelength and Phase Velocity

References **15**-19

Thomas Koryu Ishii
Marquette University

15.1 Generic Relations

15.1.1 Equivalent Circuit

The equivalent circuit of a generic transmission line in monochromatic single frequency operation is shown in Figure 15.1 [1–3], where $\dot{Z}$ is a series impedance per unit length of the transmission line (Ω/m) and $\dot{Y}$ is a shunt admittance per unit length of the transmission line (S/m). For a uniform, nonlinear transmission line, either $\dot{Z}$ or $\dot{Y}$ or both $\dot{Z}$ and $\dot{Y}$ are functions of the transmission line voltage and current, but both $\dot{Z}$ and $\dot{Y}$ are not functions of location on the transmission line. For a nonuniform, linear transmission line, both $\dot{Z}$ and $\dot{Y}$ are functions of location, but not functions of voltage and current on the transmission line. For a nonuniform and nonlinear transmission line, both $\dot{Z}$ and $\dot{Y}$ or $\dot{Z}$ or $\dot{Y}$ are functions of the voltage, the current, and the location on the transmission line.

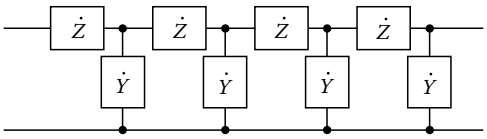


FIGURE 15.1 Equivalent circuit of a generic transmission line.

15.1.2 Transmission Line Equations

Ohm's law of a transmission line, which is the amount of voltage drop on a transmission line per unit distance of voltage transmission is expressed as

$$\frac{d\dot{V}}{dz} = -\dot{I}\dot{Z} \text{ (V/m)} \tag{15.1}$$

where

$\dot{V}$ is the transmission line voltage (V)
$\dot{I}$ is the transmission line current (A)
Z is the series impedance per unit length of the transmission line (Ω)
z is a one-dimensional coordinate placed in parallel to the transmission line (m)

The equation of current decrease per unit length is

$$\frac{d\dot{I}}{dz} = -\dot{Y}\dot{V} \text{ (A/m)} \tag{15.2}$$

where $\dot{Y}$ is the shunt admittance per unit distance of the transmission line (S/m).

Combining Equations 15.1 and 15.2, the Telegrapher's equation or Helmholtz's wave equation for the transmission line voltage is [1–3]

$$\frac{d^2\dot{V}}{dz^2} - \dot{Z}\dot{Y}\dot{V} = 0 \text{ (V/m)}^2 \tag{15.3}$$

The Telegrapher's equation or Helmholtz's wave equation for the transmission line current is [1–3]

$$\frac{d^2\dot{I}}{dz^2} - \dot{Z}\dot{Y}\dot{I} = 0 \text{ (A/m}^2\text{)} \tag{15.4}$$

15.1.3 General Solutions and Propagation Constant

The general solution of the Telegrapher's equation for transmission line voltage is [1–3]

$$\dot{V} = \dot{V}_F\varepsilon^{-\dot{\gamma}z} + \dot{V}_R\varepsilon^{\dot{\gamma}z} \tag{15.5}$$

where

$|\dot{V}_F|$ is the amplitude of the voltage waves propagating in $+z$-direction
$|\dot{V}_R|$ is the amplitude of the voltage waves propagating $-$z-direction
$\dot{\gamma}$ is the propagation constant of the transmission line [1]

$$\dot{\gamma} = \pm\sqrt{\dot{Z}\dot{Y}} = \alpha + \dot{\gamma}\beta \ (\mathrm{m}^{-1}) \tag{15.6}$$

where
+ sign is for forward propagation or propagation in $+z$-direction
− sign is for the backward propagation or propagation in $-z$-direction
α is the attenuation constant and it is the real part of propagation constant $\dot{\gamma}$

$$\alpha = \Re\dot{\gamma} = \Re\sqrt{\dot{Z}\dot{Y}} \ (\mathrm{m}^{-1}) \tag{15.7}$$

In Equation 15.6, β is the phase constant and it is the imaginary part of the propagation constant $\dot{\gamma}$:

$$\beta = \Im\dot{\gamma} = \Im\sqrt{\dot{Z}\dot{Y}} \ (\mathrm{m}^{-1}) \tag{15.8}$$

15.1.4 Characteristic Impedance

The characteristic impedance of the transmission line is [1–3]

$$Z_0 = \sqrt{\frac{\dot{Z}}{\dot{Y}}} \ (\Omega) \tag{15.9}$$

15.1.5 Wavelength

The wavelength of the transmission line voltage wave and transmission line current wave is

$$\lambda = \frac{2\pi}{\beta} = \frac{2\pi}{\Im\sqrt{\dot{Z}\dot{Y}}} \ (\mathrm{m}) \tag{15.10}$$

15.1.6 Phase Velocity

The phase velocity of voltage wave propagation and current wave propagation is

$$v_\mathrm{p} = f\lambda = \frac{\omega}{\beta} \ (\mathrm{m/s}) \tag{15.11}$$

where f is the frequency of operation, and the phase constant is

$$\beta = \frac{2\pi}{\lambda} \ (\mathrm{m}^{-1}) \tag{15.12}$$

15.1.7 Voltage Reflection Coefficient at the Load

If a transmission line of characteristic impedance Z_0 is terminated by a mismatched load impedance $\dot{Z}_\mathrm{L}$, as shown in Figure 15.2, a voltage wave reflection occurs at the load impedance $\dot{Z}_\mathrm{L}$. The voltage reflection coefficient is [1–3]

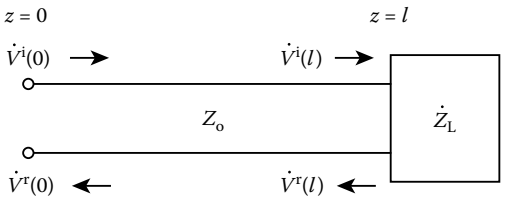


FIGURE 15.2 Incident wave $\dot{V}^{i}$ and reflected wave $\dot{V}^{r}$.

$$\dot{\rho}(l) = \frac{\dot{V}^{r}(l)}{\dot{V}^{i}(l)} = \frac{\dot{Z}_L - Z_0}{\dot{Z}_L + Z_0} = \frac{\tilde{\dot{Z}}_L - 1}{\tilde{\dot{Z}}_L + 1} \tag{15.13}$$

where

$\dot{\rho}(l)$ is the voltage reflection coefficient at $z = l$
$\tilde{\dot{Z}} = \tilde{\dot{Z}}_L / Z_0$ is the normalized load impedance
$\dot{V}^{i}(l)$ is the incident voltage at the load at $z = l$

When Z_0 is a complex quantity $\dot{Z}_0$, then

$$\dot{\rho}(l) = \frac{\dot{Z}_L - Z_0^{*}}{\dot{Z}_L + Z_0^{*}} \tag{15.14}$$

where Z_0^{*} is a conjugate of $\dot{Z}_0$.

15.1.8 Voltage Reflection Coefficient at the Input

Input Figure 15.2 $\dot{V}^{i}(l)$ is caused by $\dot{V}^{i}(0)$, which is the incident voltage at the input $z = 0$ of the transmission line. $\dot{V}^{r}(l)$ produces, $\dot{V}^{r}(0)$, which is the reflected voltage at the input $z = 0$. The voltage reflection coefficient at the input occurs, then, by omitting transmission line loss [1–3]

$$\dot{\rho} = \frac{\dot{V}^{r}(0)}{\dot{V}^{i}(0)} = \dot{\rho}(l)\varepsilon^{-2j\beta l} \tag{15.15}$$

15.1.9 Input Impedance

At the load $z = l$, from Equation 15.13,

$$\tilde{\dot{Z}}_L = \frac{1 + \dot{\rho}(l)}{1 - \dot{\rho}(l)} \tag{15.16}$$

or

$$\dot{Z}_L = Z_0 \frac{1 + \dot{\rho}(l)}{1 - \dot{\rho}(l)} \tag{15.17}$$

At the input of the transmission line $z = 0$

$$\tilde{\dot{Z}}(0) = \frac{1 + \dot{\rho}(0)}{1 - \dot{\rho}(0)} \tag{15.18}$$

Using Equation 15.15,

$$\tilde{\dot{Z}}(0) = \frac{1 + \dot{\rho}(l)\varepsilon^{-2j\beta l}}{1 - \dot{\rho}(l)\varepsilon^{-2j\beta l}} \tag{15.19}$$

Inserting Equation 15.13 [1–3]

$$\tilde{\dot{Z}}(0) = \frac{\tilde{\dot{Z}} + j\tan\beta l}{1 + j\tilde{\dot{Z}}_{\mathrm{L}}\tan\beta l} \tag{15.20}$$

or

$$\tilde{\dot{Z}}(0) = \frac{\dot{Z}_{\mathrm{L}} + jZ_0\tan\beta l}{Z_0 + j\dot{Z}_{\mathrm{L}}\tan\beta l} \tag{15.21}$$

If the line is lossy and

$$\dot{\gamma} = \alpha + j\beta \tag{15.22}$$

then [1]

$$\dot{Z}(0) = \dot{Z}_0^{\star}\frac{\dot{Z}_{\mathrm{L}} + \dot{Z}_0^{\star}\tanh\dot{\gamma}l}{\dot{Z}_0^{\star} + \dot{Z}_{\mathrm{L}}\tanh\dot{\gamma}l} \tag{15.23}$$

where $\dot{Z}_0^{\star}$ is the complex conjugate of $\dot{Z}_0$.

15.1.10 Input Admittance

The input admittance at $z = 0$ is

$$\dot{Y}(0) = \dot{Y}_0^{\star}\frac{\dot{Y}_{\mathrm{L}} + \dot{Y}_0^{\star}\tanh\dot{\gamma}l}{\dot{Y}_0^{\star} + \dot{Y}_{\mathrm{L}}\tanh\dot{\gamma}l} \tag{15.24}$$

where

$\dot{Y}(0)$ is the input admittance of the transmission line

$\dot{Y}_0$ is the characteristic admittance of the transmission line, which is

$$\dot{Y}_0 = \frac{1}{\dot{Z}_0} \tag{15.25}$$

and $\dot{Y}_0^*$ is the conjugate of $\dot{Y}_0 \cdot \dot{Y}_L$ is the load admittance; i.e.,

$$\dot{Y}_L = \frac{1}{\dot{Z}_L} \tag{15.26}$$

When the line is lossless,

$$\dot{\gamma} = j\beta \tag{15.27}$$

then [1–3]

$$\dot{Z}(0) = \frac{\dot{Z}_L + jZ_0 \tan \beta l}{Z_0 + jZ_L \tan \beta l} \tag{15.28}$$

15.2 Two-Wire Lines

15.2.1 Geometric Structure

A structural diagram of the cross-sectional view of a commercial two-wire line is shown in Figure 15.3. As observed in this figure, two parallel conductors, in most cases made of hard-drawn copper, are positioned by a plastic dielectric cover.

15.2.2 Transmission Line Parameters

In a two-wire line

$$\dot{Z} = R + jX = R + j\omega L \ (\Omega/\text{m}) \tag{15.29}$$

where
$\dot{Z}$ is the impedance per unit length of the two-wire line (Ω/m)
R is the resistance per unit length (Ω/m)
X is the reactance per unit length (Ω/m)
$\omega = 2\pi f$ is the operating angular frequency (s^{-1})
L is the inductance per unit length (H/m)

For a two-wire line made of hard-drawn copper [4]

$$R = 8.42 \frac{\sqrt{f}}{a} \ (\mu\Omega/\text{m}) \tag{15.30}$$

where f is the operating frequency and a is the radius of the conductor [4]:

$$L = 0.4 \ \ln \frac{b}{a} \ (\mu\text{H/m}) \tag{15.31}$$

where b is the wire separation or the center-to-center distance of the two-wire line as illustrated in Figure 15.3.

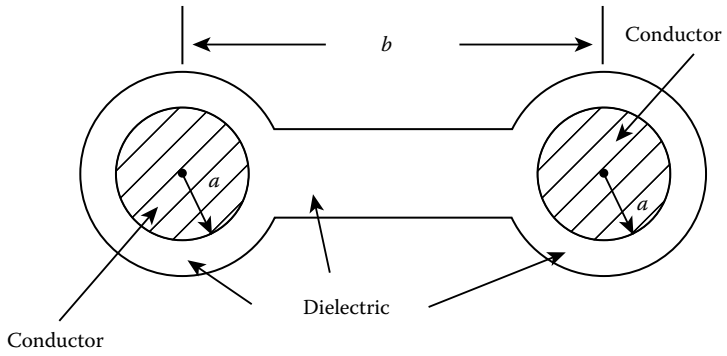


FIGURE 15.3 Cross-sectional view of a two-wire line.

$$\dot{Y} = G + jB = G + j\omega C \text{ (S/m)} \tag{15.32}$$

where

$\dot{Y}$ is a shunt admittance per unit length of the two-wire line (S/m)
G is a shunt conductance per unit length of the two-wire line (S/m)
B is a shunt susceptance per unit length (S/m)
C is a shunt capacitance per unit length (F/m)

$$G = \frac{3.14\sigma_d}{\cosh^{-1}\left(\frac{b}{2a}\right)} \text{ (pS/m)} \tag{15.33}$$

where σ_d is the insulation conductivity of the plastic dielectric surrounding the two parallel conductors, and

$$C = \frac{27.8\varepsilon_r}{\cosh^{-1}\left(\frac{b}{2a}\right)} \text{ (pF/m)} \tag{15.34}$$

where ε_r is the relative permittivity of the plastic insulating material [5].

If R and G are negligibly small, the characteristic impedance is [6]

$$Z_0 = 277 \log_{10} \frac{b}{a} \ (\Omega) \tag{15.35}$$

The attenuation constant of a generic two-wire line is, including both R and G [1],

$$\alpha = \frac{R}{2Z_0} + \frac{GZ_0}{2} \ (\text{m}^{-1}) \tag{15.36}$$

and the phase constants is [1]

$$\beta = \left\{ \frac{(\omega LB - RG) + \sqrt{(RG - \omega LB)^2 + (\omega LG + BR)^2}}{2} \right\}^{\frac{1}{2}} (\text{m}^{-1}) \tag{15.37}$$

15.2.3 Wavelength and Phase Velocity

The wavelength on a lossless two-wire line ($R = 0$, $G = 0$) is

$$\lambda_0 = \frac{\omega}{\beta_0} = \frac{1}{\sqrt{LC}} \text{ (m)} \tag{15.38}$$

where β_0 is the phase constant of the lossless two-wire line

$$\beta_0 = \omega\sqrt{LC} \text{ (m}^{-1}\text{)} \tag{15.39}$$

The wavelength on a lossy two-wire line ($R \neq 0$, $G \neq 0$) is

$$\lambda = \frac{2\pi}{\beta} = \left\{\frac{8\pi^2}{(\omega LB - RG) + \sqrt{(RG - \omega LB)^2 + (\omega LG + BR)^2}}\right\}^{\frac{1}{2}} \text{ (m)} \tag{15.40}$$

The phase velocity of transverse electromagnetic (TEM) waves on a lossless two-wire line is

$$v_0 = f\lambda_0 = \frac{\omega}{\beta_0} = \frac{1}{\sqrt{LC}} \text{ (m/s)} \tag{15.41}$$

The phase velocity of TEM waves on a lossy two-wire line is

$$v = f\lambda = \frac{\omega}{\beta} = \left\{\frac{2\omega^2}{(\omega LB - RG) + \sqrt{(RG - \omega LB)^2 + (\omega LG + BR)^2}}\right\}^{\frac{1}{2}} \text{ (m/s)} \tag{15.42}$$

15.3 Coaxial Lines

15.3.1 Geometric Structure

A generic configuration of a coaxial line is shown in Figure 15.4. The center and outer conductors are coaxially situated and separated by a coaxial insulator. Generally, coaxial lines are operated in the TEM mode, in which both the electric and magnetic fields are perpendicular to the direction of propagation. The propagating electric fields are in the radial direction and propagating magnetic fields are circumferential to the cylindrical surfaces.

In Figure 15.4, a is the radius of the center conductor, b is the inner radius of the outer conductor, and c is the outer radius of the outer conductor.

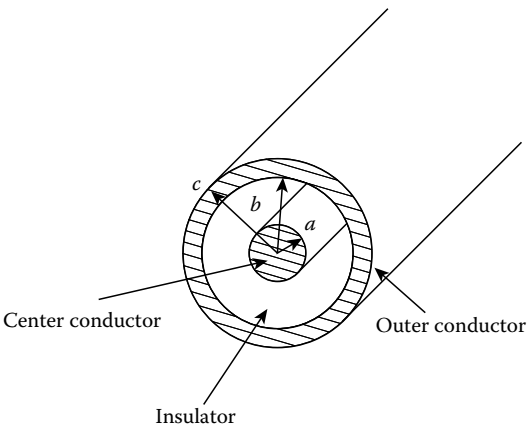


FIGURE 15.4 Generic configuration of a coaxial line.

15.3.2 Transmission Line Parameters

The series resistance per unit length of the line for copper is [7],

$$R = 4.16\sqrt{f}\left(\frac{1}{a} + \frac{1}{b}\right) \text{ } (\mu\Omega/\text{m}) \tag{15.43}$$

The series inductance per unit length is

$$L = 0.2 \ \ln\frac{b}{a} \ (\mu\text{H/m}) \tag{15.44}$$

The shunt conductance per unit length is

$$G = \frac{6.28\sigma_i}{\ln(b/a)} \ (\text{S/m}) \tag{15.45}$$

where σ_i is the conductivity of the *insulator* between the conductors. The shunt capacitance per unit length is

$$C = \frac{55.5\varepsilon_r}{\ln(b/a)} \ (\text{pF/m}) \tag{15.46}$$

where ε_r is the relative permittivity of the insulator. When the loss of the line is small, the characteristic impedance of the coaxial line is

$$Z_0 = \frac{138}{\sqrt{\varepsilon_r}} \ \log_{10}\frac{b}{a} \ (\Omega) \tag{15.47}$$

when the line is lossy [1]

$$\dot{Z}_0 = \sqrt{\frac{\dot{Z}}{\dot{Y}}} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} = R_0 + jX_0 \tag{15.48}$$

$$R_0 = \frac{\left\{RG + \omega^2 LC + \sqrt{(RG + \omega^2 LC)^2 + (\omega LG - \omega RC)^2}\right\}^{\frac{1}{2}}}{\sqrt{G^2 + \omega^2 C^2}} \tag{15.49}$$

$$X_0 = \frac{1}{2R_0} \cdot \frac{\omega LG - \omega CR}{G^2 + \omega^2 C^2} \tag{15.50}$$

The propagation constant of the coaxial line is

$$\dot{\gamma} = \alpha + j\beta \tag{15.51}$$

The attenuation constant is [1]

$$\alpha = \frac{\omega LG + \omega CR}{2\beta} \tag{15.52}$$

where the phase constant is

$$\beta = \left\{\frac{(\omega^2 LC - RG) + \sqrt{(RG - \omega^2 LC)^2 + (\omega LG + \omega RC)^2}}{2}\right\}^{\frac{1}{2}} \tag{15.53}$$

15.3.3 Wavelength and Phase Velocity

The phase velocity on the coaxial line is

$$v_p = \frac{\omega}{\beta} \tag{15.54}$$

The wavelength on the line is

$$\lambda_1 = \frac{2\pi}{\beta} \tag{15.55}$$

15.4 Waveguides

15.4.1 Rectangular Waveguides

15.4.1.1 Geometric Structure

A rectangular waveguide is a hollow conducting pipe of rectangular cross section as depicted in Figure 15.5. Electromagnetic microwaves are launched inside the waveguide through a coupling antenna at the transmission site. The launched waves are received at the receiving end of the waveguide by a coupling antenna. In this case, a rectangular coordinate system is set up on the rectangular waveguide (Figure 15.5). The *z*-axis is parallel to the axis of the waveguide and is set coinciding with the lower left corner of the waveguide. The wider dimension of the cross section of the waveguide *a* and the narrower dimension of the cross section of the waveguide is *b*, as shown in the figure.

15.4.1.2 Modes of Operation

The waveguide can be operated in either *H*- or *E*-mode, depending on the excitation configuration. An *H*-mode is a propagation mode in which the magnetic field, *H*, has a *z*-component, *Hz*, as referred to in

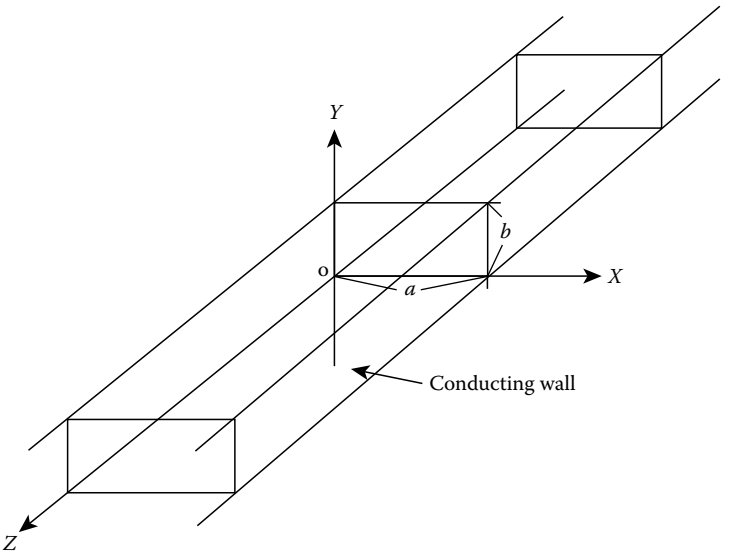


FIGURE 15.5 A rectangular waveguide and rectangular coordinate system.

Figure 15.5. In this mode, the electric field, E, is perpendicular to the direction of propagation, which is the $+z$-direction. Therefore, an H-mode is also called a transverse electric (TE) mode. An E-mode is a propagation mode in which the electric field, E, has a z-component, E_z, as referred to in Figure 15.5. In this mode, the magnetic field, H, is perpendicular to the direction of propagation, which is the $+z$-direction. Therefore, an E-mode is also called a transverse magnetic (TM) mode.

Solving Maxwell's equations for H-modes [1],

$$\dot{H}_z = \dot{H}_0 \cos\frac{m\pi x}{a} \cos\frac{n\pi y}{b} \varepsilon^{-\dot{\gamma}z+j\omega t} \tag{15.56}$$

where $|\dot{H}_0|$ is the amplitude of $\dot{H}_z$. Waveguide loss was neglected. Constants m and n are integral numbers 0, 1, 2, 3, ... and are called the mode number, and $\dot{\gamma}$ is the propagation constant. Both m and n cannot equal 0 simultaneously. Solving Maxwell's equations for E-modes [1],

$$\dot{E}_z = \dot{E}_0 \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} \varepsilon^{-\dot{\gamma}z+j\omega t} \tag{15.57}$$

where $|\dot{E}_0|$ is the amplitude of $\dot{E}_z$. Neither m nor n can equal 0. The waveguide loss was neglected.

An H-mode is expressed as the H_{mn}-mode or TE_{mn}-mode. An E-mode is expressed as the E_{mn}- or TM_{mn}-mode.

15.4.2 Waveguide Parameters

Propagation constant $\dot{\gamma}$ of a rectangular waveguide made of a good conductor [1]:

$$\dot{\gamma} = j\beta_0\sqrt{1-\left(\frac{\lambda}{\lambda_c}\right)^2} \ (\mathrm{m}^{-1}) \tag{15.58}$$

where β_0 is the phase constant of free space, which is

$$\beta_0 = \frac{2\pi}{\lambda} \ (\mathrm{m}^{-1}) \tag{15.59}$$

where

λ is the wavelength in free space

λ_c is the cutoff wavelength of the waveguide

Electromagnetic waves with $\lambda > \lambda_c$ cannot propagate inside the waveguide. It is given for both E_{mn}-mode and H_{mn}-mode operation by [1]

$$\lambda_c = \frac{2}{\sqrt{\left(\frac{m}{a}\right)^2+\left(\frac{n}{b}\right)^2}} \ (\mathrm{m}) \tag{15.60}$$

This means that if the waveguide is made of a good conductor, the attenuation constant

$$\alpha \approx 0 \ (\mathrm{m}^{-1}) \tag{15.61}$$

and the phase constant is

$$\beta_g = \beta_0\sqrt{1-\left(\frac{\lambda}{\lambda_c}\right)^2} \tag{15.62}$$

The wavelength in the waveguide, i.e., waveguide wavelength λ_g, is longer than the free-space wavelength λ:

$$\lambda_g = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \tag{15.63}$$

Then, the speed of propagation $v_p > c = f\lambda$.

$$v_p = f\lambda_g = \frac{f\lambda}{\sqrt{1 - \left(\frac{\lambda}{\lambda c}\right)^2}} = \frac{c}{\sqrt{1 - \left(\frac{\lambda}{\lambda c}\right)^2}} \tag{15.64}$$

For an H_{mn}-mode, the wave impedance is

$$\eta_H = \frac{-\dot{E}_y}{\dot{H}_x} = \frac{\dot{E}_x}{\dot{H}_y} = \frac{\sqrt{\frac{\mu_0}{\varepsilon_0}}}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \ (\Omega) \tag{15.65}$$

For an E_{mn}-mode the wave impedance is

$$\eta_E = \frac{-\dot{E}_y}{\dot{H}_x} = \frac{\dot{E}_x}{\dot{H}_y} = \sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2} \Big/ \sqrt{\frac{\mu_0}{\varepsilon_0}} \ (\Omega) \tag{15.66}$$

15.4.3 Circular Waveguides

15.4.3.1 Geometric Structure

A circular waveguide is a hollow conducting pipe of circular cross section, as depicted in Figure 15.6. Electromagnetic microwaves are launched inside the waveguide through a coupling antenna at the transmission site. The launched waves are received at the receiving end of the waveguide by a coupling

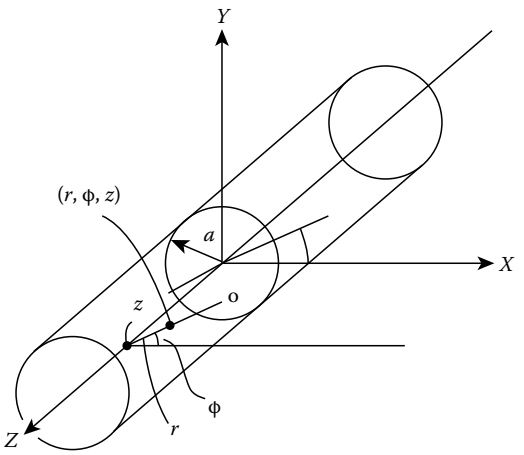


FIGURE 15.6 A circular waveguide and cylindrical coordinate system.

antenna. In this case, a circular coordinate system (r, ϕ, z) is set up in the circular waveguide, as depicted in Figure 15.6. The z-axis is coincident with the axis of the cylindrical waveguide. The inside radius of the circular waveguide is a.

15.4.3.2 Modes of Operation

The circular waveguide can be operated in either H- or E-mode, depending on the excitation configuration. An H-mode is a propagation mode in which the magnetic field, H, has a z-component, H_z, as referred to in Figure 15.6. In this mode the electric field, E, is perpendicular to the direction of propagation, which is the $+z$-direction. Therefore, an H-mode is also called a TE mode. In the mode $E_z = 0$. An E-mode is a propagation mode in which the electric field, E, has a z-component, E_z, as referred to in Figure 15.6. In this mode the magnetic field, H, is perpendicular to the direction of propagation, which is the $+z$-direction. Therefore, an E-mode is also called a TM mode.

Solving Maxwell's equations [1],

$$\dot{H}_z = \dot{H}_0 J_n\left(k'_{cm} r\right) \cos n\phi \varepsilon^{-\dot{\gamma}z + j\omega t} \tag{15.67}$$

where

$|\dot{H}_0|$ is the amplitude of $\dot{H}_z$

n and m are integral numbers 0, 1, 2, 3, ... and are called the mode number

$J_n(k'_{cm}r)$ is the Bessel function of nth order, with the argument that $k'_{cm}r$

k'_{cm} is the mth root of $J'_n(k_{cm}a) = 0$, which is

$$k'_{cm} = \frac{u'_{nm}}{a} \tag{15.68}$$

where u'_{nm} is the mth root of the derivative of the Bessel function of order n, i.e., $J'_n(x) = 0$, where x is a generic real argument. The propagation constant is $\dot{\gamma}$.

Solving Maxwell's equations for E-modes,

$$\dot{E}_z = \dot{E}_0 J_n(k_{cm} r) \cos n\phi \varepsilon^{-\dot{\gamma}z + j\omega t} \tag{15.69}$$

k_{cm} is an mth root of $J_n(k_c a) = 0$, which is

$$k_{cm} = \frac{u_{nm}}{a} \tag{15.70}$$

where u_{nm} is the mth root of the Bessel function of order n, i.e., $J_n(x) = 0$, where x is a generic real argument.

An H-mode in a circular waveguide is expressed as the H_{nm}-mode or the TM_{nm}-mode. An E-mode is expressed as the E_{nm}-mode or the TM_{nm}-mode.

15.4.3.3 Waveguide Parameters

The propagation constant $\dot{\gamma}$ of a circular waveguide made of a good conductor is [1]

$$\dot{\gamma} = j\beta_0 \sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)} \ (\mathrm{m}^{-1}) \tag{15.71}$$

where β_0 is the phase constant of free space, which is

$$\beta_0 = \frac{2\pi}{\lambda} \ (\mathrm{m}^{-1}) \tag{15.72}$$

where
λ is the wavelength in free space
λ_c is the cutoff wavelength of the waveguide

Electromagnetic waves with $\lambda > \lambda_c$ cannot propagate inside the waveguide. It is given for an H_{nm}-mode [1]

$$\lambda_{CH} = \frac{2\pi a}{u'_{nm}} \ (\mathrm{m}) \tag{15.73}$$

For E_{nm}-mode operation,

$$\lambda_{CE} = \frac{2\pi a}{u_{nm}} \ (\mathrm{m}) \tag{15.74}$$

This means that if the waveguide is made of a good conductor, the attenuation constant is

$$\alpha \approx 0 \ (\mathrm{m}^{-1}) \tag{15.75}$$

and the phase constant is

$$\beta_g = \beta_0 \sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2} \ (\mathrm{m}^{-1}) \tag{15.76}$$

The waveguide wavelength is

$$\lambda_g = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \ (\mathrm{m}^{-1}) \tag{15.77}$$

The speed of propagation (phase velocity) is

$$v_p = f\lambda_g = \frac{c}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \ (\mathrm{m/s}) \tag{15.78}$$

For an H_{nm}-mode, the wave impedence is

$$\begin{aligned} \eta_H &= \frac{\dot{E}_r}{\dot{H}_\phi} = \frac{-\dot{E}\phi}{\dot{H}_r} = \frac{j\omega\mu_0}{\dot{\gamma}} \\ &= \frac{\omega\mu_0}{\beta_0 \sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \ (\Omega) \end{aligned} \tag{15.79}$$

For an E_{nm}-mode, the wave impedance is

$$\eta_E = \frac{\dot{E}_r}{\dot{H}_\phi} = \frac{-\dot{E}_\phi}{\dot{H}_r} = \frac{\dot{\gamma}}{j\omega\varepsilon_0} = \frac{\beta_0\sqrt{1-\left(\frac{\lambda}{\lambda_c}\right)^2}}{\omega\varepsilon_0}\ (\Omega) \tag{15.80}$$

15.5 Microstrip Lines

15.5.1 Geometric Structure

Figure 15.7 presents a general geometric structure of a microstrip line. A conducting strip of width, w, and thickness, t, is laid on an insulating substrate of thickness, H, and permittivity, $\varepsilon = \varepsilon_0\varepsilon_r$. The dielectric substrate has a groundplate underneath, as presented in Figure 15.7.

15.5.2 Transmission Line Parameters

The characteristic impedance, Z_0, of a microstrip line, as shown in Figure 15.7, is given by [8–10]

$$Z_0 = \frac{42.4}{\sqrt{\varepsilon_r+1}}\ln\left\{1+\left(\frac{4H}{w'}\right)\left[\left(\frac{14+8/\varepsilon_r}{11}\right)\left(\frac{4H}{W'}\right)+\sqrt{\left(\frac{14+8\varepsilon_r}{11}\right)^2\left(\frac{4H}{w'}\right)+\frac{1+1/\varepsilon_r}{2}\pi^2}\right]\right\} \tag{15.81}$$

where w' is an effective width of the microstrip, which is given by

$$w' = w + \frac{1+1/\varepsilon_r}{2}\cdot\frac{t}{\pi}\ln\frac{10.87}{\sqrt{\left(\frac{t}{H}\right)^2+\left(\frac{1/\pi}{w/t+1.10}\right)^2}} \tag{15.82}$$

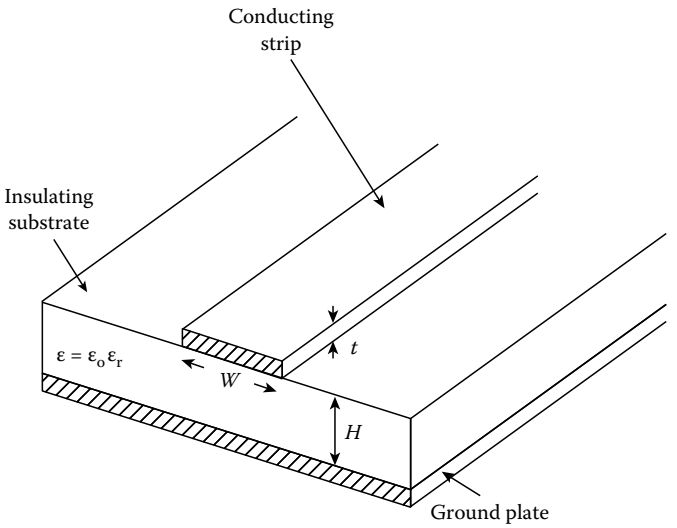


FIGURE 15.7 Geometric structure of a microstrip line.

The attenuation constant of a microstrip line is

$$\alpha = \frac{p + p'}{2} \frac{Z_{01}}{Z_0} \frac{2\pi}{\lambda_0} \text{ (Np/m)} \tag{15.83}$$

where

$$Z_{01} \equiv 30 \ \ln \left\{ 1 + \frac{1}{2} \left(\frac{8H}{w'} \right) \left[\frac{8H}{w'} + \sqrt{\left(\frac{8H}{w'} \right)^2 + \pi^2} \right] \right\} \tag{15.84}$$

and

$$p \equiv 1 - \frac{Z_{01}}{Z_{0\delta}} \tag{15.85}$$

where

$$Z_{0\delta} \equiv 30 \ \ln \left\{ 1 + \frac{1}{2} \left(\frac{8(H + \delta)}{w'} \right) \left[\frac{8H + \delta)}{w'} + \sqrt{\left(\frac{8H + \delta}{w'} \right)^2 + \pi^2} \right] \right\} \tag{15.86}$$

and

$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}} \tag{15.87}$$

is the skin depth of the conducting strip of conductivity σ

$$p' \equiv \frac{P_k}{1 - \frac{1/q - 1}{\varepsilon_r}} \tag{15.88}$$

where

$$\varepsilon_r = \varepsilon_r' - j\varepsilon_r'' \tag{15.89}$$

and

$$P_K = \sin \left[\tan^{-1} \frac{\varepsilon_r''}{\varepsilon_r'} \right] \tag{15.90}$$

and

$$q = \frac{1}{\varepsilon_r - 1} \left[\left(\frac{Z_{01}}{Z_0} \right)^2 - 1 \right] \tag{15.91}$$

15.5.3 Wavelength and Phase Velocity

The transmission line wavelength is

$$\lambda_1 = \frac{Z_0}{Z_{01}}\lambda_0 \tag{15.92}$$

The phase constant of the microstrip line is then

$$\beta = \frac{2\pi}{\lambda_0}\frac{Z_{01}}{Z_0} \tag{15.93}$$

The phase velocity of electromagnetic waves on the microstrip line is

$$v_p = 3 \times 10^8 \frac{Z_0}{Z_{01}} \tag{15.94}$$

15.6 Coplanar Waveguide

15.6.1 Geometric Structure

A cross-sectional view of a coplanar waveguide (CPW) is given in Figure 15.8. The waveguide consists of a narrow, central conducting strip of width $s(m)$ and very wide conducting plates on both sides of the central conducting strip, with gap widths w. These conductors are developed on a surface of dielectric substrate of thickness d, as presented in Figure 15.8. The electromagnetic waves propagate in the gap between the outer conducting plates and the center conducting strip.

15.6.2 Transmission Line Parameters

The attenuation constant of a CPW is given by [11]

$$\alpha = \left(\frac{\pi}{2}\right)^5 \cdot 2 \cdot \frac{1 - \frac{\varepsilon_{\text{eff}(f)}}{\varepsilon_r}}{\sqrt{\frac{\varepsilon_{\text{eff}(f)}}{\varepsilon_r}}} \cdot \frac{(s+2w)^2 \varepsilon_r^{3/2}}{c^3 K'(k) K(k)} \text{ (Np/m)} \tag{15.95}$$

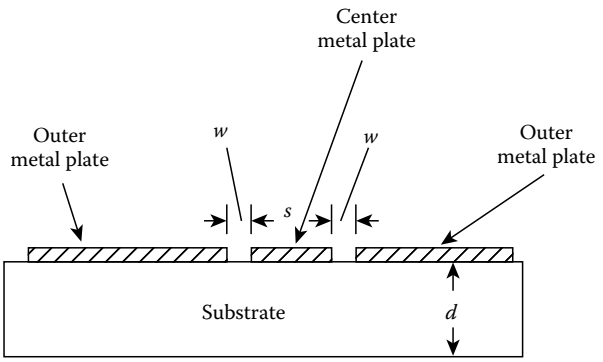


FIGURE 15.8 Cross-sectional view of a CPW.

where the effective dielectric constant is given by

$$\sqrt{\varepsilon_{\text{eff}}(f)} = \sqrt{\varepsilon_q} + \frac{\sqrt{\varepsilon_{\text{r}}} - \sqrt{\varepsilon_q}}{1 + a\left(\frac{f}{f_{\text{TE}}}\right)^{-b}} \tag{15.96}$$

and

$$\varepsilon_q = \frac{\varepsilon_{\text{r}} + 1}{2} \tag{15.97}$$

ε_{r} is the relative permittivity of the substrate material,

$$f_{\text{TE}} = \frac{c}{4d\sqrt{\varepsilon_{\text{r}} - 1}} \tag{15.98}$$

is the TE mode cutoff frequency,

$$k \equiv s/(s + 2w) \tag{15.99}$$

$K(k)$ is the complete elliptic integral of the first kind of the argument k, and c is the speed of light in vacuum, which is 3×10^8 m/s. The parameter a is [11]

$$a \approx \log^{-1}\left[u \log\frac{s}{w} + v\right] \tag{15.100}$$

$$u \approx 0.54 - 0.64q + 0.015q^2 \tag{15.101}$$

$$v \approx 0.43 - 0.86q + 0.54q^2 \tag{15.102}$$

$$q \approx \log\frac{s}{d} \tag{15.103}$$

The parameter b is an experimentally determined constant

$$b \approx 1.8 \tag{15.104}$$

$$K'(k) = K\left(\sqrt{1 - k^2}\right) \tag{15.105}$$

The phase constant of the CPW is

$$\beta(f) = 2\pi\frac{f}{c}\sqrt{\varepsilon_{\text{eff}}(f)}\ (\text{rad/m}) \tag{15.106}$$

The characteristic impedance of the CPW is [11]

$$Z_0 = \frac{120\pi}{\sqrt{\varepsilon_{\text{eff}}(f)}}\ \frac{K'(k)}{4K(k)}\ (\Omega) \tag{15.107}$$

15.6.3 Wavelength and Phase Velocity

The wavelength of electromagnetic waves propagating on a CPW is obtained from Equation 15.106:

$$\lambda_l = \frac{2\pi}{\beta(f)} = \frac{c}{f} \cdot \frac{1}{\sqrt{\varepsilon_{\text{eff}}}(f)} = \frac{\lambda}{\sqrt{\varepsilon_{\text{eff}}(f)}} \text{ (m)} \tag{15.108}$$

The phase velocity of the waves on the CPW is, then,

$$v_{\text{p}} = f\lambda_1 = \frac{c}{\sqrt{\varepsilon_{\text{eff}}(f)}} \text{ (m/s)} \tag{15.109}$$

References

1. T. K. Ishii, *Microwave Engineering*, San Diego, CA: Harcourt, Brace, Jovanovich, 1989.
2. J. R. Wait, *Electromagnetic Wave Theory*, New York: Harper & Row, 1985.
3. V. F. Fusco, *Microwave Circuits*, Englewood Cliffs, NJ: Prentice Hall, 1987.
4. L. A. Ware and H. R. Reed, *Communications Circuits*, New York: John Wiley & Sons, 1949.
5. E. A. Guillemin, *Communications Networks*, New York: John Wiley & Sons, 1935.
6. F. E. Terman, *Radio Engineering*, New York: McGraw-Hill, 1941.
7. H. J. Reich, P. F. Ordung, H. L. Krauss, and J. G. Skalnik, *Microwave Theory and Techniques*, Princeton, NJ: D. Van Nostrand, 1953.
8. H. A. Wheeler, Transmission-line properties of parallel strips separated by a dielectric sheet, *IEEE Trans. MTT*, MTT-13, 172–185, Mar. 1965.
9. H. A. Wheeler, Transmission-line properties of parallel strips by a conformal-mapping approximation, *IEEE Trans. MTT*, MTT-12, 280–289, May 1964.
10. H. A. Wheeler, Transmission-line properties of a strip on a dielectric sheet on a plane, *IEEE Trans. MTT*, MTT-25, 631–647, Aug. 1977.
11. M. Y. Frankel, S. Gupta, J. A. Valdmanis, and G. A. Mourou, Terahertz attenuation and dispersion characteristics of coplanar transmission lines. *IEEE Trans. MTT*, 39(6), 910–916, June 1991.

16

Multiconductor Transmission Lines

Daniël De Zutter
Ghent University

Luc Martens
Ghent University

16.1 Introduction: Frequency versus Time Domain Analysis **16**-1
16.2 Telegrapher's Equations for Uniform Multiconductor Transmission Lines **16**-3
Generalities • Low-Frequency or Quasitransverse Electromagnetic Description • Analytical Expressions for Some Simple Multiconductor Transmission Line Configurations
References **16**-8

16.1 Introduction: Frequency versus Time Domain Analysis

Multiconductor transmission lines (MTL), or multiconductor buses as they are also often called, are found in almost every electrical packaging technology and on every technology level from digital chips, over MMICs (monolithic microwave integrated circuits) to MCMs (multichip modules), boards, and backplanes. MTL are electrical conducting structures with a constant cross section (the x-, y-plane) which propagates signals in the direction perpendicular to that cross section (the z-axis) (see also Figure 16.1). Being restricted to a constant cross section we are in fact dealing with the so-called uniform MTL. The more general case using a nonuniform cross section is much more difficult to handle and constitutes a fully three-dimensional problem.

It is not the purpose of this chapter to give a detailed account of the physical properties and the use of the different types of MTL. The literature on this subject is abundant and any particular reference is bound to be both subjective and totally incomplete. Hence, we put forward only Refs. [1–3] as references here, as they contain a wealth of information and additional references.

In the frequency domain, i.e., for harmonic signals, solution of Maxwell's sourceless equations yields a number of (evanescent and propagating) modes characterized by modal propagation factors $\exp(\pm j \cdot \beta z)$ and by a modal field distribution, which depends only upon the (x, y) coordinates of the cross section. In the presence of losses and for evanescent modes β can take complex values, and $j\beta$ is then replaced by $\gamma = \alpha + j\beta$ (see Equation 16.3). In the propagation direction, the modal field amplitudes essentially behave as voltage and current along a transmission line. This immediately suggests that MTL should be represented on the circuit level by a set of coupled circuit transmission lines. The relationship between the typical circuit quantities, such as voltages, currents, coupling impedances, and signal velocities, on the one hand, and the original field quantities (modal fields and modal propagation factors) is not straightforward [4]. In general, the circuit quantities will be frequency dependent. The frequency domain circuit model parameters can be used as the starting point for time domain analysis of

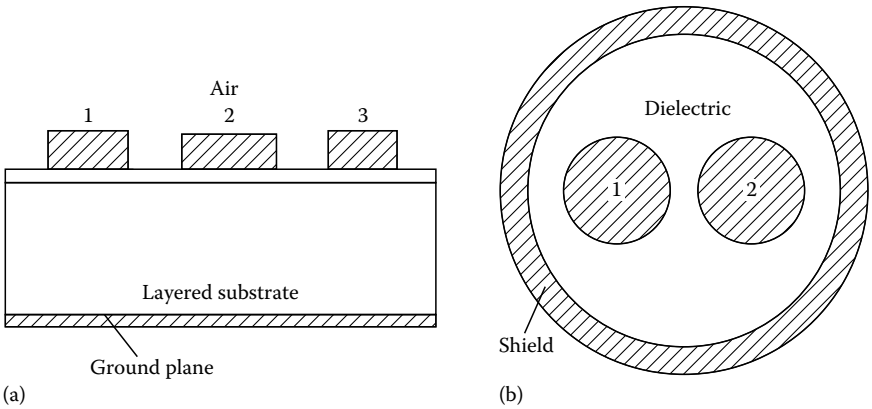


FIGURE 16.1 Two examples of cross sections of multiconductor lines.

networks, including multiconductor lines. This is, again, a vast research topic with important technical applications.

Section 16.2 describes the circuit modeling in the frequency domain of uniform MTL based on the Telegrapher's equations. The meaning of the voltages and currents in these equations is explained both at lower frequencies in which the quasi transverse electromagnetic (TEM) approach is valid as well as in the so-called full-wave regime valid for any frequency. The notions TEM, quasi-TEM, and full wave are elucidated. We introduce the capacitance, inductance, resistance, and conductance matrices together with the characteristic impedance matrix of the coupled transmission line model. Finally, for some simple MTL configurations analytical formulas are presented expressing the previous quantities and the propagation factors as a function of the geometric and electrical parameters of these configurations.

It would be a formidable task to give a comprehensive overview of all the methods that are actually used for the time domain analysis of MTL. In the remaining part of this paragraph a very short overview (both for uniform and nonuniform structures) is presented along with some references. In the case of linear loads and drivers, frequency domain methods in combination with (fast) Fourier transform techniques are certainly most effective [5–7]. In the presence of nonlinear loads and drivers other approaches must be used. Simulations based on harmonic balance techniques [8,9] are, again, mainly frequency domain methods. All signals are approximated by a finite sum of harmonics and the nonlinear loads and drivers are taken into account by converting their time domain behavior to the frequency domain. Kirchhoff laws are then imposed for each harmonic in an iterative way. Harmonic balance techniques are not very well suited for transient analysis or in the presence of strong nonlinearities, but are excellent for mixers, amplifiers, filters, etc. Many recent efforts were directed toward the development of time domain simulation methods (for both uniform and nonuniform interconnection structures) based on advanced convolution-type approaches. It is, of course, impossible to picture all the ramifications in this research field. We refer the reader to a recent special issue of *IEEE Circuits and Systems Transactions* [10], to the ''Simulation techniques for passive devices and structures'' section of a special issue of *IEEE Microwave Theory and Techniques Transactions* [11], and to a 1994 special issue of the *Analog Integrated Circuits and Signal Processing Journal* [12] and to the wealth of references therein.

Both frequency and time domain experimental characterization techniques for uniform and nonuniform multiconductor structures can be found in Chapter 17.

16.2 Telegrapher's Equations for Uniform Multiconductor Transmission Lines

16.2.1 Generalities

Figures 10.1a and b show the cross sections of two general coupled lossy MTLs consisting of $N+1$ conductors. These conductors either can be perfectly conducting or exhibit finite conductivity. Their cross section remains constant along the propagation or longitudinal direction z. The $(N+1)$th conductor is taken as reference conductor. In many practical cases, this will be the ground plane at the top or bottom (Figure 16.1a) of the layered dielectric in which the conductors are embedded or is the shielding surrounding the other conductors (Figure 16.1b). We restrict the analysis to the frequency domain, i.e., all field components and all voltages and currents have a common time dependence, exp $(j\omega t)$, which is suppressed in the sequel. The generalized Telegrapher's equations governing the circuit representation of the MTL of Figure 16.1 in terms of a set of C-coupled circuit transmission lines is given by Ref. [4]:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{V}}{\mathrm{d}z} + \mathbf{Z}\mathbf{I} &= 0 \\ \frac{\mathrm{d}\mathbf{I}}{\mathrm{d}z} + \mathbf{Y}\mathbf{V} &= 0 \end{aligned} \tag{16.1}$$

where

- $\mathbf{V}$ and $\mathbf{I}$ are column vectors, the C elements of which are the voltages and currents of the circuit model
- $\mathbf{Z}$ and $\mathbf{Y}$ are the $C \times C$ impedance and admittance matrices

Equation 16.1 is a good circuit description of the wave phenomena along a MTL if only the fundamental modes of the corresponding field problem are of importance. In that case $C = N$ ($C = 3$ in Figure 16.1a and $C = 2$ in Figure 16.1b) if a ground plane is present, and $C = N - 1$ in the absence of a ground plane. For the relationship between the actual electromagnetic field description in terms of modes and the circuit model (Equation 16.1), we refer the reader to Refs. [4,13]. The general solution to Equation 16.1 is given by

$$\begin{aligned} \mathbf{V}(z) &= 2\left(\mathbf{I}_{\mathrm{m}}^{\mathrm{T}}\right)^{-1} e^{-j\boldsymbol{\beta}z}\mathbf{K}^{+} + 2\left(\mathbf{I}_{\mathrm{m}}^{\mathrm{T}}\right)^{-1} e^{j\boldsymbol{\beta}z}\mathbf{K}^{-} \\ \mathbf{I}(z) &= \mathbf{I}_{\mathrm{m}} e^{-j\boldsymbol{\beta}z}\mathbf{K}^{+} - \mathbf{I}_{\mathrm{m}} e^{j\boldsymbol{\beta}z}\mathbf{K}^{-} \end{aligned} \tag{16.2}$$

where

- $\mathbf{K}^{+}$ and $\mathbf{K}^{-}$ are column vectors with C elements
- $\boldsymbol{\beta}$ is a diagonal $C \times C$ matrix with the propagation factors β_f ($f = 1, 2, \ldots, C$) of the C fundamental eigenmodes as diagonal elements

This matrix $\boldsymbol{\beta}$ reduces to a single propagation factor for a single transmission line (see Equation 15.6). For the calculation of the fields and of the propagation constants many different methods can be found in the literature [14]. Solution (Equation 16.2) of the differential equations (Equation 16.1) is the extension of the corresponding Equation 15.5 for a single transmission line to the coupled line case. It also consists of waves respectively traveling in positive and negative z-directions. The $\mathbf{I}_{\mathrm{m}}$ is a $C \times C$ matrix, the columns of which are the current eigenvectors of the circuit model. The following relationships hold:

$$\begin{aligned}\mathbf{Z} &= j\omega\mathbf{L} + \mathbf{R} = 2j\left(\mathbf{I}_{\mathrm{m}}^{\mathrm{T}}\right)^{-1}\beta(\mathbf{I}_{\mathrm{m}})^{-1}\\ \mathbf{Y} &= j\omega\mathbf{C} + \mathbf{G} = \frac{j}{2}\mathbf{I}_{\mathrm{m}}\beta\mathbf{I}_{\mathrm{m}}^{\mathrm{T}}\end{aligned} \tag{16.3}$$

L, **R**, **C**, and **G** are the $C \times C$ (frequency-dependent) inductance, resistance, capacitance, and conductance matrices. The $C \times C$ characteristic impedance matrix of the transmission line models is given by $Z_{\mathrm{char}} = 2\left(\mathbf{I}_{\mathrm{m}}^{\mathrm{T}}\right)^{-1}(\mathbf{I}_{\mathrm{m}})^{-1}$. The matrix $\mathbf{Z}_{\mathrm{char}}$ replaces the simple characteristic impedance number of Equation 15.9. In general the mapping of the wave phenomenon onto the circuit model equations (Equations 16.1 through 16.3) depends on the choice of $\mathbf{I}_{\mathrm{m}}$. We refer the reader to the detailed discussions in Ref. [4]. For MTLs the most adopted definition for the elements of $\mathbf{I}_{\mathrm{m}}$ is [15]

$$I_{\mathrm{m},jf} = \oint_{\mathrm{c}_j} \mathbf{H}_{\mathrm{tr},f} \cdot \mathrm{d}\mathbf{l}, \quad j, f = 1, 2, \ldots, C \tag{16.4}$$

where

c_j is the circumference of conductor j

$\mathbf{H}_{\mathrm{tr},f}$ is the transversal component of the magnetic field of eigenmode f

This means that $I_{\mathrm{m},jf}$ is the total current through conductor j due to eigenmode f. This definition is used in the power-current impedance definition for microstrip and stripline problems [9]. For slotline circuits, a formulation that parallels the one given above must be used, but in this case it makes much more sense to introduce the voltage eigenvectors and to define them as line integrals of the electric field (see Appendix B of Ref. [16]).

As **Z** and **Y** in Equation 16.3 are frequency dependent, the time domain equivalent of Equation 16.1 involves convolution integrals between **Z** and **I** on the one hand and **Y** and **V** on the other hand. The propagation factors β_f are also frequency dependent, hence the signal propagation will show dispersion, i.e., the signal waveform becomes distorted while propagating.

16.2.2 Low-Frequency or Quasitransverse Electromagnetic Description

In the previous section, the reader was given a very general picture of the MTL problem, valid for any frequency. This analysis is the so-called full-wave analysis. The present section is restricted to the low-frequency or quasistatic regime. Here, the cross section of the MTL is small with respect to the relevant wavelengths, the longitudinal field components can be neglected and the transversal field components can be found from the solution of an electrostatic or magnetostatic problem in the cross section of the MTL. A detailed discussion of the theoretical background can be found in Ref. [17]. In the quasi-TEM limit and in the absence of losses **R** and **G** are zero and **L** and **C** become frequency independent and take their classical meaning. Both skin-effect losses and small dielectric losses can be accounted for by a perturbation approach [18]. In that case a frequency dependent **R** and **G** must be reintroduced. If **R** and **G** are zero and **L** and **C** are frequency independent, the following Telegrapher's equations hold:

$$\begin{aligned}\frac{\partial \mathbf{v}}{\partial z} &= -\mathbf{L} \cdot \frac{\partial \mathbf{i}}{\partial t}\\ \frac{\partial \mathbf{i}}{\partial z} &= -\mathbf{C} \cdot \frac{\partial \mathbf{v}}{\partial t}\end{aligned} \tag{16.5}$$

Equation 16.5 is the time domain counterpart of Equation 16.1. We have replaced the capital letters for voltages and currents with lower case letters to distinguish between time and frequency domains. **L** and **C** are related to the total charge Q_i per unit length carried by each conductor and to the total magnetic flux F_i between each conductor and the reference conductor:

$$\mathbf{Q} = \mathbf{C} \cdot \mathbf{V}$$
$$\mathbf{F} = \mathbf{L} \cdot \mathbf{I} \tag{16.6}$$

where $\mathbf{Q}$ and $\mathbf{F}$ are $C \times 1$ column vectors with elements Q_i and F_i, respectively. For a piecewise homogeneous medium, one can prove that the inductance matrix $\mathbf{L}$ can be derived from an equivalent so-called vacuum capacitance matrix $\mathbf{C}_v$ with $\mathbf{L} = \mathbf{C}_v^{-1}$ and where $\mathbf{C}_v$ is calculated in the same way as $\mathbf{C}$, but with the piecewise constant ε everywhere replaced by the corresponding value of $1/\mu$. For nonmagnetic materials, this operation corresponds with taking away all dielectrics and working with vacuum, thus explaining the name of the matrix $\mathbf{C}_v$. Other properties of $\mathbf{C}$ and $\mathbf{L}$ are

$\mathbf{C}$ and $\mathbf{L}$ are symmetric, i.e., $C_{ij} = C_{ji}$ and $L_{ij} = L_{ji}$.
$\mathbf{C}$ is real, $C_{ii} > 0$ and $C_{ij} < 0$ $(i \neq j)$.
$\mathbf{l}$ is real, $L_{ii} > 0$ and $L_{ij} > 0$.

The propagation factors β_f which form the elements of β in Equation 16.2 are now given by the eigenvalues of $(\mathbf{LC})^{1/2}$ or equivalently of $(\mathbf{CL})^{1/2}$. The current eigenvectors which form the columns of $\mathbf{I}_m$ are now solutions of the following eigenproblem (where ω is the circular frequency):

$$\omega^2(\mathbf{CL})\mathbf{I} = (\beta_f)^2\mathbf{I} \tag{16.7}$$

The corresponding eigenvoltages are solutions of

$$\omega^2(\mathbf{LC})\mathbf{V} = (\beta_f)^2\mathbf{V} \tag{16.8}$$

Hence, corresponding voltage and current eigenmodes propagate with the same propagation factors and as $\mathbf{L}$, $\mathbf{C}$, $\mathbf{V}$, and $\mathbf{I}$ are frequency independent, β_f is proportional with ω and can be rewritten as $\beta_f = \omega\beta'_f$ proving that the propagation is nondispersive with velocity $v_f = 1/\beta'_f$. Remember that the subindex f takes the values 1, 2,..., C, i.e., for a three-conductor problem above a ground plane ($N = C = 3$, see Figure 16.1a), three distinct propagation factors and corresponding eigenmode profiles exist for currents and voltages. Note, however, that for the same β_f the eigenvector for the currents differs from the eigenvector of the voltages.

We conclude this section by remarking that, for MTL embedded in a homogeneous medium (such as the simple stripline or the coaxial cable with homogeneous filling), $(LC) = \varepsilon\mu\mathbf{1}$, where $\mathbf{1}$ is the unity matrix. Thus, the eigenmodes are purely TEM, i.e., electric and magnetic fields have only transversal components and the longitudinal ones are exactly zero. All propagation factors β_f take the same value $[c/(\varepsilon, \mu_r)^{1/2}]$, where c is the velocity of light in vacuum. Note, however, that even for identical β_f different eigenmodes will be found.

Numerical calculation of $\mathbf{L}$ and $\mathbf{C}$ can be performed by many different numerical methods (see the reference section of [18]), and for sufficiently simple configurations analytical formulas are available. For a line with one conductor and a ground plane ($N = C = 1$) the characteristic impedance $Z_0 = (L/C)^{1/2}$ and the signal velocity v_p is $v_p = (LC)^{-1/2}$.

16.2.3 Analytical Expressions for Some Simple Multiconductor Transmission Line Configurations

16.2.3.1 Symmetric Stripline

Sections 15.3 through 15.6 presented a number of MTL consisting of a single conductor and a ground plane ($N = C = 1$). Here, we add another important practical example, the symmetric stripline configuration of Figure 16.2. We restrict ourselves to the lossless case. A perfectly conducting strip of width w and

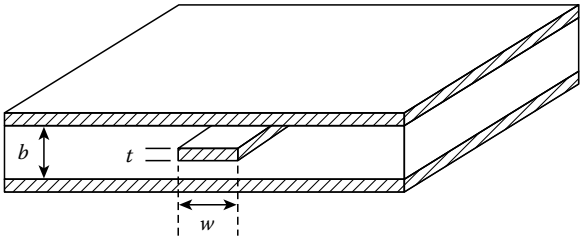


FIGURE 16.2 Stripline configuration.

thickness t is symmetrically placed between two perfectly conducting ground planes with spacing b. The insulating substrate has a permittivity $\varepsilon = \varepsilon_0\varepsilon_r$, and is nonmagnetic. This case has a single fundamental mode. The characteristic impedance Z_0 is given by Ref. [19]:

$$Z_0\sqrt{\varepsilon_r} = 30\ln\left\{1+\frac{4}{\pi}\frac{b-t}{W'}\left[\frac{8}{\pi}\frac{b-t}{W'}+\sqrt{\left(\frac{8}{\pi}\frac{b-t}{W'}\right)^2+6.27}\right]\right\} \tag{16.9}$$

with

$$\frac{W'}{b-t} = \frac{W}{b-t}+\frac{\Delta W}{b-t} \tag{16.10}$$

where

$$\frac{\Delta W}{b-t} = \frac{x}{\pi(1-x)}\left\{1-\frac{1}{2}\ln\left[\left(\frac{x}{2-x}\right)^2+\left(\frac{0.0796x}{W/b+1.1x}\right)^m\right]\right\} m = 2\left[1+\frac{2}{3}\frac{x}{1-x}\right]^{-1},\ x = t/b$$

For $W'/(b-t) < 10$ (Equation 16.9) is 0.5% accurate. The signal velocity v_p is given by $c/(\varepsilon_r)^{1/2}$, where c is the velocity of light in vacuum. The corresponding L and C are given by $L = Z_0/v_p$ and $C = 1/(Z_0 v_p)$.

16.2.3.2 Coupled Striplines

The configuration is depicted in Figure 16.3. It consists of the symmetric combination of the structure of Figure 16.2. There are now two fundamental TEM modes ($N = C = 2$). The even mode (index e) corresponds to the situation in which both central conductors are placed at the same voltage (speaking in low-frequency terms, of course). The odd mode (index o) corresponds to the situation where the

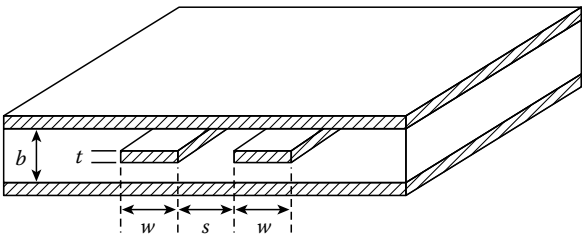


FIGURE 16.3 Coupled, symmetric stripline configuration.

central conductors are placed at opposite voltages. The impedances of the modes (respectively, $Z_{0,e}$ and $Z_{0,o}$) are given by

$$Z_{0e,o} = \frac{30\pi(b-t)}{\sqrt{\varepsilon_r}\left(W + \frac{bC_f}{2\pi}A_{e,o}\right)} \tag{16.11}$$

with

$$A_e = 1 + \frac{\ln(1 + \tanh\ \theta)}{\ln\ 2}$$

$$A_o = 1 + \frac{\ln(1 + \coth\ \theta)}{\ln\ 2}$$

$$\theta = \frac{\pi S}{2b}$$

$$C_f(t/b) = 2\ln\left(\frac{2b-t}{b-t}\right) - \frac{t}{b}\ln\left[\frac{t(2b-t)}{(b-t)^2}\right]$$

The signal velocity is the same for both modes ($c/(\varepsilon_r)^{1/2}$), and the L and C of both modes can be found by replacing Z_0 in Section 16.2.3.1 by $Z_{0,e}$ and $Z_{0,o}$, respectively.

16.2.3.3 Coupled Microstrip Lines

The configuration is depicted in Figure 16.4. It consists of the symmetric combination of the structure of Figure 15.7. Again, we have two fundamental modes, but the modes are hybrid, i.e., not purely TEM. Much work has been done on this configuration. The formulas proposed in the literature are quite lengthy; and we refer the reader to Refs. [20,21]. Reference [20] gives a very good overview together with some simple approximations, and Ref. [21] gives the most accurate formulas, taking into account the frequency dependence. It is important to remark here that the two impedances, $Z_{0,e}$ and $Z_{0,o}$, can be found. They depend upon frequency. Both modes now have a different velocity. The data found in literature are typically expressed in terms of the effective dielectric constant. The modal field lines are both found in the air above the substrate and in the substrate itself. Hence, the field experiences an effective dielectric constant that is smaller than the dielectric constant of the substrate. The effective dielectric constant for the even mode ($\varepsilon_{r,e}$) will be higher than for the odd mode ($\varepsilon_{r,o}$), and are frequency dependent. The corresponding modal velocities are given by $c/(\varepsilon_{r,e})^{1/2}$ and $c/(\varepsilon_{r,o})^{1/2}$.

16.2.3.4 Two-Wire Line

See Section 15.2.

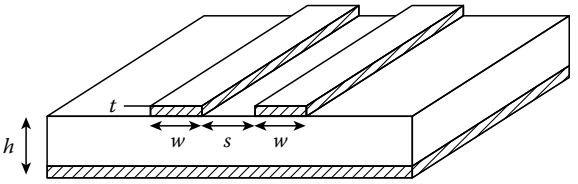


FIGURE 16.4 Coupled, symmetric microstrip configuration.

References

1. T. Itoh, Ed., *Numerical Techniques for Microwave and Millimeter-Wave Passive Structures*, New York: John Wiley & Sons, 1989.
2. J. A. Kong, Ed., *Progress in Electromagnetics Research*, Vols. 1–5, New York: Elsevier, 1989–1991.
3. C. F. Coombs, Ed., *Printed Circuits Handbook*, 3rd ed., New York: McGraw-Hill, 1988.
4. N. Faché, F. Olyslager, and D. De Zutter, *Electromagnetic and Circuit Modelling of Multiconductor Transmission Lines*, Oxford Engineering Series 35, Oxford: Clarendon Press, 1993.
5. T. R. Arabi, T. K. Sarkar, and A. R. Djordjevic, Time and frequency domain characterization of multiconductor transmission lines, *Electromagnetics*, 9(1), 85–112, 1989.
6. J. R. Griffith and M. S. Nakhla, Time domain analysis of lossy coupled transmission lines, *IEEE Trans. MTT*, 38(10), 1480–1487, Oct. 1990.
7. B. J. Cooke, J. L. Prince, and A. C. Cangellaris, *S*-parameter analysis of multiconductor, integrated circuit interconnect systems, *IEEE Trans. Computer-Aided Design*, CAD-11(3), 353–360, Mar. 1992.
8. V. Rizzoli et al., State of the art and present trends in nonlinear microwave CAD techniques, *IEEE Trans. MTT*, 36(2), 343–363, Feb. 1988.
9. R. Gilmore, Nonlinear circuit design using the modified harmonic balance algorithm, *IEEE Trans. MTT*, 34(12), 1294–1307, Dec. 1986.
10. *IEEE Trans. Circuits Syst. Trans., I: Fundamental Theory and Applications*, Special Issue on Simulation, Modelling and Electrical Design of High-Speed and High-Density Interconnects, 39(11), Nov. 1992.
11. *IEEE Trans. MTT*, Special Issue on Process-Oriented Microwave CAD and Modeling, 40(7), July 1992.
12. *Analog Integrated Circuits Signal Process.*, Special Issue on High-Speed Interconnects, 5(1), 1–107, Jan. 1994.
13. F. Olyslager, D. De Zutter, and A. T. de Hoop, New reciprocal circuit model for lossy waveguide structures based on the orthogonality of the eigenmodes, *IEEE Trans. MTT*, 42(12), 2261–2269, Dec. 1994.
14. F. Olyslager and D. De Zutter, Rigorous boundary integral equation solution for general isotropic and uniaxial anisotropic dielectric waveguides in multilayered media including losses, gain and leakage, *IEEE Trans. MTT*, 41(8), 1385–1392, Aug. 1993.
15. R. H. Jansen and M. Kirschning, Arguments and an accurate model for the power-current formulation of microstrip characteristic impedance, *Arch. Elek. Übertragung*, 37(3/4), 108–112, 1983.
16. T. Dhaene and D. De Zutter, CAD-oriented general circuit description of uniform coupled lossy dispersive waveguide structures, *IEEE Trans. MTT*, Special Issue on Process-Oriented Microwave CAD and Modeling, 40(7), 1445–1554, July 1992.
17. I. V. Lindell, On the quasi-TEM modes in inhomogeneous multiconductor transmission lines, *IEEE Trans. MTT*, 29(8), 812–817, 1981.
18. F. Olyslager, N. Faché, and D. De Zutter, New fast and accurate line parameter calculation of general multiconductor transmission lines in multilayered media, *IEEE Trans, MTT*, MTT-39(6), 901–909, June 1991.
19. H. A. Wheeler, Transmission line properties of a stripline between parallel planes, *IEEE Trans. MTT*, 26, 866–876, Nov. 1978.
20. T. Edwards, *Foundations for Microstrip Circuit Design*, 2nd ed., Chichester, UK: John Wiley & Sons, 1992.
21. M. Kirschning and R. H. Jansen, Accurate wide-range design equations for the frequency-dependent characteristics of parallel coupled microstrip lines, *IEEE Trans. MTT*, 32(1), 83–90, Jan. 1984.

17

Time and Frequency Domain Responses

17.1 Time-Domain Reflectometry **17**-1
Principles • One-Port Time-Domain Reflectometry • Time-Domain Reflectometry Pictures for Typical Loads • Time-Domain Reflectometric Characterization of an Interconnection Structure
17.2 Frequency-Domain Network Analysis..... **17**-6
Introduction • Network Analyzer: Block Diagram • Measurement Errors and Calibration
References **17**-9

Luc Martens
Ghent University

Daniël De Zutter
Ghent University

17.1 Time-Domain Reflectometry

17.1.1 Principles

Time-domain reflectometry (TDR) is used to characterize interconnections in the time domain. The setup essentially consists of a time domain step generator and a digital sampling oscilloscope (Figure 17.1) [1]. The generator produces a positive-going step signal with a well-defined rise time. The step is applied to the device under test. The reflected and the transmitted signals are shown on the oscilloscope. Measuring the reflected signal is called TDR; the transmitted signal is measured using the time-domain transmission (TDT) option.

The characteristic impedance levels and delay through an interconnection structure can be derived from the TDR measurements. The TDT measurement gives information about the losses (decrease of magnitude) and degradation of the rise time (filtering of high-frequency components). The TDR/TDT measurements also are used to extract an equivalent circuit consisting of transmission lines and lumped elements.

The fundamentals of TDR are discussed in detail in Refs. [2,3]. Reference [4] describes the applications of TDR in various environments including PCB/backplane, wafer/hybrids, IC packages, connectors, and cables.

17.1.2 One-Port Time-Domain Reflectometry

Figure 17.2 demonstrates that the device under test is a simple resistor with impedance Z_L. In this case, a mismatch with respect to the reference or system impedance Z_0 exists. A reflected voltage wave will appear on the oscilloscope display algebraically added to the incident wave. The amplitude of the reflected voltage wave, E_r, is determined by the reflection coefficient of the load impedance, Z_L, with respect to the system impedance Z_0:

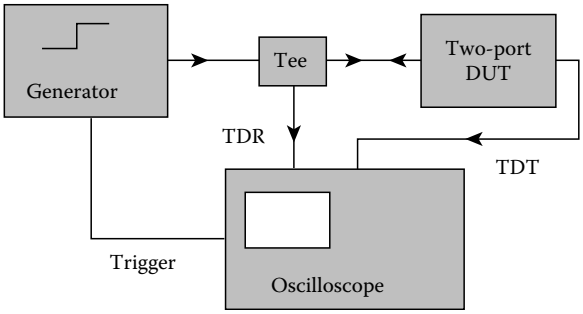


FIGURE 17.1 Setup of TDR/TDT measurements. DUT = device under test.

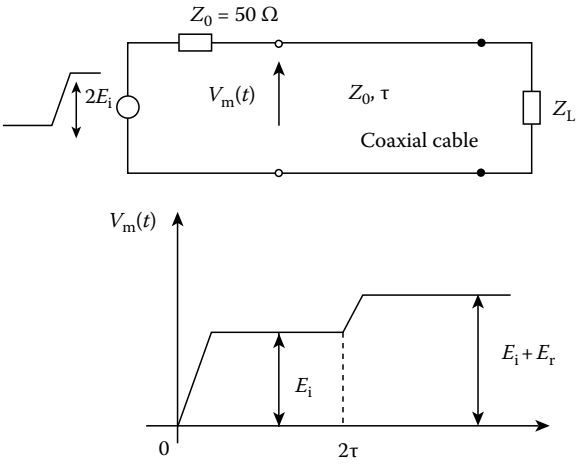


FIGURE 17.2 TDR measurement of an impedance Z_L.

$$E_r = \rho E_i = \frac{Z_L - Z_0}{Z_L + Z_0} E_i \tag{17.1}$$

Figure 17.2 also depicts the time-domain picture shown on the oscilloscope for a load, the impedance Z_L that is larger than Z_0. From the measurement of the magnitude E_r of the reflected voltage wave, the load impedance Z_L can be derived.

17.1.3 Time-Domain Reflectometry Pictures for Typical Loads

The simplest loads to be measured are the open circuit and the short circuit. For ideal open-circuit and short-circuit loads the reflection coefficient is, respectively, 1 and −1. This means that the measured voltage doubles in the first case and goes to zero in the second case, when the reflected voltage wave arrives at the oscilloscope (Figure 17.3).

For any other real load impedance, the reflection coefficient lies between −1 and 1. If the real load impedance is larger than the reference impedance, the reflected voltage wave is a positive-going step signal. In this case the amplitude of the voltage is increased when the reflected wave is added to the input step (Figure 17.4). The reverse happens when the load impedance is lower than the reference impedance.

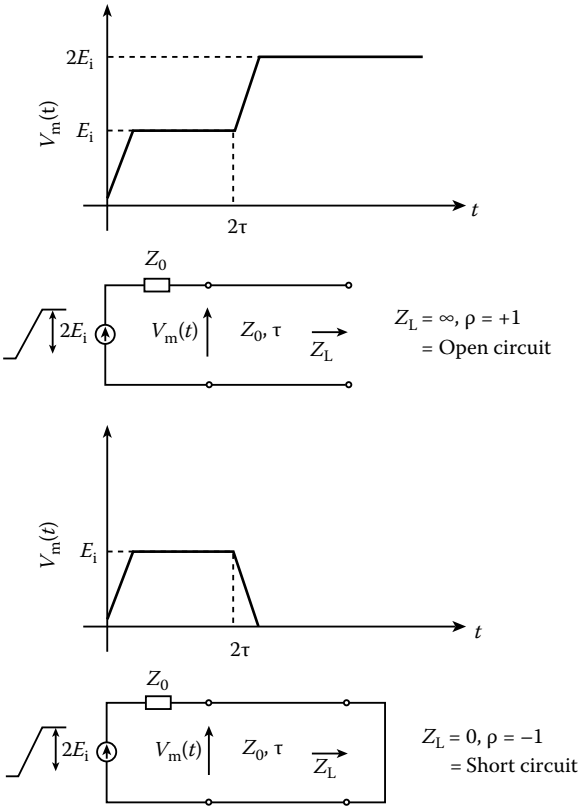


FIGURE 17.3 TDR pictures of an open- and a short-circuit termination.

For complex load impedances, the step response is more complicated. For example, in the case of a series connection of a resistance and an inductance or a parallel connection of a resistance and a capacitance, a first-order step response is obtained. From the two pictures in Figure 17.5, we learn that a series inductance gives a positive dip, while the capacitance produces a negative dip.

17.1.4 Time-Domain Reflectometric Characterization of an Interconnection Structure

One of the advantages of TDR is its ability to determine impedance levels and delays through an interconnection structure with multiple discontinuities. An example is shown in Figure 17.6 for a microstrip line connected to the measurement cable. We assume a perfect junction of the two transmission lines. The line is terminated in a load with impedance Z_L. Observe that two mismatches produce reflections that can be analyzed separately. The mismatch at the junction of the two transmission lines generates a reflected wave $E_{r1} = \rho_1 E_i$. Similarly, the mismatch at the load creates a reflection due to its reflection coefficient ρ_2. Both reflection coefficients are defined as:

$$\rho_1 = \frac{Z_0' - Z_0}{Z_0' + Z_0}$$
$$\rho_2 = \frac{Z_L - Z_0'}{Z_L + Z_0'} \tag{17.2}$$

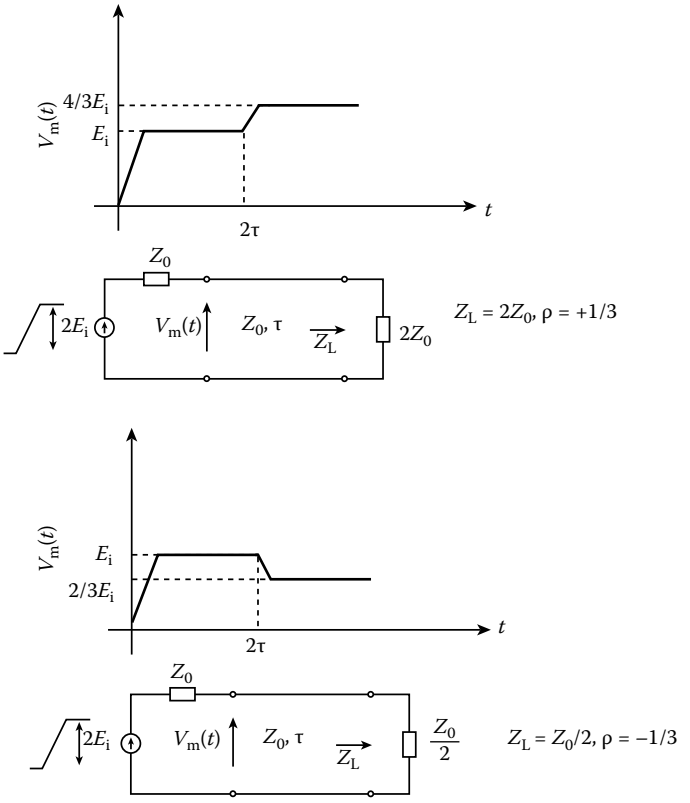


FIGURE 17.4 TDR pictures of real impedance terminations ($Z_L = 2Z_0$ and $Z_L = Z_0/2$).

After a time τ, the reflection at the junction of the transmission lines occurs. The voltage wave associated with this reflection adds to the oscilloscope's picture at the time instant 2τ. The voltage wave that propagates further in the microstrip line is $(1 + \rho_1)\, E_i$ and is incident on Z_L. The reflection at Z_L occurs at the time $\tau + \tau'$ and is given by

$$E_{rL} = \rho_2(1 + \rho_1)E_i \tag{17.3}$$

After a time $\tau + 2\tau'$, a second reflection is generated at the junction. The reflection is now determined by the reflection coefficient $\rho_1' = -\rho_1$. The voltage wave E_{r2} that is transmitted through the junction and propagates in the direction of the generator adds to the time-domain picture at time instant $2\tau + 2\tau'$ and is given by

$$E_{r2} = (1 + \rho_1)E_{rL} = (1 - \rho_1)\rho_2(1 + \rho_1)E_i = \left(1 - \rho_1^2\right)\rho_2 E_i \tag{17.4}$$

If ρ_1 is small in comparison to 1, then

$$E_{r2} \approx \rho_2 E_i \tag{17.5}$$

which means that ρ_2 can be determined from the measurement of E_{r2}.

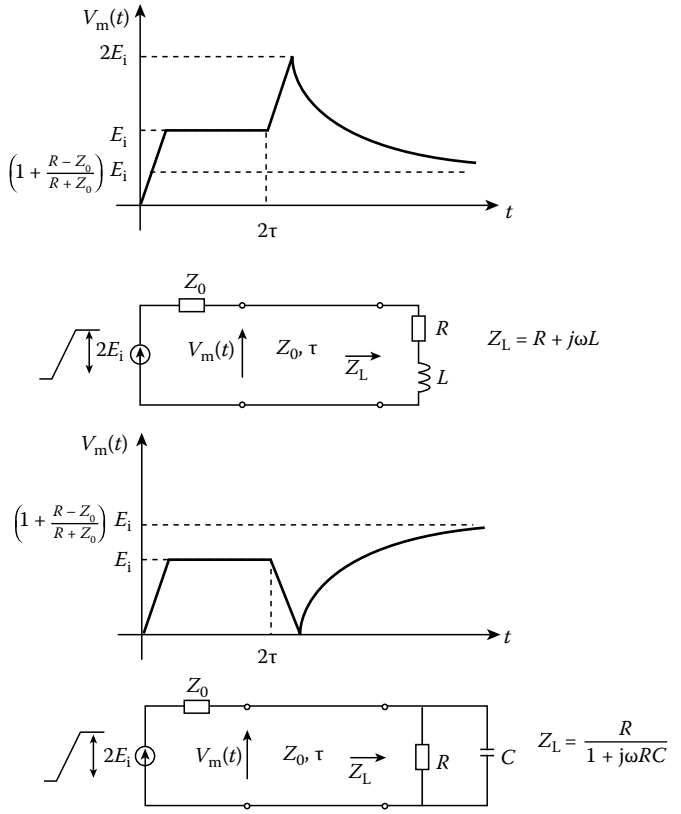


FIGURE 17.5 TDR pictures of two complex impedance terminations.

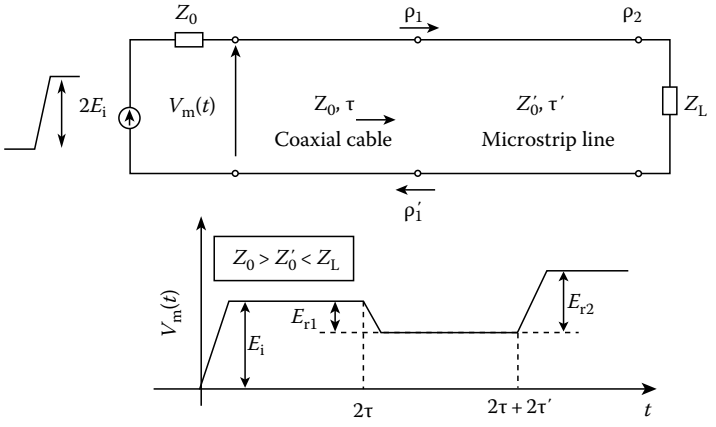


FIGURE 17.6 TDR measurement of a microstrip line terminated in an impedance Z_L.

In this example, the measurement cable was perfectly matched to the generator impedance so that no reflection occurred at the generator side, which simplifies the time-domain picture. In the case of an interconnection with many important discontinuities (high reflection coefficient), multiple reflections can prevent a straightforward interpretation of the oscilloscope's display.

17.2 Frequency-Domain Network Analysis

17.2.1 Introduction

A distributed circuit can also be analyzed in the frequency domain. At low frequencies the circuits are characterized by their *Z*- or *Y*-parameters. At high frequencies, circuits are better characterized by *S*-parameters. We focus only on *S*-parameter characterization.

The network analyzer is an instrument that measures the *S*-parameters of a two-port device (Figure 17.7) [5]. The reference impedance for the *S*-matrix is the system impedance, which is the standard 50 Ω. A sinusoidal signal is fed to the device under test and the reflected and transmitted signals are measured. If the network analyzer is of the vectorial type, it measures the magnitude as well as the phase of the signals.

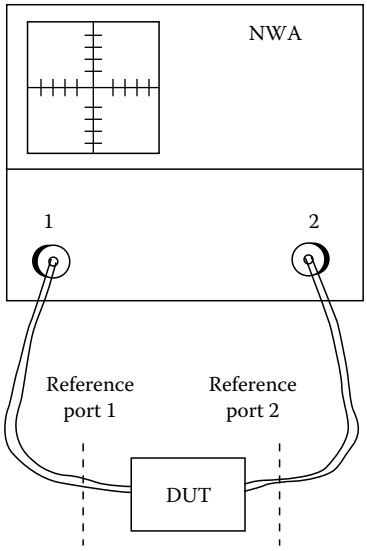


FIGURE 17.7 Network analyzer (NWA) setup.

17.2.2 Network Analyzer: Block Diagram

In operation, the source is usually set to sweep over a specified bandwidth (maximally 45 MHz to 26.5 GHz for the HP8510B network analyzer). A four-port reflectometer samples the incident, reflected, and transmitted waves, and a switch allows the network analyzer to be driven from either port 1 or port 2.

A powerful internal computer is used to calculate and display the magnitude and phase of the *S*-parameters or other quantities such as voltage standing wave radio, return loss, group delay, impedance, etc. A useful feature is the capability of determining the time-domain response of the circuit by calculating the inverse Fourier transform of the frequency-domain data.

Figure 17.8 depicts the network analyzer in the reflection measurement configuration. A sinusoidal signal from a swept source is split by a directional coupler into a signal that is fed to a reference channel and a signal that is used as input for the device under test. The reflected signal is fed back through the directional coupler to the test channel. In the complex ratio measuring unit, the amplitude and phase of the test and reference signals are compared.

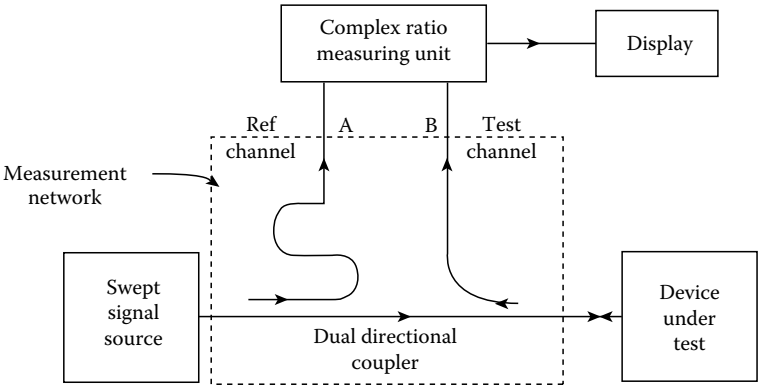


FIGURE 17.8 Block diagram for reflection measurements with the network analyzer.

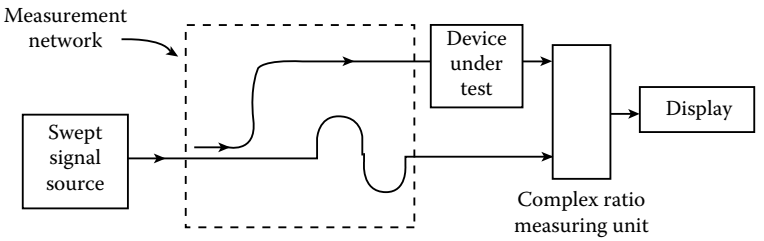


FIGURE 17.9 Block diagram for transmission measurements with the network analyzer.

The principle of the transmission measurement is the same as for the reflection configuration (Figure 17.9). In this setup the signal transmitted through the device under test is fed to the test channel. The other part of the signal of the swept source is transmitted through the directional coupler and fed to the reference channel. Again, amplitude and phase of the two signals are compared.

An important issue concerning the measurement of *S*-parameters is the calibration of the results. In order to perform a good calibration, the nature of the measurement errors must be well understood. These items are the subject of Section 17.2.3.

17.2.3 Measurement Errors and Calibration

17.2.3.1 Measurement Errors

The network analyzer measurement errors [6] can be classified in two categories:

1. *Random errors*—These are nonrepeatable measurement variations that occur due to noise, environmental changes, and other physical changes in the test setup. These are any errors that the system itself may not be able to measure.
2. *Systematic errors*—These are repeatable errors. They include the mismatch and leakage terms in the test setup, the isolation characteristic between the reference and test signal paths, and the system frequency response.

In most *S*-parameters measurements, the systematic errors are those that produce the most significant measurement uncertainty. Because each of these errors produces a predictable effect upon the measured data, their effects can be removed to obtain a corrected value for the test device response. The procedure of removing these systematic errors is called calibration. We illustrate here the calibration for one-port measurements. The extension to two-ports is mathematically more involved, but the procedure is fundamentally the same.

17.2.3.2 One-Port Calibration: Error Correction Network

In the configuration for one-port measurements, the real network analyzer can be considered as a connection of a ''perfect'' network analyzer and a two-port whose parameters are determined by the systematic errors. ρ_m is the measured reflection coefficient, while ρ_a is the actual one (Figure 17.10). The three independent coefficients of the error correction network [7] are

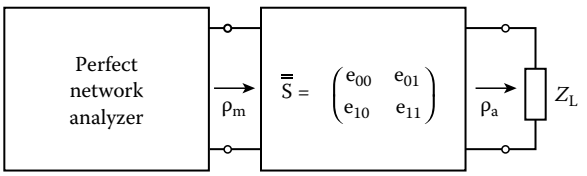


FIGURE 17.10 Error correction network for calibration of the *S*-parameter measurements.

1. *Directivity* e_{00}—The vector sum of all leakage signals appearing at the network analyzer test input due to the inability of the signal separation device to absolutely separate incident and reflected waves, as well as residual effects of cables and adapters between the signal separation device and the measurement plane
2. *Source impedance mismatch* e_{11}—The vector sum of the signals appearing at the network analyzer test input due to the inability of the source to maintain absolute constant power at the test device input as well as cable and adaptor mismatches and losses outside the source leveling loop
3. *Tracking or frequency response* $e_{10}e_{01}$—The vector sum of all test setup variations in the magnitude and phase frequency response between the reference and test signal paths

The relations between ρ_m and ρ_a are given by

$$\rho_m = e_{00} + \frac{e_{01}e_{10}\rho_a}{1 - e_{11}\rho_a} \tag{17.6}$$

$$\rho_a = \frac{\rho_m - e_{00}}{e_{11}(\rho_m - e_{00}) + e_{10}e_{01}} \tag{17.7}$$

The error coefficients are determined by measuring standard loads. In the case of the short-open-load (SOL) calibration, a short-circuit, open-circuit, and matched load are measured [5]. The standards are characterized over a specified frequency range (e.g., from 45 MHz to 26.5 GHz for the network analyzer HP8510B). The best-specified standards are coaxial. The quality of the coaxial standards is determined by the precision of the mechanical construction. The short circuit and the load are in most cases nearly perfect ($\rho_a^1 = -1$ and $\rho_a^3 = 0$, respectively). The open circuit behaves as a frequency-dependent capacitance:

$$\rho_a^2 = e^{-j2\ \mathrm{arctg}(\omega C Z_0)} \quad \text{with } C = C_0 + f^2 C_2 \tag{17.8}$$

where

Z_0 is the characteristic impedance of the coaxial part of the standard

C_0 and C_2 are specified by the constructor of the hardware

Once the reflection coefficients of the standards are specified and the standards are measured, a set of linear equations in the error coefficients is derived:

$$a\rho_a^i + b - c\rho_a^i\rho_m^i = \rho_m^i, \quad i = 1, 2, 3 \tag{17.9}$$

where $a = e_{01}e_{10} - e_{00}e_{11}$, $b = e_{00}$, and $c = -e_{11}$. This set is easily solved in the error coefficients which are then used together with the measurement of ρ_m to determine ρ_a.

Other calibration methods (thru-reflect-line [TRL], line-reflect-match [LRM]) are described in Refs. [8,9]. Analogous calibration techniques also were developed for time-domain measurements (TDR/TDT) [10]. Applications of *S*-parameter measurements on circuits on PCBs are described in Ref. [11].

17.2.3.3 Error Checking and Verification of the Calibration

Checking errors in the calibration is done by applying the calibration to a measured standard. The specified reflection response of the standard must be found. In Figure 17.11, the uncalibrated measurement of the open-circuit standard is shown. After calibration, we obtain the specified response (capacitance model) of the standard (Figure 17.11). The calibration can be further verified on a standard transmission line.

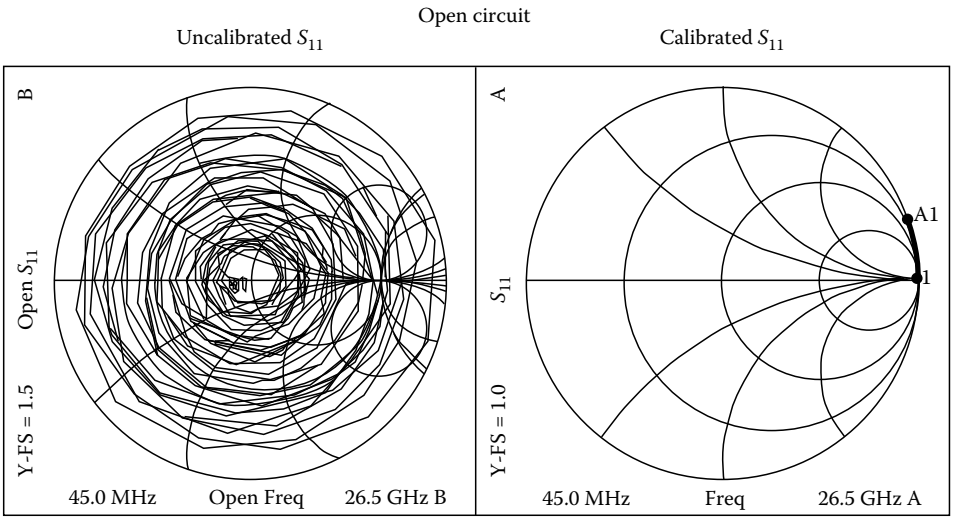


FIGURE 17.11 Smith chart representation of the uncalibrated and calibrated S_{11} data of an open circuit (measurements in the range of 45 MHz to 26.5 GHz).

References

1. A. M. Nicolson, C. L. Benett, Jr., D. Lammensdorf, and L. Susman, Applications of time domain metrology to the automation of broad-band microwave measurements, *IEEE Trans. MTT*, MTT-20(1), 3–9, Jan. 1972.
2. Agilent Technologies, TDR fundamentals for use with HP54120T digitizing oscilloscope and TDR, HP-Appl. Note 62, Apr. 1988.
3. Agilent Technologies, Improving time domain network analysis measurements, HP-Appl. Note 62-1, Apr. 1988.
4. Agilent Technologies, Advanced TDR techniques, HP-Appl. Note 62-3, May 1990.
5. R. A. Hackborn, An automatic network analyzer system, *Microwave J.*, 11, 45–52, May 1968.
6. B. Donecker, Accuracy predictions for a new generation network analyzer, *Microwave J.*, 27, 127–141, June 1984.
7. J. Williams, Accuracy enhancement fundamentals for network analyzers, *Microwave J.*, 32, 99–114, Mar. 1989.
8. G. F. Engen and C. A. Hoer, Thru-reflect-line: An improved technique for calibrating the dual six-port automatic network analyzer, *IEEE Trans. MTT*, MTT-27, 987–993, Dec. 1979.
9. D. Wiliams, R. Marks, and K.R. Phillips, Translate LRL and LRM calibrations, *Microwaves RF*, 30, 78–84, Feb. 1991.
10. T. Dhaene, L. Martens, and D. De Zutter, Calibration and normalization of time domain network analyzer measurements, *IEEE Trans. MTT*, MTT-42, 580–589, Apr. 1994.
11. P. Degraeuwe, L. Martens, and D. De Zutter, Measurement set-up for high-frequency characterization of planar contact devices, in *Proc. of 39th Automatic RF Techniques (ARFTG) Meeting*, Albuquerque, NM, pp. 19–25, June 1992.

18

Distributed RC Networks

18.1 Uniform Distributed RC Lines **18**-1
Solution in the Time Domain • Solution in the Frequency Domain • Uniform, Lossy DRC Lines
18.2 Nonuniform Distributed RC Lines **18**-8
Approximation with Concatenated Uniform Sections • Asymptotic Approximation for Large "**s**" • Lumped Element Approximation
18.3 Infinite-Length RC Lines **18**-12
Generalization of the Time Constant Representation • Generalization of the Pole-Zero Representation • Relations among $R(\zeta)$, $I_d(\Sigma)$, and the Impedance Function • Practical Calculation of the $R(\zeta)$ Function
18.4 Inverse Problem for Distributed RC Circuits **18**-19
Network Identification by Deconvolution
References **18**-24

Vladimír Székely
Budapest University of Technology and Economics

In everyday practice, we often encounter RC networks that are inherently distributed (DRC lines). All the resistors and the interconnection lines of an IC, the channel regions of the FET, and MOS transistors are DRC lines. The electrical behavior of such networks is discussed in the current section.

18.1 Uniform Distributed RC Lines

First, let us consider the so-called lossless RC lines. The structure in Figure 18.1 has only serial resistance and parallel capacitance. Both the resistance and the capacitance are distributed along the x axis. r and c denote the resistance and the capacitance pro unit length, respectively. In this section, these values are considered constants, which means that the RC line is uniform.

Consider a Δx length section of the structure. Using the notations of Figure 18.1, the following relations can be given between the currents and voltages:

$$\Delta v = -ir\Delta x, \quad \Delta i = -\frac{\partial v}{\partial t} c\Delta x \tag{18.1}$$

By taking the limit as $\Delta x \to 0$, we obtain a pair of differential equations

$$\frac{\partial v}{\partial x} = -ri, \quad \frac{\partial i}{\partial x} = -c\frac{\partial v}{\partial t} \tag{18.2}$$

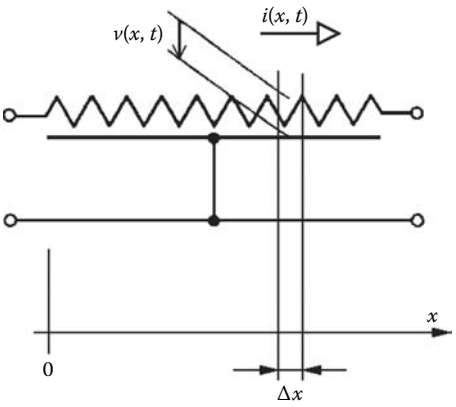


FIGURE 18.1 Lossless DRC line.

which describe the behavior of the lossless RC line. Substituting the first equation of Equation 18.2 after derivation into the second equation of Equation 18.2 results in

$$\frac{\partial v}{\partial t} = \frac{1}{rc}\frac{\partial^2 v}{\partial x^2} \tag{18.3}$$

This shows that the time-dependent voltage along the line $v(x, t)$ is determined by a homogeneous, constant-coefficient, second-order partial differential equation, with the first time derivative on one side and the second spatial derivative on the other side. This kind of equation is called a "diffusion equation," as diffusion processes are described using such equations.

18.1.1 Solution in the Time Domain

The differential Equation 18.3 is fulfilled by the simple, spatially Gauss-like function

$$v(x, t) = \frac{Q/c}{2\sqrt{\pi t/rc}} \exp\left(-\frac{x^2}{4t/rc}\right) \tag{18.4}$$

This can be proved easily by substituting the function into Equation 18.3.

Equation 18.4 describes the physical situation in which the uniform RC structure is of infinite length in both directions, and a short, impulse-like Q charge injection is applied at $x = 0$ in the $t = 0$ instant. This means a Dirac-δ excitation at $x = 0$ both in charge and voltage. As time elapses, the charge spreads equally in both directions. The extension of the charge and voltage wave is ever increasing, but their amplitude is decreasing (see Figure 18.2a).

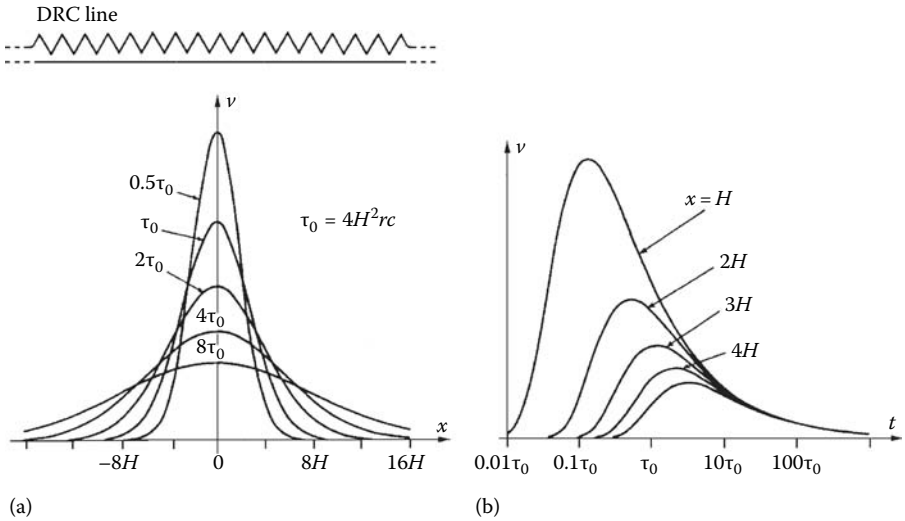


FIGURE 18.2 Effect of an impulse-like charge injection at $x = 0$. (a) Voltage distribution in subsequent time instants. (b) Voltage transients in different distances from the injection point.

The same result is represented in a different way in Figure 18.2b. The time dependence of the voltage is shown at the $x = H$, $x = 2H$, etc., spatial positions. As we move away from the $x = 0$ point, the maximum of the impulse appears increasingly later in time, and the originally sharp impulse is increasingly extended. This means that the RC line delays the input pulse and at the same time strongly spreads it. The RC line is dispersive.

18.1.1.1 Superposition

Equation 18.3 is homogeneous and linear. This means that any sum of the solutions is again a solution. Based on this fact, the problem of Figure 18.3a can be solved. At $t = 0$, the voltage distribution of the RC line is given by an arbitrary $U(x)$ function. For $t > 0$, the distribution can be calculated by dividing the $U(x)$ function into $\Delta\xi \to 0$ elementary slices. These may be considered to be individual Dirac-δ excitations and the responses given to them can be summarized by integration:

$$v(x,t) = \frac{1}{2\sqrt{\pi t/rc}} \int_{-\infty}^{\infty} U(\xi) \exp\left(-\frac{(x-\xi)^2}{4t/rc}\right) d\xi \tag{18.5}$$

Evaluating this equation for the special case of having $2U_0$ voltage on the $x < 0$ side at $t = 0$, while $x > 0$ is voltage less, results in

$$v(x,t) = \frac{2U_0}{2\sqrt{\pi t/rc}} \int_{-\infty}^{0} \exp\left(-\frac{(x-\xi)^2}{4t/rc}\right) d\xi = U_0 \operatorname{erfc}\left(\frac{x}{2\sqrt{t/rc}}\right) \tag{18.6}$$

where the integral of the Gauss function is notated by erfc(x), the complementary error function.* The originally abrupt voltage step is getting increasingly less steep with time (Figure 18.3b). In the middle at $x = 0$, the voltage remains U_0.

18.1.1.2 Semi-Infinite Line

Our next model is a bit closer to practice; the uniform RC line extends to $x \geq 0$ only. At $x = 0$ the port is characterized by the $V(t)$ voltage and the $I(t)$ current (Figure 18.4). If the line is relaxed and a $I(t) = \delta(t)$ current (a Dirac-δ current pulse) is forced to the port, a unit charge is introduced at $x = 0$. The result will

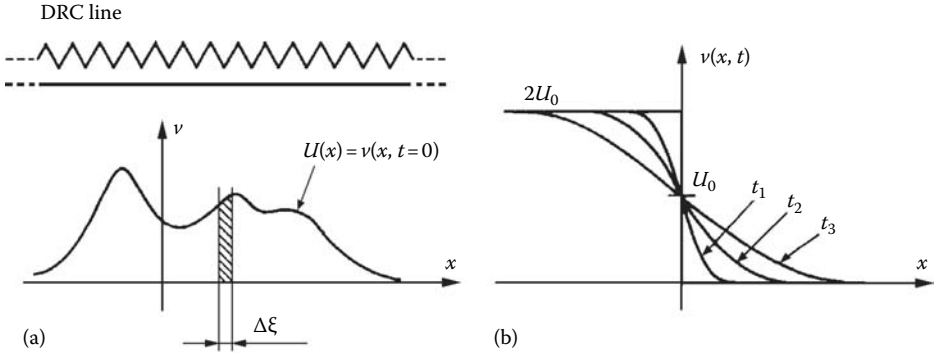


FIGURE 18.3 DRC line transients. (a) An arbitrary initial voltage distribution. (b) Solution for the initial step-function case.

* $\operatorname{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x \exp(-y^2)dy = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-y^2)dy$.

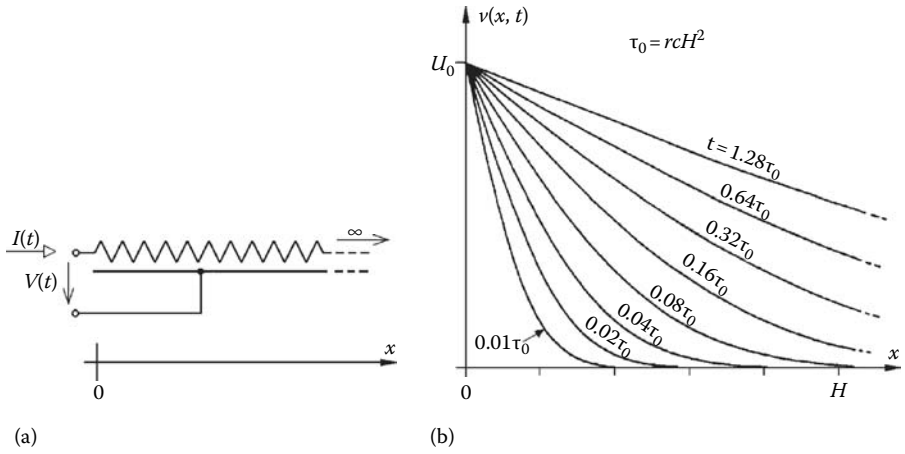


FIGURE 18.4 Semi-infinite uniform DRC line. (a) Notation. (b) Normalized solution for the initially relaxed line.

be similar to that of Figure 18.2a, but instead of symmetrical spreading the charge moves towards the positive x direction only. This means that a unit charge generates a twice-larger voltage wave

$$v(x,t) = \frac{1/c}{\sqrt{\pi t/rc}} \exp\left(-\frac{x^2}{4t/rc}\right) \tag{18.7}$$

Let us consider the case in which step function excitation is given to the port of Figure 18.4. At $t<0$, the port and the whole line are voltage free; at $t \geq 0$ a constant U_0 voltage is forced to the port. Comparing this situation with the problem of Figure 18.3b, it can be observed that the boundary conditions for the $x>0$ semi-infinite line are the same as in our current example, so that the solution must to be similar as well (see Figure 18.4b):

$$v(x,t) = U_0 \mathrm{erfc}\left(\frac{x}{2\sqrt{t/rc}}\right) \tag{18.8}$$

Applying at the $t=0$ instant an arbitrary $W(t)$ forced voltage excitation to the initially relaxed line, the response is given by the Duhamel integral as follows:

$$w(x,t) = \int_0^t \frac{\mathrm{d}W(\tau)}{\mathrm{d}\tau} \mathrm{erfc}\left(\frac{x}{2\sqrt{(t-\tau)/rc}}\right) \mathrm{d}\tau \tag{18.9}$$

18.1.1.3 Finite DRC Line

Let the DRC line of length L of Figure 18.5a be closed at $x=L$ with a short circuit. Let the line at $t<0$ be relaxed and assume a W $(t>0)$ voltage excitation at the $x=0$ port. Using the $w(x,t)$ voltage response of the semi-infinite line Equation 18.9, the response function for the short-terminated line of length L

$$\begin{aligned} v(x,t) &= \sum_{i=0}^{\infty} (-1)^i \cdot w\big(2iL + (-1)^i x, t\big) \\ &= \sum_{k=0}^{\infty} \big(w(2kL + x, t) - w(2kL + 2L - x, t)\big) \end{aligned} \tag{18.10}$$

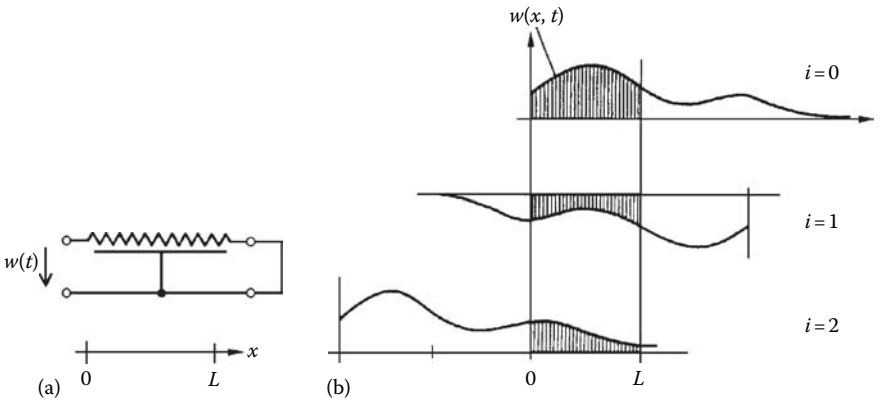


FIGURE 18.5 Finite-length uniform DRC line. (a) DRC line with short-circuit at $x = L$. (b) Visualization of the mirroring procedure.

This result is illustrated in Figure 18.5b. The $v(x, t)$ function is given as the sum of the shifted, negated, and mirrored replicas of the $w(x, t)$ function, so that it is a valid solution as well. The $x = L$ boundary condition is the short circuit $v(x = L, t) = 0$. The $i = 0$ and $i = 1$ functions are the same size with different signs at $x = L$, so they cancel each other. The same is true for $i = 2$ and 3, and so on. The $x = 0$ boundary condition $v = W(t)$ is fulfilled by the $i = 0$ function, while the further functions cancel each other in pairs (the $i = 1$ and 2, etc.).

The result can be interpreted as the response of the semi-infinite line being mirrored with negative sign on the short termination. In the case of Figure 18.5a, the termination on both the $x = 0$ and $x = L$ ends is assured by zero impedance short circuit; the resultant voltage function comes from the successive back and forth mirroring between these two ''mirrors.''

It is easy to understand that a termination with an open circuit results in mirroring without sign change. (At this termination the current equals to zero so that the dv/dx derivative equals to zero as well. This requirement is always fulfilled in the mirroring point summarizing the continuous incident function with its mirrored version.) According to this, the voltage on the open-terminated line of Figure 18.6a is

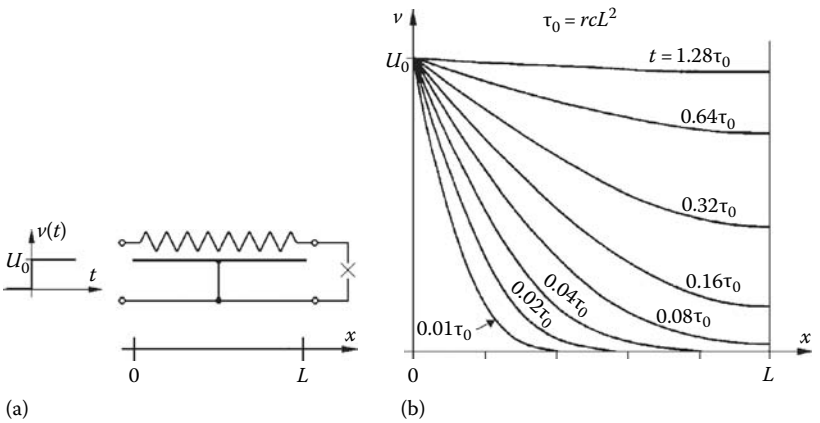


FIGURE 18.6 Finite-length uniform DRC line. (a) Open-circuit at the far end. (b) Normalized solution for the initially relaxed line.

$$v(x,t) = \sum_{k=0}^{\infty} (-1)^k (w(2kL + x, t) + w(2kL + 2L - x, t)) \tag{18.11}$$

which in the case of step function excitation with U_0 amplitude is

$$v(x,t) = U_0 \sum_{k=0}^{\infty} (-1)^k \left[\text{erfc}\left(\frac{2kL + x}{2\sqrt{t/rc}}\right) + \text{erfc}\left(\frac{2kL + 2L - x}{2\sqrt{t/rc}}\right)\right] \tag{18.12}$$

This function is given in Figure 18.6b for some time instants.

18.1.2 Solution in the Frequency Domain

To find the solution of the differential Equation 18.3, the following trial function can be used

$$v(x,t) = v \cdot \exp(j\omega t) \cdot \exp(\gamma x) \tag{18.13}$$

Substituting this function into (18.3) results in the following so-called dispersion equation:

$$\gamma = \sqrt{j\omega rc} = (1 + j)\frac{1}{\sqrt{2}}\sqrt{\omega rc} \tag{18.14}$$

This means that a wavelike solution exists as well. However, it is strongly collapsing because the real and imaginary parts of γ are always equal, which means that the attenuation on a path of λ wavelength is $\exp(-2\pi) \cong 1/535$.

The lossless DRC line can be considered to be a special telegraph line having neither serial inductance nor shunt conductance. The telegraph line theory can be conveniently used at the calculation of uniform DRC networks. The γ propagation constant and the $\mathbf{Z}_0$ characteristic impedance for the present case are

$$\gamma = \sqrt{\mathbf{s}rc}, \quad \mathbf{Z}_0 = \sqrt{\frac{r}{\mathbf{s}c}} \tag{18.15}$$

With these the two-port impedance parameters and chain parameters of an RC line of length L can be given as follows:

$$\mathbf{Z}_{ij} = \mathbf{Z}_0 \begin{bmatrix} \text{cth}\ \gamma L & 1/\text{sh}\ \gamma L \\ 1/\text{sh}\ \gamma L & \text{cth}\ \gamma L \end{bmatrix}, \quad \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \text{ch}\ \gamma L & \mathbf{Z}_0 \text{sh}\ \gamma L \\ \frac{1}{\mathbf{Z}_0}\text{sh}\ \gamma L & \text{ch}\ \gamma L \end{bmatrix} \tag{18.16}$$

If one of the ports is terminated by the impedance of $\mathbf{Z}_t$, the following impedance can be "seen" on the opposite port

$$\mathbf{Z}_{in} = \mathbf{Z}_0 \frac{\mathbf{Z}_t\ \text{ch}\ \gamma L + \mathbf{Z}_0\ \text{sh}\ \gamma L}{\mathbf{Z}_t\ \text{sh}\ \gamma L + \mathbf{Z}_0\ \text{ch}\ \gamma L} \tag{18.17}$$

18.1.3 Uniform, Lossy DRC Lines

In some cases, the DRC structure also has shunt conductance, which means that it is lossy. The value of this conductance for the unit length is notated by g. In such a case, without giving the details of the calculation, the $v(x,t)$ line voltage can be determined by the solution of the equation

$$\frac{\partial v}{\partial t} = \frac{1}{rc}\frac{\partial^2 v}{\partial x^2} - \frac{g}{c}v \tag{18.18}$$

The following forms of the characteristic impedance and the propagation constant can be used now in the frequency domain

$$\gamma = \sqrt{r(g + \mathbf{s}c)}, \quad \mathbf{Z}_0 = \sqrt{\frac{r}{g + \mathbf{s}c}} \tag{18.19}$$

It is an interesting fact that the charge carrier motion in the base region of homogeneously doped bipolar transistors can be described by formally similar equations, so that the intrinsic transients of the bipolar transistors can be exactly modeled by lossy DRC two-ports [5].

Example 18.1. Wiring delays

Neither the series resistance nor the stray capacitances of the interconnection leads of integrated circuits are negligible. As an example, in the case of a polysilicon line of 1 μm width $r \cong 50$ kΩ/mm, $c \cong 0.04$ pF/mm. This means that these wires should be considered to be DRC lines. The input logical levels appear on their output with a finite delay. Let us determine the delay of a wire of length L. From Equation 18.12,

$$v(L,t) = U_0 \sum_{k=0}^{\infty} 2(-1)^k \mathrm{erfc}((2k+1)\vartheta)$$

where

$$\vartheta = \frac{L}{2\sqrt{t/rc}}$$

The value of the summation over k will reach 0.9 at $\vartheta = 0.5$, so that the voltage at the end of the line will reach the 90% of U_0 after a time delay of

$$t_{\mathrm{delay}} \cong rcL^2 \tag{18.20}$$

Note that the time delay increases with the square of the length of the wire. In the case of $L = 1$ mm the time delay of this polysilicon wire is already 2 ns, which is more than the time delay of a CMOS logical gate. For lengthy wires ($>0.2 \div 0.5$ mm), metal wiring must be applied with its inherently small resistivity.

Example 18.2. Parasitic effects of IC resistors

In an IC amplifier stage the transistor is loaded with, e.g., $R = 10$ kΩ (Figure 18.7a). This resistor has been fabricated by the base diffusion, the sheet resistance is 200 Ω, and the parasitic capacitance is 125 pF/mm^2. The width of the resistor is 4 μm. Let us determine the impedance of the resistor in the 1 to 1000 MHz range.

The resistance can be realized in 4×200 μm size. The total parasitic capacitance is $C_p = 0.1$ pF. The impedance of the transistor-side port can be calculated according to Equation 18.17, considering that the opposite port is short-terminated, as

$$\mathbf{Z}_{\mathrm{port}} = \mathbf{Z}_0 \ \mathrm{th}\ \gamma L = \sqrt{\frac{r}{\mathbf{s}c}}\mathrm{th}\left(\sqrt{\mathbf{s}rc}L\right) = \sqrt{\frac{R}{\mathbf{s}C_p}}\mathrm{th}\sqrt{\mathbf{s}RC_p} \tag{18.21}$$

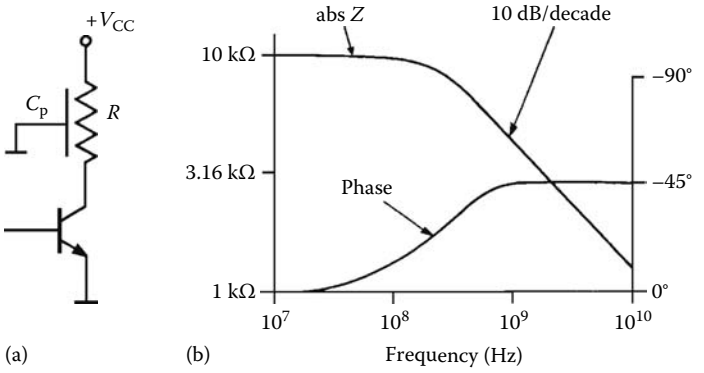


FIGURE 18.7 A simple IC amplifier stage. (a) The load resistor is in fact a DRC line. (b) The amplitude and phase plot of the load.

Using the $\mathbf{s} = j\omega$ substitution, along with the actual data, the amplitude and phase functions of Figure 18.7b can be obtained for the impedance. At 10 MHz, the phase shift caused by the parasitic capacitance is negligible, but at 100 MHz it is already considerable.

It is important to recognize that in the case of half size linewidths the size of the resistor will be only 2×100 μm, which results in one-fourth of the previous value in C_p. This means that the capacitance becomes disturbing only at four times larger frequencies.

Note in Figure 18.7b that the amplitude function shows a 10 dB/decade decay and the phase keeps to 45°, as if the load would be characterized by a "half pole." This 10 dB/decade frequency dependence often can be experienced at DRC lines.

18.2 Nonuniform Distributed RC Lines

In some cases, the capacitance and/or the resistance of the DRC line shows a spatial dependency. This happens if the width of the lead strip is modulated in order to reach some special effects (Figure 18.8a, tapered RC line). In case of a biased IC resistance, the capacitance changes along the length of the structure as well because of the voltage dependency of the junction capacitance. These structures are referred to as nonuniform DRC lines.

Let the spatially dependent resistance and capacitance pro unit length be notated by $r(x)$ and $c(x)$, respectively. The following equations can be given for the structure:

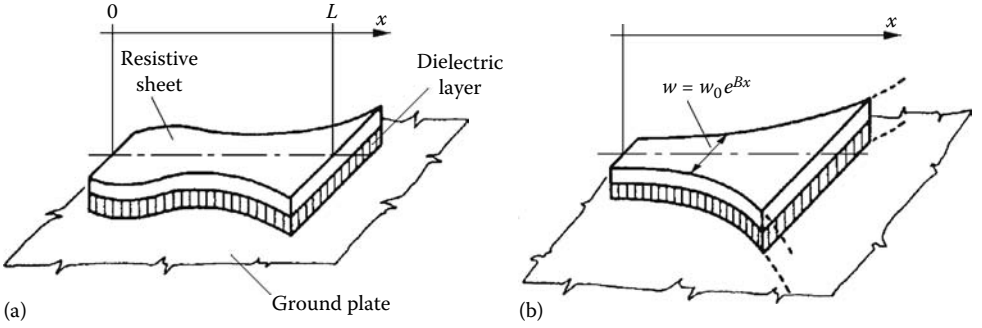


FIGURE 18.8 Nonuniform distributed RC lines. (a) Tapered line. (b) Exponentially tapered line.

$$\frac{\partial v}{\partial x} = -r(x)i, \quad \frac{\partial i}{\partial x} = -c(x)\frac{\partial v}{\partial t} \tag{18.22}$$

With these, the following differential equation can be written:

$$\frac{\partial v}{\partial t} = \frac{1}{c(x)}\frac{\partial}{\partial x}\left(\frac{1}{r(x)}\frac{\partial v}{\partial x}\right) \tag{18.23}$$

We can obtain a more convenient form if we consider as an independent variable (instead of the x spatial co-ordinate) the total ρ resistance related to a given reference point (e.g., to the $x = 0$ point), as follows:

$$\rho(x) = \int_0^x r(\xi)\mathrm{d}\xi, \quad r(x) = \frac{\partial \rho}{\partial x} \tag{18.24}$$

The variable defined this way can be considered as a kind of arc-length parameter. It has been introduced by Protonotarious and Wing [2]. With this new variable

$$\frac{\partial v}{\partial t} = \frac{1}{K(\rho)}\frac{\partial^2 v}{\partial \rho^2} \tag{18.25}$$

where

$$K(\rho) = K(\rho(x)) = \frac{c(x)}{r(x)} \tag{18.26}$$

The $K(\rho)$ function describes well the spatial parameter changes of the RC line; that is, the structure of the line. Therefore, the $K(\rho)$ function is called the **structure function.** Those DRC structures for which the $K(\rho)$ functions are the same are considered to be electrically equivalent.

Reference [2] uses the $\sigma(\rho)$ integral or cumulative version of the structure function:

$$\sigma(\rho) = \int_0^\rho K(\rho)\mathrm{d}\rho = \int_0^{x(\rho)} \frac{c(x)}{r(x)}\frac{\mathrm{d}\rho}{\mathrm{d}x}\mathrm{d}x = \int_0^{x(\rho)} c(x)\mathrm{d}x \tag{18.27}$$

This is the total capacitance related to the $x = 0$ point. This means that the cumulative structure function is the total capacitance versus total resistance map of the structure. An example of such a map is plotted in Figure 18.9.

The differential Equation 18.25 is homogeneous and linear; therefore, superposition can be used. Because this equation is of variable coefficient type, however, analytic solution can be expected only rarely. Such a case is that of the $K = K_0/\rho^4$ structure function for which

$$v(\rho, t) = \mathrm{const}\frac{1}{t^{3/2}}\exp\left(-\frac{K_0}{4\rho^2 t}\right) \tag{18.28}$$

Another form of Equation 18.25 is also known. To obtain this form, we should turn to the **s** domain with

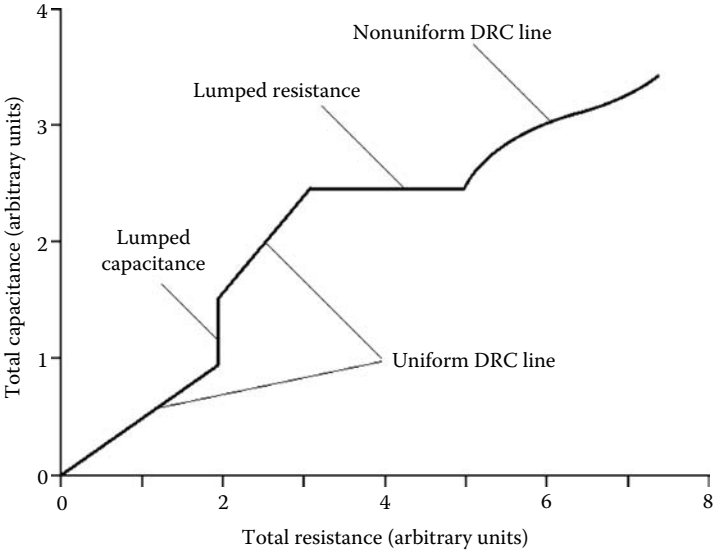


FIGURE 18.9 Cumulative structure function.

$$\mathbf{s}v = \frac{1}{K(\rho)}\frac{\partial^2 v}{\partial \rho^2} \tag{18.29}$$

Let us introduce the following new variable:

$$\mathbf{Z}(\rho) = \frac{v(\mathbf{s},\rho)}{i(\mathbf{s},\rho)} = \frac{v}{-\frac{1}{r}\frac{\partial v}{\partial x}} = -\frac{v}{\frac{\partial v}{\partial \rho}} \tag{18.30}$$

This variable is in fact the impedance of the line at the location of ρ. After rearrangements, the

$$\frac{d\mathbf{Z}}{d\rho} = 1 + \mathbf{s}K(\rho)\mathbf{Z}^2 \tag{18.31}$$

equation can be obtained. This is called the Riccati differential equation. In the case of a known $K(\rho)$ structure function, the one-port impedance of the nonuniform line can be determined from it by integration. In some cases, even the analytic solution is known. Such a case is the exponentially tapered line of Figure 18.8b, for which

$$r(x) = \frac{R_s}{w_0}\exp(-Bx), \quad c(x) = C_p w_0 \exp(Bx), \quad K(\rho) = \frac{R_s C_p}{B^2}\frac{1}{\rho^2} \tag{18.32}$$

where R_s is the sheet resistance of the structure, C_p is the capacitance per unit area, and ρ is related to the point in the infinity. If the port in the infinity is shorted, the impedance of the location of ρ is

$$\mathbf{Z}(\mathbf{s}) = \frac{\sqrt{1 + 4\mathbf{s}R_s C_p/B^2} - 1}{2\mathbf{s}R_s C_p/B^2}\rho \tag{18.33}$$

In other cases, numerical integration of Equation 18.31 leads to the solution.

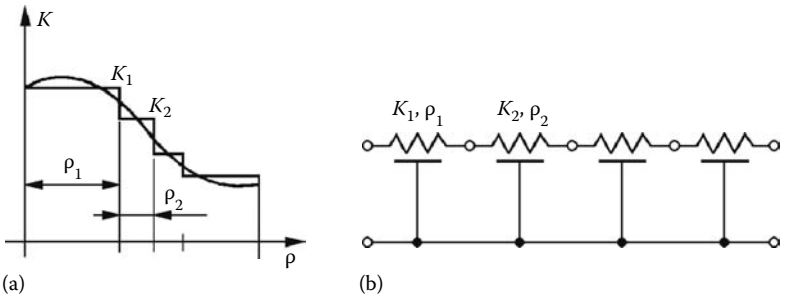


FIGURE 18.10 Approximation with concatenated uniform sections. (a) Stepwise approximation of the structure function. (b) Approximate model.

18.2.1 Approximation with Concatenated Uniform Sections

The following model can be used for approximate calculation of nonuniform structures. We split the structure function into sections (see Figure 18.10a) and use stepwise approximation. Inside the sections, $K =$ constant, so that they are uniform sections. Concatenating them according to Figure 18.10b an approximate model is obtained.

In the frequency domain, the overall parameters of the resultant two-port can be easily calculated. The chain parameter matrices of the concatenated sections have to be multiplied in the appropriate order. The time domain behavior can be calculated by inverse Laplace transformation.

18.2.2 Asymptotic Approximation for Large "s"

The chain parameters of a nonuniform DRC line can be approximately written as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cong \frac{1}{2} \exp\left(\Delta\sqrt{\mathbf{s}}\right) \begin{bmatrix} \lambda & \mu/\sqrt{\mathbf{s}} \\ \sqrt{\mathbf{s}}/\mu & 1/\lambda \end{bmatrix} \tag{18.34}$$

where

$$\Delta = \int_0^{R_0} \sqrt{K(\rho)}\,d\rho, \quad R_0 = \rho(L), \quad \lambda = \left(\frac{K(R_0)}{K(0)}\right)^{1/4}, \quad \mu = (K(R_0)K(0))^{-1/4}$$

This approximation is valid for large **s** values and for sufficiently smooth function $K(\rho)$ [2].

18.2.3 Lumped Element Approximation

Distributed RC networks can be approximated by lumped element RC networks as well. The case of a lossless line is depicted in Figure 18.11a. The element values can be determined in either of the following two ways.

1. In the case of a known structure spatial discretization can be used. The nonuniform line must be split into sections of width h (see Figure 18.11b). A node of the network is associated with the middle of each section. The total capacitance of the section must be calculated, and this gives the value of the lumped capacitance connected to the node of the section. The resistance between the middle points of two adjacent sections must be calculated, and this has to be connected

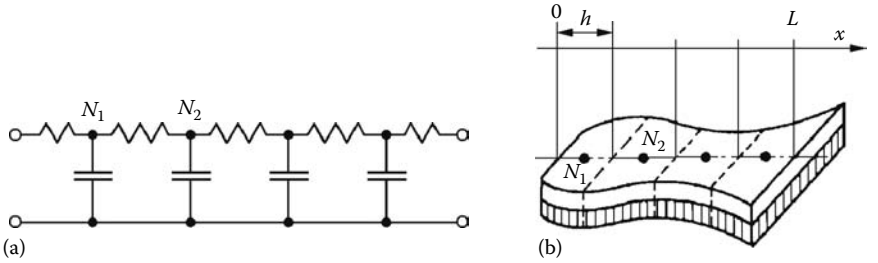


FIGURE 18.11 Lumped element approximation. (a) Network model. (b) The line split into sections.

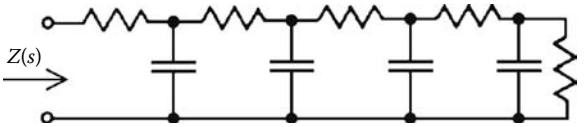


FIGURE 18.12 Cauer equivalent circuit (ladder structure).

between the nodes of the appropriate sections. It is obvious that the accuracy can be increased by decreasing h. The price is the increasing number of lumped elements. With $h \to 0$, we obtain the exact model.

2. When we know the impedance function, we can build the model using the pole-zero pattern of the network. For example, let us investigate a uniform RC line of finite length L, short-circuited at the far end. The corresponding impedance expression, according to Equation 18.21, is

$$\mathbf{Z}(\mathbf{s}) = \frac{1}{\sqrt{\mathbf{s}K_0}} \text{th}\, R_0\sqrt{\mathbf{s}K_0} \tag{18.35}$$

where $K_0 = c/r$, $R_0 = r \cdot L$. This function has poles and zeroes on the negative real axis in an infinite number. The zero and pole frequencies are

$$\sigma_{zi} = (2i)^2 \frac{\pi^2}{4} \frac{1}{R_0^2 K_0}, \quad \sigma_{pi} = (2i+1)^2 \frac{\pi^2}{4} \frac{1}{R_0^2 K_0} \tag{18.36}$$

where $i = 1, 2, \ldots, \infty$. Neglecting all the poles and zeroes situated well above the frequency range of interest and eliminating successively the remainder poles and zeroes from Equation 18.37 impedance function the element values of the ladder network in Figure 18.12 (Cauer equivalent) can be obtained from

$$\mathbf{Z}(\mathbf{s}) = R_0 \frac{\prod_{i=1}^{z} (1 + \mathbf{s}/\sigma_{zi})}{\prod_{i=1}^{p} (1 + \mathbf{s}/\sigma_{pi})} \tag{18.37}$$

18.3 Infinite-Length RC Lines

It was demonstrated earlier in the chapter that the DRC network can be described with the help of the pole-zero set of its impedance, as in the case of lumped element circuits. However, the number of these

poles and zeroes is infinite. The infinite-length DRC lines generally do not have this property. For this network category the exact description by discrete poles and zeroes is not possible.

For example, let us consider an infinitely long uniform DRC line. Its input impedance is the characteristic impedance:

$$\mathbf{Z}(\mathbf{s}) = \sqrt{\frac{r}{\mathbf{s}c}} \tag{18.38}$$

Evidently, this impedance function does not have poles and zeroes on the negative σ axis. This is the general case for a more complex, nonuniform distributed network if the length of the structure is infinite. The characteristic feature of these impedance functions is that $\sqrt{j\omega}$ factors appear in them. This is why in the logarithmic amplitude vs. frequency diagram (Bode plot) regions with 10 dB/decade slope appear, as pointed out in Ref. [1].

This section provides a generalization of the pole and zero notions and the time-constant representation in order to make them suitable to describe infinitely long distributed one-ports as well.

Before developing new ideas let us summarize the normal, well-known descriptions of a lumped element RC one-port. The port impedance of such a circuit is described by a rational function with real coefficients, as

$$\mathbf{Z}(\mathbf{s}) = R_0 \frac{(1+\mathbf{s}/\sigma_{z1})(1+\mathbf{s}/\sigma_{z2})\cdots(1+\mathbf{s}/\sigma_{zn-1})}{(1+\mathbf{s}/\sigma_{p1})(1+\mathbf{s}/\sigma_{p2})\cdots(1+\mathbf{s}/\sigma_{pn})} \tag{18.39}$$

where R_0 is the overall resistance, σ_p are the poles, and σ_z are the zeroes (as absolute values). The pole and zero values, together with the overall resistance value, hold all the information about the one-port impedance. Thus, an unambiguous representation of this impedance is given by a set of pole and zero values, and an overall resistance value. This will be called the "pole–zero representation."

Expression 18.39 can be rearranged as

$$\mathbf{Z}(\mathbf{s}) = \sum_{i=1}^{n} \frac{R_i}{1+\mathbf{s}/\sigma_{pi}} = \sum_{i=1}^{n} \frac{R_i}{1+\mathbf{s}\tau_i} \tag{18.40}$$

where

$$\tau_i = 1/\sigma_{pi} \tag{18.41}$$

which corresponds directly to the $v(t)$ voltage response for a step-function current excitation:

$$v(t) = \sum_{i=1}^{n} R_i(1-\exp(-t/\tau_i)) \tag{18.42}$$

In this case, the impedance is described in terms of the τ_i time-constants of its response and of the R_i magnitudes related to it. This will be called the time-constant representation.

18.3.1 Generalization of the Time Constant Representation*

A lumped element one-port can be represented by a finite number of τ time-constants and R magnitudes. A graphic representation of this is demonstrated in Figure 18.13. Each line of this plot represents a time constant, and the height of the line is proportional to the magnitude. This figure can be regarded as some

* Portions reprinted with permission from Székely, V., *IEEE Trans. Circuits Syst.*, 38, 711, 1991. © 1991 IEEE.

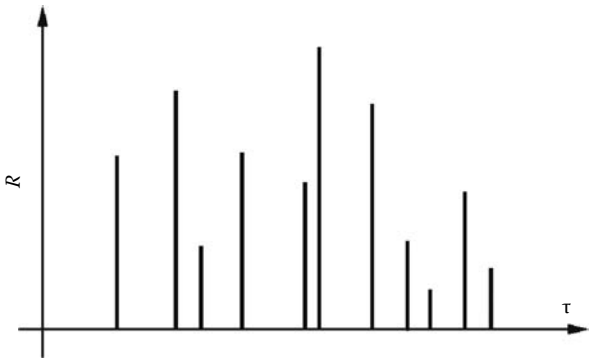


FIGURE 18.13 A lumped element one-port can be represented with a discrete set of time constants. (From Székely, V., *IEEE Trans. Circuits Syst.*, 38, 711, 1991. With permission.)

kind of a spectrum, the spectrum of the time constants that appeared in the step-function response of the network.

The port-impedance of a lumped element network has discrete ''spectrum lines'' in finite number. An infinite distributed network has no discrete lines, but it can be described with the help of a continuous time constant spectrum. The physical meaning of this idea is that in a general response any time constant can occur in some amount, some density, so that a density spectrum may suitably represent it.

We define the spectrum function by first introducing a new, logarithmic variable for the time and the time constants:

$$z = \ln t, \quad \zeta = \ln \tau \tag{18.43}$$

Let us consider a DRC one-port, the response of which contains numerous exponentials having different time constants and magnitudes. The ''time constant density'' is defined as

$$R(\zeta) = \lim_{\Delta\zeta \to 0} \frac{\text{sum of magnitudes between } \zeta \text{ and } \zeta + \Delta\zeta}{\Delta\zeta} \tag{18.44}$$

From this definition directly follows the fact that the step-function response can be composed from the time-constant density:

$$v(t) = \int_{-\infty}^{\infty} R(\zeta)[1 - \exp(-t/\exp(\zeta))]\mathrm{d}\zeta \tag{18.45}$$

This integral is obviously the generalization of the summation in Equation 18.42. If the $R(\zeta)$ density function consists of discrete lines (Dirac-δ pulses), Equation 18.42 is given back.

Using the logarithmic time variable in the integral of Equation 18.45

$$v(z) = \int_{-\infty}^{\infty} R(\zeta)[1 - \exp(-\exp(z - \zeta))]\mathrm{d}\zeta \tag{18.46}$$

a convolution-type differential equation is obtained. Differentiating both sides with respect to z, we obtain

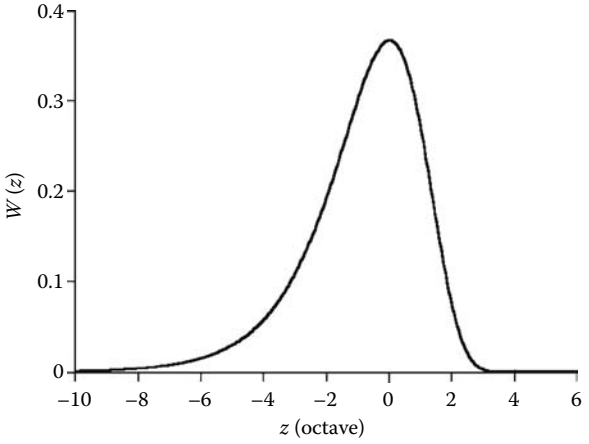


FIGURE 18.14 The shape of the $W(z)$ function. (From Székely, V., *IEEE Trans. Circuits Syst.*, 38, 711, 1991. With permission.)

$$\frac{\mathrm{d}}{\mathrm{d}z}\nu(z) = R(z) \otimes W(z) \tag{18.47}$$

where

$$W(z) = \exp(z - \exp(z)) \tag{18.48}$$

is a fixed weighting function with shape depicted in Figure 18.14, and $\otimes$ is the symbol of the convolution operation [3].

It can be proved that the area under the function $W(z)$ is equal to unity

$$\int_{-\infty}^{\infty} W(z)\mathrm{d}z = 1 \tag{18.49}$$

This means that

$$\int_{-\infty}^{\infty} R(z)\mathrm{d}z = \nu(t \to \infty) = R_0 \tag{18.50}$$

where R_0 is the zero-frequency value of the impedance. In other words, the finite step-function response guarantees that the time-constant density has finite integral.

18.3.2 Generalization of the Pole-Zero Representation

The task is now to substitute the pole-zero pattern of the lumped network with a continuous (eventually excepting some discrete points) function to describe the general distributed parameter network.

As emphasized above, the Bode plot of a distributed parameter network frequently shows regions with a 10 dB/decade slope. Figure 18.15 presents such an amplitude diagram. Using poles and zeroes, we can only approximate this behavior. If we place to point ω_1 a pole, the Bode plot turns to the decay of 20 dB/decade, which is too steep. If a zero is placed, the diagram returns to the zero-slope. However, if we alternate poles and zeroes in a manner that the mean value of the slope should give the prescribed one,

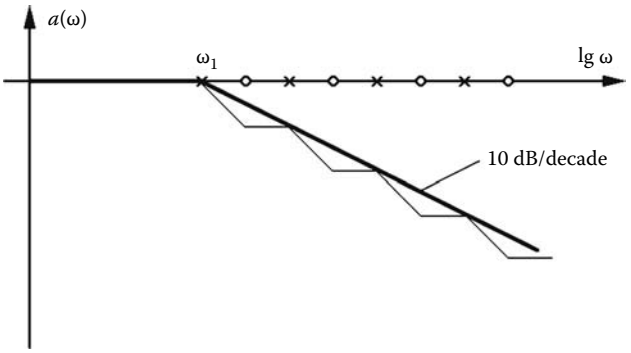


FIGURE 18.15 The 10 dB/decade decay of a DRC line amplitude plot can be approximated with an alternative sequence of poles and zeroes. (From Székely, V., *IEEE Trans. Circuits Syst.*, 38, 711, 1991. With permission.)

then any slope can be approximated. (For the previously-mentioned case, if the zeroes are situated exactly midway between the adjacent poles, then the mean slope is 10 dB/decade.) The suitability of the approximation depends on the density of poles and zeroes and can be improved by increasing the density. In this case, the network-specific information is not carried by the number of poles and zeroes (their number tends to infinity), but by the relative position of the zeroes between the adjacent poles.

An alternative interpretation is also possible. The pair of a neighboring pole and zero constitutes a dipole. The ''intensity'' of that dipole depends on the distance between the pole and the zero. If they coincide and cancel each other, then the intensity is equal to zero. If the zero is situated at the maximal distance from the pole (i.e., it is at the next pole), the intensity reaches its maximal value. We choose this to be the unity.

For later convenience, we turn to a logarithmic variable on the negative σ-axis:

$$\Sigma = \ln(-\sigma) \tag{18.51}$$

Let us investigate a $\Delta\Sigma$ interval of the logarithmic Σ-axis bounded by two adjacent poles. The distance between the left-hand pole and the inner zero is $\delta\Sigma$. Now, suppose that the density of the poles tends to infinity; i.e., $\Delta\Sigma$ becomes infinitely small. In this case the ''dipole intensity function'' is

$$I_d(\Sigma) = \lim_{\Delta\Sigma \to 0} \frac{\delta\Sigma}{\Delta\Sigma} \tag{18.52}$$

Considering that the poles and zeros of an RC port-impedance alternate, it follows that $0 \leq I_d \leq 1$. For an infinite, distributed RC two-pole the dipole intensity generally has regions in which the I_d value is between 0 and 1. For example, if the Bode plot shows a slope of 10 dB/decade, the value of I_d equals 0.5. This occurs in the case of an infinite, uniform RC line. For discrete circuits, the I_d function has only two possible values: 0 or 1.

18.3.3 Relations among $R(\zeta)$, $I_d(\Sigma)$, and the Impedance Function

Obviously, we needed one-to-one relations between the time constant density or the dipole intensity representation and the impedance expression of the one-port. (For the lumped element case Equations 18.39 and 18.40 give these correspondences).

Rather simple relations exist among the $R(\zeta)$, $I_d(\Sigma)$, and the $\mathbf{Z}(\mathbf{s})$ impedance function (see below). An interesting feature of these relations is a striking mathematical symmetry: the same expression couples

the time constant density to the impedance and the dipole intensity to the logarithmic impedance. The detailed proofs of the relations presented here are given in Ref. [4].

If the $\mathbf{Z}(\mathbf{s})$ complex impedance function is known, $R(\zeta)$ or $I_d(\Sigma)$ can be calculated as*

$$R(\zeta) = \frac{1}{\pi} \operatorname{Im} \mathbf{Z}(\mathbf{s} = -\exp(-\zeta)) \tag{18.53}$$

$$I_d(\Sigma) = \frac{1}{\pi} \operatorname{Im} (\ln \mathbf{Z}(\mathbf{s} = -\exp(\Sigma))) \tag{18.54}$$

If the $R(\zeta)$ or $I_d(\Sigma)$ function is known

$$\mathbf{Z}(\mathbf{S}) = R_0 - \int_{-\infty}^{\infty} R(-x) \frac{\exp(\mathbf{S} - x)}{1 + \exp(\mathbf{S} - x)} dx \tag{18.55}$$

$$\ln \mathbf{Z}(\mathbf{S}) = \ln R_0 - \int_{-\infty}^{\infty} I_d(x) \frac{\exp(\mathbf{S} - x)}{1 + \exp(\mathbf{S} - x)} dx \tag{18.56}$$

where $\mathbf{S}$ is the complex-valued logarithm of the complex frequency:

$$\mathbf{S} = \ln \mathbf{s} \tag{18.57}$$

Using the integral Equations 18.55 and 18.56, however, we must keep at least one from the two conditions:

1. $\mathbf{s}$ is not located on the negative real axis.
2. If $\mathbf{s}$ is located on the negative real axis then, at this point, and in a $\varepsilon \to 0$ neighborhood, $R(\zeta)$ or $I_d(\Sigma)$ must be equal to 0.

Note that Equations 18.53 through 18.56 are closely related to the Cauchy integral formula of the complex function theory. Substituting Equation 18.53 into 18.55 and exploiting some inherent properties of the RC impedance functions after some mathematics, the Cauchy integral results. The same is true for Equations 18.54 and 18.56.

An important feature of the transformations of Equations 18.53 and 18.55 is that they are linear. This means that the $\mathbf{Z}(\mathbf{s}) \times R(\zeta)$ transformation and the summation are interchangeable.

18.3.4 Practical Calculation of the $R(\zeta)$ Function†

Equation 18.53 suggests that only the $j\omega$ imaginary frequency has to be replaced by the $\mathbf{s} = -\exp(-z)$ complex frequency, and then the imaginary part of the calculated complex response multiplied by $1/\pi$ provides the time-constant spectrum.

However, the procedure is not as simple as that because of the use of Equation 18.53. This equation requires a great amount of caution. As the equation shows, the imaginary part of the $\mathbf{Z}$ impedance has to be calculated along the negative real axis of the complex plane. Along this axis, singularities usually lie: the poles of the network equation of lumped circuits or some singular lines in the case of distributed

* For Equation 12.53 $\mathbf{Z}(\mathbf{s} \to \infty) = 0$ is supposed.

† Portions reprinted, with permission, from Székely, V. and Rencz, M., *IEEE Trans. Compon. Pack. Technol.*, 23(3), 587, 2000. © 2000 IEEE.

systems. These singularities can prevent the use of Equation 18.53 for the calculation of the time-constant spectrum.

We can overcome these difficulties by adapting an approximate solution. In order to walk around the ''dangerous'' area, we have to avoid following the negative real axis. A line that is appropriately close to this axis might be used instead [9], like

$$\mathbf{s} = -(\cos\delta + j\sin\delta)\exp(-z) \tag{18.58}$$

Obviously, the δ angle has to be very small, not more than 2°–5°. Even if this angle is small, an error is introduced into the calculation. It can be proven that the calculated $R_C(z)$ time-constant spectrum can be expressed with the exact one by the following convolution equation:

$$R_C(z) = \frac{\pi - \delta}{\pi} R(z) \otimes e_r(z) \tag{18.59}$$

where

$$e_r(z) = \frac{1}{\pi - \delta} \frac{\sin\delta \exp(-z)}{1 - 2\cos\delta\exp(-z) + \exp(-2z)} \tag{18.60}$$

This function is a narrow pulse of unity area. The error of the calculation is represented by this function. Diminishing δ the $e_r(z)$ function becomes narrower and narrower. Thus, any accuracy requirement can be fulfilled by choosing an appropriately small δ angle. The half-value width, which is a measure of the resolution, is given by

$$\Delta_e = 2\ln\left(2 - \cos\delta + \sqrt{(2-\cos\delta)^2 - 1}\right) \cong 2\delta \tag{18.61}$$

If, for example, $\delta = 2°$, then the resolution is 0.1 octave, which means that two poles can be distinguished if the ratio between their frequencies is greater than 1.072.

Obviously, the calculated result has to be corrected with the factor of $\pi/(\pi - \delta)$.

Example 18.3

A tight analogy exists between electrical conductance and heat flow. Heat-conducting media, which can be characterized with distributed heat resistance and distributed heat capacitance, behave similarly to the electrical DRC networks. The analogous quantities are as follows:

Voltage → Temperature
Current → Power flow
Resistance → Thermal resistance
Capacitance → Heat capacitance

In the simplest model law, 1 V voltage corresponds to 1°C, 1 A current to 1 W power, etc., but different mapping can be applied as well.

The described analogy means that the tool set that is used to treat DRC networks can be applied to the calculation of heat-flow problems as well. This fact provides a direct way to calculate time-constant spectra in thermal field solver programs. These are thermal simulation tools suitable to solve the model equations in the **s**-domain. By using the substitution of Equation 18.58, some of these programs calculate directly the thermal time-constant spectrum of different structures [9]. As an example, a transistor package, presented in Figure 18.16a, was simulated. The time-constant spectrum calculated by the field

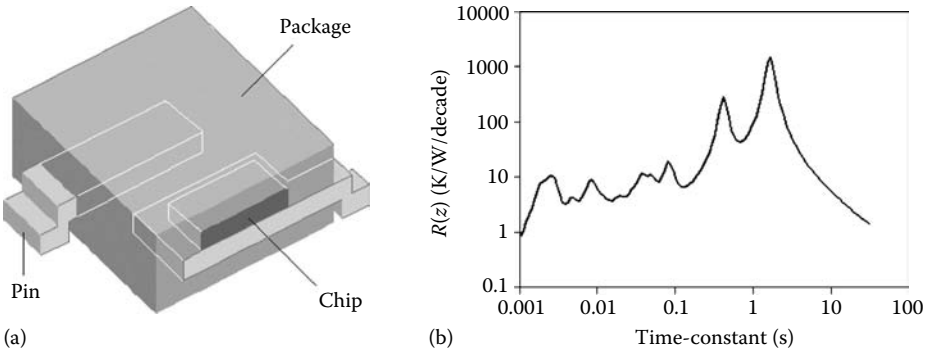


FIGURE 18.16 A transistor package and its time-constant spectrum. (a) Simulated transistor package. (b) Time-constant spectrum calculated by the field solver. (From Székely, V. and Rencz, M., *IEEE Trans. Compon. Pack. Technol.*, 23, 587, 2000. With permission.)

solver is plotted in Figure 18.16b. It is clearly visible that besides the two dominant time-constants a large number of further time-constants appear in the spectrum.

18.4 Inverse Problem for Distributed RC Circuits

Equation 18.47 offers a direct way to determine the time-constant density from the (measured or calculated) response function, which means that it is a method for the identification of RC one-ports. Using Equation 18.47 for a measured time domain response function, the time-constant density of the one-port impedance can be determined. By using this method, equivalent circuits can be constructed easily. This possibility is of considerable practical importance [8]; however, only approximate results can be obtained because the calculation leads to the inverse operation of the convolution. This operation can be done only approximately.

A possibility exists for identification in the frequency domain as well. Introducing the $\Omega = \ln \omega$ notation for the frequency axis, convolution-type equations can be found [8] between the $R(\zeta)$ time-constant density and the $\mathbf{Z}(\omega)$ complex impedance:

$$-\frac{\mathrm{d}}{\mathrm{d}\Omega}\mathrm{Re}\,\mathbf{Z}(\Omega) = R(\zeta = -\Omega) \otimes W_{\mathrm{R}}(\Omega) \tag{18.62}$$

$$-\mathrm{Im}\,\mathbf{Z}(\Omega) = R(\zeta = -\Omega) \otimes W_{\mathrm{I}}(\Omega) \tag{18.63}$$

where the weight-functions are

$$W_{\mathrm{R}}(\Omega) = \frac{2 \cdot \exp(2\ \Omega)}{(1 + \exp(2\ \Omega))^2} \tag{18.64}$$

$$W_{\mathrm{I}}(\Omega) = \frac{\exp(\Omega)}{1 + \exp(2\ \Omega)} \tag{18.65}$$

Moreover, a direct convolution relation exists between the Bode diagram of the impedance and the dipole intensity. Considering the Bode amplitude and phase diagrams, i.e., by using $\ln(\mathbf{Z}(\Omega)) = \ln \mathrm{abs}(\mathbf{Z}(\Omega)) + j \cdot \mathrm{arcus}(\mathbf{Z}(\Omega))$, we obtain

$$-\frac{\mathrm{d}}{\mathrm{d}\Omega}\ln \mathrm{abs}(\mathbf{Z}(\Omega)) = I_{\mathrm{d}}(\Omega) \otimes W_{\mathrm{R}}(\Omega) \tag{18.66}$$

$$-\text{arcus}(\mathbf{Z}(\Omega)) = I_d(\Omega) \otimes W_I(\Omega) \tag{18.67}$$

These equations may also be used for identification.

18.4.1 Network Identification by Deconvolution

All the relations between the $R(\zeta)$ time-constant spectrum and the different network responses are of convolution type Equations 18.47, 18.62, and 18.63. Knowing some kind of network responses the inverse operation of the convolution: the deconvolution leads to the $R(\zeta)$ function. The same is true for Equations 18.66 and 18.67 of the $I_d(\Sigma)$ dipole intensity. This means that the problem of identification of DRC networks is reduced to a deconvolution step. This method, called network identification by deconvolution (NID), is discussed in detail in Ref. [8], together with the appropriate deconvolution methods.

An important fact is that if we know the time response or only the real or the imaginary part of the frequency response, the network can be completely identified. (Noise effects produce practical limits—see later.)

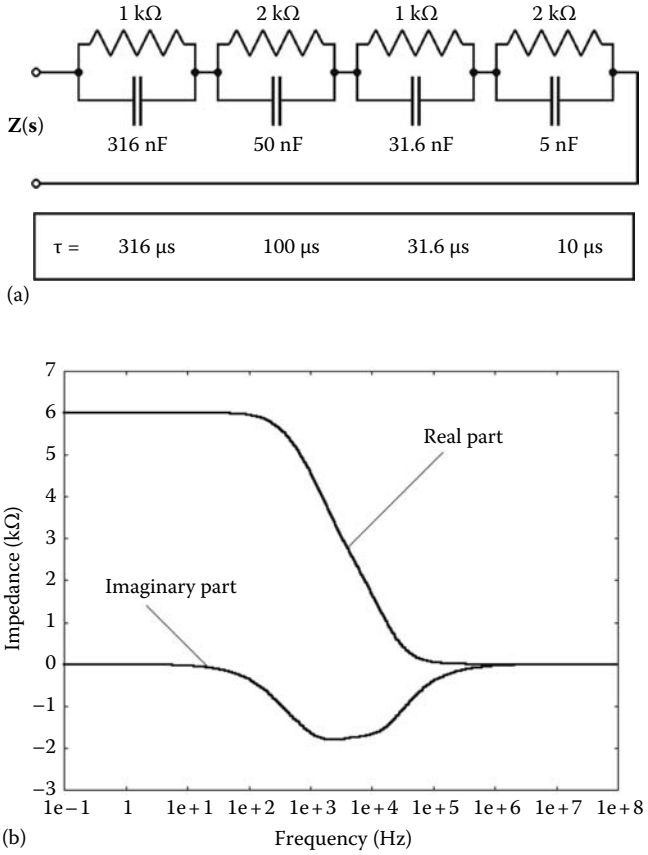


FIGURE 18.17 RC ladder and the frequency response of the $\mathbf{Z}(\omega)$ impedance. (a) Investigated RC network. (b) Real and the imaginary parts of the impedence. (From Székely, V., *IEEE Trans. Circuits Syst.—I. Theory Appl.*, 45, 244, 1998. With permission.)

Example 18.4

For the sake of simplicity, a "lumped circuit problem" will be discussed first. The investigated RC network is given in Figure 18.17a. We have calculated the frequency response of the **Z**(**s**) port-impedance of this circuit by using a standard circuit-simulator program, with a 40 point/frequency-decade resolution. Both the real and the imaginary parts of this impedance are plotted in Figure 18.17b.

In order to apply Equation 18.62, the derivative of the real part was calculated numerically. The result is shown in Figure 18.18a. In the next step, this function was deconvolved by the $W_R(\Omega)$ function. The result is plotted in Figure 18.18b. This function is the approximate time-constant density of the network. We expect that this function depicts the pole-pattern of the circuit. This is in fact obtained: the four peaks of the function are lying at $f = 497.7$, 1,585, 4,908, and 15,850 Hz. These

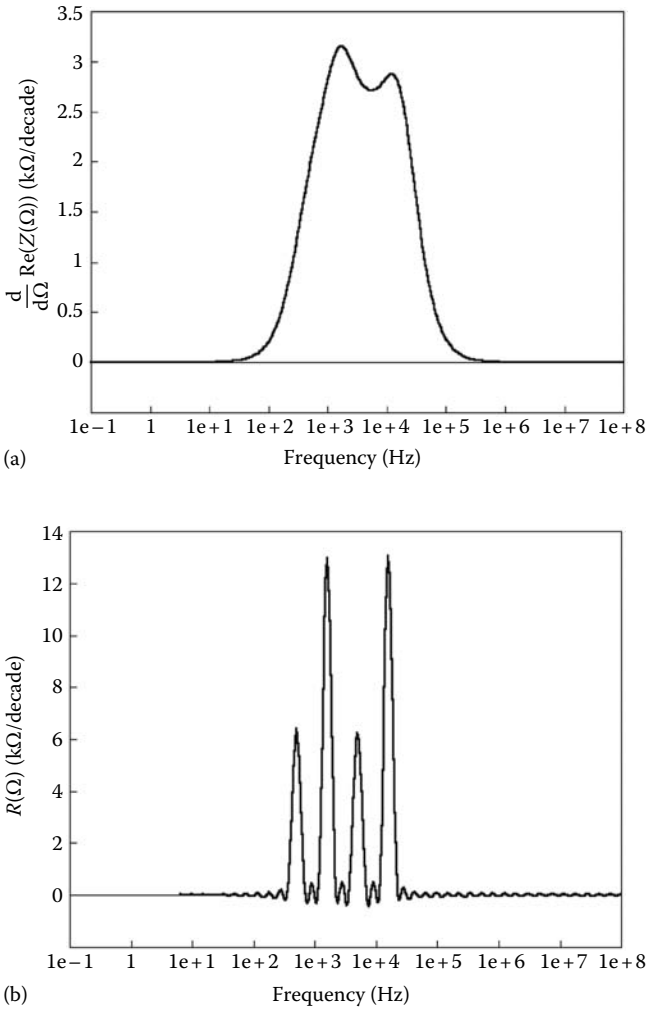


FIGURE 18.18 Identification steps. (a) Derivative of Real(**Z**). (b) The identified time-constant spectrum. (From Székely, V., *IEEE Trans. Circuits Syst.—I. Theory Appl.*, 45, 244, 1998. With permission.)

values correspond to the time constants of 320, 100.4, 32.43, and 10.04 μs, respectively. The ratios of the peak areas are about 1:2:1:2. These data agree well with the actual parameters of the circuit in Figure 18.17a.*

Notice that the noise corrupting the $\mathbf{Z}(\omega)$ function considerably affects the result of the identification. In order to reach 1 octave resolution of $R(\zeta)$ along the frequency axis, about 68 dB noise separation is needed in $\mathbf{Z}(\omega)$. Detailed discussion of the noise effects on the identification can be found in Ref. [8].

Example 18.5

As a second example, let us discuss the thermal identification of a semiconductor package + heat sink structure. The analogy between electrical current and heat flow introduced in Example 18.3 will be applied again.

Between a semiconductor chip and its ambient a complex distributed thermal structure exists consisting of many elements. The main parts of it are the chip itself, the soldering, the package, mounting to the heat sink, the heat sink itself, and the ambience. This is obviously a distributed thermal RC network, the input-port of which is the top surface of the chip and the far end is the ambience (the world). Thus, the structure can be considered practically infinite. This means that we have to examine a nonuniform infinite-length DRC network.

Investigations in the time domain require recording the thermal step-response of the system. A thermal test chip can be used for this purpose, containing appropriate heating elements that can assure step-function power excitation. The temperature rise is measured by the forward voltage change of a pn junction integrated into the test chip as well. This is the thermal response function. Such a thermal response is plotted in Figure 18.19a. The time range of the measurement is strikingly wide: 9 decades, from 10 μs to some thousand s. This is indispensable since the thermal time constants of the heat-flow structure vary over a wide range.

According to Equation 18.47, after numerical derivation of the step-response and by a consecutive deconvolution, the time constant density function $R(z)$ can be obtained (see Figure 18.19b). Because of the quantization noise and measuring error the deconvolution operation can be done only approximately with a $1 \div 1.5$ octave resolution. A suitable algorithm is discussed in Ref. [7]. Figure 18.19b illustrates that, in the 100 μs to 10 s interval, time constants spread over a relative wide range. This refers to the distributed structure of the chip and the package. At $\tau \approx 1000$ s, a relatively sharp, distinct time constant appears. This can be identified as originating from the heat capacitance of the whole heat sink and the heat sink-ambience thermal resistance.

Splitting the resultant time constant spectrum into $\Delta\tau$ time slots, each of these slots can be approximated by a Dirac-δ spectrum line proportional in height to the appropriate slot area. These give the data of a lumped element approximation according to Equation 18.40. Now, the equivalent circuit of the heat-flow structure can be generated either in Foster or in Cauer normal form.

Using the Cauer-approximation of the DRC line we can calculate the approximate $K(\rho)$ and $\sigma(\rho)$ structure functions. From these functions, the heat-conducting cross-section areas, the heat flow path length, etc. can be derived. This means that geometric and physical data of the heat-flow structure can be extracted and checked with the help of an electrical measurement. The structure function calculated from the measurement results of Figure 18.19 is plotted in Figure 18.20. It is easy to read out, e.g., the C_{hs} heat capacitance of the heat sink. For more details, see Refs. [6,10].

---

* Example reprinted with permission from Székely, V., *IEEE Trans. Circuits Syst.—I. Theory Appl.*, 45(3), 244, 1998. © 1998 IEEE.

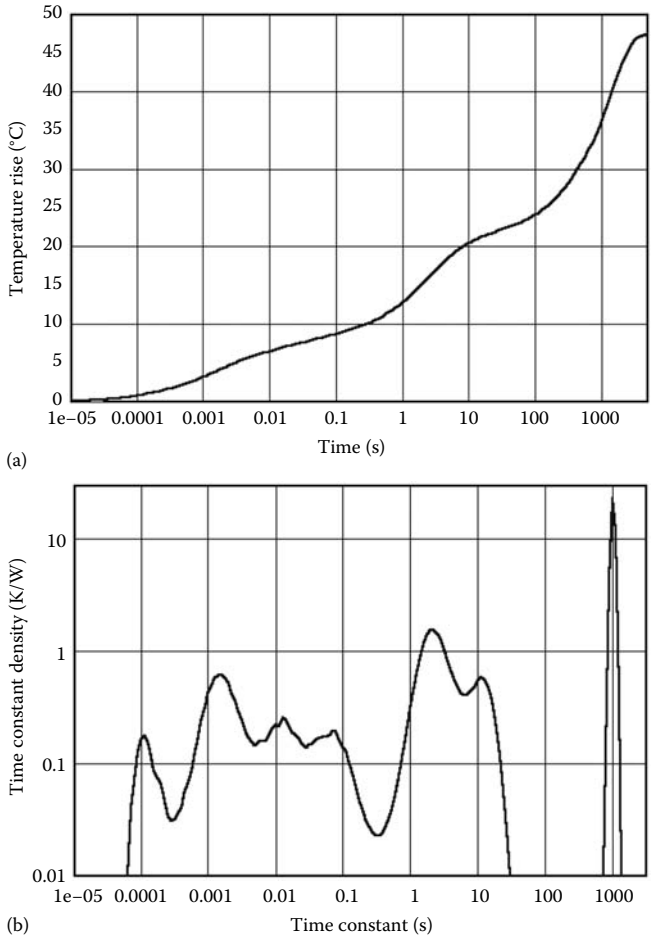


FIGURE 18.19 Thermal identification of a package + heat sink structure. (a) Thermal response between 10 μs and 4000 s. (b) The $R(z)$ time-constant density function.

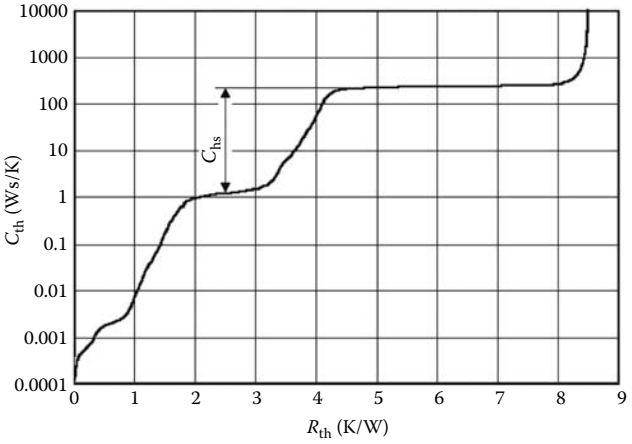


FIGURE 18.20 Cumulative structure function of the package + heat sink structure identified from the response function.

References

1. M. S. Ghausi and J. J. Kelly, *Introduction to Distributed Parameter Networks*, New York: Holt, Rinehart & Winston, 1968.
2. E. N. Protonotarios and O. Wing, Theory of nonuniform RC lines, Part I, *IEEE Transactions on Circuit Theory*, 14, 2–12, Mar. 1967.
3. D. G. Gardner, J. C. Gardner, G. Laush, and W. W. Meinke, Method for the analysis of multi-component exponential decay curves, *Journal of Chemical Physics*, 31(4), 978–986, Oct. 1959.
4. V. Székely, On the representation of infinite-length distributed RC one-ports, *IEEE Transactions on Circuits and Systems*, 38, 711–719, July 1991.
5. R. L. Pritchard, *Electrical Characteristics of Transistors*, New York: McGraw-Hill, 1967.
6. V. Székely and T. Van Bien, Fine structure of heat flow path in semiconductor devices: A measurement and identification method, *Solid-State Electronics*, 31, 1363–1368, Sept. 1988.
7. T. J. Kennett, W. V. Prestwich, and A. Robertson, Bayesian deconvolution. I: Convergent properties, *Nuclear Instruments and Methods*, 151, 285–292, 1978.
8. V. Székely, Identification of RC networks by deconvolution: Chances and limits, *IEEE Transactions on Circuits and Systems—I. Theory and Applications*, 45(3), 244–258, Mar. 1998.
9. V. Székely and M. Rencz, Thermal dynamics and the time constant domain, *IEEE Transactions on Components and Packaging Technologies*, 23(3), 587–594, Sept. 2000.
10. V. Székely, Restoration of physical structures: An approach based on the theory of RC networks, *Proceedings of the ECCTD'99, European Conference on Circuit Theory and Design*, Stresa, Italy, 29 Aug.–2 Sept. 1999, pp. 1131–1134.

19

Synthesis of Distributed Circuits

19.1 Generic Relations **19**-1
19.2 Synthesis of a Capacitance **19**-3
19.3 Synthesis of an Inductance **19**-4
19.4 Synthesis of a Resistance **19**-5
19.5 Synthesis of Transformers **19**-7
19.6 Synthesis Examples **19**-9
Series *L*–*C* Circuit • Parallel *L*–*C* Circuit • Series *L*–*C*–*R* Circuit • Parallel *L*–*C*–*R* Circuit • Low-Pass Filters • High-Pass Filters • Bandpass Filters • Bandstop Filters • Further Comments on Distributed Circuit Filters
19.7 Synthesis of Couplers **19**-18
Generic Relations • Proximity Couplers • Quarter-Wavelength Couplers • Lange Couplers
References **19**-21

Thomas Koryu Ishii
Marquette University

19.1 Generic Relations

The starting procedure for the synthesis of distributed circuits is the same as for the conventional synthesis of lumped parameter circuits. If a one-port network is to be synthesized, then a desired driving-point immittance $H(s)$ must be defined first, where

$$s = \sigma + j\omega \tag{19.1}$$

is the complex frequency, σ is the damping coefficient of the operating signal, and ω is the operating angular frequency. If a two-port network is to be synthesized, then a desired transmittance $T(s)$ must be defined first.

According to conventional principles of network synthesis [1], for the one-port network, $H(s)$ is represented by

$$H(s) = \frac{P(s)}{Q(s)} = \frac{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0} \tag{19.2}$$

where

a_n and b_m are constants determined by the network parameters
$Q(s)$ is a driving function
$P(s)$ is the response function

for a two-port network

$$T(s) = \frac{P(s)}{Q(s)} = \frac{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0}{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0} \tag{19.3}$$

Both $H(s)$ and $T(s)$ should be examined for realizability [1] before proceeding.

If the summation of even-order terms of $P(s)$ is $M_1(s)$ and the summation of odd-order terms of $P(s)$ is $N_1(s)$, then

$$P(s) = M_1(s) + N_1(s) \tag{19.4}$$

Similarly,

$$Q(s) = M_2(s) + N_2(s) \tag{19.5}$$

For a one-port network the driving-point impedance is synthesized by [1]

$$Z(s) = \frac{N_1(s)}{M_2(s)} \tag{19.6}$$

or

$$Z(s) = \frac{M_1(s)}{N_2(s)} \tag{19.7}$$

For a two-port network [1], if $P(s)$ is even, the transadmittance is

$$y_{21} = \frac{P(s)/N_2(s)}{1 + [M_2(s)/N_2(s)]} \tag{19.8}$$

the open-circuit transfer admittance is

$$y_{21} = \frac{P(s)}{N_2(s)} \tag{19.9}$$

and the open-circuit output admittance is

$$y_{22} = \frac{M_2(s)}{N_2(s)} \tag{19.10}$$

If $P(s)$ is odd,

$$y_{21} = \frac{P(s)/N_2(s)}{1 + [M_2(s)/N_2(s)]} \tag{19.11}$$

$$y_{21} = \frac{P(s)}{M_2(s)} \tag{19.12}$$

and

$$y_{22} = \frac{N_2(s)}{M_2(s)} \tag{19.13}$$

In both cases,

$$y_{11} = \frac{y_{21}(s)}{n} \tag{19.14}$$

where n is the current-ratio transfer function from port 1 to port 2. From these y- or z-parameters, the required values for the network components, i.e., L, C, and R, can be determined [1].

In high-frequency circuits the L, C, and R may be synthesized using distributed circuit components. The synthesis of distributed components in microstrip line and circuits is the emphasis of this chapter.

19.2 Synthesis of a Capacitance

If the required capacitive impedance is $-jX_C\ \Omega$, the capacitance is

$$C = \frac{1}{\omega X_C} \tag{19.15}$$

where

$$X_C > 0 \tag{19.16}$$

and ω is the operating angular frequency. In a distributed circuit the capacitance C is often synthesized using a short section of a short-circuited transmission line of negligibly small transmission line loss. If the characteristic impedance of such a transmission line is Z_0, the operating transmission line wavelength is λ_l, and the length of the transmission line is l in meters, then [2]

$$jX_C = jZ_0 \tan\frac{2\pi l}{\lambda_l} \tag{19.17}$$

where

$$\frac{\lambda_l}{4} < l < \frac{\lambda_l}{2}$$

or, more generally,

$$\frac{n\lambda_l}{2} + \frac{\lambda_l}{4} < l < \frac{n\lambda_l}{2} + \frac{\lambda_l}{2} \tag{19.18}$$

and n is an integer. Detailed information on Z_0 and λ_l is given in Chapter 15. Combining Equations 19.15 and 19.17,

$$C = \frac{1}{\omega Z_0 \tan\frac{2\pi l}{\lambda_l}} \tag{19.19}$$

In practical synthesis, the transmission line length must be determined. The design equation is

$$l = \frac{\lambda_l}{2\pi} \tan^{-1}\left(\frac{1}{\omega C Z_0}\right) \tag{19.20}$$

Thus, the capacitance C can be synthesized using a section of a short-circuited transmission line.

If an open-circuited transmission line is used instead, then

$$l = \frac{\lambda_l}{2}\left[\frac{1}{\pi}\tan^{-1}\frac{1}{\omega C Z_0} + \frac{1}{2}\right] \tag{19.21}$$

Equation 19.19 is valid provided that Equation 19.20 is used for l or,

$$l = \frac{\lambda_l}{2}\tan^{-1}\ \omega C Z_0 \tag{19.22}$$

where

$$\frac{n\lambda_l}{2} < l < \frac{\lambda_l}{4} + \frac{n\lambda_l}{2} \tag{19.23}$$

If an open-circuited transmission line section is used instead of the short-circuited transmission line section, just add $\lambda_l/4$ to the line length.

19.3 Synthesis of an Inductance

If the required inductive impedance is $+jX_L\ \Omega$, the inductance is

$$L = \frac{X_L}{\omega} \tag{19.24}$$

where

$$X_L > 0 \tag{19.25}$$

and ω is the operating angular frequency. In a distributed circuit the inductance L is often synthesized using a short section of a short-circuited transmission line of negligibly small transmission line loss. If the characteristic impedance of such a transmission line is Z_0, the operating transmission line wavelength is λ_l, and the length of transmission line is l meters, then [2]

$$jX_L = jZ_0 \tan\frac{2\pi l}{\lambda_l} \tag{19.26}$$

where

$$0 < l < \frac{\lambda_l}{4} \tag{19.27}$$

or, more generally,

$$\frac{n\lambda_l}{2} < l < \frac{n\lambda_l}{2} + \frac{\lambda_l}{4} \tag{19.28}$$

and n is an integer. Detailed information on Z_0 and λ_l is given in Chapter 15.

Combining Equations 19.24 and 19.26

$$L = \frac{Z_0}{\omega} \tan 2\pi \frac{l}{\lambda_l} \tag{19.29}$$

In practical synthesis, the transmission line length must be designed. The design equation is

$$l = \frac{\lambda_l}{2\pi} \tan^{-1} \frac{\omega L}{Z_0} \tag{19.30}$$

Thus, the inductance L can be synthesized using a section of a short-circuited transmission line.

If an open-circuited transmission line is used instead, then

$$l = \frac{\lambda_l}{2} \left(\frac{1}{\pi} \tan^{-1} \frac{\omega L}{Z_0} + \frac{1}{2} \right) \tag{19.31}$$

Equation 19.29 is valid for the open-circuited transmission line provided that Equation 19.31 is used for l or

$$l = \frac{\lambda_l}{2\pi} \tan^{-1} \frac{Z_0}{\omega L} \tag{19.32}$$

where

$$\frac{n\lambda_l}{2} + \frac{\lambda_l}{4} < l < \frac{\lambda_l}{2} + \frac{n\lambda_l}{2} \tag{19.33}$$

If an open-circuited transmission line is used instead of the short-circuited transmission line section, just add $\lambda_l/4$ to the short-circuited transmission line design.

19.4 Synthesis of a Resistance

A distributed circuit resistance can be synthesized using a lossy transmission line of length l with the line end short or open-circuited. When the line end is short-circuited, the input impedance of the line of length l and characteristic impedance $\dot{Z}_0$ is [2]

$$\dot{Z}_i = \dot{Z}_0 \tanh \dot{\gamma} l \tag{19.34}$$

where the propagation constant is

$$\dot{\gamma} = \alpha + j\beta \tag{19.35}$$

α is the attenuation constant, and β is the phase constant of the line. Assuming Z_0 is real, the input impedance becomes a pure resistance, i.e., the reactance becomes zero, when

$$\beta l = \frac{2\pi l}{\lambda_l} = \frac{1}{2}\pi, \frac{3}{2}\pi, \frac{5}{2}\pi, \ldots \tag{19.36}$$

When

$$\beta l = (2n+1)\frac{\pi}{2} \tag{19.37}$$

and where

$$n = 0, 1, 2, 3, \ldots \tag{19.38}$$

the short-circuited transmission line is at antiresonance. When

$$\beta l = 2n\frac{\pi}{2} \tag{19.39}$$

and where

$$n = 1, 2, 3, \ldots \tag{19.40}$$

the transmission line is at resonance.

When the transmission line is at antiresonance, the input impedance is [2]

$$\dot{Z}_{\mathrm{i}} = \dot{Z}_0 \frac{1 + \varepsilon^{-\alpha(2n+1)\frac{\lambda_{\mathrm{l}}}{2}}}{1 - \varepsilon^{-\alpha(2n+1)\frac{\lambda_{\mathrm{l}}}{2}}} \tag{19.41}$$

If

$$\dot{Z}_0 \approx Z_0 \tag{19.42}$$

the input resistance is

$$R_{\mathrm{i}} \approx Z_0 \frac{1 + \varepsilon^{-\alpha(2n+1)\frac{\lambda_{\mathrm{l}}}{2}}}{1 - \varepsilon^{-\alpha(2n+1)\frac{\lambda_{\mathrm{l}}}{2}}} \tag{19.43}$$

where

$$l = (2n + 1)\frac{\lambda_{\mathrm{l}}}{4} \tag{19.44}$$

When the transmission line section is at resonance, the input resistance is

$$R_{\mathrm{i}} \approx Z_0 \frac{1 - \varepsilon^{-\alpha n\lambda_{\mathrm{l}}}}{1 + \varepsilon^{-\alpha n\lambda_{\mathrm{l}}}} \tag{19.45}$$

if Equation 13.41 holds, where

$$l = n\frac{\lambda_{\mathrm{l}}}{2} \tag{19.46}$$

From transmission line theory, α and β can be determined from the transmission line model parameters as [2]

$$\alpha = \frac{\omega LG + BR}{2\beta} \tag{19.47}$$

$$\beta = \left\{ \frac{\omega LB - RG + \sqrt{(RG - \omega LB)^2 + (\omega LG + BR)^2}}{2} \right\}^{\frac{1}{2}} \tag{19.48}$$

where

L is the series inductance per meter
G is the shunt conductance per meter
B is the shunt susceptance per meter
R is the series resistance per meter of the transmission line section

The characteristic impedance of a lossy line is [2]

$$\dot{Z}_0 = R_0 + jX_0 \tag{19.49}$$

$$R_0 = \frac{\left\{ RG + \omega LB + \sqrt{(RG - \omega LB)^2 + (\omega LG - BR)^2} \right\}^{\frac{1}{2}}}{\sqrt{2}\sqrt{G^2 + B^2}} \tag{19.50}$$

$$X_0 = \frac{1}{2R_0} \cdot \frac{\omega LG - BR}{G^2 + B^2} \tag{19.51}$$

or simply from Equation 13.34

$$l = \frac{1}{\dot{\gamma}} \tanh^{-1} \frac{R_i}{\dot{Z}_0} \tag{19.52}$$

to determine l for desired R_i for synthesis.

In practical circuit synthesis, convenient, commercially available, surface-mountable chip resistors are often used. Integrated circuit resistors are monolithically developed. Therefore, the technique described here is seldom used.

19.5 Synthesis of Transformers

An impedance $\dot{Z}_1$ can be transformed into another impedance $\dot{Z}_2$ using a lossless or low-loss transmission line of length l and characteristic impedance Z_0 [2]:

$$\dot{Z}_2 = Z_0 \frac{\dot{Z}_1 + jZ_0 \tan \frac{2\pi l}{\lambda_l}}{Z_0 + j\dot{Z}_1 \tan \frac{2\pi l}{\lambda_l}} \tag{19.53}$$

where λ_l is the wavelength on the transmission line. Solving Equation 19.53 for l,

$$l = \frac{\lambda_l}{2\pi} \tan^{-1} \frac{Z_0(\dot{Z}_2 - \dot{Z}_1)}{j\left(Z_0^2 - \dot{Z}_1\dot{Z}_2\right)} \tag{19.54}$$

In Equation 19.53, if

$$l = \frac{\lambda_l}{4} \tag{19.55}$$

then

$$\dot{Z}_2 = Z_0^2/\dot{Z}_1 \tag{19.56}$$

This is a quarter wavelength transmission line transformer. A low-impedance $\dot{Z}_1$ is transformed into a new high-impedance $\dot{Z}_2$, or vice versa. A capacitive $\dot{Z}_1$ is transformed into an inductive $\dot{Z}_2$, or vice versa. However, Z_0 is usually real, which restricts the available transformed impedances.

An admittance $\dot{Y}_1$ can be transformed into another admittance $\dot{Y}_2$ using a lossless or low-loss transmission line of length l and characteristic admittance $\dot{Y}_0$ [2]

$$\dot{Y}_2 = Y_0 \frac{\dot{Y}_1 + jY_0 \tan\frac{2\pi l}{\lambda_l}}{Y_0 + j\dot{Y}_1 \tan\frac{2\pi l}{\lambda_l}} \tag{19.57}$$

Solving Equation 19.56 for l,

$$l = \frac{\lambda_l}{2\pi} \tan^{-1} \frac{Y_0\left(\dot{Y}_2 - \dot{Y}_1\right)}{j\left(Y_0^2 - \dot{Y}_1\dot{Y}_2\right)} \tag{19.58}$$

In Equation 19.57, if

$$l = \frac{\lambda_1}{4} \tag{19.59}$$

then

$$\dot{Y}_2 = Y_0^2/\dot{Y}_1 \tag{19.60}$$

This is a one quarter-wavelength transmission line transformer. A low-admittance $\dot{Y}_1$ is transformed into a new high-admittance $\dot{Y}_2$, or vice versa. A capacitive $\dot{Y}_1$ is transformed into an inductive $\dot{Y}_2$, or vice versa. However, Y_0 is usually real, which restricts the available transformed admittances.

The transforming method Equations 19.53 and 19.57 cannot transform $\dot{Z}_1$ to every possible value of $\dot{Z}_2$, especially when the value of Z_0 is given. In practice, this is often the case. If this is the case and $\dot{Z}_1$ is complex, use Equation 19.53 first to transform $\dot{Z}_1$ to $\dot{Z}_2'$, where

$$\dot{Z}_2' = R_2' + jX_2' \tag{19.61}$$

and

$$R_2' = R_2 \tag{19.62}$$

where

$$\dot{Z}_2 = R_2 + jX_2 \tag{19.63}$$

Then, synthesize $X_2 - X_2'$ and add it in series at the input of the transmission line, as depicted in Figure 19.1. Then,

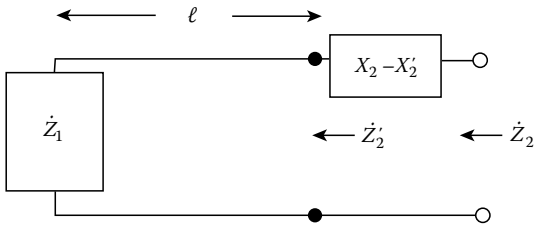


FIGURE 19.1 Synthesis of $\dot{Z}_1$ to any $\dot{Z}_2$ transformer.

$$\begin{aligned}\dot{Z}_2 &= \dot{Z}_2' + j(X_2 - X_2') \\ &= R_2' + jX_2' + jX_2 - jX_2' \\ &= R_2' + jX_2 \\ &= R_2 + jX_2 \end{aligned} \tag{19.64}$$

Thus, $\dot{Z}_1$ can be transformed to any $\dot{Z}_2$ desired.

In a similar manner, to transform a shunt admittance $\dot{Y}_1$ to any other shunt admittance $\dot{Y}_2$, first transform $\dot{Y}_1$ to $\dot{Y}_2'$ so that

$$\dot{Y}_2' = G_2' + jB_2' \tag{19.65}$$

and

$$G_2' = G_2 \tag{19.66}$$

where

$$\dot{Y}_2 = G_2 + jB_2 \tag{19.67}$$

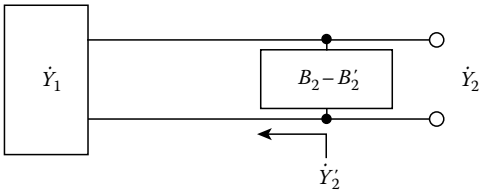


FIGURE 19.2 Synthesis of any $\dot{Y}_2$, from $\dot{Y}_1$ transformation.

Then, synthesize $B_2 - B_2'$ and add it in shunt at the input of the transmission line as depicted in Figure 19.2. Then,

$$\begin{aligned}\dot{Y}_2 &= \dot{Y}_2' + j(B_2 - B_2') \\ &= G_2' + jB_2' + jB_2 - jB_2' \\ &= G_2' + jB_2 = G_2 + jB_2 \end{aligned} \tag{19.68}$$

Thus, $\dot{Y}_1$ can be transformed to any $\dot{Y}_2$ desired.

19.6 Synthesis Examples

Distributed circuits can be synthesized using a number of transmission line sections. The transmission line sections include waveguides, coaxial lines, two-wire lines, and microstrip lines and coplanar waveguides. In this section, distributed circuit synthesis examples of distributed circuits using microstrip lines are presented for convenience. Similar techniques can be utilized for other types of transmission lines.

19.6.1 Series *L–C* Circuit

A series *L–C* circuit as illustrated in Figure 19.3 is considered to be a phase delay line in distributed line technology. The series impedance of an *L–C* circuit is $j(\omega L - 1/\omega C)$. If the load is a resistance of R_L Ω, then the phase delay in the output voltage is

$$\Delta\theta = \tan^{-1} \frac{\omega L - \frac{1}{\omega C}}{R_L} \tag{19.69}$$

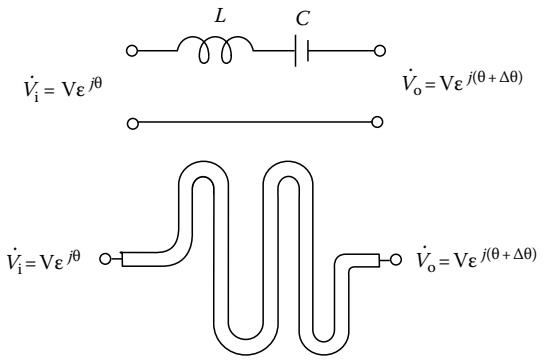


FIGURE 19.3 Synthesis of a series *L–C* circuit using a microstrip meander line.

The phase "delay" should be interpreted algebraically. Depending on the size of ωL and ωC, $\Delta\theta$ can be either "+" or "−." If it is delay, the sign of $\Delta\theta$ is "−" and if it is advance, the sign of $\Delta\theta$ must be "+" in Figure 19.3.

If the phase constant of the microstrip line is

$$\beta = \frac{2\pi}{\lambda_i} \tag{19.70}$$

where λ_1 is the wavelength on the line, then the synthesizing equation is

$$\Delta\theta = \beta l \tag{19.71}$$

or

$$l = \frac{\Delta\theta}{\beta} = \frac{\lambda_1}{2\pi}\tan^{-1}\frac{\omega L - \frac{1}{\omega C}}{R_L} \tag{19.72}$$

Usually, the delay line takes the form of a meander line in microstrip line, as illustrated in Figure 19.3.

19.6.2 Parallel *L–C* Circuit

A parallel *L–C* circuit (see Figure 19.4) can be synthesized using distributed circuit components. In this figure, the open-circuited microstrip line length l_1 represents inductive shunt admittance and the synthesis equation is given by Equation 19.31. In Figure 19.4 another open-circuited microstrip line length l_2 represents capacitive shunt admittance. The synthesis equation for this part is given by Equation 19.21.

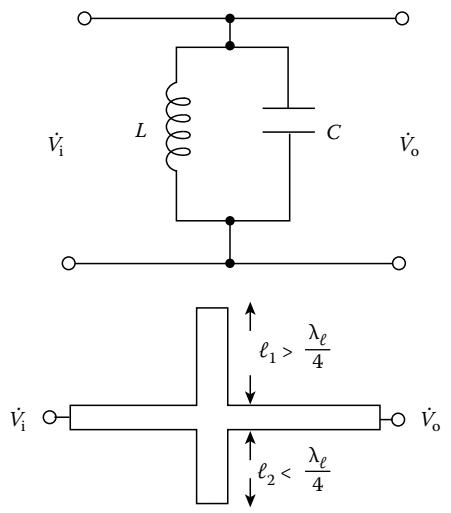


FIGURE 19.4 Synthesis of a parallel *L–C* circuit using a microstrip line circuit.

19.6.3 Series *L–C–R* Circuit

If the load resistance connected to the output circuit of a series *L–C–R* circuit (see Figure 19.5) is R_L, then the output voltage $\dot{V}_o$ of this network across R_L due to the input voltage $\dot{V}_i$ is

$$V_o = \frac{R_L\dot{V}_i}{R_L + R + j\left(\omega L - \frac{1}{\omega C}\right)} = \frac{R_L\dot{V}_i}{Z\varepsilon^{j\phi}} \tag{19.73}$$

where

$$Z = \sqrt{(R_L + R)^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} \tag{19.74}$$

$$\phi = \tan^{-1} \frac{\omega L - \frac{1}{\omega C}}{R_L + R} \tag{19.75}$$

Thus,

$$\dot{V}_o = \frac{R_L}{Z} \dot{V}_i \varepsilon^{-j\phi} \tag{19.76}$$

In a distributed circuit, such as the microstrip line in Figure 19.5,

$$V_o = \dot{V}_i \varepsilon^{-\alpha l} \varepsilon^{-j\beta l} \tag{19.77}$$

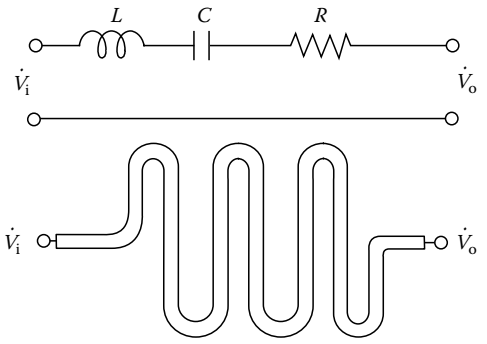


FIGURE 19.5 Synthesis of a series *L–C–R* circuit.

where

- α is the attenuation constant
- β is the phase constant
- l is the total length of the microstrip line

Then, comparing Equations 13.76 with 13.75,

$$\varepsilon^{-\alpha l} = \frac{R_L}{Z} \tag{19.78}$$

or

$$\alpha l = \ln \frac{Z}{R_L} \tag{19.79}$$

and

$$\beta l = \phi \tag{19.80}$$

19.6.4 Parallel *L–C–R* Circuit

A parallel *L–C–R* circuit can be synthesized using the microstrip line (see Figure 19.6). In this figure, the microstrip of length l_1, characteristic impedance $\dot{Z}_{01}$, attenuation constant α_1, and phase constant β_1 represents a parallel circuit of inductance L and $2R$. Another microstrip line of length l_2, characteristic impedance $\dot{Z}_{02}$, attenuation constant α_2, and phase constant β_2 represents a parallel circuit of capacitance C and $2R$.

The input admittance of the open-circuited lossy microstrip line is

$$\dot{Y}_i = \frac{1}{\dot{Z}_{01}} \tanh(\alpha_1 + j\beta_1) l_1 \tag{19.81}$$

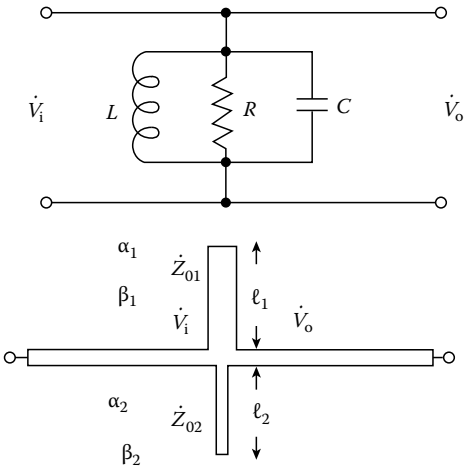


FIGURE 19.6 Synthesis of a parallel *L–C–R* circuit.

In a parallel L–$2R$ circuit, the parallel admittance is

$$Y_{\mathrm{i}} = \frac{1}{2R} + \frac{1}{j\omega L} \tag{19.82}$$

The synthesis equation for the inductive microstrip line is

$$(\alpha_1 + j\beta_1)l_1 = \tanh^{-1}\dot{Z}_{01}\left(\frac{1}{2R} - j\frac{1}{\omega L}\right) \tag{19.83}$$

Similarly, for the capacitive microstrip line,

$$(\alpha_2 + j\beta_2)l_2 = \tanh^{-1}\dot{Z}_{02}\left(\frac{1}{2R} + j\omega C\right) \tag{19.84}$$

To synthesize this distributed circuit, Equations 19.82 and 19.83 must be solved. These are transcendental equations of an excess number of unknowns to be determined by trial and error. Therefore, a digital computer is needed. In many cases, lossless lines are used for L and C and a shunt chip resistor is used for R to avoid complications.

19.6.5 Low-Pass Filters

If the load resistance of a low-pass filter, as presented in Figure 19.7 is R_{L}, then the admittance across R_{L} is

$$\dot{Y}_{\mathrm{L}} = \frac{1}{R_{\mathrm{L}}} + j\omega C \tag{19.85}$$

If this is synthesized using the microstrip line shown in Figure 19.7, then the normalized admittance of the microstrip line is

$$\tilde{\dot{Y}}_{\mathrm{L}} \equiv \frac{\dot{Y}_{\mathrm{L}}}{Y_0} = \frac{\frac{1}{R_{\mathrm{L}}} + j\omega C}{Y_0} \tag{19.86}$$

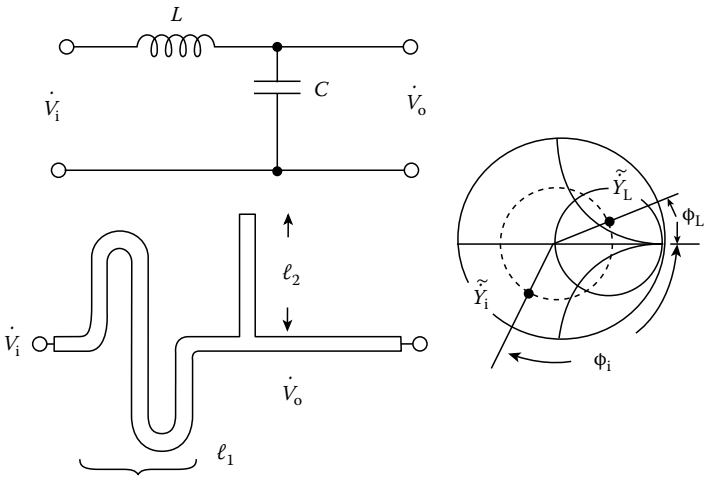


FIGURE 19.7 Synthesis of a low-pass filter.

where Y_0 is the characteristic admittance of the microstripline. $\tilde{\dot{Y}}_L$ should be plotted on a Smith chart as presented in Figure 19.7, and its angle of reflection coefficient ϕ_L must be noted. The input admittance of this circuit is

$$\begin{aligned}\dot{Y}_i &= \frac{1}{j\omega L + \frac{R_L\frac{1}{j\omega C}}{R_L+\frac{1}{j\omega C}}} \\ &= \frac{R_L + j\{\omega C R_C^2(1-\omega^2 LC) - \omega L\}}{R_L^2(1-\omega^2 LC)^2 + (\omega L)^2}\end{aligned} \tag{19.87}$$

This admittance is normalized as

$$\tilde{\dot{Y}}_i = \dot{Y}_i / Y_0 \tag{19.88}$$

and it must be plotted on the Smith chart (Figure 19.7). The phase angle of the reflection coefficient ϕ_i at this point must be noted. Therefore, the synthesis equation is

$$\beta_1 l_1 = \phi_L + \phi_i \tag{19.89}$$

In Figure 19.7, the meander line of length l_1, representing the inductance L, and a straight microstrip line of length l_2, representing the capacitance C, are joined to synthesize a microwave low-pass filter. The synthesis of the shunt capacitance C in Figure 19.7 is accomplished using an open-circuited microstrip line of the phase constant β_2. The length of the microstrip line l_2 is determined from Equation 19.21. If the low-pass filter takes on a π-shape, the microstrip version of synthesis will be similar to Figure 19.8. The microstrip line stub l_3 can be synthesized in the same way as the capacitive stub l_2.

19.6.6 High-Pass Filters

If a high-pass filter consists of a capacitance C and an inductance L, as shown in Figure 19.9, then normalized admittance across the load resistance R_L is

$$\tilde{\dot{Y}}_L = \dot{Y}_L / Y_0 = \left(\frac{1}{R_L} - j\frac{1}{\omega L}\right) \Big/ Y_0 \tag{19.90}$$

where Y_0 is the characteristic admittance of the microstrip line to be used for synthesis. This must be plotted on the Smith chart, as depicted in Figure 19.9, and the angle of reflection coefficient ϕ_L should be noted.

Normalized admittance at the input is

$$\begin{aligned}\tilde{\dot{Y}}_i \equiv \dot{Y}_i / Y_0 &= \frac{1}{\frac{1}{j\omega C} + \frac{(j\omega L)R_L}{j\omega L + R_L}} \Big/ Y_0 \\ &= \frac{(\omega^2 LC)^2 R_L + j\omega\{R_L^2 C(1-\omega^2 LC) + \omega^2 L^2 C\}}{R_L^2(1-\omega^2 LC)^2 + (\omega L)^2} \Big/ Y_0\end{aligned} \tag{19.91}$$

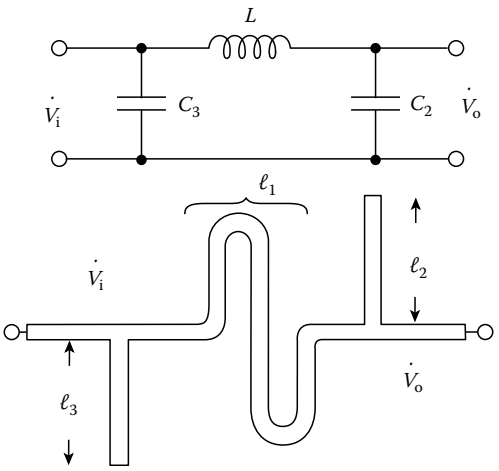


FIGURE 19.8 Synthesis of a π-network low-pass filter.

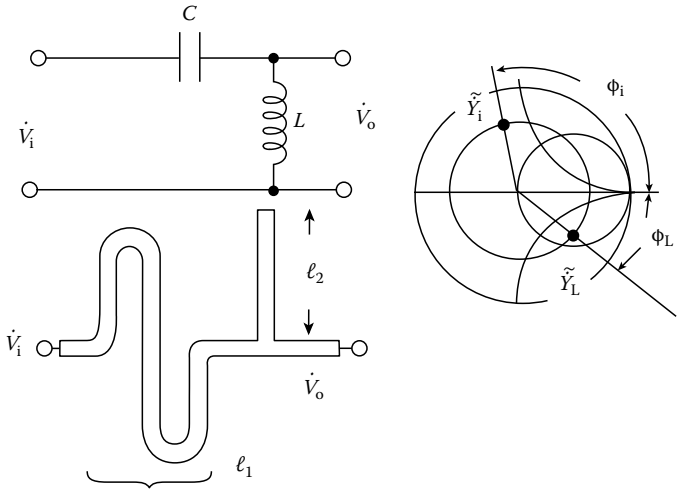


FIGURE 19.9 Synthesis of a high-pass filter.

Plot this on the Smith chart (Figure 19.9), and the angle of the voltage reflection coefficient ϕ_i should be noted.

The phase delay of the distributed microstrip line is

$$\dot{v}_0 = \dot{v}_i \varepsilon^{-j\beta l_1} \tag{19.92}$$

where β is a phase constant of the microstrip line and

$$\beta = 2\pi/\lambda_l \tag{19.93}$$

where λ_l is the wavelength on the microstrip line.

In Equation 19.91, l_1 is the length of microstrip line representing the *C*-section of the low-pass filter. Then, the synthesis equation is

$$\beta l_1 = 360^\circ - (\phi_i + \phi_L) \tag{19.94}$$

The shunt inductance *L* in Figure 19.9 is synthesized using Equation 19.31. Thus, a high-pass filter can be synthesized using a distributed microstrip line section. If the low-pass filter is in a π-network form, then microstrip line configuration will be at the designed frequency (see Figure 19.10).

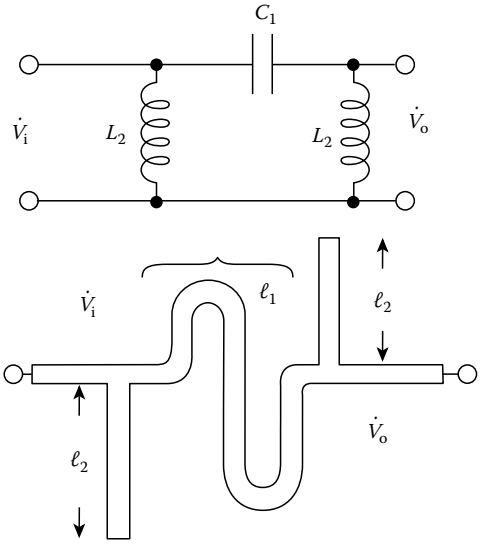


FIGURE 19.10 Synthesis of a π-network high-pass filter.

19.6.7 Bandpass Filters

A bandpass filter may be a series connection of an inductor *L* and a capacitor of capacitance *C*, as shown in Figure 19.11. This figure is identical to Figure 19.3.

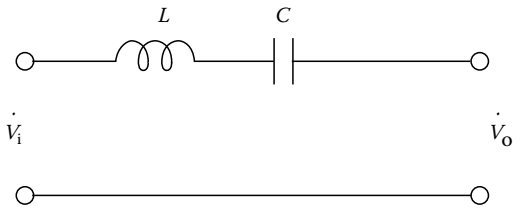


FIGURE 19.11 Configuration of a series *L*–*C* bandpass filter.

Therefore, the microstrip line synthesis equation is Equation 19.72.

If the bandpass filter is in the shape of a π-network (Figure 19.12a), then the series L_1C_1, represented by the meander microstrip line of length l_5, can be designed using Equation 19.72. Both of the shunt L_2 and C_2 arms are identical to Figure 19.4. Therefore, this part is synthesized by utilizing the technique in Section 19.6.2, which results in the configuration presented in Figure 19.12b. An alternative distributed version of Figure 19.12a is given in Figure 19.12c. In this figure, the technique in λ_l is the microstrip line wavelength at the center of the desired passband. The frequency bandwidth is determined by the quality factor of the resonating element.

19.6.8 Bandstop Filters

Various types of bandstop filters are available. A schematic diagram of a series type bandstop filter is shown in Figure 13.13. It is basically a resonant circuit inserted in series with the transmission line. The center frequency of the bandstop is the resonant frequency of the resonant circuit. The loaded quality factor of the resonator determines the bandstop width.

In the distributed microstrip line the resonator is a section of a half-wavelength microstrip line resonator of either low impedance (Figure 19.13a) or high impedance (Figure 19.13b), or a dielectric resonator placed in proximity to the microstrip line, (see Figure 19.13c) or a ring resonator placed in

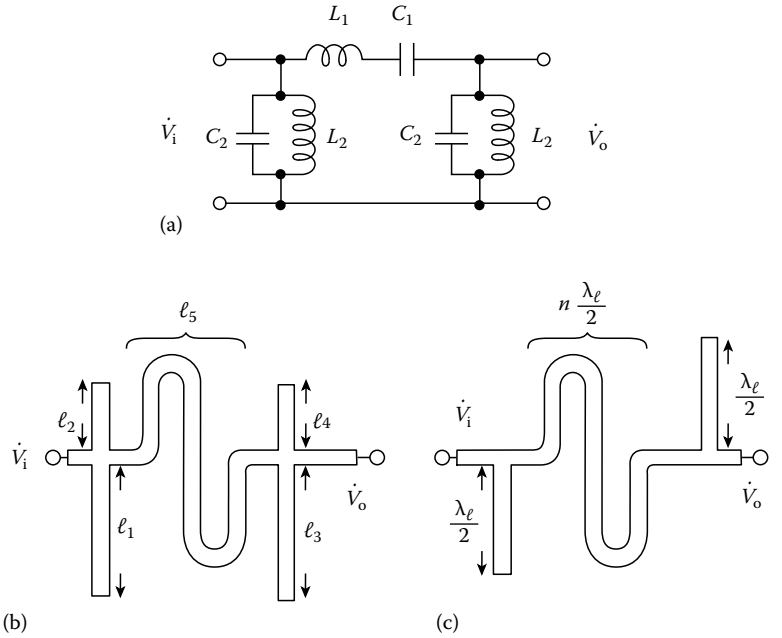


FIGURE 19.12 Synthesis of a distributed microstrip line π-network bandpass filter. (a) π-network. (b) Configuration of (a). (c) Alternative distributed version of (a).

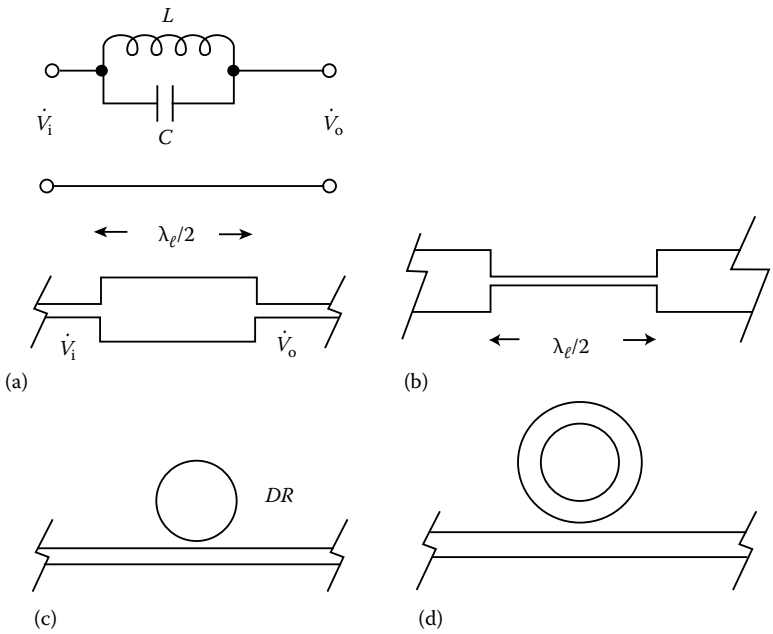


FIGURE 19.13 Synthesis of distributed series type bandstop filters. (a) A section of a half-wavelength microstrip line resonator of low impedance. (b) A section of a half-wavelength microstrip line resonator of high impedance. (c) A dielectric resonator placed in proximity to the microstrip line. (d) A ring resonator placed in proximity to the microstrip line.

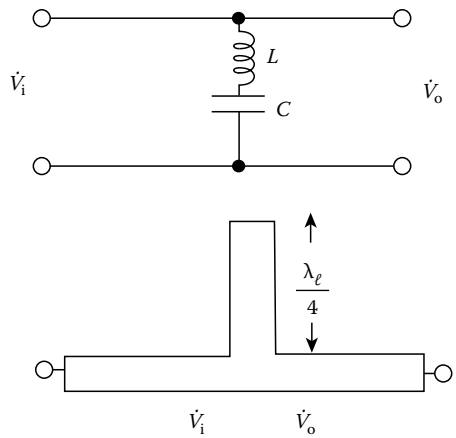


FIGURE 19.14 Synthesis of distributed parallel type bandstop filters.

proximity to the microstrip line (Figure 19.13d). The resonant frequency of the resonators is the center frequency of the bandstop. The bandwidth is determined by the loaded Q and coupling of the resonator.

A bandstop filter can also be synthesized in microstrip line if the filter is a parallel type (Figure 19.14). Simply attaching a one quarter-wavelength, open-circuited stub will create a bandstop filter. The center of the bandstop has a wavelength of λ_l (Figure 19.14).

A bandstop filter may take a π-network form, as illustrated in Figure 19.15, by the combination of the $\lambda_l/2$ resonator and $\lambda_l/4$ open-circuited stubs. The frequency bandwidth of the filter depends on the quality factor of the resonating elements.

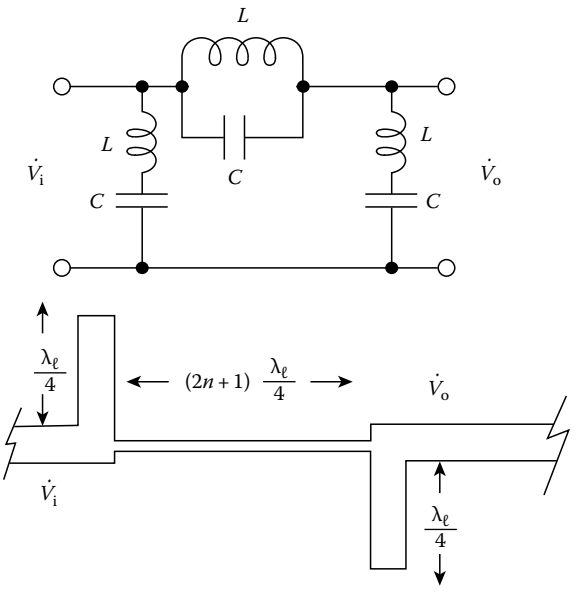


FIGURE 19.15 Synthesis of distributed π-network bandstop filters.

19.6.9 Further Comments on Distributed Circuit Filters

If the unit sections of filters previously described are treated as lumped parameter filters, they can be cascaded into a number of stages to obtain desired filter characteristics, as exemplified by Chebyshev and Butterworth filters. The center frequencies of the stages can be staggered to obtain the desired frequency bandwidth [5].

A radio frequency choke is a special case of a bandstop filter. Cascading of alternate $\lambda_l/2$ sections of high and low impedance (Figure 19.16a) is common. This is called the high-low filter. A microstrip line bandpass filter can be made using a number of $\lambda_l/2$ microstrip line resonators, as shown in Figure 19.16b [5].

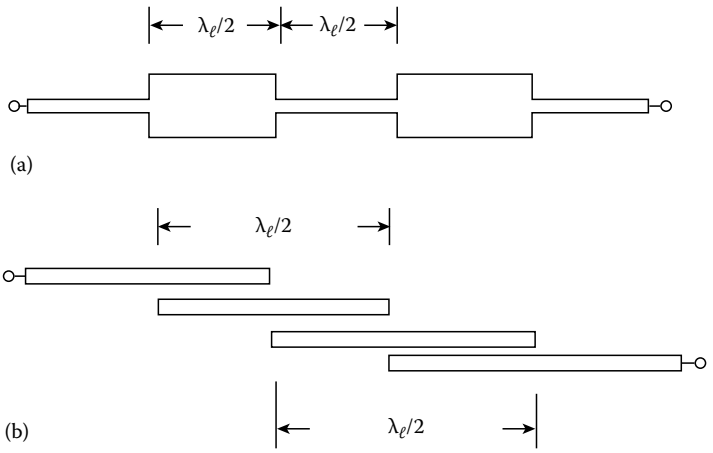


FIGURE 19.16 Microstrip line filters. (a) Bandstop high-low filter. (b) Bandpass coupled resonator filter.

19.7 Synthesis of Couplers

19.7.1 Generic Relations

A coupler is a circuit with coupling between a primary circuit and a secondary circuit. In distributed circuit technology, it often takes the form of coupling between two transmission line circuits. The coupling may be done by simply placing the secondary transmission line in proximity to the primary transmission line, as presented in Figure 19.17a and c, or both the primary and the secondary transmission lines are physically connected, as shown in Figure 19.17b. These examples of distributed coupler circuits are shown in microstrip line forms. No lumped parameter equivalent circuit exists for these distributed coupler circuits. Therefore, the previously described synthesis techniques cannot be used.

19.7.2 Proximity Couplers

Coupling of two transmission lines is accomplished simply by existing proximity to each other. If the transmission lines are microstrip lines (Figure 19.17a or 19.18), the coupler is also called an edge coupler.

The coupling factor (CF) of a coupler is defined as [3]

$$\mathrm{CF} = \frac{P_4}{P_1} \tag{19.95}$$

where

P_1 is the microwave power fed to port 1

P_4 is the microwave power output at port 4.

If the power coupling factor per meter of the line edge is $k(x)$ in the coupling region of length l, then [3]

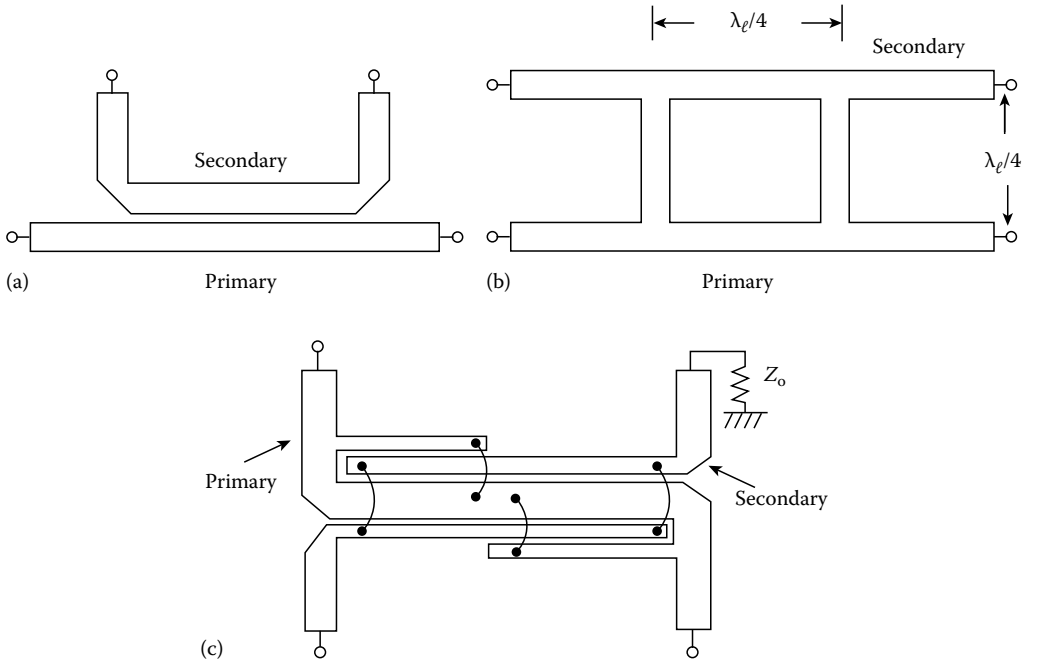


FIGURE 19.17 Distributed circuit couplers. (a) Proximity coupler. (b) Quarter-wavelength coupler. (c) Lange coupler.

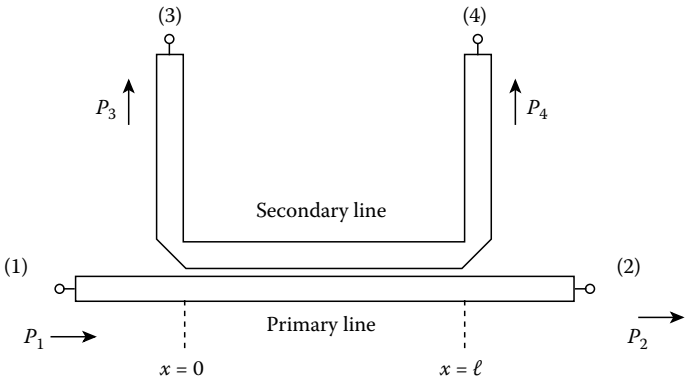


FIGURE 19.18 Microstrip line proximity coupler.

$$\mathrm{CF} = \left| \int_0^l k(x)\varepsilon^{-j\beta x} \mathrm{d}x \right| \tag{19.96}$$

This can be expressed in decibels as

$$\mathrm{CF\ (dB)} = 10 \log \frac{P_4}{P_1} = 10 \log \left| \int_0^l k(x)\varepsilon^{-j\beta x} \mathrm{d}x \right| \tag{19.97}$$

The directivity of a coupler is defined as [3]

$$\mathrm{dir} = \frac{P_3}{P_4} \tag{19.98}$$

$$\mathrm{dir} = \frac{1}{\mathrm{CF}} \left| \int_0^l k(x)\varepsilon^{-j2\beta x} \mathrm{d}x \right| \tag{19.99}$$

The decibel expression of directivity, then, is

$$\begin{aligned} \mathrm{dir\ (dB)} &= 10 \log_{10} \frac{P_3}{P_4} \\ &= 10 \log_{10} \frac{1}{\mathrm{CF}} \left| \int_0^l k(x)\varepsilon^{-j2\beta x} \mathrm{d}x \right| \end{aligned} \tag{19.100}$$

Generally, $P_3 \neq P_4$; therefore, the coupling has a directional property. This is the reason that this type of coupler is termed the directional coupler.

Insertion loss (IL) of a coupler is defined as

$$\mathrm{IL} = \frac{P_2}{P_1} = \frac{P_1 - (P_3 + P_4)}{P_1} \tag{19.101}$$

Inserting Equations 13.95 and 13.98 into Equation 19.101,

$$\text{IL} = 1 - \text{CF}(1 + \text{dir}) \tag{19.102}$$

or

$$\text{IL (dB)} = 10\log\{1 - \text{CF}(1 + \text{dir})\} \tag{19.103}$$

A complete analytical synthesis of microstrip line couplers is not available. However, computer software based on semiempirical equations is commercially available [5].

19.7.3 Quarter-Wavelength Couplers

The coupling factor of a quarter-wavelength coupler (Figure 19.19) is defined as [3]

$$\text{CF} = \frac{P_4}{P_1} \tag{19.104}$$

The amount of P_4 or coupling factor can be controlled by the width of microstrip lines connecting the secondary line to the primary line.

$$\text{CF (dB)} = 10\log_{10} P_4/P_1 \tag{19.105}$$

The directivity is

$$\text{dir} = \frac{P_3}{P_4} \tag{19.106}$$

or

$$\text{dir (dB)} = 10\log\frac{P_3}{P_4} \tag{19.107}$$

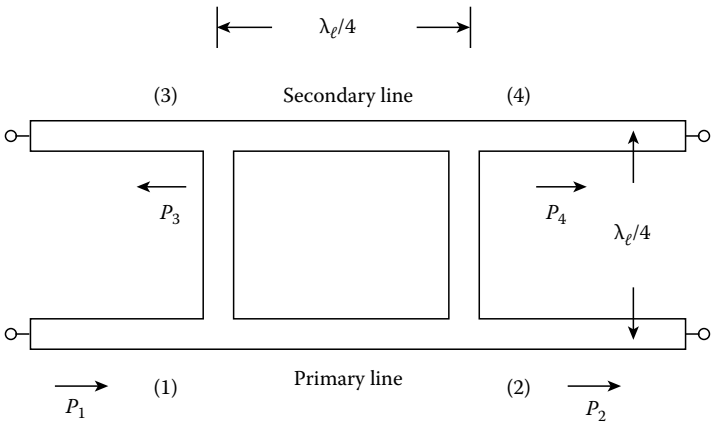


FIGURE 19.19 Quarter-wavelength coupler.

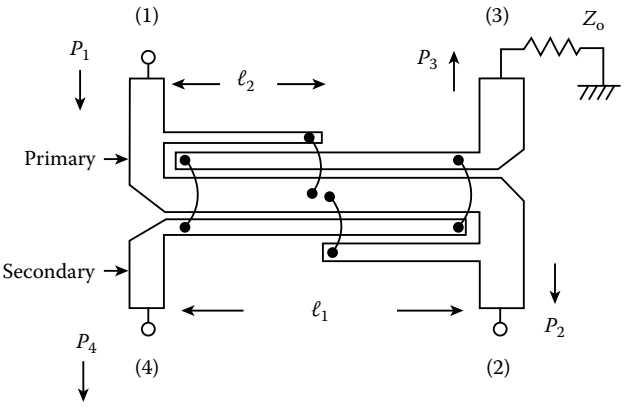


FIGURE 19.20 Lange coupler.

At the precise design operating frequency, P_3 should be zero. The insertion loss for this coupler is

$$\text{IL} = \frac{P_1 - (P_3 + P_4)}{P_1} = 1 - \text{CF}(1 + \text{dir}) \tag{19.108}$$

19.7.4 Lange Couplers

A schematic diagram of a generic Lange coupler [4] is given in Figure 19.20. In this figure, the primary transmission line is from port (1) to port (2). The secondary transmission line is from port (3) to port (4). In Lange couplers, both the primary and the secondary lines are coupled by edge coupling or proximity coupling. It should be noted that the transmission lines are not physically connected to each other by bonding wires. The bond wires on both ends form connections of the secondary transmission line from port (3) to port (4). The bond wires in the middle of the structure are the electrical potential equalizing bonding in the primary transmission line. Adjusting the values of l_1 and l_2 creates a variety of coupling factors. One of the objectives of Lange couplers is to increase the coupling factor between the primary lines and the secondary line. By comparing the structure to that of a straight edge-coupled coupler (see Figure 19.18), the Lange coupler in Figure 19.20 has more length of coupling edge. Therefore, a higher coupling factor is obtainable [5].

References

1. F. F. Kuo, *Network Analysis and Synthesis*, New York: Wiley, 1962.
2. T. K. Ishii, *Microwave Engineering*, San Diego, CA: Harcourt Brace Jovanovich, 1989.
3. F. J. Tischer, *Mikrowellen-Messtechnik*, Berlin: Springer-Verlag, 1958.
4. J. Lange, Interdigitated stripline quadrature hybrid, *IEEE Trans. MTT*, 17(11), 1150, 1151, Dec. 1969.
5. V. F. Fusco, *Microwave Circuit*, Englewood Cliffs, NJ: Prentice Hall, 1987.

Index

A

Additive white Gaussian noise (AWGN) channel, 14-70
Analogic supercomputer chip, cellular neural networks
  global analogic program unit (GAPU), 13-8
  global architecture, 13-6–13-7
  local analog memory (LAM), 13-6
Analysis Program for Linear Active Circuits (APLAC), 12-8–12-10
Artificial neural network, approximation
  feed-forward artificial neural network, 9-25
  neuron, 9-24–9-25
  training process, 9-24
Autonomous continuous-time dynamical systems, 14-3
Autonomous dynamic circuits
  DC-operating points, convergence
    analog neural network, 7-16–7-17
    cocontent, 7-16
    Lyapunov function, 7-14–7-16, 7-18–7-20
    resistive *N*-port, 7-17
    state equations, 7-14, 7-18
  nonlinear capacitor and inductor, 7-13

B

Bandpass filters, 19-14–19-15
Bandstop filters, 19-15–19-17
Bifurcation and chaos
  blue sky catastrophe, 14-23
  capacity (D_0 dimension), 14-10–14-11
  chaotic steady-state, 14-10
  Chua's oscillator
    double-scroll Chua's attractor, 14-33–14-34
    intermittency route, 14-25–14-26
    period-doubling route, 14-23–14-25
    state equations, 14-43–14-44
    time-domain waveforms, 14-34–14-35
    topological conjugacy, 14-44–14-45
  communication
    chaos shift keying, 14-65–14-69
    chaotic masking, 14-65
    cryptographic purpose, 14-71
    miscellaneous, 14-70–14-71
    modulation schemes, 14-69–14-70
    signal spreading, 14-69
  correlation (D_2) dimension
    definition, 14-13
    iterative method, 14-11–14-12
  electrical and electronic circuits
    continuous-time dynamical systems, 14-2
    discrete-time dynamical systems, 14-4
    nonautonomous dynamical systems, 14-3–14-4
  equilibrium point
    stability, 14-13–14-14
    zero-dimensional object, 14-7
  Hopf bifurcation, 14-23
  intermittency route, 14-25
  ω-limit set, 14-6
  Lyapunov exponents
    discrete-time systems, 14-21
    equilibrium point stability, 14-20
    steady-state solutions, 14-21–14-22
  parameter space diagrams, 14-26–14-27
  period-doubling bifurcation, 14-23–14-24
  periodic-doubling route, 14-25
  periodic steady-state
    chaotic steady-state, 14-10
    Fourier transform, 14-7
    power spectrum, 14-8
    quasiperiodicity, 14-8–14-10
    sampling interval, 14-10
    subharmonic, 14-8
  pseudorandom sequence generation, 14-64–14-65
  quasiperiodic (torus breakdown) route, 14-25–14-26
  saddle-node bifurcation, 14-23
  Shil'nikov's theorem, 14-17–14-19
  steady-state trajectories
    discrete-time systems, 14-14
    eigenvalues, 14-14–14-15
    eigenvectors, eigenspaces, stable and unstable manifolds, 14-15

Poincaré sections, 14-16–14-17
stability, limit cycle, 14-15
Bi-i camera-computer, 13-17
Binary phase shift keying (BPSK), 14-69–14-70
Bipolar transistor model, 11-9–11-11
Blackman's formula
input admittance, 4-3
input impedance, 4-1–4-2, 4-5
output impedance, 4-5
return difference, 4-1–4-4
series-parallel feedback configuration, 4-4
short circuit current gain, 4-5
voltage gain, 4-6
Blecher's procedure
short circuit admittance matrix, 5-3
transistor, common emitter configuration, 5-2–5-3
Brayton–Moser equations, 10-18–10-19

C

Cauchy integral formula, 18-17
Cellular neural networks (CNN)
analogic supercomputer chip
global analogic program unit (GAPU), 13-8
global architecture, 13-6–13-7
local analog memory (LAM), 13-6
Bi-i camera-computer, 13-17
biology, relation, 13-11
definition and classification
grid structure and trillion operations per second, 13-2
typical models, 13-3
image processing
bank notes recognition, 13-9
dollar bill recognition process, 13-10
halftoning and smoothing function, 13-9
partial differential equations, 13-10–13-11
simple circuit structure
capacitor voltage, 13-4
dynamics, 13-5
simple first-order cell, 13-3–13-4
template library, 13-5
unity-gain nonlinear sigmoid characteristics, 13-5–13-6
voltage-controlled current source (VCCS) amplifier, 13-3
template library
AVERAGE, pixel intensity, 13-12
black-and-white edge detector, 13-14–13-15
convex corner detector, 13-13
DELDIAG1, 13-13–13-14
DIAG, 13-14
grayscale contour detector, 13-12–13-13
logical AND function, 13-12
logical OR function, 13-16
MATCH, 13-15
PEELIPIX, 13-16
Chaos shift keying (CSK)
block diagram, 14-65–14-66
Chua's circuits, masking, 14-68
data transmission, 14-65
encoding, 14-66
signal correlation, 14-69
waveform, 14-67
Chaotic circuit synchronization
linear mutual coupling, 14-57–14-58
Pecora–Carroll drive-response concept
cascaded drive-response systems, 14-61–14-62
conditional Lyapunov exponent (CLE), 14-60–14-61
dynamical system, 14-59–14-60
linear time-varying system, 14-60
stable subsystem, 14-61
Chaotic on–off keying (COOK), 14-69–14-70
Chua's circuit, 9-6, 10-20–10-21
chaos manifestations
broadband noise-like power spectrum, 14-36
sensitive dependence, initial conditions, 14-34–14-36
time domian randomness, 14-36
D_{-1} and D_1 dynamics, 14-30–14-31
D_0 dynamics, 14-30
dimensionless coordinates and α–β parameter-space diagram
C_1 bifurcation sequence, 14-40
Hopf bifurcation, 14-41
state equations, 14-40
state-space behaviors, 14-40, 14-42
experimental steady-state solutions, 14-37–14-38
global dynamics, 14-32
piecewise-linear dynamics
DC equilibrium points, 14-29
eigenvalues, 14-29–14-30
practical realization
Chua's oscillator, 14-37–14-38
component list, 14-36–14-37
simulation
analog devices' AD712 macromodel, 14-39
ODE solver, 14-38
PSPICE, 14-40–14-42
SPICE deck, 14-40–14-41
state equations, 14-28
steady-states and bifurcations
boundary crisis, 14-34
double-scroll Chua's attractor, 14-33, 14-35
equilibrium point and Hopf bifurcation, 14-32
period-doubling cascade and periodic windows, 14-33
spiral Chua's attractor, 14-33–14-34
Chua's oscillator
circuit, 12-5, 12-11
double-scroll Chua's attractor, 14-34–14-35
intermittency route, 14-25–14-26
period-doubling route, 14-23–14-25
state equations, 14-43–14-44
time-domain waveforms, 14-34–14-35

topological conjugacy
class $\mathscr{C}$, 14-44
eigenvalues-to-parameters mapping algorithm, 14-46–14-47
equivalent eigenvalue parameters, 14-45
Circuit equations, dimensionless forms
Desloge's approach, 10-14, 10-16
extension groups, 10-11–10-12
impedance, RLC parallel circuit, 10-10
Lie algebra, 10-12
multiparameter Lie groups, 10-10
Pi-theorem, 10-13
Q-factor, 10-10
scale transformations, 10-11
tunnel-diode circuit equations, 10-14
tunnel-diode current, 10-15
Circuit feedback factor, 1-6
Circuit-simulation programs
APLAC, 12-8–12-10
DYNAST, 12-17–12-18
ESACAP, 12-14–12-17
NAP2, 12-10–12-14
SPICE, 12-4–12-7
CLE, *see* conditional Lyapunov exponent
Closed-loop damping factor
loop gain, 1-8–1-9
transfer function, 1-8
underdamped closed-loop amplifiers, 1-9
Coaxial lines
geometric structure, 15-8
transmission line parameters, 15-8–15-9
wavelength and phase velocity, 15-10
Coefficient matrix Y(*s*), *see* Indefinite-admittance matrix
Collective computation circuitry
euclidean distance, 8-31–8-32
normalization operation, 8-32
T-norm operator, 8-32–8-33
Conditional Lyapunov exponent (CLE), 14-60–14-61
COOK, *see* Chaotic on–off keying
Coplanar waveguide (CPW)
geometric structure, 15-17
transmission line parameters, 15-17–15-18
wavelength and phase velocity, 15-19
Coupling factor (CF), 19-18

D

DAE, *see* Differential-algebraic equation
Differential-algebraic equation (DAE), 10-9
Differential chaos shift keying, 14-69–14-70
Differential equation and state-space representations
Chua's circuit, 9-6
mathematical modeling, 9-3
partial differential equation representation, 9-6
physical linear circuit, 9-3
state-variable representation, 9-5
state vector, system, 9-4
system state variables, 9-3
vector-valued nonlinear function, 9-4
Directional coupler, 19-19
Dispersion equation, 18-6
Distributed circuits synthesis
capacitance
characteristic impedance, 19-3
design equation, 19-3–19-4
open-circuited transmission line, 19-4
short-circuited transmission line, 19-3–19-4
conventional principle
one-port network, 19-1–19-2
two-port network, 19-2–19-3
couplers
Lange coupler, 19-21
proximity coupler, 19-18–19-20
quarter-wavelength, 19-20–19-21
inductance
characteristic impedance, 19-4
design equation and open-circuited transmission line, 19-5
microstrip lines
bandpass filters, 19-14–19-15
bandstop filters, 19-15–19-17
filters, 19-17
high-pass filters, 19-13–19-14
low-pass filters, 19-12–19-13
parallel *L*–*C* circuit, 19-10
parallel *L*–*C*–*R* circuit, 19-11–19-12
series *L*–*C* circuit, 19-9–19-10
series *L*–*C*–*R* circuit, 19-10–19-11
resistance
antiresonance, 19-6
characteristic impedance, R_0, 19-7
input impedance, 19-6
lossy transmission line, 19-5
resonance, 19-6
surface-mountable chip resistors, 19-7
transformers
admittance transformation, 19-9
characteristic admittance, 19-8
quarter wavelength transmission line, 19-7–19-8
transforming method, 19-8–19-9
Distributed RC networks
asymptotic approximation, 18-11
cumulative structure function, 18-9–18-10
infinite-length
pole-zero representation, 18-15–18-16
time constant, 18-13–18-15
inverse problem
Bode amplitude and phase diagrams, 18-19
network identification by deconvolution (NID), 18-20
lumped element approximation, 18-11–18-12
Riccati differential equation, 18-10
uniform
differential equation, 18-1–18-2
diffusion equation, 18-2
Dirac-δ excitation, 18-2–18-3

finite DRC line, 18-4–18-6
frequency domain, 18-6
Gauss-like function, 18-2
line transients, 18-3
lossless RC lines, 18-1–18-2
lossy DRC lines, 18-6–18-7
semi-infinite line, 18-3–18-4
Double-scroll Chua's attractor, 14-33, 14-35
Dual-loop feedback architectures
series–series/shunt–shunt feedback amplifier
closed-loop transconductance, 2-21
closed-loop transresistance, 2-22
closed-loop voltage gain, 2-22, 2-24
loop gain, 2-21
Norton equivalent input, 2-23
open-loop transresistance, 2-21–2-22
output and input resistances, 2-22, 2-25
Thévenin equivalent output circuit, 2-23
series–shunt/shunt–series feedback amplifier
AC schematic diagram, 2-28
circuit voltage gain, 2-25
current gain, 2-27
driving-point input and output resistance, 2-27–2-28
h-parameters, 2-25
system-level diagram, 2-25–2-26
voltage gain, 2-27–2-28
DYNAmic Simulation Tool (DYNAST), 12-17–12-18

E

Ebers–Moll equations, 7-3
Edge coupler, 19-18
Engineering System and Circuit Analysis (ESACAP), 12-14–12-17
Equicofactor matrix, 3-3
ESACAP, *see* Engineering System and Circuit Analysis

F

Feedback amplifier theory
analysis methods, 1-2–1-3
global single-loop feedback
closed-loop damping factor, 1-7–1-9
driving-point I/O resistances, 1-7
frequency invariant feedback factor, 1-9–1-10
frequency variant feedback factor, 1-10–1-11
return ratio, 1-6
indefinite-admittance matrix
current gain, 3-7
driving-point impedance, 3-5
equicofactor matrix, 3-3
input impedance, 3-7
Kirchhoff's current law, 3-3
Laplace-transformed potentials, 3-1
n-terminal network, symbolic representation, 3-2
second-order cofactor, 3-4
transfer admittance, 3-7
transfer impedance, 3-5
transistor hybrid-pi equivalent network, 3-4
transistor, small-signal equivalent model, 3-2
voltage gain, 3-6–3-7
null return difference, 3-11–3-13
pole splitting open-loop compensation
damping factor and gain frequency, 1-15
open-loop amplifier, 1-11–1-13
pole frequency, 1-14
voltage transfer function, 1-13
return difference
amplifier voltage gain, 3-11
definition, 3-7
loop transmission, 3-9
returned voltage, 3-8–3-9
return ratio, 3-9–3-10
voltage-controlled current source, 3-7, 3-9
signal flow analysis
circuit feedback factor, 1-6
critical transimpedance parameter, 1-4
null return ratio, 1-4–1-5
resultant voltage gain, 1-4
return ratio, 1-4–1-5
Feigenbaum number, 14-25
Fourier transform, 14-7
Frequency-domain network analysis
error correction network, 17-7–17-8
network analyzer (NWA), block diagram
inverse Fourier transform, 17-6
transmission, 17-7
random errors, 17-7
Smith chart representation, S_{11} data, 17-9
S-parameter, 17-6
systematic errors, 17-7
Frequency invariant feedback factor dominant pole approximation, 1-9
open and closed loop 3 dB bandwidths, 1-9–1-10
Frequency shift keying (FSK), 14-66
Frequency variant feedback factor, 1-10–1-11
Full-wave analysis, 16-4

G

GaAs FET model, 11-12–11-13
Gaussian and bell-shaped basis functions
fuzzy membership functions, 8-8
variance parameter, 8-7
Global analogic program unit (GAPU), 13-8
Goal dynamics, synchronization, 14-57

H

Hankel operator, 9-16
Hermite linear basis function, 8-5–8-6, 8–35
High-low filter, 19-17
High-pass filters, 19-13–19-14
Hopf bifurcation, 14-23

I

Indefinite-admittance matrix
current gain, 3-7

driving-point impedance, 3-5
equicofactor matrix, 3-3
input impedance, 3-7
Kirchhoff's current law, 3-3
Laplace-transformed potentials, 3-1
n-terminal network, symbolic representation, 3-2
second-order cofactor, 3-4
transfer admittance, 3-7
transfer impedance, 3-5
transistor hybrid-pi equivalent network, 3-4
transistor, small-signal equivalent model, 3-2
voltage gain, 3-6–3-7
Infinite-length RC lines
characteristic impedance, 18-13
pole–zero representation
Bode plot, 18-15
dipole intensity function, 18-16
$R(\zeta)$, $I_d(\Sigma)$ relationship and impedance function, 18-16–18-17
time constant representation
convolution-type differential equation, 18-14
lumped element one-port, 18-13–18-14
spectrum lines, 18-14
weighting function, 18-15
time-constant spectrum, 18-17–18-18
Input–output representation
coprime factorization representation, 9-7–9-8
state-variable representation, 9-6–9-7
Insertion loss (IL), 19-19–19-20

J

Jacobian matrix
eigenvectors, eigenspaces, stable and unstable manifolds, 14-15
equilibrium point stability, 14-14

L

Lange couplers, 19-21
Least-squares approximation and projections
Banach space, 9-13–9-14
discrete data fitting, 9-11
Taylor series representation, 9-12
Volterra kernel sequences, 9-11–9-12
Lie groups, 10-10, 10-12
Linear functionals approximation, 9-23–9-24
Linear state-space representation, 9-4
Linear systems identification
(ARMA model with exogenous noisy inputs) ARMAX model, 9-27–9-28
autoregressive with moving-average (ARMA) models, 9-27
behavioral approach, 9-28
mainstream theory, characteristics, 9-27
Lipschitz operator, 9-15
Local analog memory (LAM), 13-6
Local communication and control unit (LCCU), 13-8
Log–antilog multipliers, 8-25–8-26
Loop transmission matrix, 6-5, 6-13
Low-pass filters, 19-12–19-13
Lyapunov function, 7-14–7-16, 7-19

M

Microstrip lines
geometric structure, 15-15
transmission line parameters, 15-15–15-16
wavelength and phase velocity, 15-17
Microstrip lines, distributed circuits synthesis
bandpass filters, 19-14–19-15
bandstop filters, 19-15–19-17
high-low filter, 19-17
high-pass filters, 19-13–19-14
low-pass filters, 19-12–19-13
parallel *L*–*C* circuit, 19-10
parallel *L*–*C*–*R* circuit, 19-11–19-12
series *L*–*C* circuit, 19-9–19-10
series *L*–*C*–*R* circuit, 19-10–19-11
Min–max (uniform) approximation
Banach space, 9-15
Hankel operator, 9-16
Volterra polynomials, 9-14
Monolithic microwave integrated circuits (MMICs), 16-1
MOS transistor model, 11-12
Multiconductor transmission lines (MTL)
frequency *vs*. time domain analysis
fourier transform technique, 16-2
modal propagation factors, 16-1
multichip module (MCM), 16-1
Telegrapher's equations
analytical expressions, 16-5–16-7
$C \times C$ matrix, 16-3–16-4
quasitransverse electromagnetic, 16-4–16-5
wave phenomena, 16-3–16-4
Multidimensional function, approximation techniques
canonical piecewise linear representations, 8-9
multilayer perceptron, 8-11
neurofuzzy interpolation, 8-10–8-11
radial basis functions, 8-10
sectionwise piecewise-polynomial functions, 8-8–8-9
Multiparameter sensitivity
first-order perturbation, 6-14
scalar transfer function, 6-15
Taylor series expansion, 6-14
voltage gain, 6-17
Multiple-loop feedback amplifier theory
closed-loop transfer-function matrix, 6-3–6-4
equivalent network, voltage-series feedback amplifier, 6-4
fundamental matrix feedback-flow graph, 6-3
general configuration, 6-1–6-2
seven-port network, 6-4

N

NAP2 program
applications, 12-10–12-11
definition, 12-10

"net-list" input language, 12-11–12-12
time interval, current *vs.* voltage, 12-13–12-14
Network functions
Blackman's formula
input admittance, 4-3
input impedance, 4-1–4-2, 4-5
output impedance, 4-5
return difference, 4-1–4-4
series-parallel feedback configuration, 4-4
short circuit current gain, 4-5
voltage gain, 4-6
sensitivity function
amplifier equivalent network, 4-8–4-9
amplifier indefinite admittance matrix, 4-9
common-emitter transistor amplifier, 4-8
definition, 4-6
driving-point impedance, 4-9
Neuron, 9-24
Newton–Raphson method, 12-2
Nonautonomous dynamic circuits
solutions, boundedness, 7-19–7-20
time-dependent voltage and current source, 7-19
unique asymptotic behavior, 7-20–7-21
Nonlinear Analysis Program version 2, *see* NAP2 program
Nonlinear circuits, synthesis and design
bell-shaped function, 8-30–8-31
collective computation circuitry
Euclidean distance, 8-31–8-32
normalization operation, 8-32
T-norm operator, 8-32–8-33
current transfer piecewise-linear circuitry, 8-18–8-20
dynamic systems, extension, 8-33
electronic function generators, 8-2
explicit algebraic functions, 8-1
Gaussian and bell-shaped basis functions
fuzzy membership functions, 8-8
variance parameter, 8-7
log–antilog multipliers, 8-25–8-26
MOS transistors, Ohmic region, 8-28
multidimensional function, approximation techniques
canonical piecewise linear representations, 8-9
multilayer perceptron, 8-11
neurofuzzy interpolation, 8-10–8-11
radial basis functions, 8-10
sectionwise piecewise-polynomial functions, 8-8–8-9
multiplication circuitry, 8-24–8-25
piecewise-linear and piecewise-polynomial approximants
cubic B-spline, 8-7
Hermite linear basis function, 8-5–8-6
polynomic and rational functions, concepts and techniques, 8-23–8-24
scaling and aggregation circuitry
current scaling, mirror concept, 8-17
signal aggregation, Kirchhoff's current law, 8-18
voltage scaling mechanisms, 8-16
sigmoidal characteristics, 8-28–8-30
square-law multipliers, 8-26–8-27
synthesis problem, hierarchical decomposition, 8-2
transconductance multipliers, 8-27–8-28
transformation circuits
current-to-voltage transformation, 8-13
driving-point characteristics, 8-15–8-16
transfer characteristics, 8-14–8-15
voltage/charge domain transformations, 8-14
voltage-to-current transformation, 8-12–8-13
transresistance piecewise-linear circuitry, 8-20–8-21
unidimensional functions, approximation issues
design nonlinear electronic hardware, 8-2
hardware-oriented approximating functions, 8-4
voltage-to-charge transfer characteristics, piecewise-linear shaping, 8-21–8-22
Nonlinear dynamical circuit equations
normal forms
center manifold theorem, 10-8
constrained differential equations, 10-9
formal series, 10-5
hyperbolic system, 10-8
Lie bracket, 10-6
Poincarè–Andronov–Hopf bifurcation, 10-8
topological types, 10-5
truncated Taylor series, 10-6
tunnel diode characteristics, 10-9
transformation
C^k-conjugate, vector fields, 10-2
equivalence relations technique, 10-1
Jacobian matrix, 10-3–10-4
nonlinear controlled sources, 10-2
solution manifold, 10-1
Nonlinear dynamical *n*-ports equivalence
dynamic element, 10-17
memoryless, nonlinear elements, 10-18
memristor, 10-17
Nonlinear resistive *n*-ports equivalence
capacitive and inductive *n*-ports, 10-16

Chua's synthesis approach, 10-17
Δ−Y/Y−Δ transformations, 10-16
passive *i–v* curve, 10-17
Nonlinear systems identification
dynamic systems, time series embedding theory, 9-31
Gaussian white noise, 9-30
Poincaré map, 9-31
Fock space, Volterra series, 9-29–9-30
least-squares approximation technique, 9-28
Volterra polynomial, 9-29
Nonmonotone current-controlled resistor, 14-55
Nonuniform distributed RC lines
approximate calculation, 18-11
cumulative structure function, 18-9–18-10
lumped element approximation, 18-11–18-12
Riccati differential equation, 18-10
Nullator–*I*-resistor–voltage source tree, 7-9–7-10
Null return difference matrix, 6-6–6-7
Null return ratio, 1-4–1-5, 3-12, 6-6

O

Open-circuit ouput admittance, 19-2
Open-circuit reverse voltage gain, 2-3
Open-circuit transfer admittance, 19-2
Open-loop amplifier
group delay, 1-13
volt–ampere characteristics, 1-11
y-parameter model, 1-12–1-13
Open-loop gain-bandwidth product, 1-10, 1-12
Operational transconductance amplifier (OTA), 11-9
Ordinary differential equations (ODE), 14-2

P

Parallel *L–C* circuit, 19-10
Parallel *L–C–R* circuit, 19-11–19-12
Partial differential equations, 13-10–13-11
Period-doubling bifurcation, 14-23–14-24
Phase shift keying (PSK), 14-66
Picard method, 12-2
Piecewise-linear Chua diode, 14-46
Piecewise-linear (PWL) circuits
advantages, 11-1
computed-aided analysis, 11-19–11-20
current transfer, 8-18–8-20
dynamic circuits, 11-18–11-19
electronic components, 11-8–11-14
GaAs FET model, 11-12–11-13
MOS transistor model, 11-12
op-amp finite-gain model, 11-9
transistor model, 11-9–11-11
hierarchy and representations, 11-3–11-8
bipolar transistor, two-port configuration, 11-3
consistent variation property, 11-5
hysteresis one port resistor, 11-6–11-7
linear complementarity property, 11-5
resistive *n*-port, 11-3
interrelation of methods, 11-2
resistive circuits
algorithm, 11-16–11-18
structural properties, 11-14–11-15
theorem canonical PWL, 11-15–11-16
transresistance, 8-20–8-21
voltage-to-charge transfer characteristics, 8-21–8-22
π-network form, 19-16
Poincaré–Andronov–Hopf bifurcation, 10-8
Poincaré map, 14-15–14-16
Primitive circuit component catalog, 8-34
Propagation constant, 19-5
Pseudorandom sequence generation, 14-64–14-65
PSK, *see* Phase shift keying

Q

Q-factor, 10-10
Qualitative circuit analysis
autonomous dynamic circuits
DC-operating points, convergence, 7-14–7-19
nonlinear capacitor and inductor, 7-13
nonautonomous dynamic circuits
solutions, boundedness, 7-19–7-20
time-dependent voltage and current source, 7-19
unique asymptotic behavior, 7-20–7-21
resistive circuits
DC-operating points, 7-2
Ebers–Moll equations, 7-3
equilibrium points, 7-2
feedback structure, 7-7
monotonic dependent, 7-12–7-13
nonlinear resistor characteristics, 7-6
nullator and norator, 7-3
short–open circuit combinations, 7-7–7-8
V- and *I*-resistors, 7-9
voltage-controlled current source (VCCS), 7-3–7-4
voltage-controlled voltage source (VCVS), 7-2
voltages and currents, bounds, 7-11–7-12
zener diode, 7-2
Quarter-wavelength coupler, 19-20

R

Random chaotic sequence generation, 14-64
Reciprocal nonlinear dynamical circuits, equivalence
Brayton–Moser equations, 10-18–10-19
decomposition, 10-21–10-22
diffeomorphism, 10-19
mixed-potential function, 10-18
piecewise-linear *i–v* characteristics, 10-20
Resistive circuits
DC-operating points, 7-2
Ebers–Moll equations, 7-3

equilibrium points, 7-2
monotonic dependent, 7-12–7-13
nullator and norator, 7-3
solutions, number
feedback structure, 7-7
nonlinear resistor characteristics, 7-6
short–open circuit combinations, 7-7–7-8
V- and *I*-resistors, 7-9
voltage-controlled current source (VCCS), 7-3–7-4
voltage-controlled voltage source (VCVS), 7-2
voltages and currents, bounds, 7-11–7-12
zener diode, 7-2
Return difference
Blecher's procedure
short circuit admittance matrix, 5-3
transistor, common emitter configuration, 5-2–5-3
driving-point impedances
forward transfer admittance, 5-5
indefinite-admittance matrix, 5-4
feedback amplifier theory
amplifier voltage gain, 3-11
definition, 3-7
loop transmission, 3-9
returned voltage, 3-8–3-9
return ratio, 3-9–3-10
voltage-controlled current source, 3-7, 3-9
natural frequencies, 5-1
Return difference matrix, 6-5–6-6, 6-13
Return ratio, 1-4–1-5, 3-9–3-10, 6-5
Riccati differential equation, 18-10

S

Saddle-node bifurcation, 14-23
Scaling and aggregation circuitry
current scaling, mirror concept, 8-17
signal aggregation, Kirchhoff's current law, 8-18
voltage scaling mechanisms, 8-16
Sensitivity matrix
closed-loop transfer function matrix, 6-12
general feedback structure, 6-11
loop-transmission matrix, 6-13
multivariable open-loop control system, 6-11
return difference matrix, 6-13
Series *L*–*C* circuit, 19-9–19-10
Series *L*–*C*–*R* circuit, 19-10–19-11
Series–series feedback amplifier
AC schematic diagram, 2-16–2-17
balanced differential version, 2-20
closed-loop transconductance, 2-18
frequency-dependent feedback factor, 2-19
open-loop transconductance, 2-18
small signal equivalent circuit, 2-17–2-18
volt–ampere characteristics, 2-19
wye feedback subcircuit transformation, 2-20
z-parameter modeling, feedback network, 2-17
Series–series/shunt–shunt feedback amplifier
closed-loop transconductance, 2-21
closed-loop transresistance, 2-22
closed-loop voltage gain, 2-22, 2-24
loop gain, 2-21
Norton equivalent input, 2-23
open-loop transresistance, 2-21–2-22
output and input resistances, 2-22, 2-25
Thévenin equivalent output circuit, 2-23
Series–shunt feedback amplifier
circuit modeling and analysis
AC schematic diagram, 2-2
closed-loop voltage gain, 2-3, 2-6
current-controlled current source, 2-3
h-parameter equivalent circuit, 2-3–2-4
modified small-signal model, 2-5
open-circuit reverse voltage gain, 2-3
open-loop voltage gain, 2-8
resultant closed-loop amplifier, 2-2
short-circuit forward current gain, 2-4
short-circuit input resistance, 2-3
feed-forward compensation
driving-point input and output impedances, 2-9
open-loop voltage gain, 2-8
pertinent small-signal model, 2-7–2-8
resultant frequency-domain feedback factor, 2-9
Series–shunt/shunt–series feedback amplifier
AC schematic diagram, 2-28
circuit voltage gain, 2-25
current gain, 2-27
driving-point input and output resistance, 2-27–2-28
h-parameters, 2-25
system-level diagram, 2-25–2-26
voltage gain, 2-27–2-28
Shil'nikov's theorem, 14-17–14-18
Short-circuit forward current gain, 2-4
Shunt–series feedback amplifier
AC schematic diagram, 2-10
closed-loop current gain, 2-11–2-12
h-parameter and feedback network current, 2-11
open-loop current gain and loop gain, 2-12
Shunt–shunt feedback amplifier
circuit modeling and analysis
AC schematic diagram, 2-12–2-13
closed-loop transresistance, 2-14
feedback factor, 2-13
open-loop transresistance and loop gain, 2-14
y-parameter modeling, feedback network, 2-12
design considerations
balanced differential architecture, 2-16

open-loop amplifier interstage, 2-15
pole splitting, 2-14
Signal (functions) approximation
Chebyshev system, 9-18
general formulation, 9-17
interpolation, approximation
algebraic polynomial interpolation, 9-21
Hermite interpolant, 9-22
Walsh theorem, 9-23
polynomial and rational approximations, 9-18–9-19
splines and radial functions, approximation, 9-20–9-21
Simulated Program with Integrated Circuit Emphasis (SPICE) program, 12-4–12-7
Simulation, nonlinear electrical circuit
circuit-simulation programs
APLAC, 12-8–12-10
DYNAST, 12-17–12-18
ESACAP, 12-14–12-17
NAP2, 12-10–12-14
SPICE, 12-4–12-7
nonlinear algebraic equations, numerical solutions, 12-2–12-3
nonlinear differential equations, numerical integration, 12-3–12-4
Single-loop architectures, 2-29
Smale horseshoe map, 14-18–14-19
S-parameter, 17-6
SPICE program, *see* Simulated Program with Integrated Circuit Emphasis program
Spiral Chua's attractor
broadband ''noise-like'' power spectrum, 14-36
period-doubling cascade, 14-33
time domain randomness, 14-36
Square-law multipliers, 8-26–8-27
State-variable representation, 9-5
System approximation
least-squares approximation and projections
Banach space, 9-13–9-14
discrete data fitting, 9-11
Taylor series representation, 9-12
Volterra kernel sequences, 9-11–9-12
min–max (uniform) approximation
Banach space, 9-15
Hankel operator, 9-16
Volterra polynomials, 9-14
System identification
general formulation, 9-26
least-squares operator approximation, 9-27
linear systems
(ARMA model with exogenous noisy inputs) ARMAX model, 9-27–9-28
autoregressive with moving-average (ARMA) models, 9-27
behavioral approach, 9-28
mainstream theory, characteristics, 9-27
nonlinear dynamic systems, time series
embedding theory, 9-31
Gaussian white noise, 9-30
Poincaré map, 9-31
nonlinear system
Fock space, Volterra series, 9-29–9-30
least-squares approximation technique, 9-28
Volterra polynomial, 9-29
optimal recovery problem, 9-27

T

TDR, *see* Time-domain reflectometry
Telegrapher's equations, multiconductor transmission lines
analytical expressions, configuration
coupled microstrip lines, 16-7
coupled striplines, 16-6–16-7
symmetric stripline, 16-5–16-6
two-wire line, 16-7
C × C matrix, 16-3–16-4
fundamental modes, 16-3
quasitransverse electromagnetic, 16-4–16-5
Thévenin source voltage, 1-4–1-5
Time-domain reflectometry (TDR)
interconnection structure
impedance terminations, 17-4–17-5
reflection coefficients, 17-3
voltage wave, 17-4
one-port
load impedance, 17-2
reflected voltage wave amplitude, 17-1
open-circuit and short-circuit loads
load impedance, 17-2–17-3
reflection coefficient, 17-2
principles, 17-1
Torus trajectory, winding numbers, 14-50–14-51
Transconductance multipliers, 8-27–8-28
Transfer-function matrix
amplifier input admittance, 6-9
Blackman's impedance formula, 6-7
feedback amplifier voltage gain, 6-9
impedance matrix, 6-8
return difference matrix, 6-9–6-10
Transformation circuits
current-to-voltage transformation, 8-13
driving-point characteristics, 8-15–8-16
transfer characteristics, 8-14–8-15
voltage/charge domain transformations, 8-14
voltage-to-current transformation, 8-12–8-13
Transmission lines
circular waveguides
geometric structure, 15-12–15-13
operation modes, 15-13
waveguide parameters, 15-13–15-15
coaxial lines
geometric structure, 15-8
transmission line parameters, 15-8–15-9

wavelength and phase velocity, 15-10
coplanar waveguide
geometric structure, 15-17
transmission line parameters, 15-17–15-18
wavelength and phase velocity, 15-19
generic relations
equivalent circuit, 15-1–15-2
impedance, 15-3–15-5
input admittance, 15-5–15-6
line equations, 15-2
load, voltage reflection coefficient, 15-3–15-4
solutions and propagation constant, 15-2–15-3
voltage reflection coefficient, input, 15-4
wavelength and phase velocity, 15-3
microstrip lines
geometric structure, 15-15
line parameters, 15-15–15-16
wavelength and phase velocity, 15-17
rectangular waveguides
geometric structure, 15-10
operation modes, 15-10–15-11
two-wire lines
geometric structure, 15-6
line parameters, 15-6–15-7
wavelength and phase velocity, 15-8
waveguide parameters, 15-11–15-12
Transverse electromagnetic (TEM) approach, 16-2
Two-wire lines
geometric structure, 15-6
transmission line parameters, 15-6–15-7
wavelength and phase velocity, 15-8

U

Uniform distributed RC lines
frequency domain, 18-6
time domain
diffusion equation, 18-2
Dirac-δ excitation, 18-2–18-3
finite-length, 18-4–18-6
Gauss-like function, 18-2
line transients, 18-3
lossy lines, 18-6–18-7
semi-infinite line, 18-3–18-4
Unity-gain nonlinear sigmoid characteristics, 13-5–13-6

V

Vacuum capacitance matrix, 16-5
van der Pol neon bulb oscillator
circle map
Arnold tongue structure, 14-52–14-53
coupled nonlinear oscillators, 14-51
Devil's staircase, 14-52–14-53
mode-locked region, 14-52
parameter space diagram, 14-52–14-53
circuit model, 14-55–14-56
current waveforms, 14-49–14-50
experimental observations, mode-locking and chaos
C bifurcation sequence, 14-54–14-55
f_s bifurcation sequence, low-amplitude forcing, 14-55–14-56
normalized current pulse pattern, 14-49
sinusoidal voltage source, 14-48–14-49
subsidiary phenomenon, 14-49
winding numbers, 14-50–14-51
Vector field, 14-44–14-45
Voltage amplifier, 7-12
Voltage-controlled current source (VCCS), 7-3–7-4
Voltage-controlled voltage source (VCVS), 7-2
Volterra series representation
kernels, 9-9
nonlinear input-output (I-O) relationship, 9-8
polynomial, 9-9
state transition matrix, 9-8

W

Walsh theorem, 9-23
Winding number, 14-50–14-51

Z

Zener diode, 7-2

Upon its initial publication, *The Circuits and Filters Handbook* broke new ground. It quickly became **the** resource for comprehensive coverage of issues and practical information that can be put to immediate use. Not content to rest on his laurels, in addition to updating the second edition, editor Wai-Kai Chen divided it into tightly focused texts that made the information easily accessible and digestible. These texts have been revised, updated, and expanded so that they continue to provide solid coverage of standard practices and enlightened perspectives on new and emerging techniques.

Feedback, Nonlinear, and Distributed Circuits draws together international contributors who discuss feedback amplifier theory and explore feedback amplifier configurations. They develop Bode's feedback theory as an example of general feedback theory. The coverage then moves on to the importance of complementing numerical analysis with qualitative analysis to get a global picture of a circuit's performance. After reviewing a wide range of approximation techniques and circuit design styles for discrete and monolithic circuits, the book presents a comprehensive description of the use of piecewise-linear methods in modeling, analysis, and structural properties of nonlinear circuits. It describes the analysis and synthesis of distributed circuits, and the circuit modeling in the frequency domain of uniform multiconductor transmission lines based on the Telegrapher's equations and covers frequency and time domain experimental characterization techniques for uniform and nonuniform multiconductor structures.

Features:

- Covers single-loop and multiple-loop feedback amplifiers
- Provides a brief description of mathematical representation, approximation, and identification of nonlinear systems
- Discusses general methods for the analysis of vector fields that have at least one nonhyperbolic fixed point
- Examines the simulation or analysis of a nonlinear electrical circuit by means of a computer program
- Presents a comprehensive analysis and synthesis of distributed circuits

This volume will undoubtedly take its place as the engineer's first choice in looking for solutions to problems encountered when dealing with feedback, linear, and distributed circuits.

58819

ISBN: 978-1-4200-5881-9
90000
9 781420 058819

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

The Circuits and Filters Handbook

Third Edition

Passive, Active, and Digital Filters

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Edited by

Wai-Kai Chen

Fundamentals of Circuits and Filters

Feedback, Nonlinear, and Distributed Circuits

Analog and VLSI Circuits

Computer Aided Design and Design Automation

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Passive, Active, and Digital Filters

Edited by

Wai-Kai Chen

University of Illinois

Chicago, U. S. A.

CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the

Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5885-7 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Passive, active, and digital filters / edited by Wai-Kai Chen.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-4200-5885-7
ISBN-10: 1-4200-5885-1
1. Electric filters, Digital. 2. Electric filters, Passive. 3. Electric filters, Active. I. Chen, Wai-Kai, 1936- II. Title.

TK7872.F5P375 2009
621.3815'324--dc22 2008048117

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface ix

Editor-in-Chief xi

Contributors xiii

SECTION I Passive Filters

1 General Characteristics of Filters **1**-1
Andreas Antoniou

2 Approximation **2**-1
Artice M. Davis

3 Frequency Transformations **3**-1
Jaime Ramirez-Angulo

4 Sensitivity and Selectivity **4**-1
Igor M. Filanovsky

5 Passive Immittances and Positive-Real Functions **5**-1
Wai-Kai Chen

6 Passive Cascade Synthesis **6**-1
Wai-Kai Chen

7 Synthesis of LCM and RC One-Port Networks **7**-1
Wai-Kai Chen

8 Two-Part Synthesis by Ladder Development **8**-1
Wai-Kai Chen

9 Design of Resistively Terminated Networks **9**-1
Wai-Kai Chen

10 Design of Broadband Matching Networks **10**-1
Wai-Kai Chen

SECTION II Active Filters

11 Low-Gain Active Filters **11**-1
Phillip E. Allen, Benjamin J. Blalock, and Stephen W. Milam

12 Single-Amplifier Multiple-Feedback Filters **12**-1
F. William Stephenson

13 Multiple-Amplifier Biquads **13**-1
Norbert J. Fliege

14 The Current Generalized Immittance Converter Biquads **14**-1
Wasfy B. Mikhael

15 High-Order Filters **15**-1
Rolf Schaumann

16 Continuous-Time Integrated Filters **16**-1
Rolf Schaumann

17 Switched-Capacitor Filters **17**-1
Jose Silva-Martinez and Edgar Sánchez-Sinencio

SECTION III Digital Filters

18 FIR Filters **18**-1
M. H. Er, Andreas Antoniou, Yong Ching Lim, and Tapio Saramäki

19 IIR Filters **19**-1
Sawasd Tantaratana, Chalie Charoenlarpnopparut, Phakphoom Boonyanant, and Yong Ching Lim

20 Finite Wordlength Effects **20**-1
Bruce W. Bomar

21 VLSI Implementation of Digital Filters **21**-1
Joseph B. Evans and Timothy R. Newman

22 Two-Dimensional FIR Filters **22**-1
Rashid Ansari and A. Enis Cetin

23 Two-Dimensional IIR Filters **23**-1
A. G. Constantinides and Xiaojian Xu

24 1-D Multirate Filter Banks **24**-1
Nick G. Kingsbury and David B. H. Tay

25 Directional Filter Banks **25**-1
Jose Gerardo Rosiles and Mark J. T. Smith

26 Nonlinear Filtering Using Statistical Signal Models **26**-1
Kenneth E. Barner, Tuncer C. Aysal, and Gonzalo R. Arce

27 Nonlinear Filtering for Image Denoising **27**-1
Nasir M. Rajpoot, Zhen Yao, and Roland G. Wilson

28 Video Demosaicking Filters **28**-1
Bahadir K. Gunturk and Yucel Altunbasak

Index **IN**-1

Preface

As circuit complexity continues to increase, the microelectronic industry must possess the ability to quickly adapt to the market changes and new technology through automation and simulations. The purpose of this book is to provide in a single volume a comprehensive reference work covering the broad spectrum of filter design from passive, active, to digital. The book is written and developed for the practicing electrical engineers and computer scientists in industry, government, and academia. The goal is to provide the most up-to-date information in the field.

Over the years, the fundamentals of the field have evolved to include a wide range of topics and a broad range of practice. To encompass such a wide range of knowledge, this book focuses on the key concepts, models, and equations that enable the design engineer to analyze, design, and predict the behavior of large-scale systems that employ various types of filters. While design formulas and tables are listed, emphasis is placed on the key concepts and theories underlying the processes.

This book stresses fundamental theory behind professional applications and uses several examples to reinforce this point. Extensive development of theory and details of proofs have been omitted. The reader is assumed to have a certain degree of sophistication and experience. However, brief reviews of theories, principles, and mathematics of some subject areas are given. These reviews have been done concisely with perception.

The compilation of this book would not have been possible without the dedication and efforts of Professor Rashid Ansari and Dr. A. Enis Cetin, and most of all the contributing authors. I wish to thank them all.

Wai-Kai Chen

Editor-in-Chief



Wai-Kai Chen is a professor and head emeritus of the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. He received his BS and MS in electrical engineering at Ohio University, where he was later recognized as a distinguished professor. He earned his PhD in electrical engineering at the University of Illinois at Urbana–Champaign.

Professor Chen has extensive experience in education and industry and is very active professionally in the fields of circuits and systems. He has served as a visiting professor at Purdue University, the University of Hawaii at Manoa, and Chuo University in Tokyo, Japan. He was the editor-in-chief of the *IEEE Transactions on Circuits and Systems, Series I and II*, the president of the IEEE Circuits and Systems Society, and is the founding editor and the editor-in-chief of the *Journal of Circuits, Systems and Computers*. He received the Lester R. Ford Award from the Mathematical Association of America; the Alexander von Humboldt Award from Germany; the JSPS Fellowship Award from the Japan Society for the Promotion of Science; the National Taipei University of Science and Technology Distinguished Alumnus Award; the Ohio University Alumni Medal of Merit for Distinguished Achievement in Engineering Education; the Senior University Scholar Award and the 2000 Faculty Research Award from the University of Illinois at Chicago; and the Distinguished Alumnus Award from the University of Illinois at Urbana–Champaign. He is the recipient of the Golden Jubilee Medal, the Education Award, and the Meritorious Service Award from the IEEE Circuits and Systems Society, and the Third Millennium Medal from the IEEE. He has also received more than a dozen honorary professorship awards from major institutions in Taiwan and China.

A fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the American Association for the Advancement of Science (AAAS), Professor Chen is widely known in the profession for the following works: *Applied Graph Theory* (North-Holland), *Theory and Design of Broadband Matching Networks* (Pergamon Press), *Active Network and Feedback Amplifier Theory* (McGraw-Hill), *Linear Networks and Systems* (Brooks/Cole), *Passive and Active Filters: Theory and Implements* (John Wiley), *Theory of Nets: Flows in Networks* (Wiley-Interscience), *The Electrical Engineering Handbook* (Academic Press), and *The VLSI Handbook* (CRC Press).

Contributors

Phillip E. Allen
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia

Yucel Altunbasak
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia

Rashid Ansari
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, Illinois

Andreas Antoniou
Department of Electrical and Computer Engineering
University of Victoria
Victoria, British Columbia, Canada

Gonzalo R. Arce
Electrical and Computer Engineering Department
University of Delaware
Newark, Delaware

Tuncer C. Aysal
School of Electrical and Computer Engineering
Cornell University
Ithaca, New York

Kenneth E. Barner
Department of Electrical and Computer Engineering
University of Delaware
Newark, Delaware

Benjamin J. Blalock
Department of Electrical Engineering and Computer Science
The University of Tennessee
Knoxville, Tennessee

Bruce W. Bomar
Department of Electrical and Computer Engineering
The University of Tennessee Space Institute
Tullahoma, Tennessee

Phakphoom Boonyanant
National Electronics and Computer Technology Center
Pathumthani, Thailand

A. Enis Cetin
Department of Electrical and Electronics Engineering
Bilkent University
Ankara, Turkey

Chalie Charoenlarpnopparut
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand

Wai-Kai Chen
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, Illinois

A. G. Constantinides
Department of Electrical and Electronic Engineering
Imperial College of Science, Technology and Medicine
London, England

Artice M. Davis
Department of Electrical Engineering
San Jose State University
San Jose, California

M. H. Er
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore

Joseph B. Evans
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, Kansas

Igor M. Filanovsky
Department of Electrical Engineering and Computer Technology
University of Alberta
Edmonton, Alberta, Canada

Norbert J. Fliege
Department of Electrical Engineering and Computer Engineering
University of Mannheim
Mannheim, Germany

Bahadir K. Gunturk
Department of Electrical and Computer Engineering
Louisiana State University
Baton Rouge, Louisiana

Nick G. Kingsbury
Department of Engineering
Trinity College
University of Cambridge
Cambridge, United Kingdom

Yong Ching Lim
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore

Wasfy B. Mikhael
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, Florida

Stephen W. Milam
RF Micro-Devices
Greensboro, North Carolina

Timothy R. Newman
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, Kansas

Nasir M. Rajpoot
Department of Computer Science
University of Warwick
Coventry, United Kingdom

Jaime Ramirez-Angulo
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, New Mexico

Jose Gerardo Rosiles
Electrical and Computer Engineering Department
The University of Texas at El Paso
El Paso, Texas

Edgar Sánchez-Sinencio
Department of Electrical and Computer Engineering
Texas A&M University
College Station, Texas

Tapio Saramäki
Institute of Signal Processing
Tampere University of Technology
Tampere, Finland

Rolf Schaumann
Department of Electrical Engineering
Portland State University
Portland, Oregon

Jose Silva-Martinez
Department of Electrical and Computer Engineering
Texas A&M University
College Station, Texas

Mark J. T. Smith
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

F. William Stephenson
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia

Sawasd Tantaratana
Sirindhorn International Institute of Technology
Thammasat University
Pathumthani, Thailand

David B. H. Tay
Department of Electronic Engineering
Latrobe University
Bundoora, Victoria, Australia

Roland G. Wilson
Department of Computer Science
University of Warwick
Coventry, United Kingdom

Xiaojian Xu
School of Electronic and Information Engineering
Beihang University
Beijing, China

Zhen Yao
Department of Computer Science
University of Warwick
Coventry, United Kingdom

I

Passive Filters

Wai-Kai Chen
University of Illinois at Chicago

1 General Characteristics of Filters *Andreas Antoniou* **1**-1
Introduction • Characterization • Time-Domain Response • Frequency-Domain Analysis • Ideal and Practical Filters • Amplitude and Delay Distortion • Minimum-Phase, Nonminimum-Phase, and Allpass Filters • Introduction to the Design Process • Introduction to Realization • References

2 Approximation *Artice M. Davis* **2**-1
Introduction • Butterworth LPP Approximation • Chebyshev LPP Approximation • Bessel–Thompson LPP Approximation • Elliptic Approximation • References

3 Frequency Transformations *Jaime Ramirez-Angulo* **3**-1
Low-Pass Prototype • Frequency and Impedance Scaling • Low-Pass to High-Pass Transformation • Low-Pass to Bandpass Transformation • Low-Pass to Band-Reject Transformation

4 Sensitivity and Selectivity *Igor M. Filanovsky* **4**-1
Introduction • Definitions of Sensitivity • Function Sensitivity to One Variable • Coefficient Sensitivity • Root Sensitivities • Statistical Model for One Variable • Multiparameter Sensitivities and Sensitivity Measures • Sensitivity Invariants • Sensitivity Bounds • Remarks on the Sensitivity Applications • Sensitivity Computations Using the Adjoint Network • General Methods of Reducing Sensitivity • Cascaded Realization of Active Filters • Simulation of Doubly Terminated Matched Lossless Filters • Sensitivity of Active *RC* Filters • Errors in Sensitivity Comparisons • References

5 Passive Immittances and Positive-Real Functions *Wai-Kai Chen* **5**-1
References

6 Passive Cascade Synthesis *Wai-Kai Chen* **6**-1
Introduction • Type-E Section • Richards Section • Darlington Type-D Section • References

7 Synthesis of LCM and RC One-Port Networks *Wai-Kai Chen* **7**-1
Introduction • LCM One-Port Networks • RC One-Port Networks • References

8 Two-Part Synthesis by Ladder Development *Wai-Kai Chen* **8**-1
Introduction • LC Ladder • RC Ladder • Parallel or Series Ladders • Reference

9 Design of Resistively Terminated Networks *Wai-Kai Chen* **9**-1
Introduction • Double-Terminated Butterworth Networks • Double-Terminated Chebyshev Networks • References

10 Design of Broadband Matching Networks *Wai-Kai Chen* **10**-1
Introduction • Basic Coefficient Constraints • Design Procedure • Explicit Formulas for the RLC Load • References

1

General Characteristics of Filters

1.1 Introduction **1**-1
1.2 Characterization **1**-3
Laplace Transform • Transfer Function
1.3 Time-Domain Response **1**-6
General Inversion Formula • Inverse by Using Partial Fractions • Impulse and Step Responses • Overshoot, Delay Time, and Rise Time
1.4 Frequency-Domain Analysis **1**-10
Sinusoidal Response • Graphical Construction • Loss Function
1.5 Ideal and Practical Filters **1**-15
1.6 Amplitude and Delay Distortion **1**-16
1.7 Minimum-Phase, Nonminimum-Phase, and Allpass Filters **1**-17
Minimum-Phase Filters • Allpass Filters • Decomposition of Nonminimum-Phase Transfer Functions
1.8 Introduction to the Design Process **1**-21
The Approximation Step • The Realization Step • Study of Imperfections • Implementation
1.9 Introduction to Realization **1**-23
Passive Filters • Active Filters • Biquads • Types of Basic Filter Sections
References **1**-29

Andreas Antoniou
University of Victoria

1.1 Introduction

An electrical filter is a system that can be used to modify, reshape, or manipulate the frequency spectrum of an electrical signal according to some prescribed requirements. For example, a filter may be used to amplify or attenuate a range of frequency components, reject or isolate one specific frequency component, and so on. The applications of electrical filters are numerous, for example,

- To eliminate signal contamination such as noise in communication systems
- To separate relevant from irrelevant frequency components
- To detect signals in radios and TV's
- To demodulate signals
- To bandlimit signals before sampling
- To convert sampled signals into continuous-time signals
- To improve the quality of audio equipment, e.g., loudspeakers
- In time-division to frequency-division multiplex systems

- In speech synthesis
- In the equalization of transmission lines and cables
- In the design of artificial cochleas

Typically, an electrical filter receives an input signal or *excitation* and produces an output signal or *response*. The frequency spectrum of the output signal is related to that of the input by some rule of correspondence. Depending on the type of input, output, and internal operating signals, three general types of filters can be identified, namely, continuous-time, sampled-data, and discrete-time filters.

A continuous-time signal is one that is defined at each and every instant of time. It can be represented by a function $x(t)$ whose domain is a range of numbers (t_1, t_2), where $-\infty \leq t_1$ and $t_2 \leq \infty$. A sampled-data or impulse-modulated signal is one that is defined in terms of an infinite summation of continuous-time impulses (see Ref. [1, Chapter 6]). It can be represented by a function

$$\hat{x}(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$

where $\delta(t)$ is the impulse function. The value of the signal at any instant in the range $nT < t < (n+1)T$ is zero. The frequency spectrum of a continuous-time or sampled-data signal is given by the Fourier transform.*

A discrete-time signal is one that is defined at discrete instants of time. It can be represented by a function $x(nT)$, where T is a constant and n is an integer in the range (n_1, n_2) such that $-\infty \leq n_1$ and $n_2 \leq \infty$. The value of the signal at any instant in the range $nT < t < (n+1)T$ can be zero, constant, or undefined depending on the application. The frequency spectrum in this case is obtained by evaluating the z transform on the unit circle $|z| = 1$ of the z plane.

Depending on the format of the input, output, and internal operating signals, filters can be classified either as analog or digital filters. In analog filters the operating signals are varying voltages and currents, whereas in digital filters they are encoded in some binary format. Continuous-time and sampled-data filters are always analog filters. However, discrete-time filters can be analog or digital.

Analog filters can be classified on the basis of their constituent components as

- Passive *RLC* filters
- Crystal filters
- Mechanical filters
- Microwave filters
- Active *RC* filters
- Switched-capacitor filters

Passive *RLC* filters comprise resistors, inductors, and capacitors. Crystal filters are made of piezoelectric resonators that can be modeled by resonant circuits. Mechanical filters are made of mechanical resonators. Microwave filters consist of microwave resonators and cavities that can be represented by resonant circuits. Active *RC* filters comprise resistors, capacitors, and amplifiers; in these filters, the performance of resonant circuits is simulated through the use of feedback or by supplying energy to a passive circuit. Switched-capacitor filters comprise resistors, capacitors, amplifiers, and switches. These are discrete-time filters that operate like active filters but through the use of switches the capacitance values can be kept very small. As a result, switched-capacitor filters are amenable to VLSI implementation.

This section provides an introduction to the characteristics of analog filters. Their basic characterization in terms of a differential equation is reviewed in Section 1.2 and by applying the Laplace transform, an algebraic equation is deduced that leads to the s-domain representation of a filter. The representation of analog filters in terms of the transfer function is then developed. Using the transfer function, one can

* See Chapter 4 of *Fundamentals of Circuits and Filters*.

obtain the time-domain response of a filter to an arbitrary excitation, as shown in Section 1.3. Some important time-domain responses, i.e., the impulse and step responses, are examined. Certain filter parameters related to the step response, namely, the overshoot, delay time, and rise time, are then considered. The response of a filter to a sinusoidal excitation is examined in Section 1.4 and is then used to deduce the basic frequency-domain representations of a filter, namely, its frequency response and loss characteristic. Some idealized filter characteristics are then identified and the differences between idealized and practical filters are delineated in Section 1.5. Practical filters tend to introduce signal degradation through amplitude and/or delay distortion. The causes of these types of distortion are examined in Section 1.6. In Section 1.7, certain special classes of filters, e.g., minimum-phase and allpass filters, are identified and their applications mentioned. This chapter concludes with a review of the design process and the tasks that need to be undertaken to translate a set of filter specifications into a working prototype.

1.2 Characterization

A linear causal analog filter with input $x(t)$ and output $y(t)$ can be characterized by a differential equation of the form

$$b_n \frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n} + b_{n-1} \frac{\mathrm{d}^{n-1} y(t)}{\mathrm{d}t^{n-1}} + \cdots + b_0 y(t) = a_n \frac{\mathrm{d}^n x(t)}{\mathrm{d}t^n} + a_{n-1} \frac{\mathrm{d}^{n-1} x(t)}{\mathrm{d}t^{n-1}} + \cdots + a_0 x(t)$$

The coefficients $a_0, a_1, \ldots, a_n$ and $b_0, b_1, \ldots, b_n$ are functions of the element values and are real if the parameters of the filter (e.g., resistances, inductances, etc.) are real. If they are independent of time, the filter is time invariant. The input $x(t)$ and output $y(t)$ can be either voltages or currents. The order of the differential equation is said to be the *order* of the filter.

An analog filter must of necessity incorporate reactive elements that can store energy. Consequently, the filter can produce an output even in the absence of an input. The output on such an occasion is caused by the initial conditions of the filter, namely,

$$\left.\frac{\mathrm{d}^{n-1} y(t)}{\mathrm{d}t^{n-1}}\right|_{t=0}, \left.\frac{\mathrm{d}^{n-2} y(t)}{\mathrm{d}t^{n-2}}\right|_{t=0}, \ldots, y(0)$$

The response in such a case is said to be the *zero-input response*. The response obtained if the initial conditions are zero is sometimes called the *zero-state response*.

1.2.1 Laplace Transform

The most important mathematical tool in the analysis and design of analog filters is the Laplace transform. It owes its widespread application to the fact that it transforms differential into algebraic equations that are a lot easier to manipulate. The Laplace transform of $x(t)$ is defined as*

$$X(s) = \int_{-\infty}^{\infty} x(t) e^{-st} \mathrm{d}t$$

where s is a complex variable of the form $s = \sigma + j\omega$. Signal $x(t)$ can be recovered from $X(s)$ by applying the inverse Laplace transform, which is given by

* See Chapter 3 by J. R. Deller, Jr. in *Fundamentals of Circuits and Filters*.

$$x(t) = \frac{1}{2\pi j} \int_{C-j\infty}^{C+j\infty} X(s)e^{st}\,\mathrm{d}s$$

where C is a positive constant. A shorthand notation of the Laplace transform and its inverse are

$$X(s) = \mathcal{L}x(t) \quad \text{and} \quad x(t) = \mathcal{L}^{-1}X(s)$$

Alternatively,

$$X(s) \leftrightarrow x(t)$$

A common practice in the choice of symbols for the Laplace transform and its inverse is to use upper case for the s domain and lower case for the time domain.

On applying the Laplace transform to the nth derivative of some function of time $y(t)$, we find that

$$\mathcal{L}\left[\frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n}\right] = s^n Y(s) - s^{n-1}y(0) - s^{n-2}\left.\frac{\mathrm{d}y(t)}{\mathrm{d}t}\right|_{t=0} - \cdots - \left.\frac{\mathrm{d}^{n-1}y(t)}{\mathrm{d}t^{n-1}}\right|_{t=0}$$

Now, on applying the Laplace transform to an nth-order differential equation with constant coefficients, we obtain

$$\left(b_n s^n + b_{n-1}s^{n-1} + \cdots + b_0\right)Y(s) + \Psi_y(s) = \left(a_n s^n + a_{n-1}s^{n-1} + \cdots + a_0\right)X(s) + \Psi_x(s)$$

where

$X(s)$ and $Y(s)$ are the Laplace transforms of the input and output, respectively

$\Psi_x(s)$ and $\Psi_y(s)$ are functions that combine all the initial-condition terms that depend on $x(t)$ and $y(t)$, respectively

1.2.2 Transfer Function

An important s-domain characterization of an analog filter is its *transfer function*, as for any other linear system. This is defined as the ratio of the Laplace transform of the response to the Laplace transform of the excitation.

An arbitrary linear, time-invariant, continuous-time filter, which may or may not be causal, can be represented by the convolution integral

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)\mathrm{d}\tau = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\mathrm{d}\tau$$

where $h(t)$ is the impulse response of the filter. The Laplace transform yields

$$\begin{aligned} Y(s) &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} h(t-\tau)x(\tau)\mathrm{d}\tau\right]e^{-st}\mathrm{d}t \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t-\tau)e^{-st}x(\tau)\mathrm{d}\tau\ \mathrm{d}t \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t-\tau)e^{-st}\cdot e^{s\tau}\cdot e^{-s\tau}x(\tau)\mathrm{d}\tau\ \mathrm{d}t \end{aligned}$$

Changing the order of integration, we obtain

$$Y(s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(t-\tau)e^{-s(t-\tau)} \cdot x(\tau)e^{-s\tau} \mathrm{d}t \ \mathrm{d}\tau$$
$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(t-\tau)e^{-s(t-\tau)} \mathrm{d}t \cdot x(\tau)e^{-s\tau} \mathrm{d}\tau$$

Now, if we let $t = t' + \tau$, then $\mathrm{d}t/\mathrm{d}t' = 1$ and $t - \tau = t'$; hence,

$$Y(s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(t')e^{-st'} \mathrm{d}t' \cdot x(\tau)e^{-s\tau} \mathrm{d}\tau$$
$$= \int_{-\infty}^{\infty} h(t')e^{-st'} \mathrm{d}t' \cdot \int_{-\infty}^{\infty} x(\tau)e^{-s\tau} \mathrm{d}\tau$$
$$= H(s)X(s)$$

Therefore, the transfer function is given by

$$H(s) = \frac{Y(s)}{X(s)} = \mathcal{L}h(t) \tag{1.1}$$

In effect, the transfer function is equal to the Laplace transform of the impulse response.

Some authors define the transfer function as the Laplace transform of the impulse response. Then through the use of the convolution integral, they show that the transfer function is equal to the ratio of the Laplace transform of the response to the Laplace transform of the excitation. The two definitions are, of course, equivalent.

Typically, in analog filters the input and output are voltages, e.g., $x(t) + v_i(t)$ and $y(t) + v_o(t)$. In such a case the transfer function is given by

$$\frac{V_o(s)}{V_i(s)} = H_V(s)$$

or simply by

$$\frac{V_o}{V_i} = H_V(s)$$

However, on occasion the input and output are currents, in which case

$$\frac{I_o(s)}{I_i(s)} \equiv \frac{I_o}{I_i} = H_I(s)$$

The transfer function can be obtained through network analysis using one of several classical methods,* e.g., by using

* See Chapters 18 through 27 of this volume.

- Kirchhoff's voltage and current laws
- Matrix methods
- Flow graphs
- Mason's gain formula
- State-space methods

A transfer function is said to be *realizable* if it characterizes a stable and causal network. Such a transfer function must satisfy the following constraints:

1. It must be a rational function of s with real coefficients.
2. Its poles must lie in the left-half s plane.
3. The degree of the numerator polynomial must be equal to or less than that of the denominator polynomial.

A transfer function may represent a network comprising elements with real parameters only if its coefficients are real. The poles must be in the left-half s plane to ensure that the network is stable and the numerator degree must not exceed the denominator degree to assure the existence of a causal network.

1.3 Time-Domain Response

From Equation 1.1,

$$Y(s) = H(s)X(s)$$

Therefore, the time-domain response of a filter to some arbitrary excitation can be deduced by obtaining the inverse Laplace transform of $Y(s)$, i.e.,

$$y(t) = \mathcal{L}^{-1}\{H(s)X(s)\}$$

1.3.1 General Inversion Formula

If

1. the singularities of $Y(s)$ in the finite plane are poles,* and
2. $Y(s) \to 0$ uniformly with respect to the angle of s as $|s| \to \infty$ with $\sigma \leq C$, where C is a positive constant, then [2]

$$y(t) = \begin{cases} 0 & \text{for } t < 0 \\ \frac{1}{2\pi j} \int_{C-j\infty}^{C+j\infty} Y(s)e^{st}\,ds = \frac{1}{2\pi j} \int_{\Gamma} Y(s)e^{st}\,ds & \text{for } t \geq 0 \end{cases} \tag{1.2}$$

where Γ is a contour in the counterclockwise sense make up of the part of the circle $s = Re^{j\theta}$ to the left of line $s = C$ and the segment of the line $s = C$ that overlaps the circle, as depicted in Figure 1.1; C and R are sufficiently large to ensure that Γ encloses all the finite poles of $Y(s)$.

From the residue theorem [3] and Equation 1.2, we have

$$y(t) = \begin{cases} 0 & \text{for } t < 0 \\ \frac{1}{2\pi j} \int_{\Gamma} Y(s)e^{st}\,ds = \sum_{i=1}^{K} \operatorname*{res}_{s=p_i} Y_0(s) & \text{for } t \geq 0 \end{cases}$$

* Such a function is said to be meromorphic [2,3].

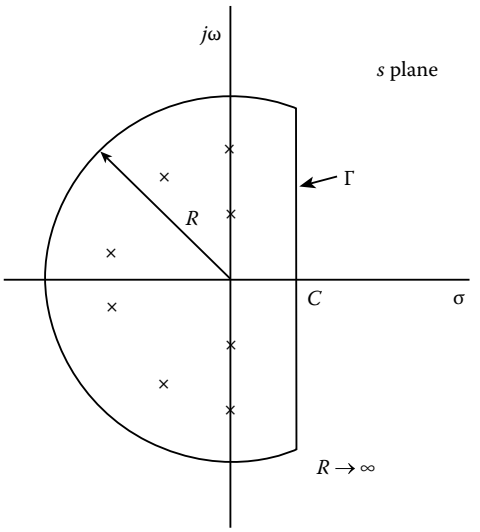


FIGURE 1.1 Contour Γ for the evaluation of the inverse Laplace transform.

where $Y_0(s) = Y(s)e^{st}$ and K is the number of poles in $Y(s)$. If $Y_0(s)$ has a pole p_i of order m_i, the residue can be obtained by using the general formula [3]

$$\operatorname*{res}_{z=p_i} Y_0(s) = \frac{1}{(m_i - 1)!} \lim_{s \to p_i} \frac{\mathrm{d}^{m_i - 1}}{\mathrm{d}s^{m_i - 1}} [(s - p_i)^{m_i} Y_0(s)]$$

Note that complex poles yield complex residues. Hence, like the poles of $Y_0(s)$, its residues occur in complex–conjugate pairs. For this reason, $y(t)$ is found to be a real function of t, as can be easily verified.

Condition 1 listed previously may not be satisfied sometimes, for example, if

$$\lim_{s \to \infty} Y(s) = A_0$$

where A_0 is a constant. In such a case, we can express $Y(s)$ as

$$Y(s) = A_0 + Y'(s)$$

where $Y'(s)$ satisfies conditions 1 and 2. Thus,

$$y(t) = A_0 \delta(t) + \mathcal{L}^{-1} Y'(s)$$

The inverse Laplace transform of $Y'(s)$ can now be obtained by using the inversion formula.

1.3.2 Inverse by Using Partial Fractions

The simplest way to obtain the time-domain response of a filter is to express $H(s)X(s)$ as a partial-fraction expansion and then invert the resulting fractions individually. If $Y(s)$ has simple poles, we can write

$$Y(s) = A_0 + \sum_{i=1}^{K} \frac{A_i}{s - p_i}$$

where A_0 is a constant and

$$A_i = \lim_{s \to p_i} [(s - p_i) Y(s)]$$

is the residue of pole $s = p_i$. On applying the general inversion formula to each partial fraction, we obtain

$$y(t) = A_0 \delta(t) + u(t) \sum_{i=1}^{K} A_i e^{p_i t}$$

where $\delta(t)$ and $u(t)$ are the impulse function and unit step, respectively.

1.3.3 Impulse and Step Responses

The response of a filter to an impulse $\delta(t)$ designated as

$$y(t) = \mathcal{R}\delta(t) \equiv h(t)$$

where $\mathcal{R}$ is an operator, is of considerable importance. Its absolute integrability guarantees the stability of the filter* and its Laplace transform, namely, $H(s)$, is the transfer function as has been shown in the section on the transfer function.

For an Nth-order, causal, linear, and time-invariant filter

$$H(s) = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_M s^M}{b_0 + b_1 s + b_2 s^2 + \cdots + b_N s^N}$$

where $M \leq N$.

The step (or unit-step) response is the output of a filter to the signal

$$u(t) = \begin{cases} 1 & \text{for } t \geq 0 \\ 0 & \text{for } t < 0 \end{cases}$$

The Laplace transform of $u(t)$ is $1/s$. Hence, the step response of an arbitrary filter is obtained as

$$y(t) = \mathcal{R}u(t) \equiv y_u(t) = \mathcal{L}^{-1}\left[\frac{H(s)}{s}\right]$$

1.3.4 Overshoot, Delay Time, and Rise Time

Three time-domain parameters of a filter are usually associated with the step response [4], namely, the overshoot, delay time, and rise time. The *overshoot* γ is the difference between the peak value and the asymptotic value of the step response in percent as $t \to \infty$. The *delay time* τ_d is the time required for the step response to reach 50% of the asymptotic value. The *rise time* τ_r is the time required for the step response to increase from 10% to 90% of the asymptotic value. These three parameters are illustrated in Figure 1.2, where $K = a_0/b_0$ is a scaling constant that normalizes the asymptotic value of the step response as $t \to \infty$ to unity.

The delay and rise times defined in terms of the step response entail quite a bit of computation. Alternative definitions of these parameters that are easier to use have been proposed by Elmore [4]. These are based on the impulse response and give accurate results if the overshoot is small. The delay time is defined as

$$\tau_{\mathrm{D}} = \int_0^\infty t h(t)\mathrm{d}t$$

and the rise time assumes the form

$$\tau_{\mathrm{R}} = \left[2\pi \int_0^\infty (t - \tau_{\mathrm{D}})^2 h(t)\mathrm{d}t\right]^{1/2} = \sqrt{2\pi}\left[\int_0^\infty t^2 h(t)\mathrm{d}t - \tau_{\mathrm{D}}^2\right]^{1/2}$$

* See Section 22.1 of *Fundamentals of Circuits and Filters.*

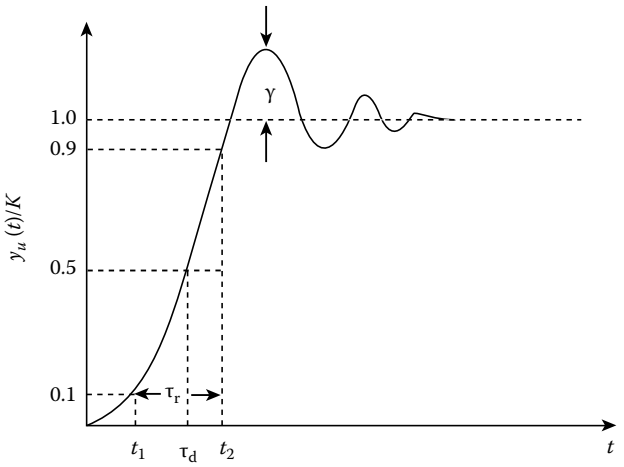


FIGURE 1.2 Overshoot, delay time, and rise time.

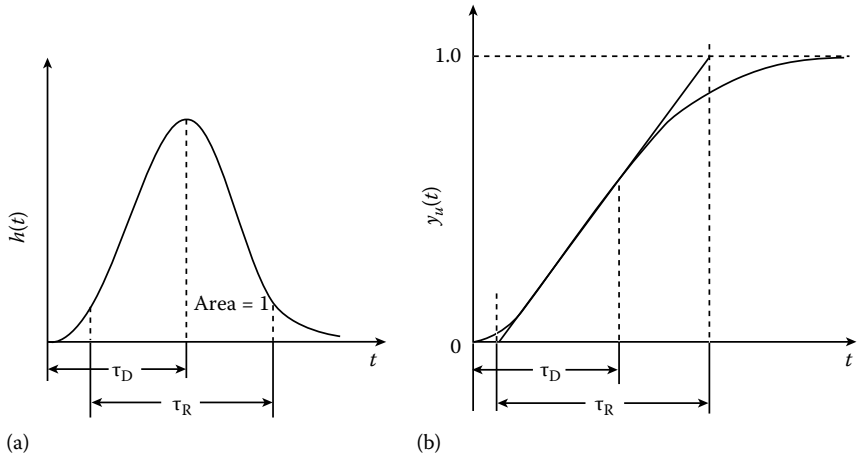


FIGURE 1.3 Physical interpretation of Elmore's definitions of delay and rise times: (a) impulse response $h(t)$ and (b) unit-step response $y_u(t)$.

The physical interpretation of these parameters is illustrated in Figure 1.3a and b. If the overshoot is small, say less than 1%, then

$$\tau_D \approx \tau_d \quad \text{and} \quad \tau_R \approx \tau_r$$

The simplification brought about by Elmore's definitions can be easily demonstrated. Consider a filter whose step response approaches unity as $t \to \infty$. Such a filter has a transfer function of the form

$$H(s) = \frac{1 + a_1 s + a_2 s^2 + \cdots + a_M s^M}{1 + b_1 s + b_2 s^2 + \cdots + b_N s^N} \tag{1.3}$$

that is, $a_0 = b_0 = 1$. From the definition of the Laplace transform,

$$\begin{aligned} H(s) &= \int_0^\infty h(t)e^{-st}\mathrm{d}t \\ &= \int_0^\infty h(t)\left(1 - st + \frac{s^2t^2}{2!} - \cdots\right)\mathrm{d}t \\ &= \int_0^\infty h(t)\mathrm{d}t - s\int_0^\infty th(t)\mathrm{d}t + \frac{s^2}{2!}\int_0^\infty t^2h(t)\mathrm{d}t - \cdots \\ &= \int_0^\infty h(t)\mathrm{d}t - s\tau_\mathrm{D} + \frac{s^2}{2!}\left(\frac{\tau_\mathrm{R}^2}{2\pi} + \tau_\mathrm{D}^2\right) - \cdots \end{aligned} \tag{1.4}$$

Alternatively, from Equation 1.3, direct division gives

$$H(s) = 1 - (b_1 - a_1)s + \left(b_1^2 - a_1b_1 + a_2 - b_2\right)s^2 + \cdots \tag{1.5}$$

Now by comparing Equations 1.4 and 1.5, we deduce

$$\int_0^\infty h(t)\mathrm{d}t = 1, \quad \tau_\mathrm{D} = b_1 - a_1$$

and

$$\tau_\mathrm{R} = \left\{2\pi\left[b_1^2 - a_1^2 + 2(a_2 - b_2)\right]\right\}^{1/2}$$

The previous definitions are based on the assumption that the unit-step response approaches unity as $t \to \infty$. If this is not the case, i.e., coefficients a_0 and b_0 are not equal to unity, then we can write

$$H(s) = KH'(s)$$

where $K = a_0/b_0$ and

$$H'(s) = \frac{1 + a_1's + a_2's^2 + \cdots + a_M's^M}{1 + b_1's + b_2's^2 + \cdots + b_N's^N}$$

Using the coefficients of $H'(s)$ in the formulas for τ_D and τ_R yields approximate values for the delay time and rise time, since these parameters are independent of the absolute value of the step response.

1.4 Frequency-Domain Analysis

The frequency response of an analog filter is deduced by finding its steady-state sinusoidal response, as we shall now demonstrate.

1.4.1 Sinusoidal Response

Consider an *N*th-order analog filter characterized by a transfer function $H(s)$. The sinusoidal response of such a filter is

$$y(t) = \mathcal{L}^{-1}[H(s)X(s)]$$

where

$$X(s) = \mathcal{L}[u(t)\sin\omega t] = \frac{\omega}{(s+j\omega)(s-j\omega)} \tag{1.6}$$

The product $H(s)X(s)$ satisfies conditions 1 and 2 imposed on the general inversion formula of Equation 1.2. Hence, for $t \geq 0$, we have

$$y(t) = \frac{1}{2\pi j}\int_{\Gamma} Y(s)e^{st}\,ds = \sum \text{res}[H(s)X(s)e^{st}] \tag{1.7}$$

where Γ is a contour enclosing the poles of $H(s)$ and $X(s)$ as in Figure 1.1.

Assuming simple poles for the transfer function, Equations 1.6 and 1.7 give

$$y(t) = \sum_{i=1}^{N} X(p_i)e^{p_i t}\operatorname*{res}_{s=p_i} H(s) + \frac{1}{2j}\left[H(j\omega)e^{j\omega t} - H(-j\omega)e^{-j\omega t}\right] \tag{1.8}$$

If the filter is assumed to be stable, then the poles are in the left-half s plane, i.e., $p_i = \sigma_i + j\omega_i$ with $\sigma_i < 0$.* As a consequence

$$\lim_{t\to\infty} e^{p_i t} = \lim_{t\to\infty}\left(e^{\sigma_i t}\cdot e^{j\omega_i t}\right) = 0$$

and since the residues of $H(s)$ are finite, the steady-state sinusoidal response is obtained from Equation 1.8 as

$$\tilde{y}(t) = \lim_{t\to\infty} y(t) = \frac{1}{2j}\left[H(j\omega)e^{j\omega t} - H(-j\omega)e^{-j\omega t}\right] \tag{1.9}$$

Equation 1.9 was deduced on the assumption that the poles of the transfer function are simple. However, it also applies for transfer functions with higher-order poles.

Now from the definition of the Laplace transform

$$H(s) = \int_{-\infty}^{\infty} h(t)e^{-st}\,dt$$

and hence

$$H(-j\omega) = \int_{-\infty}^{\infty} h(t)e^{j\omega t}\,dt = \left[\int_{-\infty}^{\infty} h(t)e^{-j\omega t}\,dt\right]^{*} = H^{*}(j\omega) \tag{1.10}$$

* See Chapter 22.1 of *Fundamentals of Circuits and Filters*.

If we write

$$H(j\omega) = M(\omega)e^{j\theta(\omega)} \tag{1.11}$$

where

$$M(\omega) = |H(j\omega)| \quad \text{and} \quad \theta(\omega) = \arg H(j\omega) \tag{1.12}$$

the steady-state sinusoidal response of the filter is obtained from Equations 1.9 through 1.12 as

$$\begin{aligned}\tilde{y}(t) &= \frac{1}{2j}\left[M(\omega)e^{j\theta(\omega)}e^{j\omega t} - M(\omega)e^{-j\theta(\omega)}e^{-j\omega t}\right] \\ &= M(\omega)\frac{1}{2j}\left[e^{j[\omega t+\theta(\omega)]} - e^{-j[\omega t+\theta(\omega)]}\right] \\ &= M(\omega)\sin[\omega t + \theta(\omega)]\end{aligned}$$

The preceding analysis has shown that the steady-state response of an analog filter to a sinusoid of unit amplitude is a sinusoid of amplitude $M(\omega)$, shifted by an angle $\theta(\omega)$. In effect, for a given frequency ω, the filter introduces a *gain* $M(\omega)$ and a *phase shift* $\theta(\omega)$.

As functions of frequency, $M(\omega)$ and $\theta(\omega)$ are known as the amplitude (or magnitude) response and phase response of the filter, respectively. The transfer function evaluated on the imaginary axis, namely, $H(j\omega)$ is the frequency response and, as was shown, its magnitude and angle are the amplitude response and phase response, respectively.

Two other quantities of a filter, which are of significant interest, are its *phase* and *group delays* These are defined as

$$\tau_p(\omega) = -\frac{\theta(\omega)}{\omega} \quad \text{and} \quad \tau_g(\omega) = -\frac{d\theta(\omega)}{d\omega}$$

respectively. For filters, the group delay is the more important of the two. As a function of frequency, $\tau_g(\omega)$ is usually referred to as the delay characteristic.

1.4.2 Graphical Construction

Consider a filter characterized by a transfer function of the form

$$H(s) = H_0\frac{N(s)}{D(s)} = H_0\frac{\prod_{i=1}^{M}(s - z_i)}{\prod_{i=1}^{N}(s - p_i)^{m_i}} \tag{1.13}$$

where H_0 is a constant. The frequency response of the filter is obtained as

$$H(j\omega) = M(\omega)e^{j\theta(\omega)} = \frac{H_0\prod_{i=1}^{M}(j\omega - z_i)}{\prod_{i=1}^{N}(j\omega - p_i)^{m_i}}$$

By letting

$$j\omega - z_i = M_{z_i} e^{j\psi_{z_i}} \tag{1.14}$$

$$j\omega - p_i = M_{p_i} e^{j\psi_{p_i}} \tag{1.15}$$

we obtain

$$M(\omega) = \frac{|H_0| \prod_{i=1}^{M} M_{z_i}}{\prod_{i=1}^{N} M_{p_i}^{m_i}} \tag{1.16}$$

and

$$\theta(\omega) = \arg H_0 + \sum_{i=1}^{M} \psi_{z_i} - \sum_{i=1}^{N} m_i \psi_{p_i} \tag{1.17}$$

where arg $H_0 = \pi$ if H_0 is negative.

The gain and phase shift $M(\omega)$ and $\theta(\omega)$ for some frequency $\omega = \omega_i$ can be determined graphically by using the following procedure:

1. Mark the zeros and poles of the filter in the s plane.
2. Draw the phasor $s = j\omega_i$, where ω_i is the frequency of interest.
3. Draw a phasor of the type in Equation 1.14 for each simple zero of $H(s)$.
4. Draw m_i phasor of the type in Equation 1.15 for each pole of order m_i.
5. Measure the magnitudes and angles of the phasors in steps 3 and 4 and use them in Equations 1.16 and 1.17 to calculate the gain $M(\omega_i)$ and phase shift $\theta(\omega_i)$, respectively.

The amplitude and phase responses of a filter can be determined by repeating the preceding procedure for frequencies $\omega = \omega_1, \omega_2, \ldots$, in the range 0 to ∞. The procedure is illustrated in Figure 1.4.

It should be mentioned that the modern approach for the analysis of filters is through the use of the many circuit analysis programs such as SPICE.* Nevertheless, the above graphical method is of interest and merits consideration for two reasons. First, it illustrates some of the fundamental properties of filters. Second, it provides a certain degree of intuition about the expected amplitude or phase response of a filter. For example, if a filter has pole close to the $j\omega$ axis, then as ω approaches the neighborhood of the pole, the magnitude of the phasor from the pole to the $j\omega$ axis decreases rapidly to a very small value and then increases as ω increases above this value. As a result, the amplitude response will exhibit a large peak in the frequency range close to the pole. On the other hand, a zero close to or on the $j\omega$ axis will lead to a notch in the amplitude response when ω is in the neighborhood of the zero.

Other situations are of interest, for example, if the poles of a filter are located in a band of the s plane below the horizontal line $s = \omega_c$ and its zeros are located above this line, then the filter will pass low-frequency and attenuate high-frequency components since $M_{zi} < M_{pi}$ if $\omega > \omega_c$ for all i. Such a filter is said to be a *low-pass* filter. If the zeros are located below the line $s = \omega_c$ and the poles above it, then the filter will pass high-frequency and attenuate low-frequency components, i.e., the filter will be a *high-pass* one.

* See Chapter 8 of *Computer Aided Design and Design Automation*, contribution of J.G. Rollins.

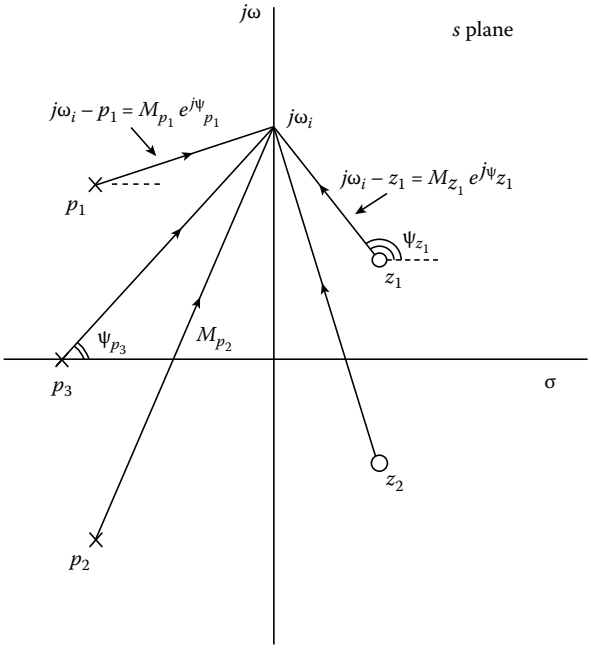


FIGURE 1.4 Graphical method for the evaluation of the frequency response.

1.4.3 Loss Function

Quite often, it is desirable to represent a filter in terms of its loss function. Consider a filter represented by the voltage transfer function

$$\frac{V_o(s)}{V_i(s)} = H(s) = \frac{N(s)}{D(s)}$$

where

$V_i(s)$ and $V_o(s)$ are the Laplace transforms of the input and output voltages, respectively
$N(s)$ and $D(s)$ are polynomials in s

The loss (or attenuation) of the filter in decibels is defined as

$$A(\omega) = 20\log\left|\frac{V_i(j\omega)}{V_o(j\omega)}\right| = 20\log\frac{1}{|H(j\omega)|} = 10\log L(\omega^2) \tag{1.18}$$

where

$$L(\omega^2) = \frac{1}{H(j\omega)H(-j\omega)}$$

$A(\omega)$ as a function of ω is the *loss characteristic*.

With $\omega = s/j$ in Equation 1.18, the function

$$L(-s^2) = \frac{D(s)D(-s)}{N(s)N(-s)}$$

can be formed. This is called the *loss function* of the filter and, as is evident, its zeros are the poles of $H(s)$ and their negatives, whereas its poles are the zeros of $H(s)$ and their negatives.

1.5 Ideal and Practical Filters

An ideal low-pass filter is one that will pass only low-frequency components. Its loss characteristic is given by

$$A(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega < \omega_c \\ \infty & \text{for } \omega_c < \omega < \infty \end{cases}$$

The frequency ranges 0 to ω_c and ω_c to ∞ are the *passband* and *stopband*, respectively. The boundary between the passband and stopband, namely, ω_c, is the *cutoff frequency*. An ideal high-pass filter will pass all components with frequencies above the cutoff frequency and reject all components with frequencies below the cutoff frequency, i.e.,

$$A(\omega) = \begin{cases} \infty & \text{for } 0 \le \omega < \omega_c \\ 0 & \text{for } \omega_c < \omega < \infty \end{cases}$$

Idealized loss characteristics can similarly be identified for bandpass and bandstop filters as

$$A(\omega) = \begin{cases} \infty & \text{for } 0 \le \omega < \omega_{c1} \\ 0 & \text{for } \omega_{c1} < \omega < \omega_{c2} \\ \infty & \text{for } \omega_{c2} \le \omega < \infty \end{cases}$$

and

$$A(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega < \omega_{c1} \\ \infty & \text{for } \omega_{c1} < \omega < \omega_{c2} \\ 0 & \text{for } \omega_{c2} \le \omega < \infty \end{cases}$$

respectively.

Practical filters differ from ideal ones in that the passband loss is not zero, the stopband loss is not infinite, and the transition between passband and stopband is gradual. Practical loss characteristics for low-pass, high-pass, bandpass, and bandstop filters assume the forms

$$A_{LP}(\omega)\begin{cases} \le A_p & \text{for } 0 \le \omega < \omega_p \\ \ge A_a & \text{for } \omega_a \le \omega \le \infty \end{cases}$$

$$A_{HP}(\omega)\begin{cases} \ge A_a & \text{for } 0 \le \omega \le \omega_a \\ \le A_p & \text{for } \omega_p \le \omega < \infty \end{cases}$$

$$A_{BP}(\omega)\begin{cases} \ge A_a & \text{for } 0 \le \omega \le \omega_{a1} \\ \le A_p & \text{for } \omega_{p1} \le \omega \le \omega_{p2} \\ \ge A_a & \text{for } \omega_{a2} \le \omega \le \infty \end{cases}$$

and

$$A_{BS}(\omega) = \begin{cases} \leq A_p & \text{for } 0 \leq \omega \leq \omega_{p1} \\ \geq A_a & \text{for } \omega_{a1} \leq \omega \leq \omega_{a2} \\ \leq A_p & \text{for } \omega_{p2} \leq \omega \leq \infty \end{cases}$$

respectively, where ω_p, ω_{p1}, and ω_{p2} are passband edges, ω_a, ω_{a1}, and ω_{a2} are stopband edges, A_p is the maximum passband loss, and A_a is the minimum stopband loss. In practice, A_p is determined from the allowable amplitude distortion (see Section 1.6) and A_a is dictated by the allowable adjacent channel interference and the desirable signal-to-noise ratio.

It should be mentioned that in practical filters the cutoff frequency ω_c is not a very precise term. It is often used to identify some hypothetical boundary between passband and stopband such as the 3 dB frequency in Butterworth filters, the passband edge in Chebyshev filters, the stopband edge in inverse-Chebyshev filters, or the geometric mean of the passband and stopband edges in elliptic filters.

If a filter is required to have a piecewise constant loss characteristic (or amplitude response) and the shape of the phase response is not critical, the filter can be fully specified by its band edges, the minimum passband and maximum stopband losses A_p and A_a, respectively.

1.6 Amplitude and Delay Distortion

In practice, a filter can distort the information content of the signal. Consider a filter characterized by a transfer function $H(s)$ and assume that its input and output signal are $v_i(t)$ and $v_o(t)$. The frequency response of the filter is given by

$$H(j\omega) = M(\omega)e^{j\theta(\omega)}$$

where $M(\omega)$ and $\theta(\omega)$ are the amplitude and phase responses, respectively.

The frequency spectrum of $v_i(t)$ is its Fourier transform, namely, $V_i(j\omega)$. Assume that the information content of $v_i(t)$ is concentrated in frequency band B given by

$$B = \{\omega: \omega_L \leq \omega \leq \omega_H\}$$

and that its frequency spectrum is zero elsewhere.

Let us assume that the amplitude response is constant with respect to band B, i.e.,

$$M(\omega) = G_0 \quad \text{for } \omega \in B \tag{1.19}$$

and that the phase response is linear, i.e.,

$$\theta(\omega) = -\tau_g\omega + \theta_o \quad \text{for } \omega \in B \tag{1.20}$$

where τ_g is a constant. This implies that the group delay is constant with respect to band B, i.e.,

$$\tau(\omega) = -\frac{d\theta(\omega)}{d\omega} = \tau_g \quad \text{for } \omega \in B$$

The frequency spectrum of the output signal $v_o(t)$ can be obtained from Equations 1.19 and 1.20 as

$$\begin{aligned} V_o(j\omega) &= H(j\omega)V_i(j\omega) = M(\omega)e^{j\theta(\omega)}V_i(j\omega) \\ &= \left[G_0 e^{-j\omega\tau_g + j\theta_0}\right]V_i(j\omega) = G_0 e^{j\theta_0}\left[e^{-j\omega\tau_g}V_i(j\omega)\right] \end{aligned}$$

and from the time-shifting theorem of the Fourier transform

$$v_o(t) = G_0 e^{j\theta_0} v_i\left(t - \tau_g\right)$$

We conclude that the amplitude response of the filter is flat and its phase response is a linear function of ω (i.e., the delay characteristic is flat) in band B, then the output signal is a delayed replica of the input signal except that a gain G_o and a constant phase shift θ_0 are introduced.

If the amplitude response of the filter is not flat in band B, then *amplitude distortion* will be introduced since different frequency components of the signal will be amplified by different amounts.

If the delay characteristic is not flat in band B, then *delay* (or *phase*) *distortion* will be introduced since different frequency components will be delayed by different amounts.

Amplitude distortion can be quite objectionable in practice and, consequently, in each frequency band that carries information, the amplitude response is required to be constant to within a prescribed tolerance. The amount of amplitude distortion allowed determines the maximum passband loss A_p.

If the ultimate receiver of the signal is the human ear, e.g., when a speech or music signal is to be processed, delay distortion is quite tolerable. However, in other applications it can be as objectionable as amplitude distortion and the delay characteristic is required to be fairly flat. Applications of this type include data transmission, where the signal is to be interpreted by digital hardware, and image processing, where the signal is used to reconstruct an image that is to be interpreted by the human eye. The allowable delay distortion dictates the degree of flatness in the delay characteristic.

1.7 Minimum-Phase, Nonminimum-Phase, and Allpass Filters

Filters satisfying prescribed loss specifications for applications where delay distortion is unimportant can be readily designed with transfer functions whose zeros are on the $j\omega$ axis or in the left-half s plane. Such transfer functions are said to be minimum-phase since the phase response at a given frequency ω is increased if any one of the zeros is moved into the right-half s plane, as will now be demonstrated.

1.7.1 Minimum-Phase Filters

Consider a filter where the zeros z_i for $i = 1, 2, \ldots, M$ are replaced by their mirror images and let the new zeros be located at $z = \bar{z}_i$, where

$$\text{Re}\ \bar{z}_i = -\text{Re}\ z_i \quad \text{and} \quad \text{Im}\ \bar{z}_i = \text{Im}\ z_i$$

as depicted in Figure 1.5. From the geometry of the new zero-pole plot, the magnitude and angle of each phasor $j\omega - \bar{z}_i$ are given by

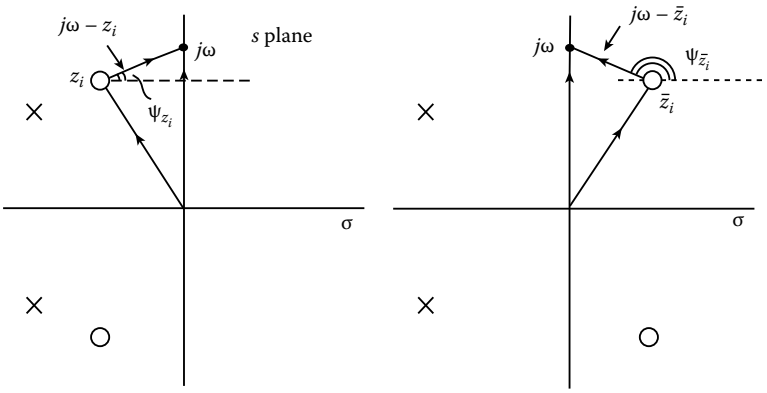


FIGURE 1.5 Zero-pole plots of minimum-phase and corresponding nonminimum-phase filter.

$$M_{\bar{z}_i} = M_{z_i} \quad \text{and} \quad \psi_{\bar{z}_i} = \pi - \psi_{z_i}$$

respectively. The amplitude response of the modified filter is obtained from Equation 1.16 as

$$\overline{M}(\omega) = \frac{|H_0| \prod_{i=1}^{M} M_{\bar{z}_i}}{\prod_{i=1}^{N} M_{p_i}^{m_i}} = \frac{|H_0| \prod_{i=1}^{M} M_{z_i}}{\prod_{i=1}^{N} M_{p_i}^{m_i}} = M(\omega)$$

Therefore, replacing the zeros of the transfer function by their mirror images leaves the amplitude response unchanged.

The phase response of the original filter is given by Equation 1.17 as

$$\theta(\omega) = \arg H_0 + \sum_{i=1}^{M} \psi_{z_i} - \sum_{i=1}^{N} m_i \psi_{p_i} \tag{1.21}$$

and since $\psi_{\bar{z}_i} = \pi - \psi_{z_i}$, the phase response of the modifier filter is given by

$$\begin{aligned} \bar{\theta}(\omega) &= \arg H_0 + \sum_{i=1}^{M} \psi_{\bar{z}_i} - \sum_{i=1}^{N} m_i \psi_{p_i} \\ &= \arg H_0 + \sum_{i=1}^{M} (\pi - \psi_{z_i}) - \sum_{i=1}^{N} m_i \psi_{p_i} \end{aligned} \tag{1.22}$$

that is, the phase response of the modified filter is different from that of the original filter. Furthermore, from Equations 1.21 and 1.22

$$\bar{\theta}(\omega) - \theta(\omega) = \sum_{i=1}^{M} (\pi - 2\psi_{z_i})$$

and since $-\pi/2 \leq -\psi_{z_i} \leq \pi/2$, we have

$$\bar{\theta}(\omega) - \theta(\omega) \geq 0$$

or

$$\bar{\theta}(\omega) \geq \theta(\omega)$$

As a consequence, the phase response of the modified filter is equal to or greater than that of the original filter for all ω.

A frequently encountered requirement in the design of filters is that the delay characteristic be flat to within a certain tolerance within the passband(s) in order to achieve tolerable delay distortion, as was demonstrated in Section 1.6. In these and other filters in which the specifications include constraints on the phase response or delay characteristic, a nonminimum-phase transfer function is almost always required.

1.7.2 Allpass Filters

An allpass filter is one that has a constant amplitude response. Consider a transfer function of the type given by Equation 1.13. From Equation 1.10, $H(-j\omega)$ is the complex conjugate of $H(j\omega)$, and hence a constant amplitude response can be achieved if

$$M^2(\omega) = H(s)H(-s)|_{s=j\omega} = H_0^2 \frac{N(s)}{D(s)} \times \left.\frac{N(-s)}{D(-s)}\right|_{s=j\omega} = H_0^2$$

Hence, an allpass filter can be obtained if

$$N(-s) = D(s)$$

that is, the zeros of such a filter must be the mirror images of the poles and vice versa. A typical zero-pole plot for an allpass filter is illustrated in Figure 1.6. A second-order allpass transfer function is given by

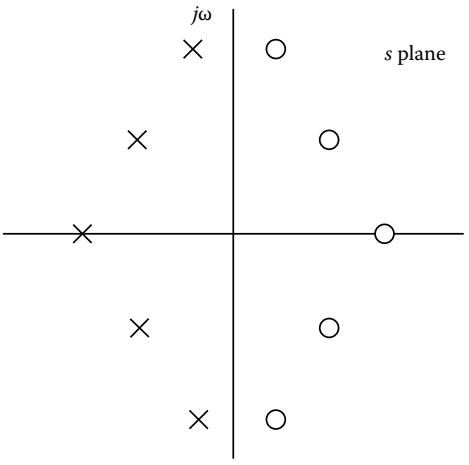


FIGURE 1.6 Typical zero-pole plot of an allpass filter.

$$H_{AP}(s) = \frac{s^2 - bs + c}{s^2 + bs + c}$$

where $b > 0$ for stability. As described previously, we can write

$$M^2(\omega) = H_{AP}(s)H_{AP}(-s)|_{s=j\omega} = \frac{s^2 - bs + c}{s^2 + bs + c} \times \left.\frac{s^2 + bs + c}{s^2 - bs + c}\right|_{s=j\omega} = 1$$

Allpass filters can be used to modify the phase responses of filters without changing their amplitude responses. Hence, they are used along with minimum-phase filters to obtain nonminimum-phase filters that satisfy amplitude and phase response specifications simultaneously.

1.7.3 Decomposition of Nonminimum-Phase Transfer Functions

Some methods for the design of filters satisfying amplitude and phase response specifications, usually methods based on optimization, yield a nonminimum-phase transfer function. Such a transfer function can be easily decomposed into a product of a minimum-phase and an allpass transfer function, i.e.,

$$H_N(s) = H_M(s)H_{AP}(s)$$

Consequently, a nonminimum-phase filter can be implemented as a cascade arrangement of a minimum-phase and an allpass filter.

The preceding decomposition can be obtained by using the following procedure:

1. For each zero in the right-half s plane, augment the transfer function by a zero and a pole at the mirror image position of the zero.
2. Assign the left-half s-plane zeros and the original poles to the minimum-phase transfer function $H_M(s)$.
3. Assign the right-half s-plane zeros and the left-hand s-plane poles generated in step 1 to the allpass transfer function $H_{AP}(s)$.

This procedure is illustrated in Figure 1.7. For example, if

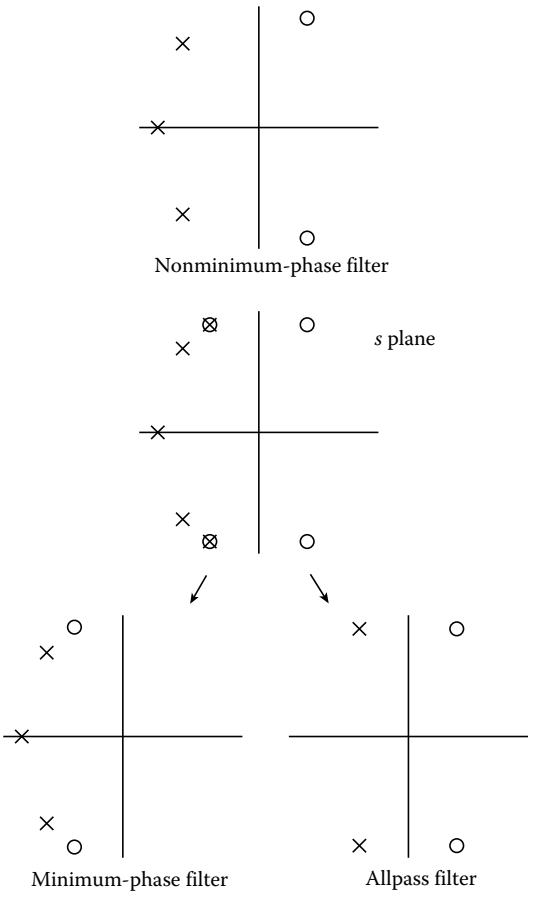


FIGURE 1.7 Decomposition of nonminimum-phase transfer function.

$$H_N(s) = \frac{(s^2+4s+5)(s^2-3s+7)(s-5)}{(s^2+2s+6)(s^2+4s+9)(s+2)}$$

then, we can write

$$H_N(s) = \frac{(s^2+4s+5)(s^2-3s+7)(s-5)}{(s^2+2s+6)(s^2+4s+9)(s+2)} \times \frac{(s^2+3s+7)(s+5)}{(s^2+3s+7)(s+5)}$$

Hence,

$$H_N(s) = \frac{(s^2+4s+5)(s^2+3s+7)(s+5)}{(s^2+2s+6)(s^2+4s+9)(s+2)} \times \frac{(s^2-3s+7)(s-5)}{(s^2+3s+7)(s+5)}$$

or

$$H_N(s) = H_M(s)H_{AP}(s)$$

where

$$H_M(s) = \frac{(s^2+4s+5)(s^2+3s+7)(s+5)}{(s^2+2s+6)(s^2+4s+9)(s+2)}$$
$$H_{AP}(s) = \frac{(s^2-3s+7)(s-5)}{(s^2+3s+7)(s+5)}$$

1.8 Introduction to the Design Process

The design of filters starts with a set of specifications and ends with the implementation of a prototype. It comprises four general steps, as follows:

1. Approximation
2. Realization
3. Study of imperfections
4. Implementation

1.8.1 The Approximation Step

The approximation step is the process of generating a transfer function that satisfies the desired specifications, which may concern the amplitude, phase, and possibly the time-domain response of the filter.

The available methods for the solution of the approximation problem can be classified as closed-form or iterative. In closed-form methods, the problem is solved through a small number of design steps using a set of closed-form formulas or transformations. In iterative methods, an initial solution is assumed and, through the application of optimization methods, a series of progressively improved solutions are obtained until some design criterion is satisfied. Closed-form solutions are very precise and entail a minimal amount of computation. However, the available solutions are useful in applications where the loss characteristic is required to be piecewise constant to within some prescribed tolerances. Iterative methods, on the other hand, entail a considerable amount of computation but can be used to design filters

with arbitrary amplitude and phase response characteristics (see Ref. [1, Chapter 14]) for the application of these methods for the design of digital filters). Some classical closed-form solutions are the so-called Butterworth, Chebyshev, and elliptic* approximations to be described in Chapter 2 by A.M. Davis.

In general, the designer is interested in simple and reliable approximation methods that yield precise designs with the minimum amount of computation.

1.8.2 The Realization Step

The *synthesis* of a filter is the process of converting some characterization of the filter into a network. The process of converting the transfer function into a network is said to be the realization step and the network obtained is sometimes called the realization.

The realization of a transfer function can be accomplished by expressing it in some form that allows the identification of an interconnection of elemental filter subnetworks and/or elements. Many realization methods have been proposed in the past that lead to structures of varying complexity and properties. In general, the designer is interested in realizations that are economical in terms of the number of elements, do not require expensive components, and are not seriously affected by variations in the element values such as may be caused by variations in temperature and humidity, and drift due to element aging.

1.8.3 Study of Imperfections

During the approximation step, the coefficients of the transfer function are determined to a high degree of precision and the realization is obtained on the assumption that elements are ideal, i.e., capacitors are lossless, inductors are free of winding capacitances, amplifiers have infinite bandwidths, and so on. In practice, however, the filter is implemented with nonideal elements that have finite tolerances and are often nonlinear. Consequently, once a realization is obtained, sometimes referred to as a *paper design*, the designer must embark on the study of the effects of element imperfections. Several types of analysis are usually called for ranging from tolerance analysis, study of parasitics, time-domain analysis, sensitivity analysis, noise analysis, etc. Tight tolerances result in high-precision filters but the cost per unit would be high. Hence the designer is obliged to determine the highest tolerance that can be tolerated without violating the specifications of the filter throughout its working life. Sensitivity analysis is a related study that will ascertain the degree of dependence of a filter parameter, e.g., the dependence of the amplitude response on a specific element. If the loss characteristic of a filter is not very sensitive to certain capacitance, then the designer would be able to use a less precise and cheaper capacitor, which would, of course, decrease the cost of the unit.

1.8.4 Implementation

Once the filter is thoroughly analyzed and found to meet the desired specifications under ideal conditions, a prototype is constructed and tested. Decisions to be made involve the type of components and packaging, and the methods are to be used for the manufacture, testing, and tuning of the filter. Problems may often surface at the implementation stage that may call for one or more modifications in the paper design. Then the realization and possibly the approximation may have to be redone.

---

* To be precise, the elliptic approximation is not a closed-form method, since the transfer function coefficients are given in terms of certain infinite series. However, these series converge very rapidly and can be treated as closed-form formulas for most practical purposes (see Ref. [1, Chapter 5]).

1.9 Introduction to Realization

Realization tends to depend heavily on the type of filter required. The realization of passive *RLC* filters differs quite significantly from that of active filters which, in turn, is entirely different from the realization of microwave filters.

1.9.1 Passive Filters

Passive *RLC* filters have been the mainstay of communications since the 1920s and, furthermore, they continue to be of considerable importance today for frequencies in the 100–500 kHz range.

The realization of passive *RLC* filters has received considerable attention through the years and it is, as a consequence, highly developed and sophisticated. It can be accomplished by using available filter-design packages such as FILSYN [5] and FILTOR [6]. In addition, several filter-design handbooks and published design tables are available [7–10].

The realization of passive *RLC* filters starts with a resistively terminated *LC* two-port network such as that in Figure 1.8. Then through one of several approaches, the transfer function is used to generate expressions for the *z* or *y* parameters of the *LC* two-port. The realization of the *LC* two-port is achieved by realizing the *z* or *y* parameters. The realization of passive filters is considered in Section I.

1.9.2 Active Filters

Since the reactance of an inductor is ωL, increased inductance values are required to achieve reasonable reactance values at low frequencies. For example, an inductance of 1 mH which will present a reactance of 6.28 kΩ at 1 MHz will present only 0.628 Ω at 100 Hz. Thus, as the frequency range of interest is reduced, the inductance values must be increased if a specified impedance level is to be maintained. This can be done by increasing the number of turns on the inductor coil and to some extent by using ferromagnetic cores of high permeability. Increasing the number of turns increases the resistance, the size, and the cost of the inductor. The resistance is increased because the length of the wire is increased [$R = (\rho \times length)/Area$], and hence the *Q* factor is reduced. The cost goes up because the cost of materials as well as the cost of labor go up, since an inductor must be individually wound. For these reasons, inductors are generally incompatible with miniaturization or microcircuit implementation.

The preceding physical problem has led to the invention and development of a class of inductorless filters known collectively as active filters. Sensitivity considerations, which will be examined in Chapter 4 by I. Filanovsky, have led to two basic approaches to the design of active filters. In one approach, the active filter is obtained by simulating the inductances in a passive *RLC* filter or by realizing a signal flow graph of the passive *RLC* filter. In another approach, the active filter is obtained by cascading a number of low-order filter sections of some type, as depicted in Figure 1.9a where Z_{o0} is the output impedance of the signal source.

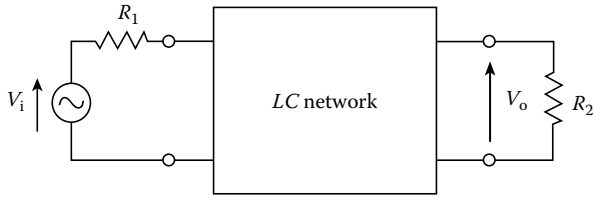


FIGURE 1.8 Passive *RLC* filter.

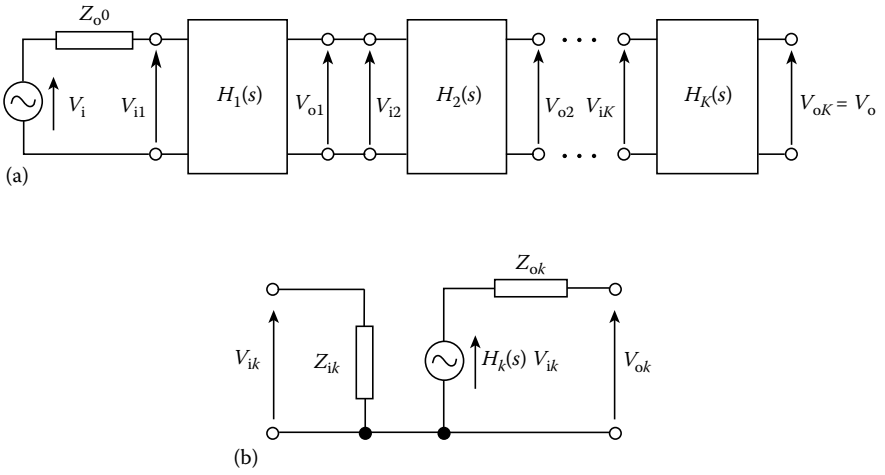


FIGURE 1.9 (a) Cascade realization and (b) Thévenin equivalent circuit of filter section.

Each filter section is made up of an interconnection of resistors, capacitors, and active elements, and by Thévenin's theorem, it can be represented by its input impedance, open-circuit voltage transfer function, and output impedance as shown in Figure 1.9b. The voltage transfer function of the configuration is given by

$$H(s) = \frac{V_o}{V_i}$$

and since the input voltage of section k is equal to the output voltage of section $k-1$, i.e., $V_{ik} = V_{o(k-1)}$ for $k = 2, 3, \ldots, K$, and $V_o = V_{oK}$ we can write

$$H(s) = \frac{V_o}{V_i} = \frac{V_{i1}}{V_i} \times \frac{V_{o1}}{V_{i1}} \times \frac{V_{o2}}{V_{i2}} \times \cdots \times \frac{V_{oK}}{V_{iK}} \tag{1.23}$$

where

$$\frac{V_{i1}}{V_i} = \frac{Z_{i1}}{Z_{o0} + Z_{i1}} \tag{1.24}$$

and

$$\frac{V_{ok}}{V_{ik}} = \frac{Z_{i(k+1)}}{Z_{ok} + Z_{i(k+1)}} H_k(s) \tag{1.25}$$

is the transfer function of the kth section. From Equations 1.23 through 1.25, we obtain

$$H(s) = \frac{V_o}{V_i} = \frac{Z_{i1}}{Z_{o0} + Z_{i1}} \prod_{k=1}^{K} \frac{Z_{i(k+1)}}{Z_{ok} + Z_{i(k+1)}} H_k(s)$$

Now if

$$|Z_{ik}| \gg |Z_{o(k-1)}|$$

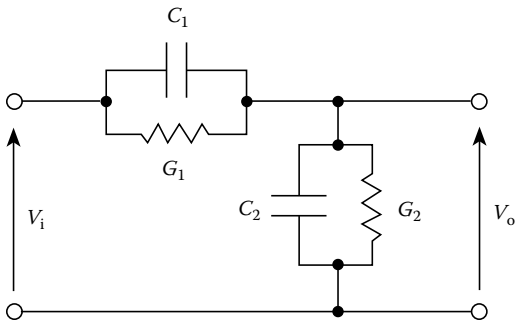


FIGURE 1.10 First-order *RC* network.

for $k = 1, 2, \ldots, K$, then the loading effect produced by section $k+1$ on section k can be neglected and hence

$$H(s) = \frac{V_o}{V_i} = \prod_{k-1}^{K} H_k(s)$$

Evidently, a highly desirable property in active filter sections is that the magnitude of the input impedance be large and/or that of the output impedance be small since in such a case the transfer function of the cascade structure is equal to the product of the transfer functions of the individual sections.

An arbitrary *N*th-order transfer function obtained by using the Butterworth, Bessel, Chebyshev, inverse-Chebyshev, or elliptic approximation can be expressed as

$$H(s) = H_0(s) \prod_{k=1}^{K} \frac{a_{2k}s^2 + a_{1k}s + a_{0k}}{s^2 + b_{1k}s + b_{0k}}$$

where

$$H_0(s) = \begin{cases} \frac{a_{10}s + a_{00}}{b_{10}s + b_{00}} & \text{for odd } N \\ 1 & \text{for even } N \end{cases}$$

The first-order transfer function $H_0(s)$ for the case of an odd-order can be readily realized using the *RC* network of Figure 1.10.

1.9.3 Biquads

From the above analysis, we note that all we need to be able to realize an arbitrary transfer function is a circuit that realizes the biquadratic transfer function

$$H_{BQ}(s) = \frac{a_2s^2 + a_1s + a_0}{s^2 + b_1s + b_0} = \frac{a_2(s + z_1)(s + z_2)}{(s + p_1)(s + p_2)} \tag{1.26}$$

where zeros and poles occur in complex conjugate pairs, i.e., $z_2 = z_1^*$ and $p_2 = p_1^*$. Such a circuit is commonly referred to as a *biquad*.

After some manipulation, the transfer function in Equation 1.26 can be expressed as

$$H_{BQ}(s) = K\frac{s^2 + (2\text{Re } z_1)s + (\text{Re } z_1)^2 + (\text{Im } z_1)^2}{s^2 + (2\text{Re } p_1)s + (\text{Re } p_1)^2 + (\text{Im } p_1)^2}$$
$$= K\frac{s^2 + (\omega_z/Q_z)s + \omega_z^2}{s^2 + (\omega_p/Q_p)s + \omega_p^2}$$

where

$K = a_2$

ω_z and ω_p are the zero and pole frequencies, respectively

Q_z and Q_p are the zero and pole *quality factors* (or Q factors for short), respectively

The formulas for the various parameters are as follows:

$$\omega_z = \sqrt{(\text{Re } z_1)^2 + (\text{Im } z_1)^2}$$
$$\omega_p = \sqrt{(\text{Re } p_1)^2 + (\text{Im } p_1)^2}$$
$$Q_z = \frac{\omega_z}{2\text{ Re } z_1}$$
$$Q_p = \frac{\omega_p}{2\text{ Re } p_1}$$

The zero and pole frequencies are approximately equal to the frequencies of minimum gain and maximum gain, respectively. The zero and pole Q factors have to do with the selectivity of the filter. A high zero Q factor results in a deep notch in the amplitude response, whereas a high pole Q factor results in a very peaky amplitude response.

The dc gain and the gain as $\omega \to \infty$ in decibels are given by

$$M_0 = 20\log|H_{BQ}(0)| = 20\log\left(K\frac{\omega_z^2}{\omega_p^2}\right)$$

and

$$M_\infty = 20\log|H_{BQ}(j\infty)| = 20\log K$$

respectively.

1.9.4 Types of Basic Filter Sections

Depending on the values of the transfer function coefficients, five basic types of filter sections can be identified, namely, low-pass, high-pass, bandpass, notch (sometimes referred to as bandreject), and allpass. These sections can serve as building blocks for the design of filters that can satisfy arbitrary specifications. They are actually sufficient for the design of all the standard types of filters, namely, Butterworth, Chebyshev, inverse-Chebyshev, and elliptic filters.

1.9.4.1 Low-Pass Section

In a *low-pass* section, we have $a_2 = a_1 = 0$ and $a_0 = K\omega_p^2$. Hence, the transfer function assumes the form

$$H_{LP}(s) = \frac{a_0}{s^2 + b_1 s + b_0} = \frac{K\omega_p^2}{s^2 + (\omega_p/Q_p)s + \omega_p^2}$$

(see Figure 1.11a)

1.9.4.2 High-Pass Section

In a *high-pass* section, we have $a_2 = K$ and $a_1 = a_0 = 0$. Hence, the transfer function assumes the form

$$H_{\mathrm{HP}}(s) = \frac{a_2 s^2}{s^2 + b_1 s + b_0} = \frac{Ks^2}{s^2 + \left(\omega_{\mathrm{p}}/Q_{\mathrm{p}}\right)s + \omega_{\mathrm{p}}^2}$$

(see Figure 1.11b)

1.9.4.3 Bandpass Section

In a *bandpass* section, we have $a_1 = K\omega_{\mathrm{p}}/Q_{\mathrm{p}}$ and $a_2 = a_0 = 0$. Hence the transfer function assumes the form

$$H_{\mathrm{BP}}(s) = \frac{a_1 s}{s^2 + b_1 s + b_0} = \frac{K\left(\omega_p/Q_{\mathrm{p}}\right)s}{s^2 + \left(\omega_{\mathrm{p}}/Q_{\mathrm{p}}\right) + \omega_{\mathrm{p}}^2}$$

(see Figure 1.11c)

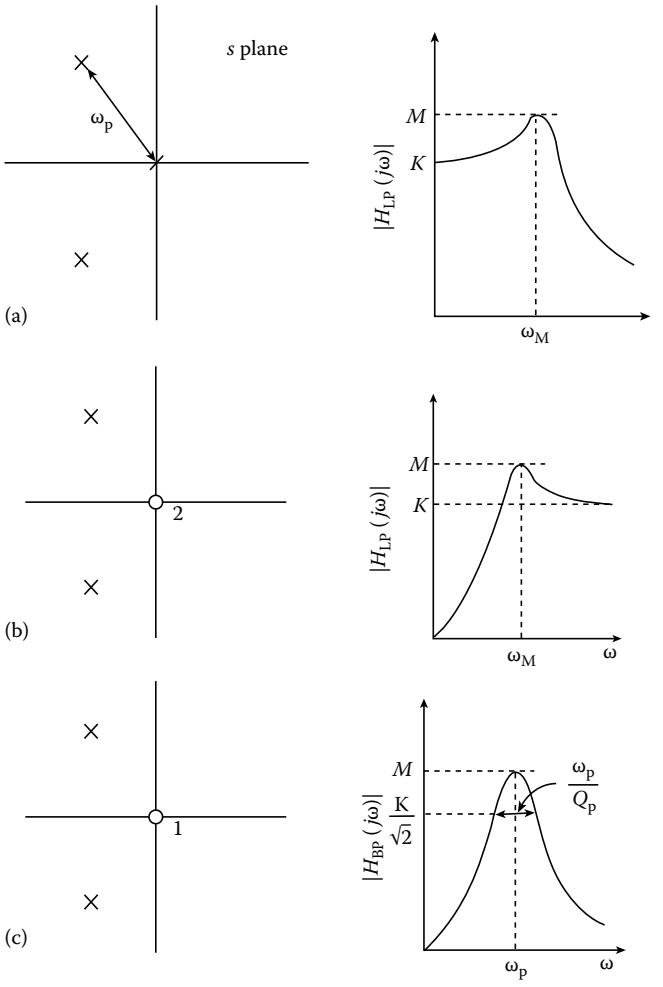


FIGURE 1.11 Basic second-order filter sections: (a) low-pass, (b) high-pass, (c) bandpass,

(*continued*)

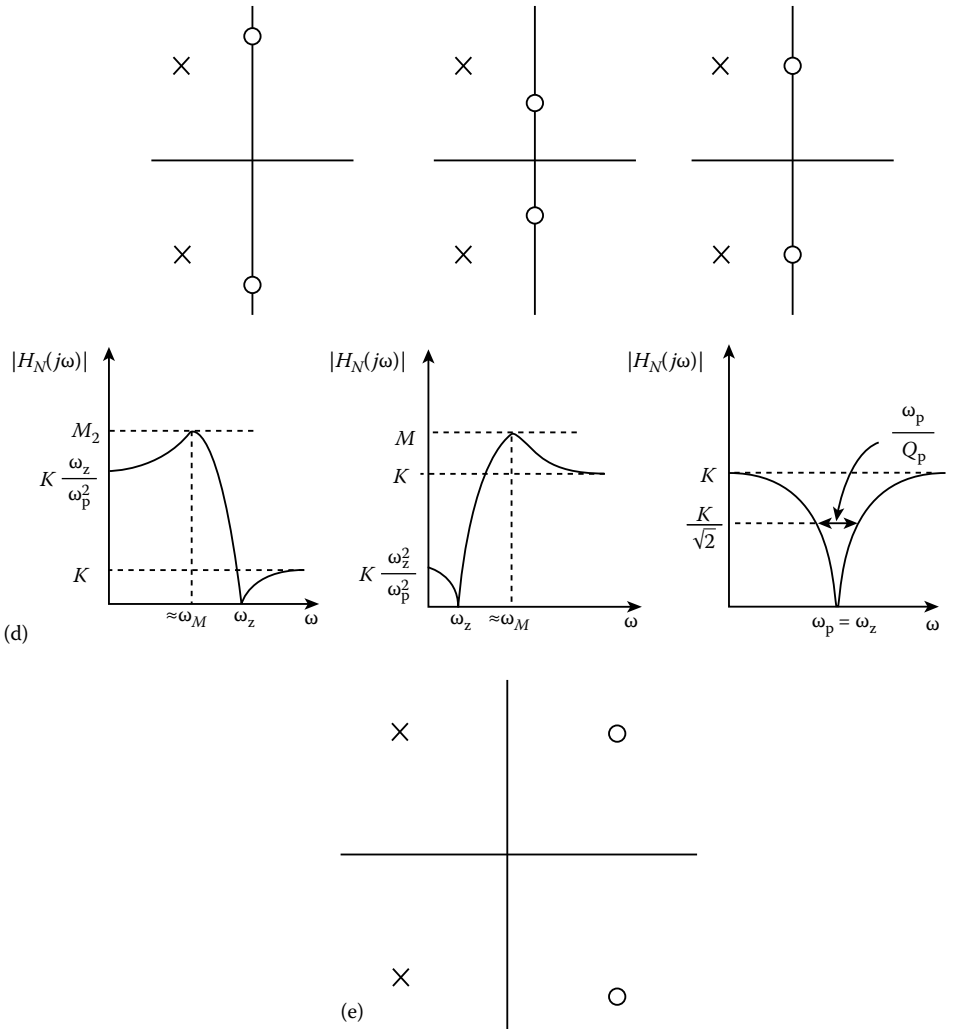


FIGURE 1.11 (continued) (d) notch, and (e) allpass.

1.9.4.4 Notch Section

In a *notch* section, we have $a_2 = K$, $a_1 = 0$, and $a_0 = K\omega_p^2$. Hence, the transfer function assumes the form

$$H_N(s) = \frac{a_2 s^2 + a_0}{s^2 + b_1 s + b_0} = \frac{K\left(s^2 + \omega_z^2\right)}{s^2 + \left(\omega_p/Q_p\right)s + \omega_p^2}$$

(see Figure 1.11d)

1.9.4.5 Allpass Section

In an *allpass* section, we have $a_2 = K$, $a_1 = -Kw_p/Q_p$, and $a_0 = K\omega_p^2$. Hence the transfer function assumes the form

$$H_{AP}(s) = \frac{a_2 s^2 + a_1 s + a_0}{s^2 + b_1 s + b_0} = \frac{K\left[s^2 - \left(\omega_p/Q_p\right)s + \omega_p^2\right]}{s^2 + \left(\omega_p/Q_p\right)s + \omega_p^2}$$

(see Figure 1.11e)

The design of active and switched-capacitor filters is treated in some detail in Section II.

References

1. A. Antoniou, *Digital Filters: Analysis, Design, and Applications*, 2nd ed. New York: McGraw-Hill, 1993.
2. R. J. Schwarz and B. Friedland, *Linear Systems*, New York: McGraw-Hill, 1965.
3. E. Kreyszig, *Advanced Engineering Mathematics*, 3rd ed. New York: Wiley, 1972.
4. R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters*, Englewood Cliffs, NJ: Prentice Hall, 1990.
5. G. Szentirmai, FILSYN—A general purpose filter synthesis program, *Proc. IEEE*, 65, 1443–1458, Oct. 1977.
6. A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive*, Portland, OR: Matrix, 1978.
7. J. K. Skwirzynski, *Design Theory and Data for Electrical Filters*, London: Van Nostrand, 1965.
8. R. Saal, *Handbook of Filter Design*, Backnang: AEG Telefunken, 1979.
9. A. I. Zverev, *Handbook of Filter Synthesis*, New York: Wiley, 1967.
10. E. Chirlian, *LC Filters: Design, Testing, and Manufacturing*, New York: Wiley, 1983.

2

Approximation

2.1 Introduction **2**-1
2.2 Butterworth LPP Approximation **2**-8
2.3 Chebyshev LPP Approximation **2**-12
2.4 Bessel–Thompson LPP Approximation **2**-18
2.5 Elliptic Approximation **2**-23
References **2**-33

Artice M. Davis
San Jose State University

2.1 Introduction

The approximation problem for filters is illustrated in Figure 2.1. A filter is often desired to produce a given slope of gain over one or more frequency intervals, to remain constant over other intervals, and to completely reject signals having frequencies contained in still other intervals. Thus, in the example shown in the figure, the desired gain is zero for very low and very high frequencies. The centerline, shown dashed, is the nominal behavior and the shaded band shows the permissible variation in the gain characteristic. Realizable circuits must always generate smooth curves and so cannot exactly meet the piecewise linear specification represented by the centerline. Thus, the realizable behavior is shown by the smooth, dark curve that lies entirely within the shaded tolerance band.

What type of frequency response function can be postulated that will meet the required specifications and, at the same time, be realizable: constructible with a specified catalog of elements? The answer depends upon the types of elements allowed. For instance, if one allows pure delays with a common delay time, summers, and scalar multipliers, a trigonometric polynomial will work; this, however, will cause the gain function to be repeated in a periodic manner. If this is permissible, one can then realize the filter in the form of an FIR digital filter or as a commensurate transmission line filter, and in fact, it can be realized in such a fashion that the resulting phase behavior is precisely linear. If one fits the required behavior with a rational trigonometric function, a function that is the ratio of two trigonometric polynomials, an economy of hardware will result. The phase, however, will unfortunately no longer be linear. These issues are discussed at greater length in Ref. [1].

Another option would be to select an ordinary polynomial in ω as the approximating function. Polynomials, however, behave badly at infinity. They approach infinity as $\omega \to \pm\infty$, a highly undesirable solution. For this reason, one must discard polynomials. A rational function of ω, however, will work nicely for the ratio of two polynomials will approach zero as $\omega \to \pm\infty$ if the degree of the numerator polynomial is selected to be of lower degree than that of the denominator. Furthermore, by the Weierstrass theorem, such a function can approximate any continuous function arbitrarily closely over any closed interval of finite length [2]. Thus, one sees that the rational functions in ω offer a suitable approximation for analog filter design and, in fact, do not have the repetitive nature of the trigonometric rational functions.

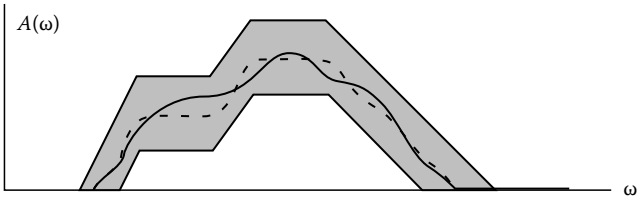


FIGURE 2.1 General approximation problem.

Suppose, therefore, that the gain function is of the form

$$A(\omega) = \frac{N(\omega)}{D(\omega)} = \frac{a_0 + a_1\omega + a_2\omega^2 + \cdots + a_r\omega^r}{b_0 + b_1\omega + b_2\omega^2 + \cdots + b_q\omega^q} \tag{2.1}$$

where $r \leq q$ for reasons mentioned above. Assuming that the filter to be realized is constrained to be constructable with real* elements, one must require that $A(-\omega) = A(\omega)$, that is, that the gain be an event function of frequency. But then, as it is straightforward to show, one must require that all the odd coefficients of both numerator and denominator be zero. This means that the gain is a function of ω^2:

$$A(\omega) = \frac{N(\omega^2)}{D(\omega^2)} = \frac{a_0 + a_1\omega^2 + \cdots + a_m\omega^{2m}}{b_0 + b_1\omega^2 + \cdots + b_n\omega^{2n}} = A(\omega^{-2}) \tag{2.2}$$

The expression has been reindexed and the constants redefined in an obvious manner. The net result is that one must approximate the desired characteristic by the ratio of two polynomials in ω^2; the objective is to determine the numerator and denominator coefficients to meet the stated specifications. Once this is accomplished one must compute the filter transfer function $G(s)$ in order to synthesize the filter [4,5]. Assuming that $G(s)$ is real (has real coefficients), then its complex conjugate satisfies $G^*(s) = G(s^*)$, from which it follows that $G(s)$ is related to $A(\omega^2)$ by the relationship

$$[G(s)G(-s)]_{s=j\omega} = G(j\omega)G^*(j\omega) = |G(j\omega)|^2 = A^2(\omega^2) \tag{2.3}$$

In fact, it is more straightforward to simply cast the original approximation problem in terms of $A^2(\omega^2)$, rather than in terms of $A(\omega)$. In this case, Equation 2.2 becomes

$$A^2(\omega^2) = \frac{N(\omega^2)}{D(\omega^2)} = \frac{a_0 + a_1\omega^2 + \cdots + a_m\omega^{2m}}{b_0 + b_1\omega^2 + \cdots + b_n\omega^{2n}} \tag{2.4}$$

Thus, one can assume that the approximation process produces $A^2(\omega^2)$ as the ratio of two real polynomials in ω^2. Since Equation 2.3 requires the substitutions $s \rightarrow j\omega$, one also has $s^2 \rightarrow -\omega^2$, and conversely. Thus, Equation 2.3 becomes

$$G(s)G(-s) = A^2(-s^2) \tag{2.5}$$

Though this has been shown to hold only on the imaginary axis, it continues to hold for other complex values of s as well by analytic continuation.†

* Complex filters are quite possible to construct, as recent work [3] shows.

† A function analytic in a region is completely determined by its values along any line segment in that region—in this case, by its value along the $j\omega$ axis.

The problem now is to compute $G(s)$ from Equation 2.5, a process known as the factorization problem. The solution is not unique; in fact, the phase is arbitrary—subject only to certain realizability conditions. To see this, just let $G(j\omega) = A(\omega)e^{j\phi(\omega)}$, where $\phi\omega$ is an arbitrary phase function. Then, Equation 2.3 implies that

$$G(j\omega)G^{*}(j\omega) = A(\omega)e^{j\phi(\omega)} \cdot A(\omega)e^{-j\phi(\omega)} = A^2(\omega) \tag{2.6}$$

If the resulting structure is to have the property of minimum phase [6], the phase function is determined completely by the gain function. If not, one can simply perform the factorization and accept whatever phase function results from the particular process chosen. As has been pointed out earlier in this chapter, it is often desirable that the phase be a linear function of frequency. In this case, one must follow the filter designed by the above process with a phase equalization filter, one that has constant gain and a phase characteristic that, when summed with that of the first filter, produces linear phase. As it happens, the human ear is insensitive to phase nonlinearity, so the phase is not of much importance for filters designed to operate in the audio range. For those intended for video applications, however, it is vitally important. Nonlinear phase produces, for instance, the phenomenon of multiple edges in a reproduced picture.

If completely arbitrary gain characteristics are desired, computer optimization is necessary [6]. Indeed, if phase is of great significance, computer algorithms are available for the simultaneous approximation of both gain and phase. These are complex and unwieldy to use, however, so for more modest applications the above approach relying upon gain approximation only suffices. In fact, the approach arose historically in the telephone industry in its earlier days in which voice transmission was the only concern, data and video transmission being unforeseen at the time. Furthermore, the frequency division multiplexing of voice signals was the primary concern; hence, a number of standard desired shapes of frequency response were generated: low pass, high pass, bandpass, and bandreject (or notch). Typical but stylized specification curves are shown in Figure 2.2. This figure serves to define the following parameters: the minimum passband gain A_p, the maximum stopband gain A_s, the passband cutoff frequency ω_p, the stopband cutoff frequency ω_s (the last two parameters are for low-pass and high-pass filters only), the center frequency ω_o, upper passband and stopband cutoff frequencies ω_{pu} and ω_{su}, and lower passband and stopband cutoff frequencies ω_{pl} and ω_{sl} (the last four parameters are for the bandpass

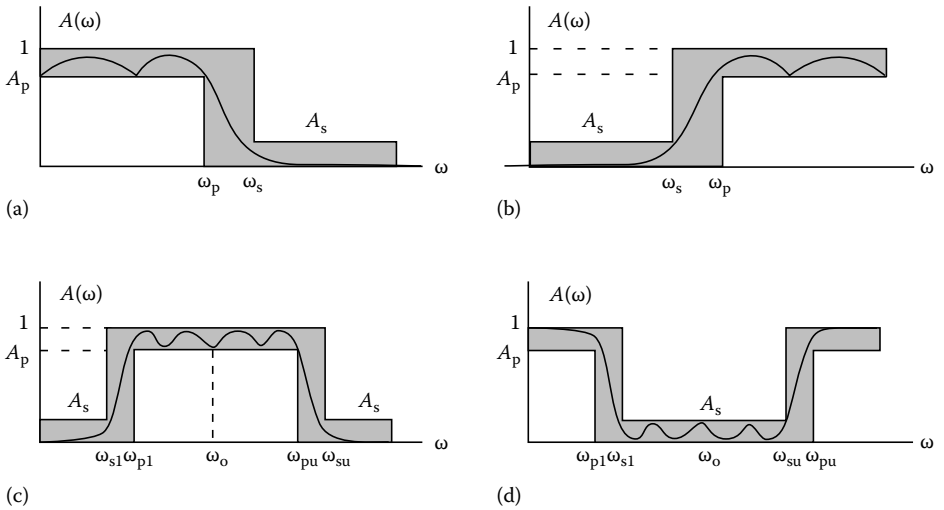


FIGURE 2.2 Catalog of basic filter types: (a) low pass, (b) high pass, (c) bandpass, and (d) bandreject.

and band-reject filters only). As shown in the figure, the maximum passband gain is usually taken to be unity. In the realization process, the transfer function scale factor is often allowed to be a free parameter that is resolved in the design procedure. The resulting "flat gain" (frequency independent) difference from unity is usually considered to be of no consequence, as long as it is not too small, thus creating signal-to-noise ratio problems. There is a fifth standard type that we have not shown: the allpass filter. It has a constant gain for all frequencies, but with a phase characteristic that can be tailored to fit a standard specification in order to compensate for phase distortion. Frequency ranges where the gain is relatively large are called passbands and those where the gain is relatively small, stopbands. Those in between—where the gain is increasing or decreasing—are termed transition bands.

In order to simplify the design even more, one bases the design of all other types of filter in terms of only one: the low pass. In this case, one says that the low-pass filter is a prototype. The specifications of the desired filter type are transformed to those of an equivalent low-pass prototype, and the transfer function for this filter is determined to meet the transformed specifications. Letting the low-pass frequency be symbolized by Ω and the original by ω, one sets

$$\Omega = f(\omega) \tag{2.7}$$

The approximation problem is then solved in terms of Ω. Letting

$$p = j\Omega \tag{2.8}$$

one then has $G(p)$, the desired transfer function.

Two approaches are now possible. One is to apply the inverse transformation, letting

$$s = j\omega = jf^{-1}(\Omega) = jf^{-1}(p/j) \tag{2.9}$$

thus obtaining the desired transfer function

$$G(s) = G(j\omega) = G[jf^{-1}(\Omega)] = G[jf^{-1}(p/j)] \tag{2.10}$$

The other consists of designing the circuit to realize the low-pass prototype filter, then transform *each of the elements* from functions of p to functions of s by means of the complex frequency transformation

$$s = j\omega = jf^{-1}(\Omega) = jf^{-1}(p/j) \tag{2.11}$$

As it happens, the transformation $p = f(s)$ has a special form. It can be shown that if f is real for real values of s, thereby having real parameters, and maps the imaginary axis of the p plane into the imaginary axis of the s plane then it must be an odd function; furthermore, if it is to map *positive real rational functions* into those of like kind (necessary for realizability with R, L, and C elements, as well as possibly ideal transformers*), it must be a *reactance function*. (See Ref. [7] for details.) Since it is always desirable from an economic standpoint to design filters of minimum order, it is desirable that the transformation be of the smallest possible degree. As a result, the following transformations are used:

$$(\text{lpp} \leftrightarrow \text{lp}) \quad p = ks \tag{2.12}$$

$$(\text{lpp} \leftrightarrow \text{hp}) \quad p = \frac{k}{s} \tag{2.13}$$

* For active realizations, those containing dependent sources, this condition is not necessary.

$$(\text{lpp} \leftrightarrow \text{bp}) \quad p = \frac{s^2 + \omega_o^2}{Bs} \tag{2.14}$$

$$(\text{lpp} \leftrightarrow \text{br}) \quad p = \frac{Bs}{s^2 + \omega_o^2} \tag{2.15}$$

where the parameters k, B, and ω_o are real constants to be determined by the particular set of specifications. We have used the standard abbreviations of lpp for *low-pass prototype*, lp for *low pass*, hp for *high pass*, bp for *bandpass*, and br for *bandreject*. Often the letter *f* is added; for example, one might use the acronym brf for *bandreject filter*. The reason for including the transformation in Equation 2.12 is to allow standardization of the lpp. For instance, one can transform from an lpp with, say, a passband cutoff frequency of 1 rad/s to a low-pass filter with a passband cutoff of perhaps 1 kHz.

As a simple example, suppose a bandreject filter were being designed and that the result of the approximation process were

$$H(p) = \frac{1}{p+1} \tag{2.16}$$

Then the br transfer function would be

$$H(s) = [H(p)]_{p=[Bs/(s^2+\omega_o^2)]} = \frac{1}{\frac{Bs}{s^2+\omega_o^2} + 1} = \frac{s^2 + \omega_o^2}{s^2 + Bs + \omega_o^2} \tag{2.17}$$

(The parameters ω_o and B would be determined by the bandreject specifications.) As one can readily see, a first-order lpp is transformed into a second-order brf. In general, for bandpass and bandreject design, the object transfer function is of twice the order of the lpp. Since the example is so simple, it can readily be seen that the circuit in Figure 2.3 realizes the lpp voltage gain function in Equation 2.16. If one applies the transformation in Equation 2.15 the 1-Ω resistor maps into a 1-Ω resistor, but the 1 F capacitor maps into a combination of elements having the admittance

$$Y(p) = p = \frac{1}{\frac{s}{B} + \frac{\omega_o^2}{Bs}} \tag{2.18}$$

But this is simply the series connection of a capacitor of value B/ω_o^2 farads and an inductor of value $1/B$ henrys. The resulting bandreject filter is shown in Figure. 2.4.

The only remaining "loose end" is the determination of the constant(s) in the appropriate transformation equation selected appropriately from Equations 2.12 through 2.15. This will be done here for the

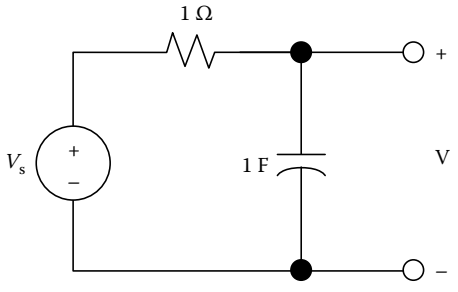


FIGURE 2.3 Low-pass prototype.

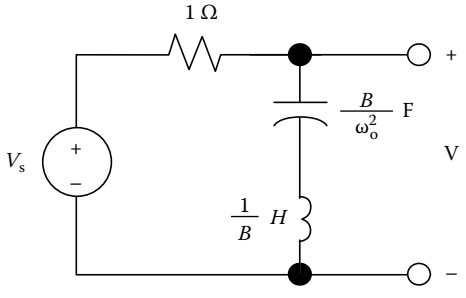


FIGURE 2.4 Resulting bandreject filter.

lpp × bp transformation (Equation 2.14). It is typical, and the reader should have no difficulty working out the other cases. Substituting $p = j\Omega$ and $s = j\omega$ in Equation 2.14, one gets

$$j\Omega = \frac{-\omega^2 + \omega_o^2}{jB\omega} \tag{2.19}$$

or

$$\Omega = \frac{\omega^2 - \omega_o^2}{B\omega} = \frac{\omega}{B} - \frac{\omega_o^2}{B\omega} \tag{2.20}$$

This clearly shows that $\omega = \pm\omega_o$ maps into $\Omega = 0$ and $\omega = \pm\infty$ into $\Omega = \pm\infty$. However, as $\omega \to 0+$, $\Omega \to -\infty$ and as $\omega \to 0-$, $\Omega \to +\infty$. Of perhaps more interest is the inverse transformation. Solving Equation 2.20 for ω in terms of Ω, one finds that*

$$\omega = \frac{B\Omega}{2} + \sqrt{\left[\frac{B\Omega}{2}\right] + \omega_o^2} \tag{2.21}$$

Now, consider pairs of values of Ω, of which one is the negative of the other. Letting ω_+ be the image of Ω with $\Omega > 0$ and ω_- be the image of $-\Omega$, one has

$$\omega_+ = \frac{B\Omega}{2} + \sqrt{\left[\frac{B\Omega}{2}\right] + \omega_o^2} \tag{2.22}$$

and

$$\omega_- = \frac{-B\Omega}{2} + \sqrt{\left[\frac{B\Omega}{2}\right] + \omega_o^2} \tag{2.23}$$

Subtracting, one obtains

$$\omega_+ - \omega_- = B\Omega \tag{2.24}$$

$$\omega_+\omega_- = \omega_o^2 \tag{2.25}$$

Thus, the geometric mean of ω_+ and ω_- is the parameter ω_o; furthermore, the lpp frequencies $\Omega = 1$ rad/s map into points whose difference is the parameter B. Recalling that $A(\Omega)$ has to be an even function of Ω, one sees that the gain magnitudes at these two points must be identical. If the lpp is designed so that $\Omega = 1$ rad/s is the "bandwidth" (single-sided), then the object bpf will have a (two-sided) bandwidth of B rad/s.

An example should clarify things. Figure 2.5 shows a set of bandpass filter gain specifications. Some slight generality has been allowed over those shown in Figure 2.2 by allowing the maximum stopband gains to be different in the two stopbands.

The graph is semilog: the vertical axis is linear with a dB scale and the horizontal axis is a log scale (base 10). The −0.1 dB minimum passband gain, by the way, is called the *passband ripple* because actual

* The negative sign on the radical gives $\omega < 0$ and the preceding treatment only considers positive ω.

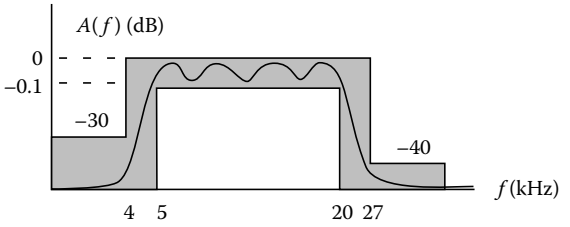


FIGURE 2.5 Bandpass filter specifications.

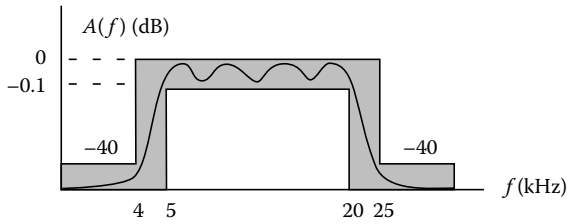


FIGURE 2.6 Bandpass filter specifications.

realized response is permitted to "ripple" back and forth between 0 and −0.1 dB. Notice that the frequency has been specified in terms of kilohertz—often a practical unit. Equation 2.20, however, can be normalized to any unit without affecting Ω. Thus, by normalizing ω to $2\pi \times 10^3$, one can substitute f in kilohertz for ω; the parameters ω_o (replaced of f_o symbolically) and B will then be in kilohertz also.

Now, notice that $5 \times 20 = 100 \neq 4 \times 27 = 108$, so the specifications are not geometrically symmetric relative to any frequency. Somewhat arbitrarily choosing $f_0 = \sqrt{5 \times 20} = 10$ kHz, one can force the specifications to have geometric symmetry by the following device: simply reduce the upper stopband cutoff frequency from 27 to 25 kHz. Then force the two stopband attenuations to be identical by decreasing the −30 dB lower stopband figure to −40 dB. This results in the modified specifications shown in Figure 2.6. If one chooses to map the upper and lower passband cutoff frequencies to $\Omega = 1$ rad/s (a quite typical choice, as many filter design catalogs are tabulated under this assumption), one then has

$$B = 20 - 5 = 15\,\text{kHz} \tag{2.26}$$

This fixes the parameters in the transformation and the lpp stopband frequency can be determined from Equation 2.20:

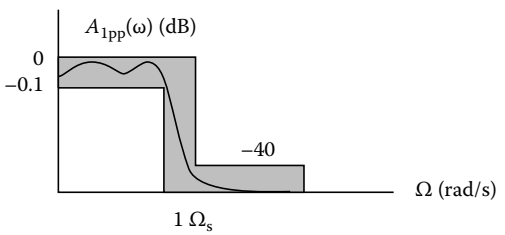


FIGURE 2.7 Bandpass filter specifications.

$$\Omega_s = \frac{25^2 - 10^2}{15 \times 25} = \frac{25}{15} - \frac{100}{375} = 1.498\,\text{rad/s} \tag{2.27}$$

The lpp specifications then assume the form shown in Figure 2.7. Once the lpp approximation problem is solved, one can then transform either the derived transfer function or the synthesized circuit back up to bandpass form since the parameters of the transformation are known.

2.2 Butterworth LPP Approximation

For performing lpp approximations, it is more convenient to work with the *characteristic function* $k(\omega)$ than with the gain function. It is defined by the equation

$$A^2(\omega) = \frac{1}{1 + K^2(\omega)} \tag{2.28}$$

Although Ω was used in Section 2.1 to denote lpp frequency, the lower case ω is used here and throughout the remainder of the section. No confusion should result because frequency will henceforth always mean lpp frequency. The main advantage in using the characteristic function is simply that it approximates zero over any frequency interval for which the gain function approximates unity. Further, it becomes infinitely large when the gain becomes zero. These ideas are illustrated in Figure 2.8. Notice that $K(\omega)$ can be either positive or negative in the passband for it is squared in the defining equation. The basic problem in lpp filter approximation is therefore to find a characteristic function that approximates zero in the passband, approximates infinity in the stopband, and makes the transition from one to the other rapidly. Ideally, it would be exactly zero in the passband, then become abruptly infinity for frequencies in the stopband.

The *n*th-order Butterworth approximation is defined by

$$K(\omega) = \omega^n \tag{2.29}$$

This characteristic function is sketched in Figure 2.9 for two values of n—one small and the other large. As is easily seen, the larger order provides a better approximation to the idea "brick wall" lpp response. Notice, however, that $K(1) = 1$ regardless of the order; hence $A(1) = 0.5$ (-3 dB) regardless of the order.

It is conventional to define the *loss function* $H(s)$ to be reciprocal of the gain function:

$$H(s) = \frac{1}{G(s)} \tag{2.30}$$

Letting $s = j\omega$ and applying Equation 2.28 results in

$$|H(j\omega)|^2 - K^2(\omega) = 1 \tag{2.31}$$

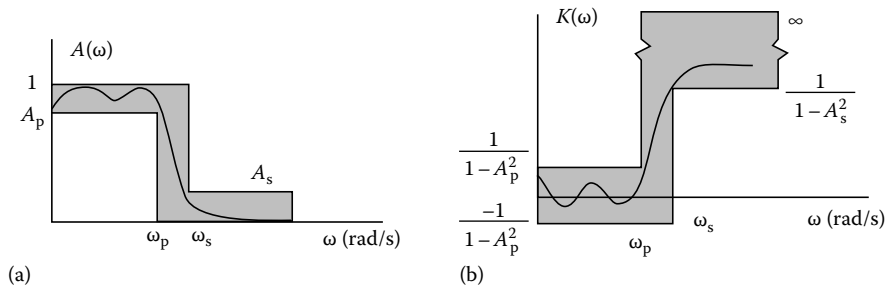


FIGURE 2.8 Filter specifications in terms of the characteristic function: (a) gain and (b) characteristic function.

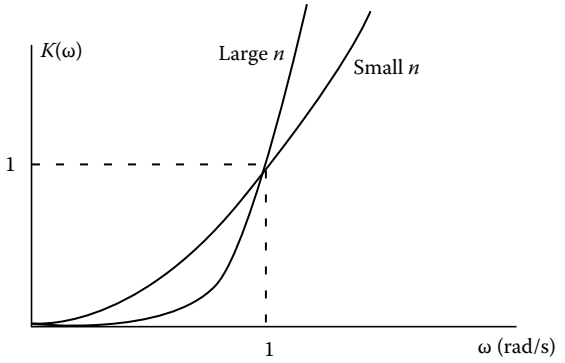


FIGURE 2.9 Butterworth characteristic function.

which is one form of *Feldtkeller's equation*, a fundamental equation in the study of filters. The loss approximates unity wherever the characteristic function approximates zero and infinity when the latter approximates infinity.

The loss function can be used to illustrate a striking property of the Butterworth approximation. Taking the kth derivative of Equation 2.31, one has

$$\frac{d^k|H(j\omega)|^2}{d\omega^k} = \frac{d^k K^2(\omega)}{d\omega^k} = \frac{d^k \omega^{2n}}{d\omega^k} = \frac{(2n)!}{k!}\omega^{2n-k} \tag{2.32}$$

This has the value zero at $\omega = 0$ for $k \leq 2n - 1$. It is the unique polynomial having this property among the set of all monic* polynomials of order $2n$ or lower having the value zero at the origin. But this means that the square of the Butterworth characteristic function $K(\omega)$ is the flattest of all such polynomials at $\omega = 0$. Since adding one to $K^2(\omega)$ produces the loss function $|H(j\omega)|^2$, the same is true of it relative to the set of all loss functions having the value unity at the origin. For this reason, the Butterworth approximation is often called the *maximally flat magnitude* (or MFM) approximation.

The passband ripple parameter A_p is always $1/\sqrt{2}$ for a Butterworth lpp; note that if a different ripple parameter is desired, one must treat the corresponding filter as a general low-pass filter, then apply Equation 3.8 of this book. The value of the parameter in that frequency transformation is determined by the requirement that the frequency at which the desired filter assumes the value $1/\sqrt{2}$ map into a lpp passband frequency of 1 rad/s. The required order is determined from the equation

$$\frac{1}{\sqrt{1+\omega_s^{2n}}} \leq A_s \tag{2.33}$$

or, rearranged,

$$n \geq \frac{\log\left[\frac{1}{A_s^2} - 1\right]}{2\log(\omega_s)} \tag{2.34}$$

The value of n is, of course, chosen to be the smallest integer greater than the expression on the right-hand side of Equation 2.34.

* A monic polynomial is one where the leading coefficient (highest power of ω) is unity.

Since only one parameter is in the Butterworth approximation, Equation 2.34 completely determines $A(\omega)$. That is, the MFM is a one-parameter approximation. The only remaining item of concern is the determination of $G(s)$, for synthesis of the actual filter requires knowledge of the transfer function. As was pointed out in Section 2.1, this is the factorization problem. In general, the solution is given by Equation 2.5, repeated here for convenience as Equation 2.35:

$$G(s)G(-s) = A^2(-s^2) \tag{2.35}$$

In the present case, we have

$$G(s)G(-s) = \frac{1}{1+(-s^2)^n} = \frac{1}{1+(-1)^n s^{2n}} \tag{2.36}$$

How does one find $G(s)$ from this equation? The solution merely lies in applying the restriction that the resulting filter is to be stable. This means that the poles of $G(s)G(-s)$ that lie in the right-half plane must be discarded and the remaining ones assigned to $G(s)$. In this connection, observe that any poles on the imaginary axis must be off even multiplicity since $G(s)G(-s)$ is an even function (or, equivalently, since $A^2(\omega^2) = |G(j\omega)|^2$ is nonnegative). Furthermore, any even-order poles of $G(s)G(-s)$ on the imaginary axis could only result from one or more poles of $G(s)$, itself, at the same location. But such a $G(s)$ represents a filter that is undesirable because, at best, it is only marginally stable. As will be shown, this situation does not occur for Butterworth filters.

The problem now is merely to find all the poles of $G(s)G(-s)$, then to sort them. These poles are located at the zeros of the denominator in Equation 2.36. Thus, one must solve

$$1 + (-s^2)^n = 0 \tag{2.37}$$

or, equivalently,

$$s^{2n} = (-1)^{n-1} \tag{2.38}$$

Representing s in polar coordinates by

$$s = \rho e^{j\phi} \tag{2.39}$$

one can write Equation 2.38 in the form

$$\rho^{2n} e^{j2n\phi} = e^{j(n-1)\pi} \tag{2.40}$$

This has the solution

$$\rho = 1 \tag{2.41}$$

and

$$\phi = \frac{\pi}{2} + (2k-1)\frac{\pi}{2n} \tag{2.42}$$

where k is any integer. Of course, only those values of ϕ between 0 and 2π rad are to be considered unique.

As an example, suppose that $n = 2$; that is, one is interested in determining the transfer function of a second-order Butterworth filter. Then the unique values of ϕ determined from Equation 2.42 are $\pi/4$, $3\pi/4$, $5\pi/4$, and $7\pi/4$. All other values are simply these four with integer multiples of 2π added. The last two represent poles in the right-half plane, so are simply discarded. The other two correspond to poles at $s = -0.707 \pm j0.707$. Letting $D(s)$ be the numerator polynomial of $G(s)$, then, one has

$$D(s) = s^2 + \sqrt{2}s + 1 \tag{2.43}$$

The poles of $G(s)G(-s)$ are sketched in Figure 2.10. Notice that if one assigns the left-half plane poles to $G(s)$, then those in the right-half plane will be those of $G(-s)$.

Perhaps another relatively simple example is in order. To this end, consider the third-order Butterworth transfer function $n = 3$. In this case, the polar angles of the poles of $G(s)$ are at $\pi/3$, $2\pi/3$, π, $4\pi/3$, $5\pi/3$, and 2π. This pole pattern is shown in Figure 2.11. The denominator polynomial corresponds to those in the left-half plane. It is

$$D(s) = (s+1)\left(s^2 + s + 1\right) = s^3 + 2s^2 + 2s + 1 \tag{2.44}$$

The pattern for the general nth order case is similar—all the poles of $G(s)G(-s)$ lie on the unit circle and are equally spaced at intervals of π/n radians, but are offset by $\pi/2$ radians relative to the positive real axis. A little thought will convince one that this means that no poles ever fall on the imaginary axis.

The factorization procedure determines the denominator polynomial in $G(s)$. But what about the numerator? Since the characteristic function is a polynomial, it is clear that $G(j\omega)$, and hence $G(s)$ itself, will have a constant numerator. For this reason, the Butterworth approximation is referred to as an *all-pole* filter. Sometimes it is also called a *polynomial* filter, referring to the fact that the characteristic function is a polynomial. As was mentioned earlier, the constant is usually allowed to float freely in the synthesis process and is determined only at the conclusion of the design process. However, more can be said. Writing

$$G(s) = \frac{a}{D(s)} \tag{2.45}$$

one can apply Equation 2.36 to show that $a^2 = 1$, provided that $|G(0)|$ is to be one, as is the case for the normalized lpp. This implies that

$$a = \pm 1 \tag{2.46}$$

If a passive unbalanced (grounded) filter is desired the positive sign must be chosen. Otherwise, one can opt for either.

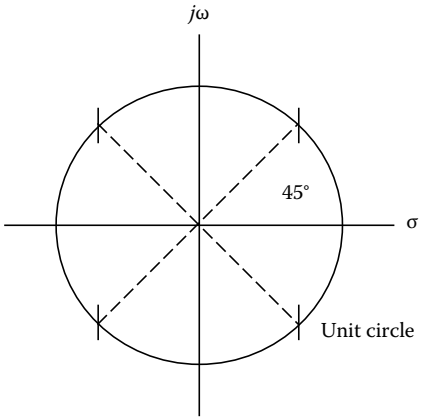


FIGURE 2.10 Poles of the second-order Butterworth transfer function.

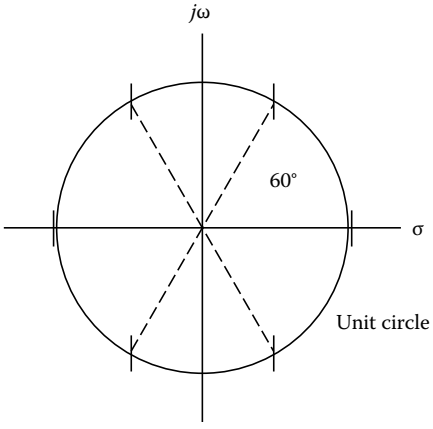


FIGURE 2.11 Poles of the third-order Butterworth transfer function.

2.3 Chebyshev LPP Approximation

The main advantage of the Butterworth approximation is that it is simple. It does not, however, draw upon the maximum approximating power of polynomials. In fact, a classical problem in mathematics is to approximate a given continuous function on a closed bounded interval with a polynomial of a specified maximum degree. One can choose to define the error of approximation in many ways, but the so-called minimax criterion seems to be the most suitable for filter design. It is the minimum value, computed over all polynomials of a specified maximum degree, of the maximum difference between the polynomial values and those of the specified function. This is illustrated in Figure 2.12. The minimax error in this case occurs at ω_x. It is the largest value of the magnitude of the difference between the function values $f(\omega)$ and those of a given candidate polynomial $p_n(\omega)$. The polynomial of best fit is the one for which this value is the smallest.

The basic lpp approximation problem is to pick the characteristic function to be that polynomial of a specified maximum degree no more than, say, n, which gives the smallest maximum error of approximation *to the constant value* 0 over the interval $0 \leq \omega \leq 1$ (arbitrarily assuming that the passband cutoff is to be 1 rad/s). In this special case, the solution is known in closed form: it is the Chebyshev* polynomial of degree n. Then, $K(\omega)$ is polynomial $\varepsilon T_n(\omega)$, where

$$T_n(\omega) = \cos\left[n \cos^{-1}(\omega)\right] \tag{2.47}$$

and ε is the minimax error (a constant).

It is perhaps not clear that $T_n(\omega)$ is actually a polynomial; however, upon computing the first few by applying simple trigonometric identities one has the results shown in Table 2.1. In fact, again by calling upon simple trigonometric identities, one can derive the general recurrence relation

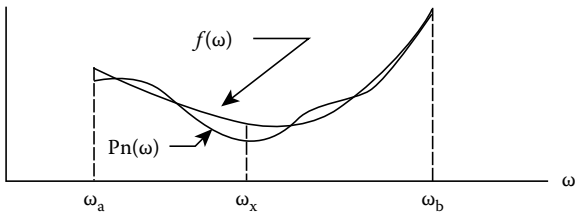


FIGURE 2.12 Minimax error criterion.

TABLE 2.1 Chebyshev Polynomial of Degree n

n	$T_n(\omega)$
0	1
1	ω
2	$2\omega^2 - 1$
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$

* If the name of the Russian mathematician is transliterated from the French, in which the first non-Russian translations were given, it is spelled Tchebychev.

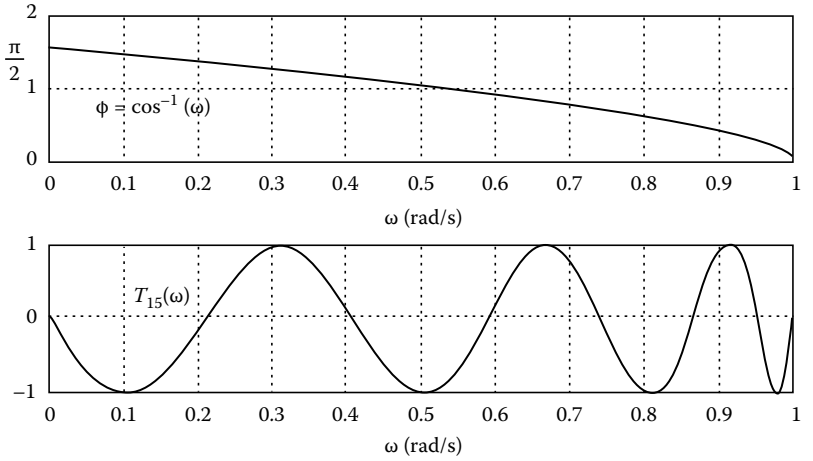


FIGURE 2.13 Behavior of the Chebyshev polynomials.

$$T_n(\omega) = 2\omega T_{n-1}(\omega) - T_{n-2}(\omega); \quad n \geq 2 \tag{2.48}$$

The Chebyshev polynomials have an enormous number of interesting properties and to explore them all would require a complete monograph. Among those of the most interest for filtering applications, however, are these. First, from the recursion relationship (Equation 2.48) one can see that $T_n(\omega)$ is indeed a polynomial of order n; furthermore, its leading coefficient is 2^{n-1}. If n is even, $T_n(\omega)$ is an even polynomial in ω and if n is odd, $T_n(\omega)$ is an odd polynomial. The basic definition in Equation 2.47 clearly shows that the extreme values of $T_n(\omega)$ over the interval $0 \leq \omega \leq 1$ are ± 1. Some insight into the behavior of the Chebyshev polynomials can be obtained by making the transformation $\phi = \cos^{-1}(\omega)$. Then, $T_n(\phi) = \cos(n\phi)$, a trigonometric function that is quite well known. The behavior of T_{15}, for example, is shown in Figure 2.13. The basic idea is this: the Chebyshev polynomial of nth order is merely a cosine of "frequency" $n/4$, which "starts" at $\omega = 1$ and "runs backward" to $\omega = 0$. Thus, it is always 1 at $\omega = 1$ and as ω goes from 1 rad/s to 0 rad/s backward, it goes through n quarter-periods (or $n/4$ full periods). Thus, at $\omega = 0$ the value of this polynomial will be either 0 or ± 1, depending upon the specific value of n. If n is even, an integral number of half-periods will have been described and the resulting value will be ± 1; if n is odd, an integral number of half-periods plus a quarter-period will have been described and the value will be zero.

Based on the foregoing theory, one sees that the best approximation to the ideal lpp characteristic over the passband is, for a given passband tolerance, ε, given by

$$A^2(\omega) = \frac{1}{1 + \varepsilon^2 T_n^2(\omega)} \tag{2.49}$$

It is, of course, known as the Chebyshev approximation and the resulting filter as the Chebyshev lpp of order n. The gain magnitude $A(\omega)$ is plotted for $n = 5$ and $\varepsilon = 0.1$ in Figure 2.14. The passband behavior looks like ripples in a container of water, and since the crests are equally spaced above and below the average value, it is called equiripple behavior. In the passband, the maximum value is 1 and the minimum value is

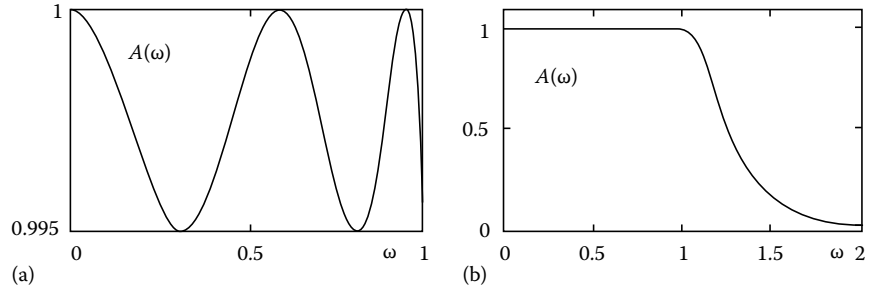


FIGURE 2.14 Frequency response of a fifth-order Chebyshev lpp: (a) passband and (b) overall.

$$A_{\min} = \frac{1}{\sqrt{1+\varepsilon^2}} \tag{2.50}$$

The passband ripple is usually specified as the peak to peak variation in dB. Since the maximum value is one, that is 0 dB, this quantity is related to the *ripple parameter* ε through the equation

$$\text{passband ripple in dB} = 20\log\sqrt{1+\varepsilon^2} \tag{2.51}$$

The Chebyshev approximation is the best possible among the class of all-pole filters—*over the passband*. But what about its stopband behavior? As was pointed out previously, it is desirable that—in addition to approximating zero in the passband—the characteristic function should go to infinity as rapidly as possible in the stopband. Now it is a happy coincidence that the Chebyshev polynomial goes to infinity for $\omega > 1$ faster than any other polynomial of the same order. Thus, the Chebyshev approximation is the best possible among the class of polynomial, or all-pole, filters.

The basic definition of the Chebyshev polynomial works fine for values of ω in the passband, where $\omega \leq 1$. For larger values of ω, however, $\cos^{-1}(\omega)$ is a complex number. Fortunately, there is an alternate form that avoids complex arithmetic. To derive this form, simply recognize the complex nature of $\cos^{-1}(\omega)$ explicitly and write

$$x = \cos^{-1}(\omega) \tag{2.52}$$

One then has*

$$\omega = \cos(x) = \cos[\,j(jx)] = \cosh(jx) \tag{2.53}$$

so

$$jx = \cosh^{-1}(\omega) \tag{2.54}$$

Thus, one can also write

$$T_n(\omega) = \cos[jn(jx)] = \cosh[n(jx)] = \cosh[n\cosh^{-1}(\omega)] \tag{2.55}$$

* Since $\cos(x) = \cosh(jx)$.

This result is used to compute the required filter order. Assuming as usual that A_s is the maximum allowed stopband gain, one uses the square root of Equation 2.49 to get

$$A(\omega_s) = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2(\omega_s)}} \leq A_s \tag{2.56}$$

Solving, one has

$$T_n(\omega_s) \geq \frac{1}{\varepsilon}\sqrt{\frac{1}{A_s^2} - 1} \tag{2.57}$$

Since $\omega_s \geq 1$, it is most convenient to use the hyperbolic form in Equation 2.55:

$$n \geq \frac{\cosh^{-1}\left[\frac{1}{\varepsilon}\sqrt{\frac{1}{A_s^2} - 1}\right]}{\cosh^{-1}(\omega_s)} \tag{2.58}$$

To summarize, one first determines the parameter ε from the allowed passband ripple, usually using Equation 2.51; then one determines the minimum order required using Equation 2.58. The original filter specifications must, of course, be mapped into the lpp domain through appropriate choice(s) of the constant(s) in the transformation Equations 2.12 through 2.15. Notice that the definition of passband for the Chebyshev filter differs from that of the Butterworth unless the passband ripple is 3 dB. For the Chebyshev characteristic, the passband cutoff frequency is that frequency at which the gain goes through the value $1/\sqrt{1+\varepsilon^2}$ for the last time as ω increases.

The only remaining item is the determination of the transfer function $G(s)$ by factorization. Again, this requires the computation of the poles of $G(s)G(-s)$. Using Equation 2.49, one has

$$G(s)G(-s) = \left[A^2(\omega)\right]_{s=j\omega} = \frac{1}{1 + \varepsilon^2\left[T_n^2(\omega)\right]_{s=j\omega}} \tag{2.59}$$

Thus, the poles are at those values of s for which

$$\left[T_n^2(\omega)\right]_{s=j\omega} = \frac{-1}{\varepsilon^2} \tag{2.60}$$

or*

$$\cos\left[n\ \cos^{-1}\left(\frac{s}{j}\right)\right] = \pm j\frac{1}{\varepsilon} \tag{2.61}$$

Letting

$$\cos^{-1}(s/j) = a + jb, \text{ there results} \tag{2.62}$$

$$\cos(na)\cosh(nb) - j\sin(na)\sinh(nb) = \pm j\frac{1}{\varepsilon} \tag{2.63}$$

* Since s is complex anyway, nothing is to be gained from using the hyperbolic form.

Equating real and imaginary parts, one has

$$\cos(na)\cos(nb) = 0 \tag{2.64a}$$

and

$$\sin(na)\sinh(nb) = \pm\frac{1}{\varepsilon} \tag{2.64b}$$

Since $\cosh(nb) > 0$ for any b, one must have

$$\cos(na) = 0 \tag{2.65}$$

which can hold only if

$$a = (2k+1)\frac{\pi}{2n} \quad k \text{ any integer} \tag{2.66}$$

But, in this case, $\sin[(2k+1)\pi/2] = \pm 1$, so application of Equation 2.64b gives

$$\sinh(nb) = \pm\frac{1}{\varepsilon} \tag{2.67}$$

One can now solve for b:

$$b = \pm\frac{1}{n}\sinh^{-1}\left[\frac{1}{\varepsilon}\right] \tag{2.68}$$

Equations 2.66 through 2.68 together determine a and b, hence $\cos^{-1}(s/j)$. Taking the cosine of both sides of Equation 2.62 and using Equations 2.65 and 2.68 gives

$$\begin{aligned} s = &\pm\sin\left[(2k+1)\frac{\pi}{2n}\right]\sinh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \\ &+ j\cos\left[(2k+1)\frac{\pi}{2n}\right]\cosh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \end{aligned} \tag{2.69}$$

Letting $s = \sigma + j\omega$ as usual, one can rearrange Equation 2.69 into the form

$$\left[\frac{\sigma}{\sinh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right]}\right]^2 + \left[\frac{\omega}{\cosh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right]}\right]^2 = 1 \tag{2.70}$$

which is the equation of an ellipse in the s plane with real axis intercepts of

$$\sigma_o = \pm\sinh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \tag{2.71}$$

and imaginary axis intercepts of

$$\omega_o = \pm\cosh\left[\frac{1}{n}\sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \tag{2.72}$$

This is shown in Figure 2.15.

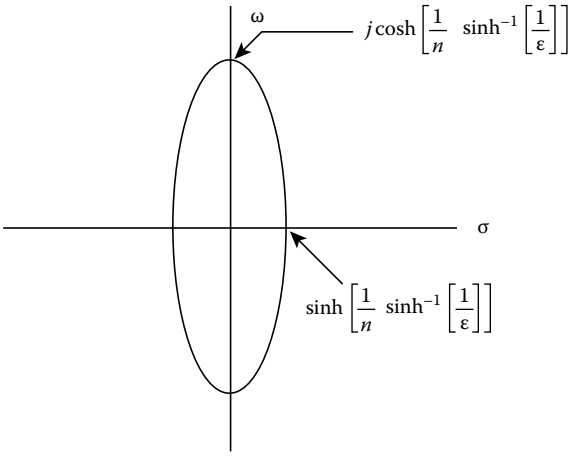


FIGURE 2.15 Pole locations for $G(s)G(-s)$ for the Chebyshev filter.

As an example, suppose that a Chebyshev lpp is to be designed that has a passband ripple of 0.1 dB, a maximum stopband gain of −20 dB, and $\omega_s = 2$ rad/s. Then, one can use Equation 2.51 to find* the ripple parameter ε:

$$\varepsilon = \sqrt{10^{0.1/10} - 1} = 0.1526 \tag{2.73}$$

Equation 2.58 gives the minimum order required:

$$n \geq \frac{\cosh^{-1}\left[\frac{1}{0.1526}\sqrt{\frac{1}{(0.1)^2} - 1}\right]}{\cosh^{-1}(2)} = \frac{\cosh^{-1}(65.20)}{\cosh^{-1}(2)} = 3.70 \tag{2.74}$$

In doing this computation by hand, one often uses the identity

$$\cosh^{-1}(x) = \ln\left[x + \sqrt{x^2 - 1}\right] \tag{2.75}$$

which can be closely approximated by

$$\cosh^{-1}(x) = \ln(2x) \tag{2.76}$$

if $x \gg 1$. In the present case, a fourth-order filter is required. The poles are shown in Table 2.2 and graphed in Figure 2.16.

By selecting the left-half plane poles and forming the corresponding factors, then multiplying them, one finds the denominator polynomial of $G(s)$ to be

$$D(s) = s^4 + 1.8040s^3 + 2.2670s^2 + 2.0257s + 0.8286 \tag{2.77}$$

* For practical designs, a great deal of precision is required for higher-order filters.

TABLE 2.2 Pole Locations

k	Real Part	Imaginary Part
0	0.2642	1.1226
1	0.6378	0.4650
2	0.6378	−0.4650
3	0.2642	−1.1266
4	−0.2642	−1.1266
5	−0.6378	−0.4650
6	−0.6378	0.4650
7	−0.2642	1.1226

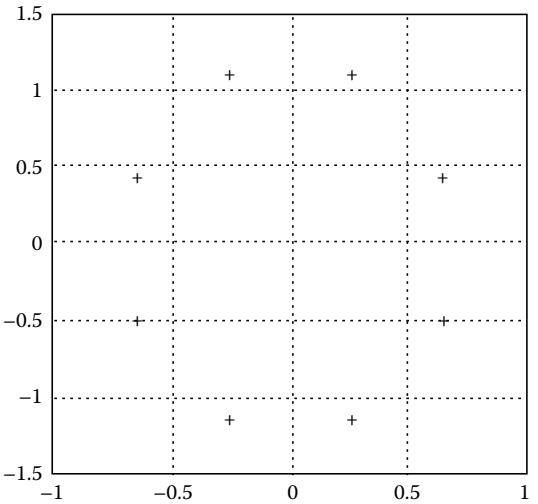


FIGURE 2.16 Pole locations for example filter.

As for the Butterworth example of Section 2.2, one finds the scale factor:

$$k = 0.8286 \tag{2.78}$$

Therefore, the complete transfer function is

$$G(s) = \frac{k}{D(s)} = \frac{0.8286}{s^4 + 1.8040s^3 + 2.2670s^2 + 2.0257s + 0.8286} \tag{2.79}$$

Of course, if completely automated algorithms are not being used to design such a filter as the one in our example, numerical assistance is required. The computation in the preceding example was performed in MATLAB®—a convenient package for many computational problems in filter design.

2.4 Bessel–Thompson LPP Approximation

Thus far, two all-pole approximations have been presented. As was pointed out in Section 2.3, the Chebyshev is better than the Butterworth—in fact, it is the best all-pole approximation available. So why bother with the Butterworth at all, other than as an item of historical interest? The answer lies in the phase. The Chebyshev filter has a phase characteristic that departs farther from linearity than that of the

Butterworth; put differently, its group delay deviates from a constant by a greater amount. Thus, the Butterworth approximation is still a viable approximation in applications where phase linearity is of some importance.

The question naturally arises as to whether there is an lpp approximation that has better phase characteristics than that of the Butterworth. The answer is yes, and that is the topic of this section, which will follow [8]—perhaps the simplest development of all.

Recall that the gain function $G(s)$ is the inversion of the loss function $H(s)$:

$$G(s) = \frac{1}{H(s)} \tag{2.80}$$

Also, recall that it was the loss function $|h(j\omega)| = 1/A(\omega)$ (or, rather, its square $|h(j\omega)|^2$) that was required to have MFM property at $\omega = 0$,

$$\left[\frac{\mathrm{d}^k |H(j\omega)|^2}{\mathrm{d}\omega^k}\right]_{\omega=0} = \left[\frac{\mathrm{d}^k \omega^{2n}}{\mathrm{d}\omega^k}\right]_{\omega=0} = \left[\frac{(2n)!}{k!}\omega^{2n-k}\right]_{\omega=0} = 0 \tag{2.81}$$

for $k = 0, 1, \ldots, 2n-1$. The question to be asked and answered in this section is whether there exists a similar approximation for the group delay $\tau_g(\omega)$ a maximally flat delay (MFD) approximation.

To answer this question, the phase will be written in terms of the loss function $H(j\omega)$. Since the latter quantity can be written in polar form as

$$H(j\omega) = \frac{1}{G(j\omega)} = \frac{1}{A(\omega)e^{j\phi(\omega)}} = \frac{1}{A(\omega)}e^{-j\phi(\omega)} \tag{2.82}$$

the (complex) logarithm is

$$\ln H(j\omega) = -\ln A(\omega) - j\phi(\omega) \tag{2.83}$$

Thus,

$$\phi(\omega) = -\mathrm{Im}\{\ln H(j\omega)\} \tag{2.84}$$

The group delay is merely the negative of the derivative of the phase, so one has

$$\begin{aligned}\tau_g(\omega) &= -\frac{\mathrm{d}\phi}{\mathrm{d}\omega} = \frac{\mathrm{d}}{\mathrm{d}\omega}\mathrm{Im}\{\ln H(j\omega)\} = \mathrm{Im}\left\{\frac{\mathrm{d}}{\mathrm{d}\omega}\ln H(j\omega)\right\} \\ &= \mathrm{Im}\left\{j\frac{\mathrm{d}}{\mathrm{d}(j\omega)}\ln H(j\omega)\right\} = \mathrm{Re}\left\{\frac{\mathrm{d}}{\mathrm{d}(j\omega)}\ln H(j\omega)\right\}\end{aligned} \tag{2.85}$$

Recalling that the even part of a complex function $F(s)$ is given by

$$\mathrm{Ev}\{F(s)\} = \frac{F(s) + F(-s)}{2} \tag{2.86}$$

one can use the symmetry property for real $F(s)$ (in the present context, $F(s)$ is assumed to be a rational function, so a real $F(s)$ is one with real coefficients) to show that

$$\mathrm{Ev}\{F(s)\}_{s=j\omega} = \frac{F(j\omega) + F(-j\omega)}{2} = \frac{F(j\omega) + F^{*}(j\omega)}{2} = \mathrm{Re}\{F(j\omega)\} \tag{2.87}$$

In this manner, one can analytically extend the group delay function $\tau_g(j\omega)$ so that it becomes a function of s:

$$\tau_g(s) = \mathrm{Ev}\left\{\frac{\mathrm{d}}{\mathrm{d}s}\ln H(s)\right\} = \mathrm{Ev}\left[\frac{H'(s)}{H(s)}\right] = \frac{1}{2}\left[\frac{H'(s)}{H(s)} + \frac{H'(-s)}{H(-s)}\right] \tag{2.88}$$

In the present case, it will be assumed that $H(s)$ is a polynomial. In this manner, an all-pole (or polynomial) filter will result. Thus, one can write

$$H(s) = \sum_{k=0}^{n} a_k^n s^k \tag{2.89}$$

The superscript on the (assumed real) coefficient matches the upper limit on the sum and is the assumed filter order. The group delay function is, thus, a real rational function:

$$\tau_g(s) = \frac{N(s)}{D(s)} \tag{2.90}$$

where $N(s)$, the numerator polynomial, and $D(s)$, the denominator polynomial, have real coefficients and are of degrees no greater than $2n-1$ and $2n$, respectively, by inspection of Equation 2.88. But, again according to Equation 2.88, $\tau_g(s)$ is an even function. Thus,

$$\tau_g(-s) = \frac{N(-s)}{D(-s)} = \tau_g(s) = \frac{N(s)}{D(s)} \tag{2.91}$$

The last equation, however, implies that

$$\frac{N(-s)}{N(s)} = \frac{D(-s)}{D(s)} = 1 \tag{2.92}$$

The last equality is arrived at by the following reasoning. $N(s)$ and $D(s)$ are assumed to have no common factors—any such have already been cancelled in the formation of $\tau_g(s)$. Thus, the two functions of s in Equation 2.92 are independent; since they must be equal for s, they must therefore equal a constant. But this constant is unity, as is easily shown by allowing s to become infinite and noting that the degrees and leading coefficients of $N(-s)$ and $N(s)$ are the same.

The implication of the preceding development is simply that $N(s)$ and $D(s)$ consist of only even powers of s. Looking at $N(s)$, for example, and letting it be written

$$N(s) = \sum_{k=0}^{2n-1} \rho_k s^k \tag{2.93}$$

one has

$$N(-s) = \sum_{k=0}^{2n-1} \rho_k (-s)^k = N(s) = \sum_{k=0}^{m} \rho_k (s)^k \tag{2.94}$$

This however implies that the ρ_k are zero for odd k. Hence $N(s)$ consists of only even powers of s. The same is true of $D(s)$ and therefore of $\tau_g(s)$. Clearly, therefore $\tau_g(\omega)$ will consist of only even powers of ω, that is, it will be a function of ω^2. Now there is to be an MFD approximation one must have by analogy with the MFM approximation in Section 2.2,

$$\frac{d^k \tau_g(\omega)}{d(\omega^2)^k} = 0 \tag{2.95}$$

for $k = 1, 2, \ldots, n-1$. The constraint is, of course, that τ_g must come from a polynomial loss function $H(s)$ whose zeros [poles of $G(s)$] all lie in the left-half plane.

It is convenient to normalize time so that the group delay T, say, at $\omega = 0$ is 1 s; this is equivalent to scaling the frequency variable ω to be ωT. Here, it will be assumed that this has been performed already. A slight difference exists between the MFD and MFM approximations; the latter approximates zero at $\omega = 0$, while the former approximates $T = 1s$ at $\omega = 0$. The two become the same, however, if one considers the function

$$\tau_g(s) - 1 = \frac{P(s)}{2H(s)H(-s)} \tag{2.96}$$

where $P(s)$ has a maximum order of $2n$. The form on the right-hand side of this expression can be readily verified by consideration of Equation 2.88, the basic result for the following derivation. Furthermore, writing $P(s)$ in the form

$$P(s) = \sum_{k=0}^{n} P_k \left(s^2\right)^k \tag{2.97}$$

it is readily seen that

$$p_o = p_1 = \cdots = p_{n-1} = 0 \tag{2.98}$$

The lowest-order coefficient is clearly zero because $\tau_g(0) = 1$; furthermore, all odd coefficients are zero since τ_g is even. Finally, all other coefficients in Equation 2.98 have to be zero if one imposes the MFD condition in Equation 2.95.*

At this stage, one can write the group delay function in the form

$$\tau_g(s) = \frac{N(s)}{D(s)} = 1 + \frac{p_n s^{2n}}{2H(s)H(-s)} = \frac{2H(s)H(-s) + p_n s^{2n}}{2H(s)H(-s)} \tag{2.99}$$

It was pointed out immediately after Equation 2.90, however, that the degree of $N(s)$ is at most $2n-1$. Hence, the coefficient of s^{2n} in the numerator of Equation 2.99 have vanished, and one therefore also has

$$2(-1)^n a_n + p_n = 0 \tag{2.100}$$

or, equivalently,

$$p_n = 2(-1)^{n+1} a_n \tag{2.101}$$

* The derivatives are easy to compute, but this is omitted here for reasons of space.

Finally, this allows one to write Equation 2.99 in the form

$$\tau_g(s) = \frac{H(s)H(-s) + (-1)^{n+1}a_n s^{2n}}{H(s)H(-s)} \tag{2.102}$$

If one now equates Equations 2.102 and 2.88 and simplifies, there results

$$H'(s)H(-s) + H'(-s)H(s) - 2H(s)H(-s) = 2(-1)^{n+1}a_n s^{2n} \tag{2.103}$$

Multiplying both sides by s^{-2n}, taking the derivative (noting that $(\mathrm{d}/\mathrm{d}s)H(-s) = -H'(-s)$), one obtains (after a bit of algebra)

$$\mathrm{Ev}\{[sH''(s) - 2(s+n)H'(s) + 2nH(s)]H(-s)\} = 0 \tag{2.104}$$

Now, if s_o is a zero of this even function, the $-s_o$ will also be a zero. Further, since all of the coefficients in $H(s)$ are real, s_0^* and $-s_0^*$ must also be zeros as well; that is, the zeros must occur in a *quadrantally symmetric* manner. Each zero must belong either to $H(-s)$, or to the factor it multiplies, or to both. The degree of the entire expression in Equation 2.40 is $2n$ and $H(-s)$ has n zeros. Thus, the expression in square brackets must have n zeros. Now here is the crucial step in the logic: if the filter being designed is to be stable, then all n zeros $H(-s)$ must be in the (open) right-half plane. This implies that the factor in square brackets must have n zeros in the (open) left-half plane. Since the expression has degree n, these zeros can be found from the equation

$$sH''(s) - 2(s+n)H'(s) + 2nH(s) = 0 \tag{2.105}$$

This differential equation can be transformed into that of Bessel; here, however, the solution will be derived directly by recursion. Using Equation 2.89 for $H(s)$, computing its derivatives, reindexing, and using Equation 2.105, one obtains, for $0 \le k \le n-1$,

$$(k+1)ka_{k+1}^n - 2n(k+1)a_{k+1}^n - 2ka_k^n + 2na_k^n = 0 \tag{2.106}$$

This produces the recursion formula

$$a_{k+1}^n = \frac{2(n-k)}{(2n-k)(k+1)}a_k^n; \quad 0 \le k \le n-1 \tag{2.107}$$

or, normalizing the one free constant so that $a_n^n = 1$ and reindexing, gives

$$a_{k+1}^n = \frac{2(n-k)}{(2n-k)(k+1)}; \quad 0 \le k \le n-1 \tag{2.108}$$

The resulting polynomials for $H(s)$ are closely allied with the Bessel polynomials. The first several are given in Table 2.3 and the corresponding gain and group delay characteristics are plotted (using MATLAB) in Figure 2.17. Notice that the higher the order, the more accurately the group delay approximates a constant and the better the gain approximates the ideal lpp; the latter behavior, however, is fairly poor. A view of the group delay behavior in the passband is shown for the third-order filter in Figure 2.18.

TABLE 2.3 Bessel Polynomials of Order n

n	$H(s)$
1	$s+1$
2	s^2+3s+3
3	$s^3+6s^2+15s+15$
4	$s^4+10s^3+45s^2+105s+105$

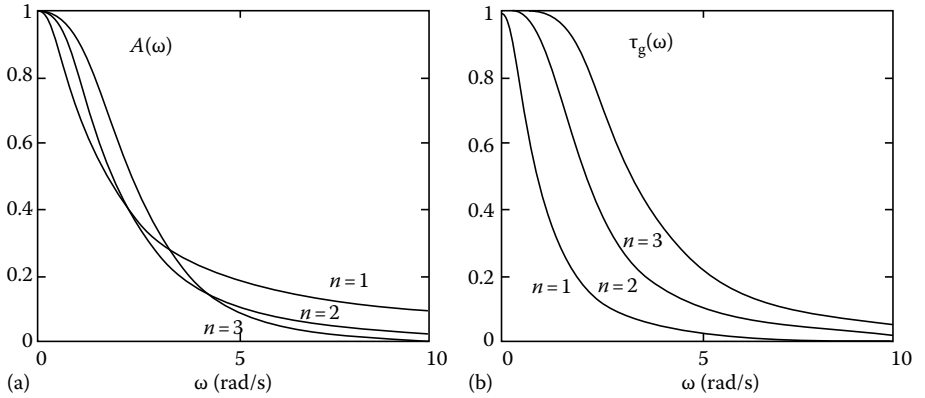


FIGURE 2.17 Gain and group delay characteristics for the Bessel–Thompson filters of orders one through three: (a) gain and (b) group delay.

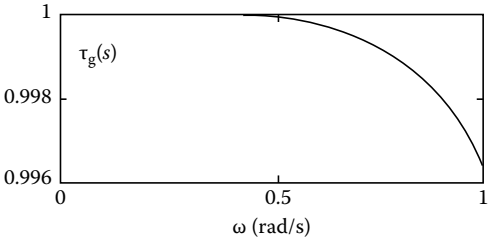


FIGURE 2.18 Group delay of third-order Bessel filter.

2.5 Elliptic Approximation

The Butterworth approximation provides a very fine approximation to the ideal "brick wall" lpp response at $\omega = 0$, but is poor for other frequencies; the Chebyshev, on the other hand, spreads out the error of approximation throughout the passband and thereby achieves a much better amplitude approximation. This is one concrete application of a general result in approximation theory known as the Weierstrass theorem [2], which asserts the possibility of uniformly approximating a continuous function on a compact set by a polynomial or by a rational function.

A compact set is the generalization of a closed and bounded interval—the setting for the Chebyshev approximation sketched in Figure 2.19. The compact set is the closed interval $[-1, 1]$,* the continuous

* Here, we are explicitly observing that the characteristic function to be approximated is even in ω.

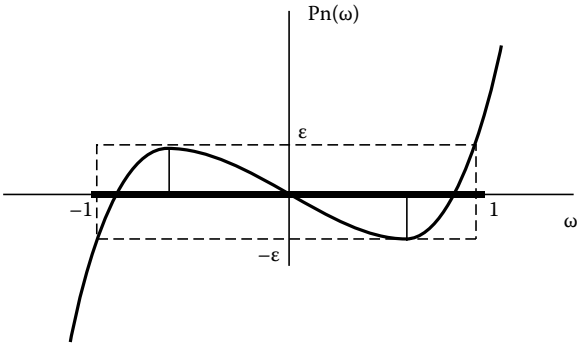


FIGURE 2.19 Chebyshev approximation.

function to be approximated is the constant 0, and the approximating function is a polynomial of degree n (in the figure n is three). A careful inspection of this figure reveals that the maximum error of approximation occurs exactly four times on a set of four discrete points within the interval and the *sign* of this maximum error alternates from one such point to its adjacent neighbor. This is no coincidence; in fact, the polynomial of best approximation of degree n is characterized by this alternation or *equiripple* property: the error function achieves its maximum value exactly $n+1$ times on the interval of approximation and the signs at these points alternate. If one finds a polynomial by any means that has this property it is the unique polynomial of best approximation.

In the present case, one simply notes that the cosine function has the required equiripple behavior. That is, as ϕ varies between $-\pi/2$ and $\pi/2$, $\cos(n\phi)$ varies between its maximum and minimum values of ± 1 a total of $n+1$ times. But $\cos(n\phi)$ is not a polynomial, and the problem (for reasons stated in the introduction) is to determine a *polynomial* having this property. At this point, one observes that if one makes the transformation

$$\phi = \cos^{-1}(\omega) \tag{2.109}$$

then, as ω varies from -1 to $+1$, ϕ varies from $-\pi/2$ to $+\pi/2$. This transformation is, fortunately, one-to-one over this range; furthermore, even more fortunately, the overall function

$$T_n(\omega) = \cos[n\cos^{-1}(\omega)] \tag{2.110}$$

is a polynomial as desired. Of course, the maximum error is ± 1, an impractically large value. This is easily rectified by requiring that the approximating polynomial be

$$p_n(\omega) = \varepsilon T_n(\omega) = \varepsilon\cos[n\cos^{-1}(\omega)] \tag{2.111}$$

In terms of gain and characteristic functions, therefore, one has

$$G(\omega) = \frac{1}{1+K^2(\omega)} = \frac{1}{1+\varepsilon^2 T_n^2(\omega)} \tag{2.112}$$

That is, $p_n(\omega) = \varepsilon T_n(\omega)$ is the best approximation to the characteristic function $K(\omega)$.

Since the gain function to be approximated is even in ω, $K(\omega)$—and, therefore, $p_n(\omega) = \varepsilon T_n(\omega)$—must be either even or odd. As one can recall, the Chebyshev polynomials have this property. For this reason, it is only necessary to discuss the situation for $\omega \geq 0$ for the extension to negative ω is then obvious.

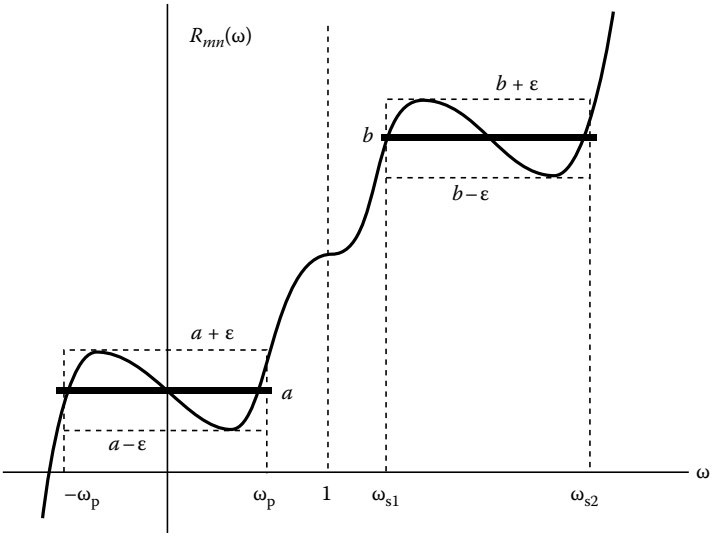


FIGURE 2.20 Chebyshev rational approximation.

The foregoing sets the stage for the more general problem, which will now be addressed. As noted previously, the Weierstrass theorem holds for more general compact sets. Thus, as a preliminary exercise, the approximation problem sketched in Figure 2.20 will be discussed. That figure shows two closed intervals as the compact set on which the approximation is to occur. The function to be approximated is assumed to be the constant a on the first interval and the constant b on the second.* In order to cast this into the filtering context, it will be assumed that the constant a is small and the constant b large. The sharpness of the transition characteristic can then be set to any desired degree by allowing ω_p to approach unity from below and ω_s^1 to approach unity from above. Notice that the frequency scaling will be different here than for the all-pole case in which the right-hand interval endpoint was allowed to be unity.

In addition to more general compact sets, the Weierstrass theorem allows the approximating function to be a general real rational function, which has been denoted by $R_{mn}(\omega)$ in the figure. This notation means that the numerator and denominator polynomials have maximum degree m and n, respectively. Now suppose that one specifies that $b = 1/a$ and suppose $R_{mn}(\omega)$ is a reciprocal function, that is, one having the property that

$$R_{mn}\left(\frac{1}{\omega}\right) = \frac{1}{R_{mn}(\omega)} \tag{2.113}$$

Then, if one determines its coefficients such that the equiripple property holds on the interval $-\omega_p \leq \omega \leq \omega_p$, it will also hold on the interval $\omega_{s1} \leq \omega \leq \omega_{s2}$. Of course, this interval is constrained such that $\omega_{s1} = 1/\omega_p$. The other interval endpoint will then extend to $+\infty$ (and, by symmetry, there will also be an interval on the negative frequency axis with one endpoint at $-1/\omega_p$ and the other at $-\infty$). This is, of course, not a compact set but the approximation theorem continues to hold anyway because it is equivalent to approximating $a = 1/b$ on the low-pass interval $[-\omega_p, \omega_p)$. Thus, one can let $a = 0$ and $b = \infty$ and simultaneously approximate the ideal characteristic function

* One should note that this function is continuous on the compact set composed of the two closed intervals.

$$K(\omega) = \begin{cases} 0; & |\omega| \le \omega_p \\ \infty; & |\omega| \ge \frac{1}{\omega_p} \end{cases} \tag{2.114}$$

with the real and even or odd (for realizability) rational function $R_{mn}(\omega)$ having the aforementioned reciprocal property. The Weierstrass theorem equiripple property for such rational functions demands that the total number of error extrema* on the compact set be $m+n+2$. (This assumes the degree of both polynomials to be relative to the variable ω^2.)

Based on the preceding discussion, one has the following form for $R_{mn}(\omega)$:

$$R_{2n,2n}(\omega) \equiv R_{2n}(\omega) = \frac{(\omega_1^2 - \omega^2)(\omega_3^2 - \omega^2)\cdots(\omega_{2n-1}^2 - \omega^2)}{(1 - \omega_1^2\omega^2)(1 - \omega_2^2\omega^2)\cdots(1 - \omega_{2n-1}^2\omega^2)} \tag{2.115}$$

$$R_{2n+1,2n}(\omega) \equiv R_{2n+1}(\omega) = \omega\frac{(\omega_2^2 - \omega^2)(\omega_4^2 - \omega^2)\cdots(\omega_{2n}^2 - \omega^2)}{(1 - \omega_2^2\omega^2)(1 - \omega_4^2\omega^2)\cdots(1 - \omega_{2n}^2\omega^2)} \tag{2.116}$$

The first is clearly an even rational function and latter odd. The problem now is to find the location of the pole and zero factors such that equiripple behavior is achieved in the passband. The even case is illustrated in Figure 2.21 for $n=2$. Notice that the upper limit of the passband frequency interval has been taken for convenience to be equal to $\sqrt{k}$; hence, the lower limit of the stopband frequency interval is $1/\sqrt{k}$. Thus,

$$k = \frac{\omega_p}{\omega_s} \tag{2.117}$$

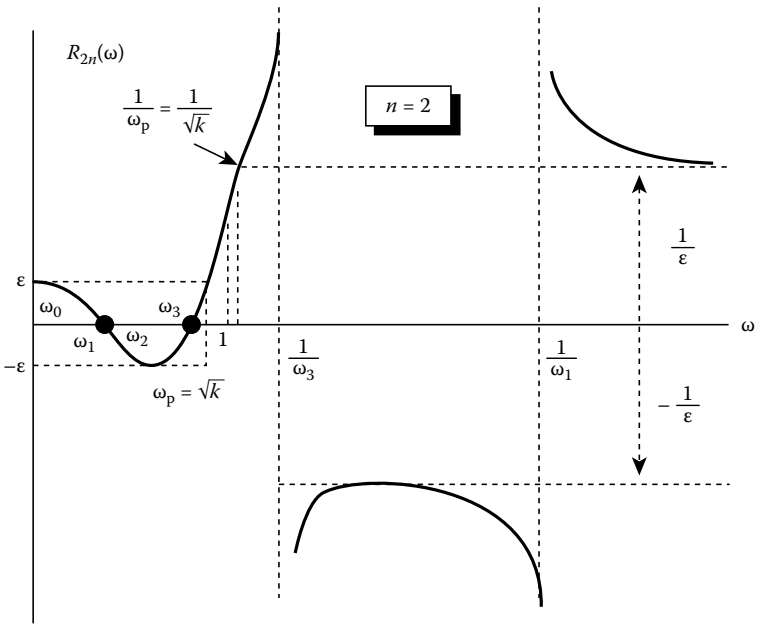


FIGURE 2.21 Typical plot of an even-order Chebyshev rational function.

* There can be degeneracy in the general approximation problem, but the constraints of the problem being discussed here preclude this from occurring.

is a measure of the sharpness of the transition band rolloff and is always less than unity. The closer it is to one, the sharper the rolloff. It is an arbitrarily specified parameter in the design. Notice that equiripple behavior in the passband implies equiripple behavior in the stopband—in the latter, the approximation is to the constant whose value is infinity and the minimum value (which corresponds to a maximum value of gain) is $1/\varepsilon$, while the maximum deviation from zero in the passband is ε. The zeros are all in the passband and are mirrored in poles, or infinite values, in the stopband. The notation ω_0 and ω_2 for the passband frequencies of maximum error has been introduced. In general, their indices will be even for even approximations and odd for odd approximations.

A sketch for the odd case would differ only in that the plot would go through the origin, that is, the origin would be a zero rather than a point of maximum error. Notice that the resulting filter will have finite gain unlike the Butterworth or Chebyshev, which continue to rolloff toward zero as ω becomes infinitely large. As noted in the figure, the approximating functions are called the *Chebyshev rational functions*.

The procedure [9] now is quite analogous to the Chebyshev polynomial approximation, though rather more complicated. One looks for a continuous periodic waveform that possess the desired equiripple behavior. Since the Jacobian elliptic functions are generalizations of the more ordinary sinusoids, it is only natural that they be investigated with an eye toward solving the problem under attack. With that in mind, some of the more salient properties will now be reviewed.

Consider the function

$$I(\phi, k) = -\frac{1}{\sqrt{1 - k^2 \sin^2(\phi)}} \tag{2.118}$$

which is plotted for two values of k in Figure 2.22. If $k = \pm 1$, the peak value is infinite; in this case, $I(\phi, k) = \sec(\phi)$. For smaller values of k the peak value depends upon k with smaller values of k resulting lower peak values. The peaks occur at odd multiples of $\pi/2$. Note that $I(\phi, k)$ has the constant value one for $k = 0$. Figure 2.23 shows the running integral of $I(\phi, k)$ for $k^2 = 0.99$ and for $k^2 = 0$. For $k^2 = 0$ the curve is a straight line (shown dashed); or other values of k^2, it deviates from a straight line by an amount that depends upon the size of k^2. Observe that the running integral has been plotted over one full period of $I(k, \phi)$, that is, from 0 to π. The running integral is given in analytical form by

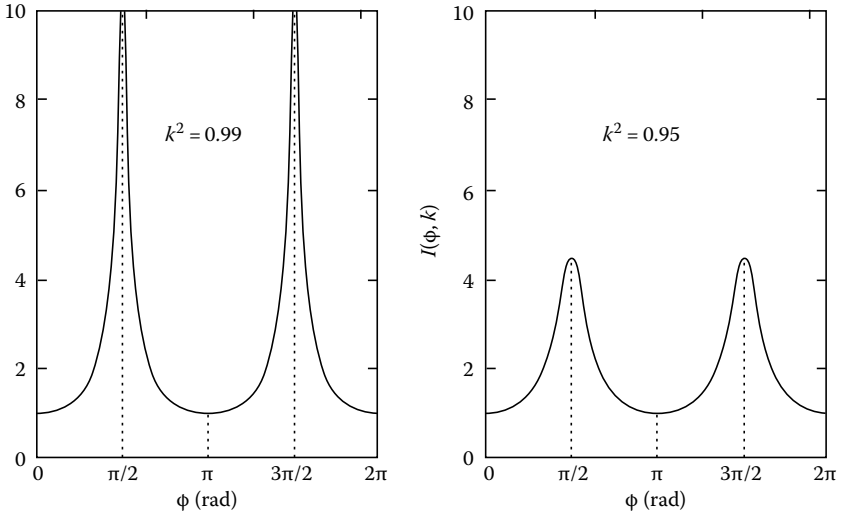


FIGURE 2.22 Plot of $I(\phi, k)$.

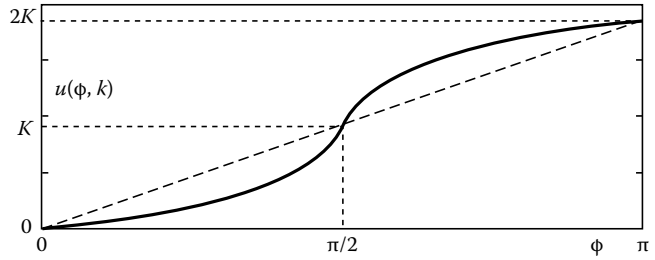


FIGURE 2.23 Running integral of I normalized to K.

$$u(\phi, k) = \int_0^{\phi} I(\alpha, k)\mathrm{d}\alpha = \int_0^{\phi} \frac{1}{\sqrt{1 - k^2 \sin^2(\alpha)}}\mathrm{d}\alpha \tag{2.119}$$

The quantity K, shown by the lowest dashed horizontal line in Figure 2.23, is the integral of $I(\phi, k)$ from 0 to $\pi/2$; that is, it is the area beneath the $I(\phi, k)$ curve in Figure 2.22 from 0 to $\pi/2$. Thus, it is the area beneath the curve to the left of the first vertical dashed line in that figure. It is given by

$$u\left(\frac{\pi}{2}, k\right) = \int_0^{\pi/2} I(\alpha, k)\mathrm{d}\alpha = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2(\alpha)}}\mathrm{d}\alpha = K \tag{2.120}$$

and is referred to as the *complete elliptic integral of the first kind.*

The sine generalization sought can now be defined. Since the running integral $u(\phi, k)$ is monotonic, it can be inverted and thereby solved for ϕ in terms of u; for each value of u there corresponds a unique value of ϕ. The elliptic sine function is defined to be the ordinary sine function of ϕ:

$$\mathrm{sn}(u, k) = \sin(\phi, k) \tag{2.121}$$

Now, inspection of Figure 2.23 shows that, as ϕ progresses from 0 to 2π, $u(\phi, k)$ increases from 0 to $4K$; since angles are unique only to within multiples of 2π, therefore, it is clear that $\mathrm{sn}(u, k)$ is periodic with period $4K$. Hence, K is a quarter-period of the elliptic sine function.

The integral in Equation 2.119 cannot be evaluated in closed form; thus, there is not a simple, compact expression for the required inverse. Therefore, the elliptic sine function can only be tabulated in numerical form or computer using numerical techniques. It is shown for $k^2 = 0.99$ in Figure 2.24 and compared with a conventional sine function having the same period ($4K$).

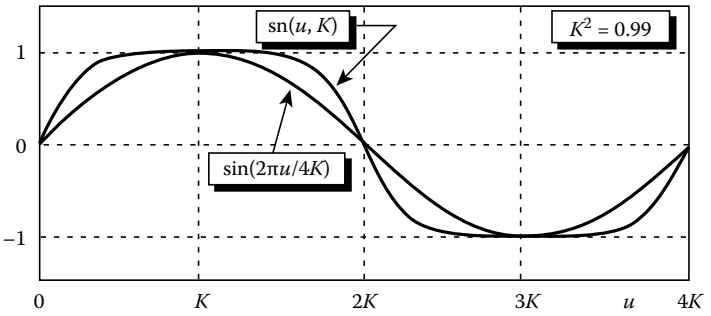


FIGURE 2.24 Plots of the elliptic and ordinary sine functions.

Recall, now, the objective of exploring the preceding generalization of the sinusoid: one is looking for a transformation that will convert the characteristic function given by Equations 2.115 and 2.116 into equivalent waveforms having the equiripple property. As one might suspect (since so much time has been spent on developing it), the elliptic sine function is precisely the transformation desired. The crucial aspect of showing this is the application of a fundamental property of sn(u, k), known as an additional formula:

$$\mathrm{sn}(u+a,k)\mathrm{sn}(u-a,k)=\frac{\mathrm{sn}^2(u,k)-\mathrm{sn}^2(a,k)}{1-k^2\mathrm{sn}^2(u,k)\mathrm{sn}^2(a,k)} \tag{2.122}$$

The right-hand side of this identity has the same form as one factor in Equations 2.115 and 2.116, that is of one zero factor coupled with its corresponding pole factor. This suggests the transformation

$$\omega=\sqrt{k}\mathrm{sn}(u,k) \tag{2.123}$$

The passband zeros then are given by

$$\omega_i=\sqrt{k}\mathrm{sn}(u_i,k) \tag{2.124}$$

and the factors mentioned previously map into

$$\frac{\omega_i^2-\omega^2}{1-\omega_i^2\omega^2}=\frac{k[\mathrm{sn}^2(u_i,k)-\mathrm{sn}^2(u,k)]}{1-k^2\mathrm{sn}^2(u_i,k)\mathrm{sn}^2(u,k)}=k\mathrm{sn}(u+u_i)\mathrm{sn}(u-u_i) \tag{2.125}$$

For specificity, the even-order case will be discussed henceforth. The odd-order case is the same if minor notational modifications are made. Thus, one sees that

$$R_{2n}(\omega)=\prod_{i=1,3,\ldots,2n-1}\frac{\omega_i^2-\omega^2}{1-\omega_i^2\omega^2}=\prod_{i=1,3,\ldots,2n-1}k\mathrm{sn}(u+u_i)\mathrm{sn}(u-u_i) \tag{2.126}$$

The u_i are to be chosen. Before doing this, it helps to simplify the preceding expression by defining $u_{-i}=-u_i$ and reindexing. Calling the resulting function $G(u)$, one has

$$G(u)=\prod_{i=-1,-3,\ldots,-(2n-1)}^{i=1,3,\ldots,2n-1}k\mathrm{sn}(u+u_i) \tag{2.127}$$

Refer now to Figure 2.24. Each of the sn functions is periodic with period $4K$ and is completely defined by its values over one quarter-period $[0, K]$. Suppose that one defines

$$u_i=i\frac{K}{2n} \tag{2.128}$$

Figure 2.25 shows the resulting transformation corresponding to Equations 2.123 and 2.124. As u progresses from $-K$ to $+K$, ω increases from $-\sqrt{k}$ to $+\sqrt{k}$ as desired. Furthermore, because of the symmetry of sn(u), the set $\{u_i\}$ forms an additive group—adding $K/2n$ to the index of any u_i results in another u_i in the set. This means that $G(u)$ in Equation 2.127 is periodic with period $K/2n$. Thus, as ω increases from $-\sqrt{k}$ to $+\sqrt{k}$, $R_{2n}(\omega)$ achieves $2n+2$ extrema, that is, positive and negative peak values. But this is sufficient for $R_{2n}(\omega)$ to be the Chebyshev rational function of best approximation.

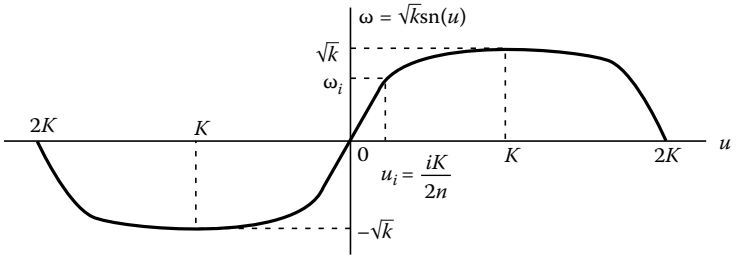


FIGURE 2.25 Elliptic sine transformation.

The symmetry of the zero distribution around the peak value of $\mathrm{sn}(u)$, that is, around $u = K$, reveals that the peak values of $R_{2n}(\omega)$ occur between the zeros; that is, recalling that in Figure 2.21 the peak values have been defined by the symbols ω_i for even i, one has

$$\omega_i = \sqrt{k}\,\mathrm{sn}\left(\frac{iK}{2n}\right); \quad i = 0, 1, 2, \ldots, 2n \tag{2.129}$$

where the odd indices correspond to the zeros and the even ones to the peak values. Note that $i = 0$ corresponds to $\omega = 0$ and $i = 2n$ to $\omega = \sqrt{k}$. This permits one to compute the minimax error:

$$\varepsilon = |R_{2n}(0)| = (\omega_1 \omega_3 \cdots \omega_{2n-1})^2 \tag{2.130}$$

As pointed out previously, the analysis for odd-order filters proceeds quite analogously. The only difference lies in the computation of ε. In this case, $R_{2n+1}(0) = 0$; thus, $\omega = 0$ is a zero—not a point of maximum deviation. One can, however, note that the quantity $\varepsilon^2 - R_{2n+1}^2(\omega)$ has double zeros at the frequencies of maximum deviation (except at $\omega = \omega_{2n+1} = \sqrt{k}$) and the same denominator as $R_{2n+1}^2(\omega)$. Hence,

$$\varepsilon^2 - R_{2n+1}^2(\omega) = \frac{\left(\omega_1^2 - \omega^2\right)^2 \left(\omega_3^2 - \omega^2\right)^2 \cdots \left(\omega_{2n+1}^2 - \omega^2\right)}{\left(1 - \omega_2^2\omega^2\right)^2 \left(1 - \omega_4^2\omega^2\right)^2 \cdots \left(1 - \omega_{2n}^2\omega^2\right)^2} \tag{2.131}$$

Notice here that $\omega_{2n+1} = \sqrt{k}$. Since $R_{2n+1}(0) = 0$, one can evaluate this expression at $\omega = 0$ to get

$$\varepsilon = (\omega_1 \omega_3 \cdots \omega_{2n-1})^2 \sqrt{k} \tag{2.132}$$

Note that one uses the zero frequencies in the even-order case and the frequencies of maximum deviation in the odd-order case.

As an example, suppose that the object is to design an elliptic filter of order $n = 2$. Further, suppose that the passband ripple cutoff frequency is to be 0.95. Then, one has

$$\omega_\mathrm{p} = \sqrt{k} = 0.95 \tag{2.133}$$

The quarter period of the elliptic sine function is

$$K = 2.9083 \tag{2.134}$$

TABLE 2.4 Zeros of Elliptic Rational Functions

i	ω_i
0	0.0000
1	0.5923
2	0.8588
3	0.9352
4	0.9500

Evaluating the zeros and points of maximum deviation of the Chebyshev rational function numerically using Equation 2.129, one obtains the values shown in Table 2.4. Thus, the required elliptic rational function is

$$R_4(\omega) = \frac{(0.3508 - \omega^2)(0.8746 - \omega^2)}{(1 - 0.3508\omega^2)(1 - 0.8746\omega^2)} \tag{2.135}$$

Finally, the maximum error is given by Equation 2.130:

$$\varepsilon = (\omega_1\omega_2)^2 = (0.5923 \times 0.9352)^2 = 0.3078 \tag{2.136}$$

Figure 2.26 shows the resulting gain plot. Observe the transmission zero at $\omega = 1/0.9352 = 1.069$, corresponding to the pole of the Chebyshev rational function located at the inverse of the largest passband zero. Also, as anticipated, the minimum gain in the passband is

$$A_p = \frac{1}{\sqrt{1+\varepsilon^2}} = \frac{1}{\sqrt{1+(0.3078)^2}} = 0.9558 \tag{2.137}$$

and the maximum stopband gain is

$$A_s = \frac{1}{\sqrt{1+\frac{1}{\varepsilon^2}}} = \frac{1}{\sqrt{1+\frac{1}{(0.3078)^2}}} = 0.2942 \tag{2.138}$$

Perhaps a summary of the various filter types is in order at this point. The elliptic filter has more flexibility than the Butterworth or the Chebyshev because one can adjust its transition band rolloff independently of the passband ripple. However, as the sharpness of this rolloff increases, as a more detailed analysis shows, the stopband gain increases and the passband ripple increases. Thus, if these

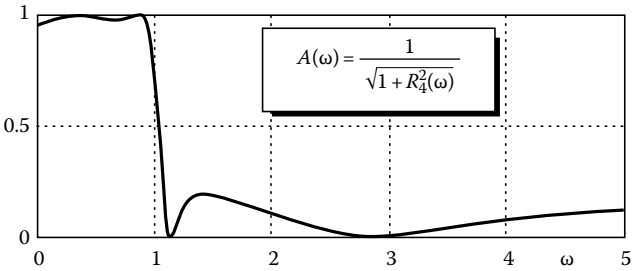


FIGURE 2.26 Gain plot for the example elliptic filter.

parameters are to be independently specified—as is often the desired approach—one must allow the order of the filter to float freely. In this case, the design becomes a bit more involved. The reader is referred to Ref. [10], which presents a simple curve for use in determining the required order. It proceeds from the result that the filter order is the integer greater than or equal to the following ratio:

$$n \geq \frac{f(M)}{f(\Omega)} \tag{2.139}$$

where

$$f(x) = \frac{K\left(\sqrt{1 - x^{-2}}\right)}{K(1/x)} \tag{2.140}$$

Lin [10] presents a curve of this function $f(x)$. K is the complete elliptic integral given in Equation 2.120. The parameters M and Ω are defined by

$$M = \sqrt{\frac{1 - 10^{-0.1A_s}}{1 - 10^{-0.1A_p}}} \tag{2.141}$$

and

$$\Omega = 1/k \tag{2.142}$$

Of course, there is still the problem of factorization. That is, now that the appropriate Chebyshev rational function is known, one must find the corresponding $G(s)$ transfer function of the filter. The overall design process is explained in some detail in Ref. [11], which develops a numerically efficient algorithm for directly computing the parameters of the transfer function.

The Butterworth and Chebyshev filters are of the all-pole variety, and this means that the synthesis of such filters is simpler than is the realization of elliptic filters, which requires the realization of transmission zeros on the finite $j\omega$ axis.

Finally, a word about phase (or group delay). Table 2.5 shows a rank ordering of the filters in terms of both gain and phase performance. As one can see, the phase behavior is inverse to the gain performance. Thus, the elliptic filter offers the very best standard approximation to be ideal lpp "brickwall" gain behavior, but its group delay deviates considerably from a constant. On the other end of the spectrum, one notes that the Bessel–Thompson filters offers excellent phase performance, but a quite high order is required to achieve a reasonable gain characteristic.

If both excellent phase and gain performance are absolutely necessary, two approaches are possible. One either uses computer optimization techniques to simultaneously approximate gain and phase, or one uses one of the filters described in this section followed by an *allpass* filter, one having unity gain over the frequency range of interest, but whose phase can be designed to have the inverse characteristic to the filter providing the desired gain. This process is known as *phase compensation*.

TABLE 2.5 A Rank Ordering of the Filters in Terms of Both Gain and Phase Performance

Filter Type	Gain Rolloff	Phase Linear
Bessel	Worst	Best
Butterworth	Poor	Better
Chebyshev	Better	Poor
Elliptic	Best	Worst

References

1. A. M. Davis, The approximation theoretic foundation of analog and digital filters, *IEEE International Symposium on Circuits and Systems,* San José, California, May, 1986.
2. T. J. Rivlin, *An Introduction to the Approximation of Functions*, New York: Dover, 1981.
3. G. R. Lang and P. O. Brackett, Complex analog filters, in *Proc. Euro. Conf. Circuit Theory, Design*, The Hague, The Netherlands, Aug. 1981, pp. 412–415.
4. W. K. Chen, *Passive and Active Filters, Theory and Implementations,* New York: Wiley, 1986.
5. L. P. Huelsman, *Theory and Design of Active RC Circuits,* New York: McGraw-Hill, 1968.
6. G. Szentirmai, *Computer-Aided Filter Design*, New York: IEEE, 1973.
7. A. M. Davis, Realizability-preserving transformations for digital and analog filters, *J. Franklin Institute*, 311(2), 111–121, Feb. 1981.
8. G. C. Temes and J. W. LaPatra, *Circuit Synthesis and Design*, New York: McGraw-Hill, 1977.
9. E. A. Guillemin, *Synthesis of Passive Networks*, New York: Wiley, 1957.
10. P. M. Lin, Single curve for determining the order of an elliptic filter, *IEEE Trans. Circuits Syst*., 37(9), 1181–1183, Sept. 1990.
11. A. Antoniou, *Digital Filters: Analysis and Design*, New York: McGraw-Hill, 1979.

3

Frequency Transformations

3.1 Low-Pass Prototype **3**-1
3.2 Frequency and Impedance Scaling **3**-2
Frequency Scaling • Impedance Scaling
3.3 Low-Pass to High-Pass Transformation **3**-5
3.4 Low-Pass to Bandpass Transformation **3**-7
3.5 Low-Pass to Band-Reject Transformation **3**-10

Jaime Ramirez-Angulo
New Mexico State University

3.1 Low-Pass Prototype

As discussed in Chapter 2, conventional approximation techniques (Butterworth, Chebyshev, Elliptic, Bessel, etc.) lead to a normalized transfer function denoted low-pass prototype (LPP). The LPP is characterized by a passband frequency $\Omega_P = 1.0$ rad/s, a maximum passband ripple A_P (or A_{max}), a minimum stopband attenuation A_s (or A_{min}), and a stopband frequency Ω_s. A_p and A_s are usually specified in decibels. Tolerance bounds (also called box constraints) for the magnitude response of an LPP are illustrated in Figure 3.1a. The ratio Ω_s/Ω_p is called the selectivity factor and it has a value Ω_s for an LPP filter. The passband and stopband edge frequencies are defined as the maximum frequency with the maximum passband attenuation A_p and the minimum frequency with the minimum stopband attenuation A_s, respectively. The passband ripple and the minimum passband attenuation are expressed by

$$A_P = 20\log\left|\frac{K}{H(\omega_P)}\right|, \quad A_s = 20\log\left|\frac{K}{H(\omega_s)}\right| \tag{3.1}$$

where K is the maximum value of the magnitude response in the passband (usually unity). Figure 3.1b shows the magnitude response of a Chebyshev LPP transfer function with specifications $A_P = 2$ dB, $A_s = 45$ dB, and $\Omega_s = 1.6$.

Transformation of transfer function. Low-pass, high-pass, bandpass, and band-reject transfer functions (denoted in what follows LP, HP, BP, and BR, respectively) can be derived from an LPP transfer function through a transformation of the complex frequency variable. For convenience, the transfer function of the LPP is expressed in terms of the complex frequency variable s, where $s = u + j\Omega$ while the transfer functions obtained through the frequency transformation (low-pass, high-pass, bandpass, or band-reject) are expressed in terms of the transformed complex frequency variable $p = \sigma + j\omega$.

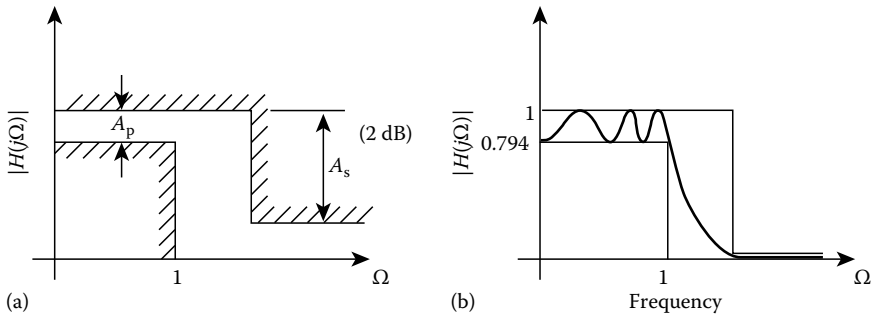


FIGURE 3.1 (a) Tolerance bounds for magnitude response of low-pass prototype and (b) Chebyshev LPP response.

The approximation of an LP, HP, BP, or BR transfer function with passband ripple (*s*) A_p and stopband attenuation (*s*) A_s involves three steps:

1. Determination of the stopband edge frequency or selectivity factor Ω_s of an LPP, which can be transformed into the desired LP, HP, BP, or BR filter.
2. Approximation of an LPP transfer function $T_{LPP}(s)$ with selectivity factor Ω_s and with same passband ripple and stopband attenuation A_p and A_s are the desired LP, HP, BP, or BR filter, respectively
3. Transformation of the LPP transfer function $T_{LPP}(s)$ into the desired transfer function (LP, HP, BP, or BR) $T(p)$ through a frequency transformation of the form

$$s = f(p) \tag{3.2}$$

Transformation of a network with LPP magnitude response into a low-pass, high-pass, bandpass, or band-rejection network can be done directly on the elements of the network. This procedure is denoted network transformation. It is very convenient in practice because element values for double terminated lossless ladder networks with LPP specifications have been extensively tabulated for some common values of A_p, A_s, and Ω_s. Also, a host of personal computer programs have become available in recent years that allow one to determine the component values of LPP ladder networks for arbitrary values A_p, A_s, and Ω_s. In what follows we study the frequency transformation $s = f(p)$ for each specific type of filter response (LP, HP, BP, and BR). We show how to calculate the selectivity factor of the equivalent LPP based on box constraint specifications for each type of filter. We then show how mapping of the imaginary frequency axis from *s* to *p* leads to LP, HP, BP, or BR magnitude responses. We analyze how poles and zeros are mapped from the *s*-plane to the *p*-plane for each transformation and finally we show the element transformations required to directly transform LPP networks into any of the filter types addressed previously.

3.2 Frequency and Impedance Scaling

3.2.1 Frequency Scaling

The simplest frequency transformation is a scaling operation expressed by

$$s = \frac{p}{\omega_o} \tag{3.3}$$

where ω_o is a frequency scaling parameter. This transformation is denoted frequency scaling and it allows one to obtain a low-pass transfer function with a nonunity passband frequency edge from an LPP transfer function.

Transformation of poles and zeros of transfer function. Consider an LPP factorized transfer function $T_{LPP}(s)$ with n poles $s_{p1}, s_{p2}, \ldots, s_{pn}$ and m zeros $s_{z1}, s_{z2}, \ldots, s_{zm}$

$$T_{LPP}(s) = K\frac{(1 - s/s_{z1})(1 - s/s_{z2})\cdots(1 - s/s_{zm})}{(1 - s/s_{p1})(1 - s/s_{p2})\cdots(1 - s/s_{pn})} \tag{3.4}$$

Using Equation 3.3, this transfer function becomes

$$T_{LP}(p) = K\frac{(1 - p/p_{z1})(1 - p/p_{z2})\cdots(1 - p/p_{zm})}{(1 - p/p_{p1})(1 - p/p_{p2})\cdots(1 - p/p_{pn})} \tag{3.5}$$

where poles and zeros (s_{zi} and s_{pj}) of the LPP transfer function $P_{LPP}(s)$ become simply poles and zeros in $T_{LP}(p)$, which are related to those of $T_{LPP}(s)$ by the scaling factor ω_o:

$$p_{pi} = \omega_o s_{pi}, \quad p_{zj} = \omega_o s_{zj} \tag{3.6}$$

To determine the magnitude (or frequency) response of the LP filter we evaluate the magnitude of the transfer function on the imaginary axis (for $s = j\Omega$). The magnitude response of the transformed transfer function $|T_{LP}(j\omega)|$ preserves a low-pass characteristic as illustrated in Figure 3.2. The frequency range from 0 to ∞ in the Ω-axis is mapped to the range 0 to ∞ in the ω-axis. A frequency and its mirror image in the negative axis $\pm\Omega$ is mapped to frequencies $\omega = \pm\omega_o\Omega$ with the same magnitude response: $|T_{LPP}(j\Omega)| = |T_{LP}(j\omega_o\Omega)|$. The passband and stopband edge frequencies $\Omega_P = 1$ rad/s and Ω_s of the LPP are mapped into passband and stopband edge frequencies $\omega_p = \omega_o$ and $\omega_s = \omega_o\Omega_s$, respectively. From this, it can be seen that for given low-pass filter specifications ω_s, ω_p the equivalent LPP is determined based on the relation $\Omega_s = \omega_s/\omega_p$, while the frequency scaling parameter ω_o corresponds to the passband edge frequency of the desired LP.

LP network transformation. Capacitors and inductors are the only elements that are frequency dependent and that can be affected by a change of frequency variable. Capacitors and inductors in an LPP network have impedances $z_c = 1/sc_n$ and $z_1 = sl_n$, respectively. Using Equation 3.3 these become $Z_c(p) = 1/pC$ and $Z_L(p) = pL$, where $C = c_n/\omega_o$ and $L = l_n/\omega_o$. The LPP to LP frequency transformation is performed directly on the network by simply dividing the values of all capacitors and inductors by the frequency scaling factor ω_o. This is illustrated in Figure 3.3a. The transformation expressed by Equation 3.3 can be applied to any type of filter and it has the effect of scaling the frequency axis without changing the shape of its magnitude response. This is illustrated in Figure 3.3, where the elements of an LPP with $A_p = 2$ dB, $A_s = 45$ dB, and selectivity $\Omega_s = 1.6$ rad/s are scaled (Figure 3.3b) to transform the network into an LP network with passband and stopband edge frequencies $\omega_p = 2\pi$ 10 krad/s and $\omega_s = 2\pi$ 16 krad/s (or $f_p = 10$ kHz and $f_s = 16$ kHz), respectively.

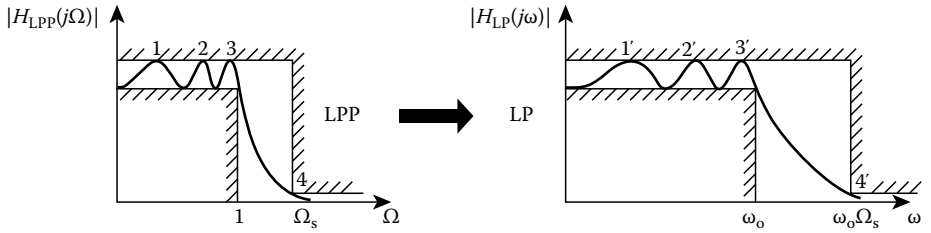


FIGURE 3.2 Derivation of low-pass response from a low-pass prototype by frequency scaling.

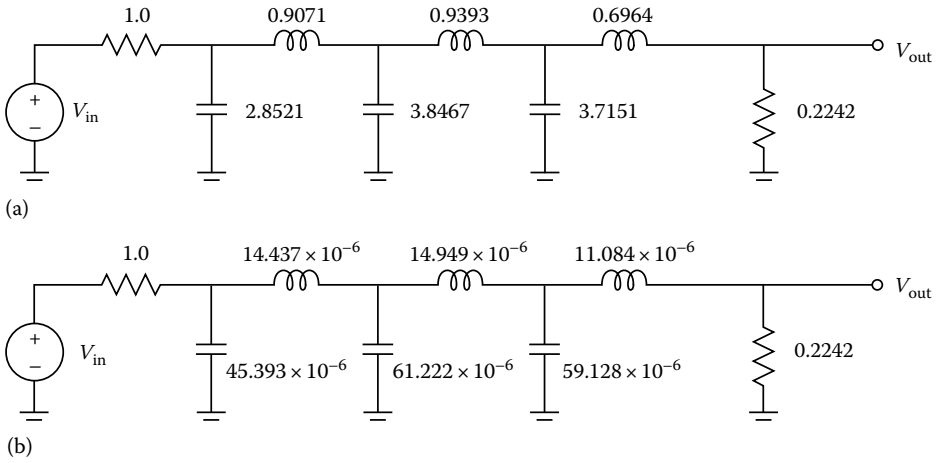


FIGURE 3.3 (a) Low-pass prototype ladder network and (b) LP network with passband frequency $f_p = 10$ kHZ derived from (a).

3.2.2 Impedance Scaling

Dimensionless transfer functions defined by ratios of voltages (V_{out}/V_{in}) or currents (I_{out}/I_{in}) remain unchanged if all impedances of a network are scaled by a common scaling factor "*a*." On the other hand, transfer functions of the transresistance type (V_{out}/I_{in}) or of the transconductance type (I_{out}/V_{in}) are simply modified by the impedance scaling factor a and $1/a$, respectively. If we denote a normalized impedance by z_n, then the impedance scaling operation leads to an impedance $Z = az_n$. When applied to resistors (r_n), capacitors (c_n), inductors (l_n), transconductance gain coefficients (g_n), and transresistance gain coefficients (r_n) result in the following relations for the elements (R, C, L, g, and r) of the impedance scaled network.

$$\begin{aligned} R &= ar_n \\ L &= al_n \\ C &= \frac{1}{a}c_n \\ g &= \frac{1}{a}g_n \\ r &= ar_n \end{aligned} \tag{3.7}$$

Dimensionless voltage-gain and current-gain coefficients are not affected by impedance scaling. Technologies for fabrication of microelectronic circuits (CMOS, bipolar, BiCMOS monolithic integrated circuits, thin-film and thick-film hybrid circuits) only allow elements values and time constants (or pole and zero frequencies) within certain practical ranges. Frequency and impedance scaling are very useful to scale normalized responses and network elements resulting from standard approximation procedures to values within the range achievable by the implementation technology. This is illustrated in the following example.

Example 3.1

The amplifier of Figure 3.4. is characterized by a one-pole low-pass voltage transfer function given by $H(s) = V_{out}/V_{in} = K(1+s/\omega_p)$, where $K = g_m r_L r_1/(r_1 + r_2)$, and $\omega_p = 1/r_L C_L$. Perform frequency and impedance scaling so that the circuit pole takes a value $\omega_p = 2\pi \times 10$ Mrad/s (or $f_p = 10$ MHz) and resistance,

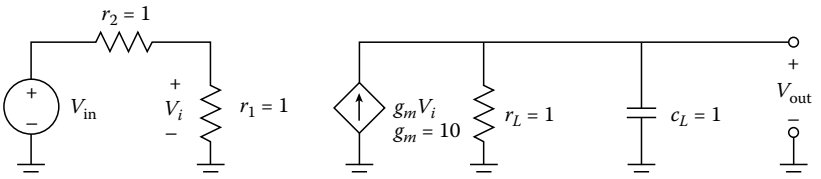


FIGURE 3.4 Normalized transconductance amplifier.

capacitance, and transconductance gain values are in the range of kΩ, pF, and μA/V, which are appropriate for the implementation of the circuit as an integrated circuit in CMOS technology.

Solution

The required location of the pole and range of values for the circuit elements can be achieved using frequency and impedance scaling factors $\omega_o = 2\pi \times 10^7$ and $a = 10^4$, respectively. These result in $R_1 = ar_1 = 10$ kΩ, $R_2 = ar_2 = 10$ kΩ, $g = g_m/a = 1000$ μA/V, $R_L = ar_L = 10$ kΩ, and $C_L = c_L/a\omega_o = 1.59$ pF.

3.3 Low-Pass to High-Pass Transformation

The LPP to high-pass transformation is defined by

$$s = \frac{\omega_o^2}{p} \tag{3.8}$$

Using this substitution in the LPP transfer function (Equation 3.4), it becomes

$$T_{HP}(P) = K\frac{p^{n-m}(p - p_{z1})(p - p_{z2})\cdots(p - p_{zm})}{(p - p_{p1})(p - p_{p2})\cdots(p - p_{pn})} \tag{3.9}$$

where the poles and zeros of Equation 3.8 are given by

$$\begin{aligned} p_{zi} &= \frac{\omega_o^2}{s_{zi}} \quad \text{for } i \in \{1, 2, \ldots, m\} \\ p_{pj} &= \frac{\omega_o^2}{s_{pj}} \quad \text{for } j \in \{1, 2, \ldots, n\} \end{aligned} \tag{3.10}$$

It can be seen that zeros and poles of $T_{HP}(p)$ are reciprocal to those of $T_{LPP}(s)$ and scaled by the factor ω_o^2. $T_{HP}(p)$ has $n - m$ zeros at $s = 0$, which can be considered to originate from $n - m$ zeros at ∞ in $T_{LPP}(s)$.

Let us consider now the transformation of the imaginary axis in s to the imaginary axis in p. For $s = j\Omega$, p takes the form $p = j\omega$ where

$$\omega = \frac{-\omega_o^2}{\Omega} \tag{3.11}$$

From Equation 3.11, it can be seen that positive frequencies in the LPP transform to reciprocal and scaled frequencies of the HP filter. Specifically, the frequency range from 0 to ∞ in Ω maps to the frequency range $-\infty$ to 0 in ω, while the range $-\infty$ to 0 in Ω maps to 0 to ∞ in ω. The passband edge frequency

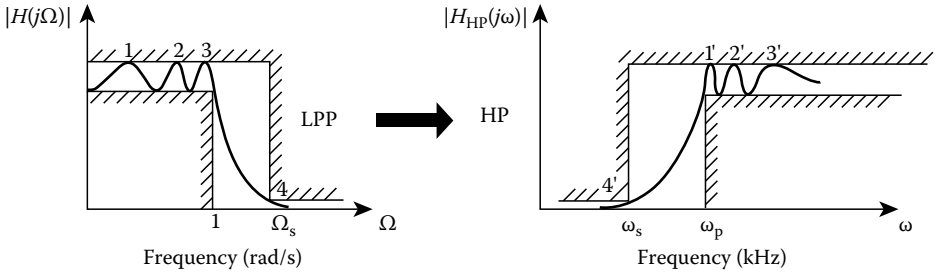


FIGURE 3.5 Transformation of a low-pass into a high-pass response.

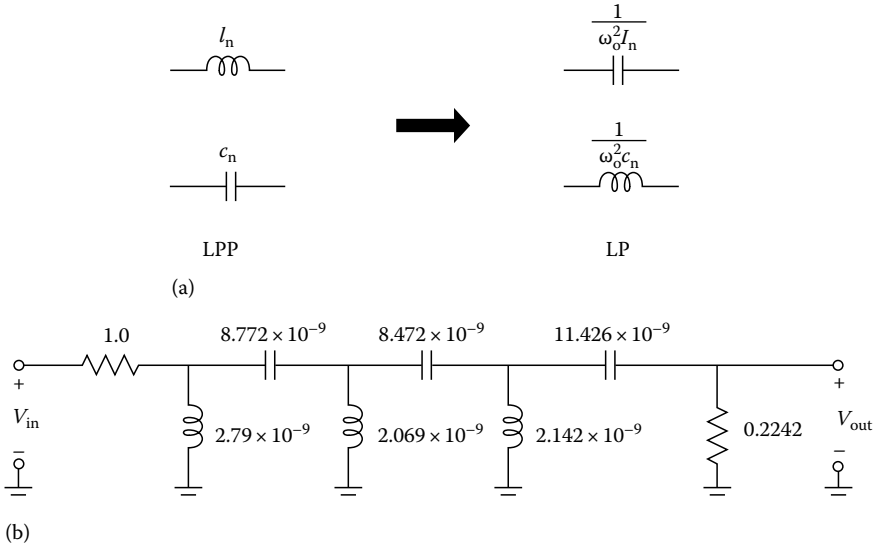


FIGURE 3.6 (a) LPP to high-pass network element transformations and (b) high-pass network derived from LPP of Figure 3.3a.

($\Omega_p = \pm 1$) and the stopband edge frequency $\pm\,\Omega_s$ of the LPP is mapped to $\omega_p = \pm\omega_o^2$ and $\omega_s = \pm\omega_o^2/\Omega_s$ in the high-pass response. This is illustrated in Figure 3.5.

The procedure to obtain the specifications of the equivalent LPP given specifications ω_p and ω_s for a HP circuit can be outlined as follows:

1. Calculate the selectivity factor of the LPP according to $\Omega_s = \omega_p/\omega_s$.
2. Approximate an LPP transfer function $T_{LPP}(s)$ with the selectivity Ω_s and the passband ripple and stopband attenuation of the desired high-pass response.
3. Perform an LPP to HP transformation either by direct substitution $p = \omega_o^2/s$ in $T_{LPP}(s)$ or by transforming poles and zeros of $T_{LPP}(s)$ using Equation 3.10.

Network transformation. Consider a capacitor c_n and an inductor l_n in an LPP network. They have impedances $z_c(s) = 1/sc_n$, $z_1(s) = sl_n$, respectively. Using Equation 3.8, these become impedances $Z_L(p) = pL$ and $Z_c = 1/pC$ in the high-pass network, where $L = 1/\omega_o^2\ c_n$ and $C = 1/\omega_o^2\ l_n$. It can be seen that an LPP to HP transformation can be done directly on an LPP network by replacing capacitors by inductors and inductors by capacitors. For illustration, Figure 3.6 shows a high-pass network with a passband edge frequency $\omega_p = 2\pi$ 20 Mrad/s or ($fp = 20$ MHz) derived from the LPP network shown in Figure 3.6a.

3.4 Low-Pass to Bandpass Transformation

If the LPP transfer function is expressed now as a rational function

$$H_{\mathrm{LPP}}(s) = K\frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_m s_m}{a_0 + a_1 s + a_2 s^2 + \cdots + a_n s^n} \tag{3.12}$$

then through the substitution

$$s = \frac{1}{\mathrm{BW}}\frac{p^2 + \omega_{\mathrm{o}}^2}{p} \tag{3.13}$$

$H_{\mathrm{LPP}}(s)$ transformed into a bandpass transfer function $H_{\mathrm{BP}}(p)$ with the form

$$H_{\mathrm{BP}}(P) = K' P^{n-m}\frac{1 + B_1 p + B_2 p^2 + \cdots + B_2 p^{2m-2} + B_1 p^{2m-1} + p^{2m}}{1 + A_1 p + A_2 p^2 + \cdots + A_2 p^{2m-2} + A_1 p^{2m-1} + p^{2m}} \tag{3.14}$$

From Equation 3.14, it can be seen that the bandpass transfer function has twice as many poles and zeros as the LPP transfer function. In addition it has $n - m$ zeros at the origin. The coefficients of the numerator and denominator polynomials are symmetric and are a function of the coefficients of $H_{\mathrm{LPP}}(s)$.

In order to obtain poles and zeros of the bandpass transfer function from the poles and zeros of $T_{\mathrm{LPP}}(p)$, three points must be considered.

First, a real pole (or zero) $s_{\mathrm{p}} = -u_{\mathrm{p}}$ of $H_{\mathrm{LPP}}(s)$ maps into a complex conjugate pair with frequency ω_{o} and Q (or selectivity) factor in $H_{\mathrm{BP}}(p)$, where $Dq = \omega_{\mathrm{o}}/(U_{\mathrm{p}}BW)$.*

Second, a pair of complex conjugate pole (or zeros) of $H_{\mathrm{LPP}}(s)$ with frequency Ω_{o} and pole-quality factor q denoted by (Ω_{o}, q) is mapped into two pairs of complex conjugate poles (or zeros) $(\omega_{\mathrm{o1}}, Q)$ and $(\omega_{\mathrm{o2}}, Q)$, where the following relations apply:

$$\begin{aligned} \omega_{\mathrm{o1}} &= \omega_{\mathrm{o}} M \\ \omega_{\mathrm{o2}} &= \frac{\omega_{\mathrm{o}}}{M} \\ Q &= \frac{a}{c}\left(M + \frac{1}{M}\right) \end{aligned} \tag{3.15a}$$

and the definitions

$$\begin{aligned} a &= \frac{\omega_{\mathrm{o}}}{\mathrm{BW}} \\ b &= \frac{\Omega_{\mathrm{o}}}{2a} \\ c &= \frac{\Omega_{\mathrm{o}}}{q} \\ M &= \sqrt{b^2 + \sqrt{(1+b^2)^2 - \frac{c}{(2a)^2}} + \sqrt{2b}\sqrt{1 + b^2 - \frac{c}{2\Omega_{\mathrm{o}}^2} + \sqrt{(1+b^2)^2 - \frac{c}{(2a)^2}}}} \end{aligned} \tag{3.15b}$$

apply.

* A complex conjugate pole pair can be expressed as $s_{\mathrm{p}}, S_{\mathrm{p}}^* = u_{\mathrm{p}} \pm j\omega_{\mathrm{p}} = \omega_{\mathrm{c}}\, e^{\pm j\theta} = (\omega_{\mathrm{c}}, Q)$, where the pole quality factor Q is given by $Q = \frac{1}{2}\cos\theta$ and $\omega_{\mathrm{c}} = (G_{\mathrm{p}}^2 + \omega_{\mathrm{p}}^2)^{\frac{1}{2}}$, $\theta = tg^{-1}\frac{\omega_{\mathrm{p}}}{G_{\mathrm{p}}}$.

Narrow-band approximation. If the condition $BW/\omega_o \ll 1$ is satisfied, then following simple transformations known as the narrow band approximation* can be used to map directly poles (or zeros) from the *s*-plane to the *p*-plane

$$p_p \approx \frac{BW}{2} s_p + j\omega_o, \quad p_z \approx \frac{BW}{2} s_z + j\omega_o \tag{3.16}$$

These approximations are valid only if the transformed poles and zeros are in the vicinity of $j\omega_o$, that is, if $|s_p - \omega_o|/\omega_o \ll 1$.

Third, in order to obtain poles and zeros of the bandpass transfer function, mapping of complex zeros on the imaginary Ω axis (s_2, $s_x^* = \pm j\Omega_z$) takes place using the same mapping relations discussed next.

Mapping of imaginary frequency axis. Consider a frequency $s = j\Omega$ and its mirror image $s = -j\Omega$ in the LPP. Using Equation 3.13, these two frequencies are mapped into four frequencies: $\pm\omega_1$ and $\pm\omega_2$, where ω_1 and ω_2 are given by

$$\begin{aligned} \omega_2 &= \Omega\frac{BW}{2} + \sqrt{\omega_o^2 + \left(\Omega\frac{BW}{2}\right)^2} \\ \omega_1 &= -\Omega\frac{BW}{2} + \sqrt{\omega_o^2 + \left(\Omega\frac{BW}{2}\right)^2} \end{aligned} \tag{3.17}$$

From Equation 3.17, the following relations can be derived:

$$\begin{aligned} \omega_2 - \omega_1 &= BW\Omega \\ \omega_1\omega_2 &= \omega_o^2 \end{aligned} \tag{3.18}$$

It can be seen that with the LPP to bandpass transformation frequencies are mapped into bandwidths. A frequency Ω and its mirror image $-\Omega$ are mapped into two pairs of frequency points that have ω_0 as center of geometry. The interval from 0 to ∞ in the positive Ω axis maps into two intervals in the ω axis: the first from ω_o to $+\infty$ on the positive ω axis and the second from $-\omega_o$ to 0 to in the negative ω axis. The interval $-\infty$ to 0 on the negative Ω axis maps into two intervals: from $-\infty$ to $-\omega_o$ in the negative ω axis and from 0 to ω_o in the positive ω axis. The LPP passband and stopband edge frequencies $\Omega_p = \pm 1$ and $+\Omega_s$ are mapped into passband edge frequencies ω_{p1}, ω_{p2}, and into stopband edge frequencies ω_{s1}, ω_{s2} that satisfy

$$\begin{aligned} \omega_{p2} - \omega_{p1} &= BW \\ \omega_{s2} - \omega_{s1} &= BW\Omega_s \end{aligned} \tag{3.19}$$

and

$$\omega_{p1}\omega_{p2} = \omega_{s1}\omega_{s2} = \omega_o^2 \tag{3.20}$$

Figure 3.7 shows mapping of frequency points 1, 2, 3, and 4 in the Ω axis to points 1′, 2′, 3′, and 4′ and 1″, 2″, 3″, 4″ in the ω axis of the bandpass response.

If the bandpass filter specifications do not satisfy Equation 3.20 (which is usually the case), then either one of the stopband frequencies or one of the passband frequencies has to be redefined so that they become symmetric w.r.t. ω_o and an equivalent LPP filter can be specified. For given passband and

* L.P. Huelsman, An algorithm for the low-pass to bandpass transformation, *IEEE Trans. Education,* EH, 72, March 1968.

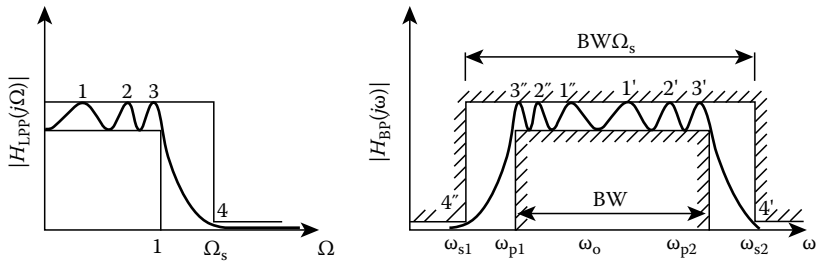


FIGURE 3.7 Low-pass to bandpass transformation.

stopband specifications ω_{p1}, ω_{p2}, ω_{s1}, ω_{s2}, A_p, A_s, the procedure to determine Ω_s for an equivalent LPP is as follows:

1. Calculate first the parameter ω_o in terms of the passband frequencies according to $\omega_o = \sqrt{\omega_{p1}\omega_{p2}}$.
2. Make the stopband frequencies geometrically symmetric with respect to ω_o determined in step 1 by redefining either ω_{s1} or ω_{s2} so that one of these frequencies becomes more constrained.* If $\omega_{s2} < \omega_o^2/\omega_{s2}$, then assign ω_{s1} the new value: $\omega_{s1} = <\omega_o^2/\omega_{s1}$. Otherwise assign ω_{s2} the new value $\omega_{s2} = \omega_o^2/\omega_{s1}$.
3. Calculate a selectivity factor of the LPP based on the redefined stopband frequency according to $\Omega_s = (\omega_{s2} - \omega_{s1})/(\omega_{p2} - \omega_{p1})$. This expression follows directly from Equations 3.19 and 3.20.
4. Calculate now the parameter ω_o in terms of the stopband frequencies, according to $\omega_o = \sqrt{\omega_{s1}\omega_{s2}}$.
5. Symmetrize the passband frequencies w.r.t. the value of ω_o determined in step 4 by constraining either ω_{p1} or ω_{p2}. If $\omega_{p1} > \omega_o^2/\omega_{p2}$, then assign ω_{p1} the new value: $\omega_{p1} > \omega_o^2/\omega_{p2}$. Otherwise assign ω_{p2} the new value $\omega_{p2} > \omega_o^2/\omega_{p1}$.
6. Calculate the selectivity factor based on the new set of passband frequencies using the same expression as in step 3.
7. Select from step 3 or step 6 the maximum selectivity factor Ω_s and determine the transformation parameters ω_o and BW from the values calculated in steps 1–3 or in 4–6, whichever sequence leads to the maximum Ω_s, which leads to the lowest order n for $T_{LPP}(s)$ and with this to the least expensive filter implementation.

Example 3.2

Consider the following nonsymmetric specifications for a bandpass filter: $\omega_{s1} = 2\pi$ 9, $\omega_{s2} = 2\pi$ 17, $\omega_{p1} = 2\pi$ 10, and $\omega_{p1} = 2\pi$ 14.4 (all frequencies specified in krad/s). Application of the above procedure leads to a new value for the upper stopband frequency $\omega_{s2} = 16$ and from this to the following parameters: $BW = 2\pi$ (14.4 −10) $= 2\pi$ 4.4, ω_o^2 $\omega_o^2 =$ 2π 9 2π 16 $= 2\pi$ 10 2π 14.4 $= (2\pi)^2$ 144, and $\Omega_s = 16 - 9/(14.4 - 10) = 1.59$.

Bandpass network transformation. Consider now the transformation of capacitors and inductors in an LPP filter. An inductor in the LPP network has an impedance $z_I(s) = sl_n$. This becomes an impedance $z_s(p) = pL_s + 1/pC_s$, where $L_s = l_n/BW$ and $C_s = BW/l_n\omega_o^2\omega_o^2$. Now consider a capacitor c_n in the LPP with admittance $Y_c(s) = sc_n$. Using the transformation (Equation 3.12), this becomes an admittance $Y_p(p) = pC_p + 1/pL_p$, where $C_p = c_n/BW$ and $L_p = BW/c_n\omega_o^2$. This indicates that to transform an LPP network into a bandpass network, inductors in the LPP network are replaced by the series connection of

* The term "constraint specifications" is used here in the sense of redefining either a stopband frequency or a passband frequency so that one of the transition bands becomes narrower, which corresponds to tighter design specifications.

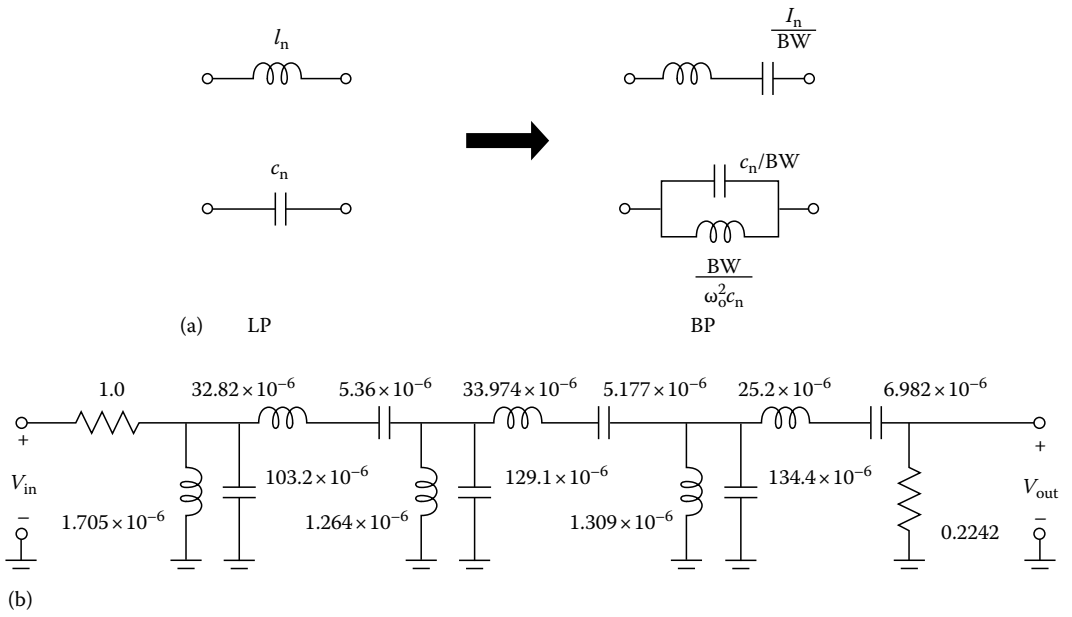


FIGURE 3.8 (a) LPP to bandpass network element transformations and (b) bandpass network derived from LPP network of Figure 3.3a.

an inductor with value L_s and a capacitor with value C_s and capacitors in the LPP are replaced by the parallel combination of a capacitor C_p and inductor L_p. This is illustrated in Figure 3.8. As with other transformations, resistors remain unchanged since they are not frequency dependent.

Example 3.3

Consider the LPP network shown in Figure 3.3a with specifications $Ap = 2$ dB, $As = 45$ dB, and $\Omega_s = 1.6$. Derive a bandpass network using the parameters calculated in Example 3.2: $\omega_o^2 = (2\pi)^2$ 144, BW = 2π 4.4 krad/s.

Solution

Straightforward application of the relations shown in Figure 3.8a leads to the network of Figure 3.8b.

3.5 Low-Pass to Band-Reject Transformation

This transformation is characterized by

$$S = \mathrm{BW}\frac{p}{p^2 + \omega_o^2} \tag{3.21}$$

and it can be best visualized as a sequence of two transformations through the intermediate complex frequency variables $s' = u' + j\Omega' \times 1$. A normalized LPP to high-pass transformations

$$s = \frac{1}{s'} \tag{3.22}$$

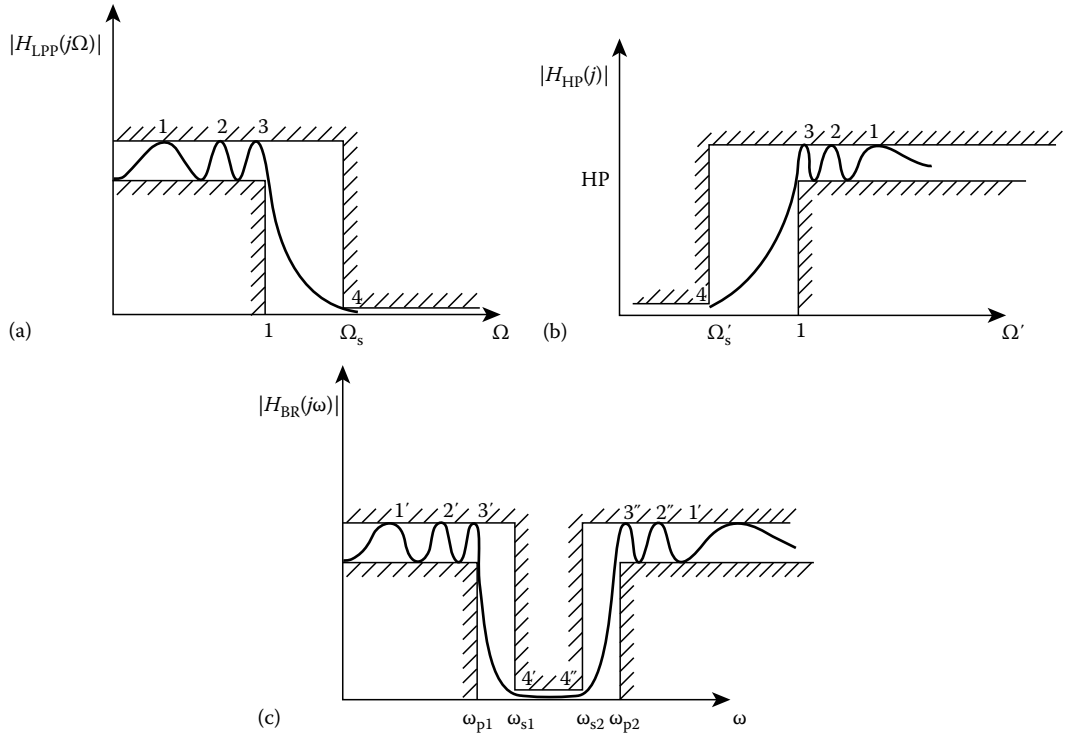


FIGURE 3.9 Low-pass to band-reject transformation: (a) Low-pass response, (b) normalized high-pass response, and (c) band-reject response derived from (a) and (b).

followed by a bandpass transformation applied to s'

$$s' = \frac{1}{\text{BW}} \frac{p^2 + \omega_o^2}{p} \tag{3.23}$$

Mapping of the imaginary ω axis to the Ω axis through this sequence of transformations leads to a band-rejection response in ω; frequency points 1, 2, 3, 4 have been singled out to better illustrate the transformation. Figure 3.9a shows the magnitude response of the LPP, Figure 3.9b shows the high-pass response obtained through Equation 3.22. This intermediate response is a normalized high-pass response with a passband extending from $\Omega'_p = 1$ to ∞ and a stopband extending from 0 to $\Omega'_s = 1/\Omega_s$. Figure 3.9c shows the magnitude response obtained by applying Equation 3.23 to the variable s'. The frequency range from $\Omega' = 0$ to ∞ in Figure 3.9b is mapped into the range from ω_o to ∞ and from $-\omega_o$ to 0. The frequency range from $-\infty$ to 0 in Ω' is mapped into the ranges from 0 to ω_o and from $-\infty$ to $-\omega_o$ in ω as indicated in Figure 3.9c. It can be seen that the bandpass transformation applied to a normalized high-pass response creates two passbands with ripple A_p from $\omega = 0$ to ω_{p1} and from ω_{p2} to ∞ and a stopband with attenuation A_s from ω_{s1} to ω_{s2}. The following conditions are satisfied:

$$\begin{aligned} \omega_{s1}\omega_{s2} &= \omega_{p1}\omega_{p2} = \omega_o^2 \\ \omega_{s2} - \omega_{s1} &= \text{BW}\Omega'_s = \text{BW}/\Omega_s \\ \omega_{p2} - \omega_{p1} &= \text{BW} \end{aligned} \tag{3.24}$$

From Equation 3.24, if ω_{s1}, ω_{s2}, ω_{p1}, ω_{p2} as well as A_p and A_s are specified for a band-rejection filter, then the selectivity factor of the equivalent LPP is calculated according to $\Omega_s = (\omega_{p2} - \omega_{p1})/(\omega_{s2} - \omega_{s1})$. Similar to the case of bandpass filters, the band-rejection specifications must be made symmetric with respect to ω_o so that an equivalent LPP can be specified. This is done following a similar procedure as for the bandpass transformation by constraining either one of the passband frequencies or one of the stopband frequencies so that the response becomes geometrically symmetric with respect to ω_o. The option leading to the largest selectivity factor (that corresponds in general to the least expensive network implementation) is then selected. In this case constraining design specifications refers to either increase ω_{p1} or ω_{s2} or to decrease ω_{p2} or ω_{s1}.

Example 3.4

Make following band-rejection filter specifications symmetric so that an equivalent LPP with the lowest Ω_s can be found: $\omega_{s1} = 2\pi$ 10, $\omega_{s2} = 2\pi$ 14.4, $\omega_{p1} = 2\pi$ 9, $\omega_{p2} = 2\pi$ 17.

Solution

Defining ω_o in terms of the passband frequencies the upper stopband frequency acquires the new value $\omega_{s2} = 2\pi$ 15.3 and the selectivity factor $\Omega_s = 1.509$ is obtained. If ω_o is defined in terms of the stopband frequencies the upper passband frequency is assigned the new value $\omega_{p2} = 2\pi$ 16 and the selectivity factor $\Omega_s = 1.59$ is obtained. Therefore, the second option with $\Omega_s = 1.59$ corresponds to the largest selectivity factor and the following transformation parameters result: $BW = \omega_{p2} - \omega_{p1} = 2\pi(16–9) = 2\pi$ 7, $\omega_o^2 = \omega_{p1}\ \omega_{p1} = (2\pi)^2$ 144 is made.

Transformation of poles and zeros of the LPP transfer function. To determine the poles and zeros of the band-rejection transfer function $H_{BR}(p)$ starting from those of $H_{LPP}(s)$, again a sequence of two transformations is required: poles and zeros of the LPP (denoted $s_{p1}, s_{p2}, \ldots, s_{pn}$ and $s_{z1}, s_{z2}, \ldots, s_{zn}$) are transformed into poles and zeros of $H_{HP}(s')$ $s'_{p1}, s'_{p2}, \ldots, s'_{z1}, s'_{z2}, \ldots, s'_{zm}$, which are reciprocal to those of the LPP. The high-pass also acquires $n - m$ zeros at the origin as explained in Sections 3.3 and 3.2. The transformations described in Section 3.4 are then applied to the high-pass poles zeros $s'_{p1}, s'_{p2}, \ldots, s'_{z1}, s'_{z2}, \ldots, s'_{zm}$.

Band-rejection network transformation. Using the transformation (Equation 3.21), an inductor in the LPP network with admittance $y_1(s) = 1/sl_n$ becomes an admittance $Y_p(p) = pC_p + 1/pL_p$, where $C_p = 1/\text{BW}\ l_n$ and $L_p = \text{BW}\ l_n/\omega_o^2$. A capacitor c_n in the LPP with impedance $z_c(s) = 1/sc_n$ becomes with Equation 3.20 an impedance $Z_s(p) = pL_s + 1/pC_s$, where $L_s = 1/c_n\text{BW}$ and $C_s = c_n\text{BW}/\omega_o^2$. To transform an LPP network into a bandpass network, capacitors in the LPP network are replaced by a series connection of an inductor with value L_s and a capacitor with value C_s, while inductors in the LPP are replaced by the parallel combination of a capacitor C_p and inductor L_p. This is illustrated in Figure 3.10a.

Example 3.5

Consider the LPP network shown in Figure 3.3a. It corresponds to the specifications: $A_p = 2$ dB, $A_s = 45$ dB, and $\Omega_s = 1.6$. Transform it into a bandpass network using the following parameters from Example 3.4: $\omega_o^2 = (2\pi)^2$ 144, $BW = 2\pi$ 7 Mrad/s (units for ω_o are microradians per second).

Solution

The circuit of Figure 3.10b is obtained applying the transformations indicated in Figure 3.10a.

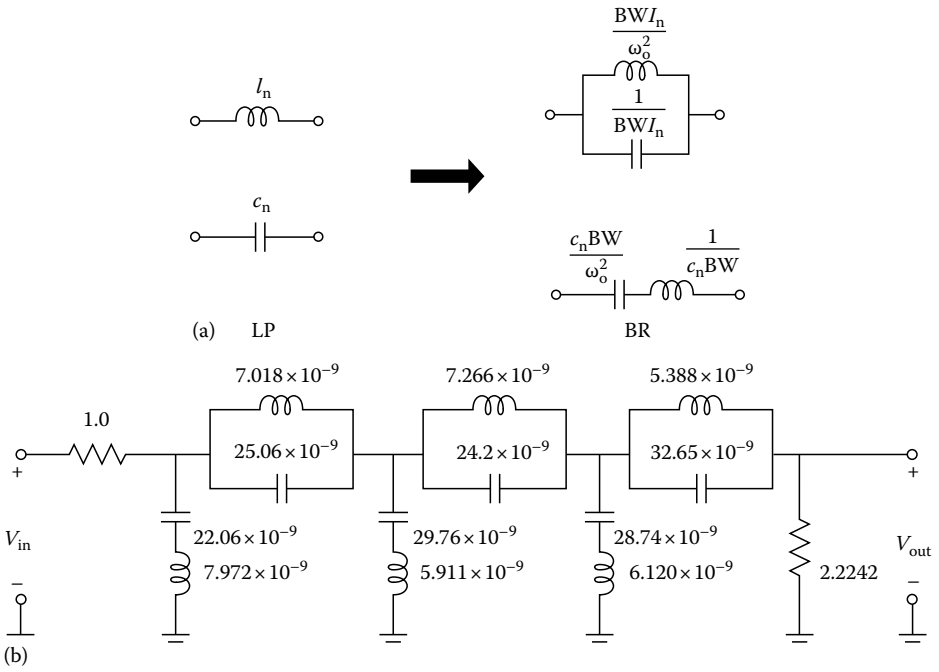


FIGURE 3.10 (a) LPP to band-reject transformations of network elements and (b) band-reject network derived from LPP of Figure 3.3a.

4

Sensitivity and Selectivity

4.1 Introduction 4-1
4.2 Definitions of Sensitivity 4-1
4.3 Function Sensitivity to One Variable 4-3
4.4 Coefficient Sensitivity 4-4
4.5 Root Sensitivities 4-5
4.6 Statistical Model for One Variable 4-7
4.7 Multiparameter Sensitivities and Sensitivity Measures 4-8
4.8 Sensitivity Invariants 4-15
4.9 Sensitivity Bounds 4-18
4.10 Remarks on the Sensitivity Applications 4-19
4.11 Sensitivity Computations Using the Adjoint Network 4-19
4.12 General Methods of Reducing Sensitivity 4-22
4.13 Cascaded Realization of Active Filters 4-23
4.14 Simulation of Doubly Terminated Matched Lossless Filters 4-28
4.15 Sensitivity of Active *RC* Filters 4-30
4.16 Errors in Sensitivity Comparisons 4-30
References 4-31

Igor M. Filanovsky
University of Alberta

4.1 Introduction

Using sensitivity one can evaluate the change in a filter performance characteristic (bandpass, *Q*-factor) or in a filter function (input impedance, transfer function) resulting from a change in a nominal value of one or more of the filter components. Hence, sensitivities and based on them sensitivity measures are used to compare different realizations of electric filters that meet the same specifications. Sensitivities can also be used to estimate the spread of the performance characteristic caused by the spread of the element values. In the design of filters, one is interested both in choosing realizations that have low sensitivities and in minimizing the sensitivities. This allows use of components with wider tolerances for a given variation or a given spread of the filter characteristic or function.

4.2 Definitions of Sensitivity

Let y be the filter performance characteristic and x be the value of the parameter of a filter element that is causing the characteristic change. The *relative sensitivity* is defined as follows:

$$\mathbf{S}_x^y(y,x) = \frac{\partial y}{\partial x}\frac{x}{y} = \frac{\partial y/y}{\partial x/x} = \frac{\partial(\ln y)}{\partial(\ln x)} \tag{4.1}$$

It is usually used to establish the approximate relationship between the relative changes $\delta y = \Delta y/y$ and $\delta x = \Delta x/x$. Here, Δy and Δx are absolute changes. The interpretation of relative changes δy and δx depends on the problem at hand. If these relative changes are small, one writes that

$$\delta y \approx \mathbf{S}_x^y(y,x)\delta x \tag{4.2}$$

(This relationship assumes that $\mathbf{S}_x^y$ is different from zero. If $\mathbf{S}_x^y = 0$, the relative changes δy and δx may be independent. This happens, for example, in the passband of doubly terminated LC networks [see below], where the passband attenuation always increases independently on the sign in the variation of a reactance element.) The argument in the parentheses of Equation 4.2, when does not involve any ambiguity, will usually be omitted, i.e., we will write simply $\mathbf{S}_x^y$. Some simple properties of the sensitivity determined by Equation 4.1 can be established by differentiation only. They are summarized in Table 4.1 [1].

One can also define two *semirelative sensitivities*

$$\mathbf{S}_x(y,x) = x\frac{\partial y}{\partial x} = \frac{\partial y}{\partial x/x} = \frac{\partial y}{\partial(\ln x)} \tag{4.3}$$

which is here frequently denoted by $\mathbf{S}_x(y)$, and

$$\mathbf{S}^y(y,x) = \frac{1}{y}\frac{\partial y}{\partial x} = \frac{\partial y/y}{\partial x} = \frac{\partial(\ln y)}{\partial x} \tag{4.4}$$

which is also denoted by $\mathbf{S}^y(x)$. Both these sensitivities can be used in a way similar to Equation 4.2 to establish the approximate relationships between one relative and one absolute change. Finally, the absolute sensitivity $\mathbf{S}(y,x)$ is simply a partial derivative of y with respect to x, i.e., it can be used to establish the relationship between absolute changes. The variable x represents the value for any component of the filter. The set of values for all the components will be denoted as $\boldsymbol{x} = \{x_i\}$, where $i = 1, 2, \ldots, n$.

TABLE 4.1 Properties of the Relative Sensitivity

Property Number	Relation	Property Number	Relation
1	$\mathbf{S}_x^{ky} = \mathbf{S}_{kx}^y = \mathbf{S}_x^y$	10	$\mathbf{S}_x^{y_1/y_2} = \mathbf{S}_x^{y_1} - \mathbf{S}_x^{y_2}$
2	$\mathbf{S}_x^x = \mathbf{S}_x^{kx} = \mathbf{S}_{kx}^{kx} = 1$	11	$\mathbf{S}_{x_1}^y = \mathbf{S}_{x_2}^y \mathbf{S}_{x_1}^{x_2}$
3	$\mathbf{S}_{1/x}^y = \mathbf{S}_x^{1/y} = -\mathbf{S}_x^y$	12[a]	$\mathbf{S}_x^y = \mathbf{S}_x^{\lvert y\rvert} + j\arg y \mathbf{S}_x^{\arg y}$
4	$\mathbf{S}_x^{y_1 y_2} = \mathbf{S}_x^{y_1} + \mathbf{S}_x^{y_2}$	13[a]	$\mathbf{S}_x^{\arg y} = \frac{1}{\arg y}\operatorname{Im}\mathbf{S}_x^y$
5	$\mathbf{S}_x^{\Pi_{i=1}^n y_i} = \sum_{i=1}^n \mathbf{S}_x^{y_i}$	14[a]	$\mathbf{S}_x^{\lvert y\rvert} = \operatorname{Re}\mathbf{S}_x^y$
6	$\mathbf{S}_x^{y^n} = n\mathbf{S}_x^y$	15	$\mathbf{S}_x^{y+z} = \frac{1}{y+z}\left(y\mathbf{S}_x^y + z\mathbf{S}_x^z\right)$
7	$\mathbf{S}_x^{x^n} = n\mathbf{S}_x^{kx^n} = n$	16	$\mathbf{S}_x^{\sum_{y=1}^n y_i} = \frac{\sum_{i=1}^n y_i \mathbf{S}_x^{y_i}}{\sum_{i=1}^n y_i}$
8	$\mathbf{S}_{x^n}^y = \frac{1}{n}\mathbf{S}_x^y$	17	$\mathbf{S}_x^{\ln y} = \frac{1}{\ln y}\mathbf{S}_x^y$
9	$\mathbf{S}_{x^n}^y = \mathbf{S}_{kx^n}^x = \frac{1}{n}$		

[a] In this relation, y is a complex quantity and x is a real quantity.

4.3 Function Sensitivity to One Variable

Let the chosen quantity y be the filter function $F(s,x)$. When it does not involve any ambiguity, this function will be denoted as $F(s)$. The element x is a passive or active element in the circuit realization of the function. The function sensitivity is defined as

$$\mathbf{S}_x^{F(s,x)} = \frac{\partial F(s,x)}{\partial x}\frac{x}{F(s,x)} \tag{4.5}$$

Under condition of sinusoidal steady state, when $s = j\omega$, the function $F(j\omega,x)$ can be represented as

$$F(j\omega,x) = |F(j\omega,x)|e^{j\arg F(j\omega,x)} = e^{-\alpha(\omega,x)+j\beta(\omega,x)} \tag{4.6}$$

and using the left-hand part of Equation 4.6, one finds that

$$\mathbf{S}_x^{F(j\omega,x)} = \mathrm{Re}\,\mathbf{S}_x^{F(j\omega,x)} + j\,\mathrm{Im}\,\mathbf{S}_x^{F(j\omega,x)} = \mathbf{S}_x^{|F(j\omega,x)|} + j\frac{\partial \arg F(j\omega,x)}{\partial x/x} \tag{4.7}$$

as follows from property 12 of Table 4.1. Thus, the real part of the function sensitivity gives the relative change in the magnitude response, and the imaginary part gives the change in the phase response, both with respect to a normalized element change. If one determines $\delta F = [F(j\omega,x) - F(j\omega,x_0)]/F(j\omega,x_0)$ and $\delta x = (x - x_0)/x_0$, where x_0 is the initial value of the element and the deflection $\Delta x = x - x_0$ is small, then Equation 4.5 is used to write

$$\delta F \approx \mathbf{S}_x^{F(s,x)}\delta x \tag{4.8}$$

And if one determines $\delta|F| = [|F(j\omega,x) - F(j\omega,x_0)|]/|F(j\omega,x_0)|$, then using Equation 4.7, one obtains

$$\delta|F| \approx \mathrm{Re}\,\mathbf{S}_x^{F(s,x)}\delta x \tag{4.9}$$

These calculations assume that the sensitivity is also calculated at $x = x_0$.

A frequently used alternate form of Equation 4.7 is obtained by using the attenuation function $\alpha(\omega,x) = \ln\left(\frac{1}{|F(j\omega,x)|}\right) = -\ln|F(j\omega,x)|$ and the phase function $\beta(\omega,x) = \arg F(j\omega,x)$ defined by the right-hand part of Equation 4.6 (this interpretation is usually used when $F(s)$ is the filter transfer function $T(s)$). In terms of these, Equation 4.7 may be rewritten as

$$\mathbf{S}_x^{F(j\omega,x)} = -\frac{\partial\alpha(\omega,x)}{\partial x/x} + j\frac{\partial\beta(\omega,x)}{\partial x/x} = -\mathbf{S}_x[\alpha(\omega),x] + j\mathbf{S}_x[\beta(\omega),x] \tag{4.10}$$

From Equations 4.7 and 4.10 one concludes that $\mathbf{S}_x^{|F(j\omega,x)|} = -\mathbf{S}_x[\alpha(\omega),x]$. Besides, using Equations 4.7 and 4.10 one can write that

$$\Delta\arg F(j\omega,x) = \Delta\beta(\omega,x) \approx \mathrm{Im}\,\mathbf{S}_x^{F(j\omega,x)}\delta x = \mathbf{S}_x[\beta(\omega),x]\delta x \tag{4.11}$$

where $\Delta\arg F(j\omega,x) = \arg F(j\omega,x) - \arg F(j\omega,x_0)$.

Usually the filter function is a ratio of two polynomials $N(s)$ and $D(s)$, i.e.,

$$F(s) = \frac{N(s)}{D(s)} \tag{4.12}$$

Then, assuming that the coefficients of $N(s)$ and $D(s)$ depend on the element x, and using Equation 4.1, one derives the following form of Equation 4.5:

$$\mathbf{S}_x^{F(s)} = x\left[\frac{\partial N(s)/\partial x}{N(s)} - \frac{\partial D(s)/\partial x}{D(s)}\right] \tag{4.13}$$

which is sometimes more convenient.

4.4 Coefficient Sensitivity

In general, a network function $F(s)$ for any active or passive lumped network is a ratio of polynomials having the form

$$F(s) = \frac{N(s)}{D(s)} = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_m s^m}{d_0 + d_1 s + d_2 s^2 + \cdots + d_n s^n} \tag{4.14}$$

Here the coefficients a_i and d_i are real and can be functions of an arbitrary filter element x. For such an element x one may define the *relative coefficient sensitivities* as follows:

$$\mathbf{S}_x^{a_i} = \frac{\partial a_i}{\partial x}\frac{x}{a_i}, \quad \mathbf{S}_x^{d_i} = \frac{\partial d_i}{\partial x}\frac{x}{d_i} \tag{4.15}$$

or the *semirelative coefficient sensitivities* (they are even more useful):

$$\mathbf{S}_x(a_i) = x\frac{\partial a_i}{\partial x}, \quad \mathbf{S}_x(d_i) = x\frac{\partial d_i}{\partial x} \tag{4.16}$$

The coefficient sensitivities defined in this way are related to the function sensitivity introduced in Section 4.3. Indeed, using Equations 4.13 and 4.16 one easily obtains that

$$\mathbf{S}_x^{F(s)} = \left[\frac{\sum_{i=0}^{m} \mathbf{s}_x(a_i)s^i}{N(s)} - \frac{\sum_{i=0}^{n} \mathbf{s}_x(d_i)s^i}{D(s)}\right] \tag{4.17}$$

or, in terms of relative sensitivities, that

$$\mathbf{S}_x^{F(s)} = \left[\frac{\sum_{i=0}^{m} \mathbf{s}_x^{a_i} a_i s^i}{N(s)} - \frac{\sum_{i=0}^{n} \mathbf{s}_x^{d_i} d_i s^i}{D(s)}\right] \tag{4.18}$$

The manner in which the filter function depends on any element x is a bilinear dependence [2]. Thus, Equation 4.14 may also be written in the form

$$F(s) = \frac{N(s)}{D(s)} = \frac{N_1(s) + xN_2(s)}{D_1(s) + xD_2(s)} \tag{4.19}$$

where $N_1(s), N_2(s), D_1(s)$, and $D_2(s)$ are polynomials with real coefficients that are not functions of the filter element x. This is true whether x is chosen to be the value of a passive resistor or capacitor, the gain of some amplifier or controlled source, etc. Only for filters with ideal transformers, ideal gyrators, and ideal negative impedance converters, the filter functions are the biquadratic functions of the ideal element parameters [2].

Because of the bilinear dependence property, there are only two ways in which a coefficient, say, a_i may depend on network element. The first of these has the form $a_i = kx$, in which case $\mathbf{S}_x^{a_i} = 1$ and $\mathbf{S}_x(a_i) = kx$; the second possible dependence for a coefficient a_i (or d_i) is $a_i = k_0 + k_1 x$, in which case $\mathbf{S}_x^{a_i} = k_1 x/(k_0 + k_1 x)$ and $\mathbf{S}_x(a_i) = k_1 x$. In this latter situation one has two cases: (1) the parities of the term are the same, and thus the magnitude of $\mathbf{S}_x^{a_i}$ is less than one and (2) the terms have opposite parities, in which case the magnitude of $\mathbf{S}_x^{a_i}$ is greater than one. In the last case the relative sensitivity $\mathbf{S}_x^{a_i}$ can have an infinite value, as a result of dividing by zero. In this case, a more meaningful measure of the change would be to use the semirelative coefficient sensitivity $\mathbf{S}_x(a_i)$.

4.5 Root Sensitivities

A filter function can also be represented as

$$F(s) = \frac{a_m \Pi_{i=0}^{m}(s - z_i)}{d_n \Pi_{i=0}^{n}(s - p_i)} \tag{4.20}$$

where z_i are zeros and p_i are poles. If $F(s)$ is also a function of the filter element x, the location of these poles and zeros will depend on this element. This dependence is described by the *semirelative root sensitivities*

$$\mathbf{S}_x(z_i) = x\frac{\partial z_i}{\partial x}, \quad \mathbf{S}_x(p_i) = x\frac{\partial p_i}{\partial x} \tag{4.21}$$

We will give calculation of the pole sensitivities only (they are used more frequently, to verify the stability) calculating the absolute change Δp_i for a given δx, the calculation of zeros follows the same pattern. Assume that p_i is a simple pole of $F(s)$, then

$$D(p_i) = D_1(p_i) + xD_2(p_i) = 0 \tag{4.22}$$

When the parameter x becomes $x + \Delta x$, the pole moves to the point $p_i + \Delta p_i$. Substituting these values in Equation 4.22 one obtains that

$$D_1(p_i + \Delta p_i) + (x + \Delta x)D_2(p_i + \Delta p_i) = 0 \tag{4.23}$$

If one uses Taylor's expansions $D_1(p_i + \Delta p_i) = D_1(p_i) + [\partial D_1(s)/\partial s]\,|_{s=p_i}\,\Delta p_i + \cdots$ and $D_2(p_i + \Delta p_i) = D_2(p_i) + [\partial D_2(s)/\partial s]\,|_{s=p_i}\,\Delta p_i + \cdots$ and substitutes them in Equation 4.23, keeping the terms of the first order of smallness, one obtains that

$$\frac{\Delta p_i}{\Delta x} = -\frac{D_2(p_i)}{D'(p_i)} \tag{4.24}$$

where $D'(p_i) = \left[\dfrac{\partial D(s)}{\partial s}\right]\Bigg|_{s=p_i}$. This result allows calculation of the pole sensitivity, which becomes

$$\mathbf{S}_x(p_i) = x\frac{\partial p_i}{\partial x} = -x\frac{D_2(p_i)}{D'(p_i)} \tag{4.25}$$

One can write that $D_1(s) = b_0 + b_1 s + b_2 s^2 + \cdots$ and $D_2(s) = c_0 + c_1 s + c_2 s^2 + \cdots$. Then, taking into consideration Equation 4.19, one can write that $D(s) = d_0 + d_1 s + d_2 s^2 + \cdots = (b_0 + xc_0) + (b_1 + xc_1)s + (b_2 + xc_2)s^2 + \cdots$. Differentiating this result, one obtains that $D_2(s) = c_0 + c_1 s + c_2 s^2 + \cdots = \partial d_0/\partial x + (\partial d_1/\partial x)s + (\partial d_2/\partial x)s^2 + \cdots = (1/x)[\mathbf{S}_x(d_0) + \mathbf{S}_x(d_1)s + \mathbf{S}_x(d_2)s^2 + \cdots]$. From

the other side, $D'(s) = \frac{\partial D(s)}{\partial s} = d_1 + 2d_2 s + 3d_3 s^2 + \cdots$. Calculating the two last expressions at $s = p_i$ and substituting them in Equation 4.25, one obtains that

$$\mathbf{S}_x(p_i) = -\frac{\sum_{j=0}^{n} p_i^j \mathbf{s}_x(d_j)}{\sum_{j=0}^{n-1} (j+1) d_{j+1} p_i^j} = -\frac{\sum_{j=0}^{n} d_j p_i^j \mathbf{s}_x^{d_j}}{\sum_{j=0}^{n-1} (j+1) d_{j+1} p_i^j} \tag{4.26}$$

The result (Equation 4.26) produces the pole sensitivity without representation of poles via coefficients of $D(s)$ (which is not possible if $n > 4$). It is convenient even for polynomials of low degree [3].

If p_i is a multiple root then the derivative $D'(p_i) = 0$ and $\mathbf{S}_x(p_i) = \infty$. But this does not mean that the variation δx causes infinitely large change of the pole location. This variation splits the multiple root into a group of simple roots. The location of these roots can be calculated in the following way. The roots are always satisfying the equation $D_1(s) + (x + \Delta x)D_2(s) = 0$, i.e., the equation $D(s) + \Delta x D_2(s) = 0$. One can rewrite the last equation as

$$1 + \delta x \left[\frac{x D_2(s)}{D(s)}\right] = 0 \tag{4.27}$$

where $\delta x = \Delta x / x$ as usual. The function $G(s) = \left[\frac{x D_2(s)}{D(s)}\right]$ can be represented as a sum of simple ratios. If the roots of $D(s)$ (i.e., the poles of $F(s)$) are simple, then

$$G(s) = \sum_{j=1}^{n} \frac{K_{ip}}{s - p_i} + K_{0p} \tag{4.28}$$

where

$K_{0p} = G(\infty)$

$K_{ip} = (s - p_i) G(s)\big|_{s=p_i}$

In the vicinity of $s = p_i$, Equation 4.27 can be substituted by

$$\lim_{s \to p_i} [1 + \delta x G(s)] = 1 + \delta x \frac{K_{ip}}{s - p_i} \tag{4.29}$$

Equating the right-hand side of Equation 4.29 to zero and substituting $s = p_i + \Delta p_i$ in this equation, one obtains that when $\Delta x \to 0$, the pole sensitivity can be calculated as

$$\mathbf{S}_x(p_i) = \frac{\partial p_i}{\partial x / x} = -K_{ip} \tag{4.30}$$

If a pole of $G(s)$ is not simple, but multiple, with multiplicity of k, then the limit form of Equation 4.28 will be

$$1 + \delta x \left[\frac{K_{ip}^{(1)}}{s - p_i} + \frac{K_{ip}^{(2)}}{(s - p_i)^2} + \cdots + \frac{K_{ip}^{(k)}}{(s - p_i)^k}\right] = 0 \tag{4.31}$$

If now $s = p_i + \Delta p_i$ is substituted in Equation 4.31 and only the largest term is kept, one finds that

$$\Delta p_i = \left[-\delta x K_{ip}^{(k)}\right]^{1/k} \tag{4.32}$$

Hence, these new simple roots of $D(s)$ are equiangularly spaced on a circle around p_i.

Similar calculations with similar results can be obtained for the zeros z_i of the function $F(s)$. They, together with the calculation for poles, allow one to establish the relationship between relative sensitivity of $F(s)$ and the sensitivities of its zeros and poles. Indeed, taking into consideration Equation 4.19, the result (Equation 4.13) can be rewritten as

$$\mathbf{S}_x^{F(s)} = \frac{xN_2(s)}{N(s)} - \frac{xD_2(s)}{D(s)} = H(s) - G(s) \tag{4.33}$$

Expanding both $H(s)$ and $G(s)$ into sums of simple ratios, one obtains

$$\mathbf{S}_x^{F(s)} = \sum_{i=1}^{n} \frac{\mathbf{S}_x(p_i)}{s - p_i} - \sum_{i=1}^{m} \frac{\mathbf{S}_x(z_i)}{s - z_i} + K_{0z} - K_{0p} \tag{4.34}$$

Here, $K_{0z} = H(\infty)$, $K_{0p} = G(\infty)$, and it is assumed that both zeros and poles of $F(s)$ are simple. Finally, a useful modification is obtained for the case when a coefficient d_k in the polynomial $D(s) = d_0 + d_1 s + d_2 s^2 + \cdots$ is considered as a variable parameter. If d_k is substituted by $d_k + \Delta d_k$, the polynomial $D(s)$ becomes $D(s) + \Delta d_k s^k$. For this case the function $G(s) = (\delta d_k s^k)/D(s)$ and one can write that

$$1 + \frac{\Delta d_k s^k}{D(s)} = 1 + \frac{\Delta d_k s^k}{(s - p_i)D'(s)} \tag{4.35}$$

Here $D'(s) = \partial D(s)/\partial s$ and it can be assumed that p_i is a simple root of $D(s)$. Hence, in the vicinity of $s = p_i + \Delta p_i$, the value of Δp_i can be obtained from the equation

$$1 + \frac{\Delta d_k p_i^k}{\Delta p_i D'(p_i)} = 0 \tag{4.36}$$

From Equation 4.36, one finds that

$$\frac{\Delta p_i}{\Delta d_k} = -\frac{p_i^k}{D'(p_i)} \tag{4.37}$$

and, if necessary, the pole-coefficient sensitivity

$$\mathbf{S}_{d_k}(p_i) = \frac{\partial p_i}{\partial d_k / d_k} = -d_k \frac{p_i^k}{D'(p_i)} \tag{4.38}$$

4.6 Statistical Model for One Variable

Assume that x is the value of a filter element. This x differs from the average value $\bar{x}$ in a way that cannot be controlled by the filter designer. This situation can be modeled by considering x as a random variable. Its statistical distribution depends on the manufacturing process. An approximate calculation is sufficient in most practical cases. If $F(s, x)$ is a filter function that depends on x, then the variation $F(s, x)$ around the average value $\bar{x}$ can be approximated by

$$F(s, x) \approx F(s, \bar{x}) + (x - \bar{x}) \frac{\partial F(s, x)}{\partial x}\Big|_{x=\bar{x}} \tag{4.39}$$

The statistical interpretation of Equation 4.29 follows. Due to its dependence on the random variable, it becomes a random variable as well. The values $\bar{x}$, $F(s,\bar{x}) = \bar{F}(s)$, and $\partial F(s,x)/\partial x$ calculated at $x = \bar{x}$ are constants. The last constant can be denoted as $\partial F(s,\bar{x})/\partial x$. Instead of x and $F(s,x)$ it is preferable to use their relative deviations from the average values, namely $\delta x = (x - \bar{x})/\bar{x}$ and $\delta F(s) = (F - \bar{F})/\bar{F}$. Then one obtains from Equation 4.39 that

$$\delta F(s) = \left[\frac{\bar{x}}{F(s,\bar{x})}\frac{\partial F(s,\bar{x})}{\partial x}\right]\delta x = \mathbf{S}_x^{F(s,\bar{x})}\delta x \tag{4.40}$$

Hence, in the first-order approximation the random variables $\delta F(s)$ and δx are proportional, and the proportionality factor is the same sensitivity of $F(s,x)$ with respect to x calculated at the average point $\bar{x}$. Thus, on the $j\omega$ axis the average and the variance of $\delta F(j\omega)$ and δx are related by

$$\mu_{\delta F} \approx \mathbf{S}_x^{F(s,\bar{x})}\mu_{\delta x} \tag{4.41}$$

and

$$\sigma_{\delta F}^2 = E\left\{\left|\frac{F(j\omega,x) - F(j\omega,\bar{x})}{F(j\omega,\bar{x})}\right|^2\right\} \approx |\mathbf{S}^{F(j\omega,\bar{x})}|^2\sigma_{\delta x}^2 \tag{4.42}$$

where $E = \{\}$ means the expected value. Here, $\mu_{\delta x}$ is the average value of δx, $\mu_{\delta F}$ is the average value of δF, and $\sigma_{\delta x}^2$ and $\sigma_{\delta F}^2$ are the dispersions of these values. If the deviation δx is bound by the modulus $M_{\delta x}$, i.e., the probability distribution is concentrated in the interval $[-M_{\delta x}, M_{\delta x}]$, then the deviation δF is bound in the first approximation by

$$|\delta F| \le M_{\delta F} \approx |\mathbf{S}_x^{F(j\omega,\bar{x})}|M_{\delta x} \tag{4.43}$$

Normally, the probability distribution of x should be centered around the average value $\bar{x}$, so that it can be assumed $\mu_{\delta x} = 0$. This implies that $\mu_{\delta F} = 0$ as well.

It is not difficult to see that Equations 4.8 and 4.40 are different by interpretation of δx and δF (since these were deflections from the nominal point, it was tacitly assumed that we were dealing with one sample of the filter, here they are random) and the point of calculation of sensitivity. The interpretation with a random variable is possible for Equation 4.9 as well. One has to determine $\delta|F| = [|F(j\omega,x)| - |F(j\omega,\bar{x})|]/|F(j\omega,\bar{x})|$, then using Equation 4.9 one can write

$$\mu_{\delta|F|} \approx \mathrm{Re}\,\mathbf{S}_x^{F(s,\bar{x})}\mu_{\delta x} \tag{4.44}$$

and

$$\sigma_{\delta|F|}^2 \approx \left(\mathrm{Re}\,\mathbf{S}_x^{F(j\omega,\bar{x})}\right)^2\sigma_{\delta x}^2 \tag{4.45}$$

The result (Equation 4.11) can also be interpreted for random variables and allows one to calculate the average and the variance of the change in the filter function argument (which is hardly ever done in filter design).

4.7 Multiparameter Sensitivities and Sensitivity Measures

The multiparameter sensitivities (sometimes [4] they are called sensitivity indices) appear as an effort to introduce generalized functions that represent the influence of all filter elements. They can be used for

comparison of different designs and should be minimized in the design process. The sensitivity measures appear as numbers (they are functionals of the parameter sensitivities) that should be minimized in the design.

First of all, the definition of function sensitivity given in Equation 4.5 is readily extended to determine the effect on the filter function of variation of more than one component. In this case $F(s, x_1, x_2, \ldots, x_n) = F(s, \mathbf{x})$ and one may write that

$$\frac{\mathrm{d}F(s,\mathbf{x})}{F(s,\mathbf{x})} = \mathrm{d}[\ln F(s,\mathbf{x})] = \sum_{i=1}^{n} \mathbf{S}_{x_i}^{F(s,\mathbf{x})} \frac{\mathrm{d}x_i}{x_i} \tag{4.46}$$

where n is the number of components being considered. Here $\mathbf{S}_{x_i}^{F(s,\mathbf{x})} = [x_i \partial F(s,\mathbf{x})]/[F(s,\mathbf{x})\partial x_i]$. From this result one directly (substituting $s = j\omega$ and separating real and imaginary parts) obtains that

$$\frac{\mathrm{d}|F(j\omega,\mathbf{x})|}{|F(j\omega,\mathbf{x})|} = \sum_{i=1}^{n} \mathrm{Re}\, \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \frac{\mathrm{d}x_i}{x_i} \tag{4.47}$$

and

$$\mathrm{d}\arg F(j,\omega) = \sum_{i=1}^{n} \mathrm{Im}\, \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \frac{\mathrm{d}x_i}{x_i} \tag{4.48}$$

The results (Equations 4.47 and 4.48) are used to evaluate the deviations of the magnitude and phase values of a given filter realization from their nominal values when the circuit elements have prescribed normalized deviations $\delta x_i = (x_i - x_{i0})/x_{i0}$ $(i = 1, 2, \cdots n)$. One can introduce a column vector of normalized deviation $[\delta x_1 \delta x_2 \cdots \delta x_n]^{\mathrm{t}}$, where t means transpose and a sensitivity row vector

$$\mathbf{S}_{\mathbf{x}}^{F(s,\mathbf{x})} = \left[\mathbf{S}_{x_1}^{F(s,\mathbf{x})} \mathbf{S}_{x_2}^{F(s,\mathbf{x})} \cdots \mathbf{S}_{x_n}^{F(s,\mathbf{x})}\right] \tag{4.49}$$

Then defining $\delta F(s,\mathbf{x}) = [F(s,\mathbf{x}) - F(s,\mathbf{x}_0)]/F(s,\mathbf{x}_0)$, one can use Equation 4.46 to write

$$\delta F(s,\mathbf{x}) \approx \mathbf{S}_{\mathbf{x}}^{F(s,\mathbf{x})} \delta\mathbf{x} \tag{4.50}$$

which is analogous to Equation 4.8. As was mentioned before, the calculation of the filter function magnitude change is traditionally considered of primary importance in filter design. Introducing $\delta|F(s,\mathbf{x})| = [|F(j\omega,\mathbf{x})| - |F(j\omega,\mathbf{x}_0)|]/|F(j\omega,\mathbf{x}_0)|$ and using Equation 4.47 one writes

$$\delta|F(j\omega,\mathbf{x})| \approx \left[\mathrm{Re}\, \mathbf{S}_{x}^{F(j\omega,\mathbf{x})}\right] \delta\mathbf{x} \tag{4.51}$$

where the row vector

$$\left[\mathrm{Re}\, \mathbf{S}_{x}^{F(j\omega,\mathbf{x})}\right] = \left[\mathrm{Re}\, \mathbf{S}_{x_1}^{F(j\omega,\mathbf{x})} \mathrm{Re}\, \mathbf{S}_{x_2}^{F(j\omega,\mathbf{x})} \cdots \mathrm{Re}\, \mathbf{S}_{x_n}^{F(j\omega,\mathbf{x})}\right] \tag{4.52}$$

is used. This vector is determined by the function $F(s,\mathbf{x})$ and its derivatives calculated at $\mathbf{x} = \mathbf{x}_0$. To characterize and compare the vectors of this type, one can introduce different vector measures that are called *sensitivity indices*. The most frequently used ones are the *average sensitivity index*

$$\psi(F) = \sum_{i=1}^{n} \mathrm{Re}\, \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \tag{4.53}$$

then the *worst-case sensitivity index*

$$\nu(F) = \sum_{i=1}^{n} \left| \operatorname{Re} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right| \tag{4.54}$$

(sometimes it is called worst-case magnitude sensitivity), and, finally, the *quadratic sensitivity index*

$$\rho(F) = \left[\sum_{i=1}^{n} \left(\operatorname{Re} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right)^2 \right]^{1/2} \tag{4.55}$$

These sensitivity indices can be considered as multiparameter sensitivities.

If we let the individual nominal values of n elements be given as x_{i0}, then we may define a tolerance constant ε_i (positive number) by the requirement that

$$x_{i0}(1 - \varepsilon_i) \le x_i \le x_{i0}(1 - \varepsilon_i) \tag{4.56}$$

Then we may define a worst-case measure of sensitivity

$$M_{\mathrm{W}} = \int_{\omega_1}^{\omega_2} \left(\sum_{i=1}^{n} \left| \operatorname{Re} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right| \varepsilon_i \right) \mathrm{d}\omega \tag{4.57}$$

The goal of the filter design should be the search for the set of tolerance constants yielding the least expensive in the production filter. This is a difficult problem, and at the design stage it can be modeled by the minimization of the chosen sensitivity measure. In the design based on the worst-case measure of sensitivity, the usual approach [2] is to choose the tolerance constants in such a way that the contributions $\left| \operatorname{Re} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right| \varepsilon_i$ are approximately equal, i.e., the elements with lower sensitivities get wider tolerance constants.

For any values of the filter elements x_i satisfying Equation 4.56, the magnitude characteristic will lie within the definite bounds that are apart from the nominal characteristic by the distance less than $\varepsilon_{i\max} \nu(F)$. If the tolerance constants are all equal to ε, then the maximum deviation from the nominal characteristic (when $\mathbf{x} = \mathbf{x}_0$) is thus given as $\varepsilon \nu(F)$. And the worst-case measure of sensitivity becomes, for this case

$$M_{\mathrm{W}} = \varepsilon \int_{\omega_1}^{\omega_2} \nu(F) \mathrm{d}\omega \tag{4.58}$$

Considering the imaginary parts of the sensitivity row vector one can introduce corresponding sensitivity indices and similar sensitivity measures for the filter function phase.

The element tolerances obtained using the worst-case sensitivity index and measures are extremely tight and this set of elements is frequently unfeasible. Besides, with given tolerances, the set of elements producing the worst-case sensitivity is never obtained in practice. A more feasible set of tolerances is obtained when one uses the sum of the squares of the individual functions. One may define a *quadratic measure of sensitivity* as

$$M_{\mathrm{Q}} = \int_{\omega_1}^{\omega_2} \left[\sum_{i=1}^{n} \left(\operatorname{Re} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right)^2 \varepsilon_i^2 \right] \mathrm{d}\omega \tag{4.59}$$

In the design using the sensitivity measure given by Equation 4.59 one also tries to get the tolerances so that the contributions of each term $\left(\mathrm{Re}\,\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right)^2\varepsilon_i^2$ are approximately equal in the considered bandwidth. Again, if the tolerances are equal, then this expression is simplified to

$$M_Q = \varepsilon^2 \int_{\omega_1}^{\omega_2} \rho^2(F)\mathrm{d}\omega \tag{4.60}$$

which is useful for comparison of different filters.

As one can see, the multivariable sensitivities appear as a result of certain operations with the sensitivity row vector components. Additional multivariable sensitivities could be introduced, for example, the sum of magnitudes of vector components, the sum of their squares, etc. The multivariable sensitivities and the measures considered above represent the most frequently used filter design in the context of filter characteristic variations.

The case of random variables can also be generalized so that imprecisions of the values of several elements are simultaneously considered. Around the nominal value $\bar{\mathbf{x}} = [\bar{x}_i]$ $(i = 1, 2, \cdots n)$ the function $F(s, \bar{\mathbf{x}})$ can be approximated as

$$F(s, \mathbf{x}) \approx F(s, \bar{\mathbf{x}}) + \sum_{i=1}^{n} (x - \bar{x}_i) \frac{\partial F(s, \mathbf{x})}{\partial x_i} \tag{4.61}$$

And from this approximation one obtains

$$\delta F(s, \mathbf{x}) \approx \sum_{i=1}^{n} \mathbf{S}_{x_i}^{F(s,\mathbf{x})} \delta x_i \tag{4.62}$$

Here, $\delta F(s, x) = [F(s, \boldsymbol{x}) - F(s, \overline{\boldsymbol{x}})]/F(s, \overline{\boldsymbol{x}})$ and $\delta x_i = (x_i - \bar{x}_i)/\bar{x}_i$ $(i = 1, 2, \cdots n)$. This result can be rewritten as

$$\delta F(s, \mathbf{x}) \approx \mathbf{S}_{\mathbf{x}}^{F(s,\mathbf{x})} \delta \mathbf{x} \tag{4.63}$$

and is completely analogous to Equation 4.50. It is different in interpretation only. The components of the column vector $[\delta x_1 \delta x_2 \cdots \delta x_n]^{\mathrm{t}}$ are the random variables now and the components of the row vector $\mathbf{S}_{\mathbf{x}}^{F(s,\mathbf{x})} = \left[\mathbf{S}_{\mathbf{x}_1}^{F(s,\mathbf{x})} \mathbf{S}_{\mathbf{x}_2}^{F(s,\mathbf{x})} \cdots \mathbf{S}_{\mathbf{x}_n}^{F(s,\mathbf{x})}\right]$ are calculated at the point $\mathbf{x} = \bar{\mathbf{x}}$.

This interpretation allow us to obtain from Equation 4.63 that on the $j\omega$ axis

$$\mu_{\delta F} = \sum_{i=1}^{n} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \mu_i \tag{4.64}$$

Here, μ_i is the average of δx_i. If all μ_i are equal, i.e., $\mu_i = \mu_x$ $(i = 1, 2, \cdots n)$, one can introduce the average sensitivity index

$$\psi(F) = \sum_{i=1}^{n} \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \tag{4.65}$$

Using Equation 4.65, the average value can be calculated as $\mu_{\delta F} = \mu_x \psi(F)$. If, in addition, the deviation of δx_i is bound by M_i, then

$$M_{\delta F} \le \sum_{i=1}^{n} \left|\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right| M_i \tag{4.66}$$

If the elements of a filter have the same precision, which means that all M_i are equal, it is reasonable to introduce the worst-case sensitivity index

$$\nu(F) = \sum_{i=1}^{n} \left|\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right| \tag{4.67}$$

so that when all M_i are equal to M_x, $M_{\delta F} = M_x \nu(F)$. Finally, one can calculate $\sigma_{\delta F}^2 = E\{[\delta F(j\omega, \mathbf{x})]^* \delta F(j\omega, \mathbf{x})\}$ (here * means complex conjugate) or

$$\sigma_{\delta F}^2 = E\left\{\left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})}\delta\mathbf{x}\right)^*\left(\mathbf{S}_{\boldsymbol{x}}^{F(j\omega,\mathbf{x})}\delta\mathbf{x}\right)\right\} \tag{4.68}$$

To take into consideration possible correlation between the components δx_i one can do the following. The value of $\left(\mathbf{S}_{\boldsymbol{x}}^{F(j\omega,\mathbf{x})}\delta\mathbf{x}\right)$ is a scalar. Then

$$\left(\mathbf{S}_{\boldsymbol{x}}^{F(j\omega,\mathbf{x})}\delta\mathbf{x}\right) = \left(\mathbf{S}_{\boldsymbol{x}}^{F(j\omega,\mathbf{x})}\delta\mathbf{x}\right)^{\mathrm{t}} = (\delta\mathbf{x})^t\left(\mathbf{S}_{\boldsymbol{x}}^{F(j\omega,\mathbf{x})}\right)^{\mathrm{t}} \tag{4.69}$$

Substituting this result into Equation 4.68 one obtains that

$$\sigma_{\delta F}^2 = E\left\{ \left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})}\right)^*(\delta\mathbf{x})^*(\delta\mathbf{x})^{\mathrm{t}}\left(\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right)^{\mathrm{t}} \right\} \tag{4.70}$$

But the components of $\delta\mathbf{x}$ are real, i.e., $(\delta\mathbf{x})^* = \delta\mathbf{x}$ and the result of multiplication in the curly brackets of Equation 4.70 is a square $n \times n$ matrix. Then $E\{(\delta\mathbf{x})^*(\delta\mathbf{x})^{\mathrm{t}}\}$ is also a square matrix

$$[\mathbf{P}] = \begin{bmatrix} \sigma_{x_1}^2 & \rho_{x_1x_2} & \cdots & \rho_{x_1x_n} \\ \rho_{x_2x_1} & \sigma_{x_2}^2 & \cdots & \rho_{x_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{x_nx_1} & \cdots & \cdots & \sigma_{x_n}^2 \end{bmatrix} \tag{4.71}$$

the diagonal elements of which are variances of δx_i and off-diagonal terms are nonnormalized correlation coefficients. Then, Equation 4.70 can be rewritten as

$$\sigma_{\delta F}^2 = \left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})}\right)^*[\mathbf{P}]\left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})}\right)^{\mathrm{t}} \tag{4.72}$$

which is sometimes [4] called the propagation-of-variance formula. In the absence of correlation between the variations δx_i the matrix $[\mathbf{P}]$ has the diagonal terms only and Equation 4.72 becomes

$$\sigma_{\delta F}^2 = \sum_{i=1}^{n} \left|\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right|^2 \sigma_{x_i}^2 \tag{4.73}$$

If all $\sigma_{x_i}^2$ are equal to σ_x^2 one can introduce a quadratic sensitivity index

$$\rho(F) = \left[\sum_{i=1}^{n} \left|\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}\right|^2\right]^{1/2} \tag{4.74}$$

and in this case $\sigma^2_{\delta F} = \rho^2(F)\sigma^2_x$ (the value $\rho^2(F)$ is sometimes called Schoeffler multivariable sensitivity). One can also introduce two sensitivity measures, namely, the worst-case sensitivity measure

$$M_{\mathrm{W}} = \int_{\omega_1}^{\omega_2} \left(\sum_{i=1}^{n} \left| \mathrm{Re}\, \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right| \mu_i \right) \mathrm{d}\omega \tag{4.75}$$

and the quadratic sensitivity measure

$$M_{\mathrm{Q}} = \int_{\omega_1}^{\omega_2} \left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})} \right)^{*} [\mathbf{P}] \left(\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})} \right)^{\mathrm{t}} \mathrm{d}\omega \tag{4.76}$$

which, when the correlation between the elements of $\delta\mathbf{x}$ is absent, becomes

$$M_{\mathrm{Q}} = \int_{\omega_1}^{\omega_2} \left[\sum_{i=1}^{n} \left| \mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} \right|^2 \sigma_i^2 \right] \mathrm{d}\omega \tag{4.77}$$

Here, for simplicity the notation $\sigma_i = \sigma_{x_i}$ is used.

The sensitivity indices and sensitivity measures introduced for the case when $\delta\mathbf{x}$ is a random vector are cumulative; they take into consideration the variation of the amplitude and phase of the filter function. For this reason some authors prefer to use the indices as they are defined in Equations 4.65, 4.67, and 4.74 and the measures as they are defined by Equations 4.75 and 4.77 for the deterministic cases as well (the deterministic case does not assume any correlation between the variations δx_i), with the corresponding substitution of μ_i by ε_i and σ_i^2 by ε_i^2. From the other side, one can take Equation 4.51 and use it for the case of random vector $\delta\mathbf{x}$ considering, for example, the variation $\delta|F(j\omega,\mathbf{x})|$ as a random variable and calculating $\mu_{\delta|F|}$ and $\sigma^2_{\delta|F|}$, which will be characteristics of this variable. In this case one can use the results (Equations 4.75, 4.77, etc.) substituting $\mathbf{S}_{x_i}^{F(s,\mathbf{x})}$ by $\mathrm{Re}\,\mathbf{S}_{x_i}^{F(s,\mathbf{x})}$. These possibilities are responsible for many formulations of multiparameter sensitivities that represent different measures of the vector $\mathbf{S}_{\boldsymbol{x}}^{F(s,\mathbf{x})}$. In the design based, for example, on Equations 4.75 and 4.77, one determines the required $\sigma^2_{\delta F}$ using the reject probability [2] depending on the ratio of $\varepsilon_{\delta F}/\sigma_{\delta F}$. Here, $\varepsilon_{\delta F}$ is the tolerance of $|\delta F(j\omega,\mathbf{x})|$ and in many cases one takes $\varepsilon_{\delta F}/\sigma_{\delta F} = 2.5$ which gives the reject probability of 0.01. Then, one determines the dispersion σ_i^2 so that the contributions of each term in Equation 4.77 are equal. Finally, using the probability function that describes the distribution of δx_i within the tolerance borders one finds these borders (if, for example, the selected element has evenly distributed values $-\varepsilon_i \le \delta x_i \le \varepsilon_i$, then $\varepsilon_i = \sqrt{3}\sigma_i$; for Gaussian distribution one frequently accepts $\varepsilon_i = 2.5\sigma_i$).

The preliminary calculation of the coefficient sensitivities is useful for finding the sensitivity measures. If, for example, one calculates a multivariable statistical measure of sensitivity then one can consider that

$$F(j\omega,\mathbf{x}) = \frac{a_0 + a_1(j\omega) + \cdots + a_m(j\omega)^m}{d_0 + d(j\omega) + \cdots + d_n(j\omega)^n} = F(j\omega,\mathbf{a},\mathbf{d},\mathbf{x}) \tag{4.78}$$

where

$$\mathbf{a} = [\, a_0 \quad a_1 \quad \cdots \quad a_m \,]^{\mathrm{t}}$$
$$\mathbf{d} = [\, d_0 \quad d_1 \quad \cdots \quad d_n \,]^{\mathrm{t}}$$

Then, the components $\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})}$ defined earlier can be rewritten as

$$\begin{aligned}\mathbf{S}_{x_i}^{F(j\omega,\mathbf{x})} &= \frac{x_i}{F(j\omega,\mathbf{x})}\left[\frac{\partial \mathbf{a}^t}{\partial x_i}\nabla_a F(j\omega,\mathbf{x}) + \frac{\partial \mathbf{d}^t}{\partial x_i}\nabla_d F(j\omega,\mathbf{x})\right] \\ &= \frac{1}{F(j\omega,\mathbf{x})}\left[\sum_{j=0}^{m}\frac{\partial F(j\omega,\mathbf{x})}{\partial a_j}\mathbf{S}_{x_i}(a_j) + \sum_{j=0}^{n}\frac{\partial F(j\omega,\mathbf{x})}{\partial d_j}\mathbf{S}_{x_i}(d_j)\right] \end{aligned} \tag{4.79}$$

where

$$\nabla_a F(j\omega,\mathbf{x}) = [\partial F(j\omega,\mathbf{x})/\partial a_0\ \partial F(j\omega,\mathbf{x})/\partial a_1 \cdots \partial F(j\omega,\mathbf{x})/\partial a_m]^{\mathrm{t}}$$

and

$$\nabla_d F(j\omega,\mathbf{x}) = [\partial F(j\omega,\mathbf{x})/\partial d_0\ \partial F(j\omega,\mathbf{x})/\partial d_1 \cdots \partial F(j\omega,\mathbf{x})/\partial d_n]^{\mathrm{t}}$$

For a given transfer function, the components of the vectors $\nabla_a F(j\omega,\mathbf{x})$ and $\nabla_d F(j\omega,\mathbf{x})$ are independent of the form of the realization or the values of the elements and can be calculated in advance. If we now define a matrix $k \times (m+1)$ $\mathbf{C}_1$ as

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{S}_{x_1}(a_0) & \mathbf{S}_{x_1}(a_1) & \cdots & \mathbf{S}_{x_1}(a_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{x_k}(a_0) & \mathbf{S}_{x_k}(a_1) & \cdots & \mathbf{S}_{x_k}(a_m) \end{bmatrix} \tag{4.80}$$

and $k \times (n+1)$ matrix $\mathbf{C}_2$ as

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{S}_{x_1}(d_0) & \mathbf{S}_{x_1}(d_1) & \cdots & \mathbf{S}_{x_1}(d_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{x_k}(d_0) & \mathbf{S}_{x_k}(d_1) & \cdots & \mathbf{S}_{x_k}(d_n) \end{bmatrix} \tag{4.81}$$

then one can rewrite

$$\left[\mathbf{S}_{\mathbf{x}}^{F(j\omega,\mathbf{x})}\right]^t = \left[\mathbf{S}_{x_1}^{F(j\omega,\mathbf{x})}\mathbf{S}_{x_2}^{F(j\omega,\mathbf{x})}\cdots\mathbf{S}_{x_k}^{F(j\omega,\mathbf{x})}\right]^t = \mathbf{C}_1\frac{\nabla_a F(j\omega,\mathbf{x})}{F(j\omega,\mathbf{x})} + \mathbf{C}_2\frac{\nabla_d F(j\omega,\mathbf{x})}{F(j\omega,\mathbf{x})} \tag{4.82}$$

Then, the multiparameter sensitivity measure can be written as

$$M_{\mathrm{Q}} = \int_{\omega_1}^{\omega_2}\left[\left(\frac{\nabla_a F}{F}\right)^{*_t}\mathbf{C}_1^t\mathbf{P}\mathbf{C}_1\left(\frac{\nabla_a F}{F}\right) + \left(\frac{\nabla_d F}{F}\right)^{*_t}\mathbf{C}_2^t\mathbf{P}\mathbf{C}_2\left(\frac{\nabla_d F}{F}\right)\right]\mathrm{d}\omega + \int_{\omega_1}^{\omega_2} 2\mathrm{Re}\left[\left(\frac{\nabla_a F}{F}\right)^{*_t}\mathbf{C}_1^t\mathbf{P}\mathbf{C}_2\left(\frac{\nabla_d F}{F}\right)\right]\mathrm{d}\omega \tag{4.83}$$

and this definition of statistical multiparameter sensitivity measure may be directly applied to a given network realization. In a similar fashion, the matrices of unnormalized coefficient sensitivities can be used with other multiparameter sensitivity measures.

4.8 Sensitivity Invariants

When one is talking about sensitivity invariants [5–7], it is assumed that for a filter function $F(s, \mathbf{x})$ there exists the relationship

$$\sum_{i=1}^{n} \mathbf{S}_{x_i}^{F(s,\mathbf{x})} = k \tag{4.84}$$

where
$\mathbf{x} = [x_1 x_2 \cdots x_n]^t$, as usual,
k is a constant

These relationships are useful to check the sensitivity calculations. In the cases considered below, this constant can have one of three possible values, namely, 1, 0, and −1, and the sensitivity invariants are obtained from the homogeneity of some of the filter functions.

The function $F(s, \mathbf{x})$ is called homogeneous of order k with respect to the vector $\mathbf{x}$ if and only if it satisfies the relationship

$$F(s, \lambda\mathbf{x}) = \lambda^k F(s, \mathbf{x}) \tag{4.85}$$

where λ is an arbitrary scalar. For the homogeneous function $F(s, \mathbf{x})$ the sensitivities are related by Equation 4.84. Indeed, if one takes the logarithm of both sides of Equation 4.85, one obtains

$$\ln F(s, \lambda\mathbf{x}) = k \ln \lambda + \ln F(s, \mathbf{x}) \tag{4.86}$$

Taking the derivative of both sides of Equation 4.86 with respect to λ one obtains that

$$\frac{1}{F(s, \lambda\mathbf{x})} \left[\sum_{i=1}^{n} \frac{\partial F(s, \lambda\mathbf{x})}{\partial(\lambda x_i)} x_i \right] = \frac{K}{\lambda} \tag{4.87}$$

Substituting $\lambda = 1$ in Equation 4.87 gives Equation 4.84.

Let the filter be a passive *RLC* circuit that includes r resistors, l inductors, and c capacitors, so that $r + l + c = n$ and $\mathbf{x} = [R_1 R_2 \cdots R_r L_1 L_2 \cdots L_l D_1 D_2 \cdots D_c]$, where $D_i = 1/C_i$. One of the frequently used operations is the impedance scaling. If the scaling operation is applied to a port impedance or a transimpedance of the filter, i.e., $F(s, \mathbf{x}) = Z(s, \mathbf{x})$, then

$$Z(s, \lambda\mathbf{x}) = \lambda Z(s, \mathbf{x}) \tag{4.88}$$

Equation 4.88 is identical to Equation 4.85, with $k = 1$. Then, one can write that

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{Z(s,\mathbf{x})} + \sum_{i=1}^{l} \mathbf{S}_{L_i}^{Z(s,\mathbf{x})} + \sum_{i=1}^{c} \mathbf{S}_{D_i}^{Z(s,\mathbf{x})} = 1 \tag{4.89}$$

Considering that $D_i = 1/C_i$ and $\mathbf{S}_{C_i}^{Z(s,\mathbf{x})} = -\mathbf{S}_{D_i}^{Z(s,\mathbf{x})}$ (see Table 4.1), this result can be rewritten as

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{Z(s,\mathbf{x})} + \sum_{i=1}^{l} \mathbf{S}_{L_i}^{Z(s,\mathbf{x})} - \sum_{i=1}^{c} \mathbf{S}_{C_i}^{Z(s,\mathbf{x})} = 1 \tag{4.90}$$

If the same scaling operation is applied to a port admittance or a transadmittance of the filter, i.e., $F(s, \mathbf{x}) = Y(s, \mathbf{x})$, then,

$$Y(s, \lambda\mathbf{x}) = \lambda^{-1} Y(s, \mathbf{x}) \tag{4.91}$$

But Equation 4.91 is identical to Equation 4.85, with $k = -1$. Then,

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{Y(s,\mathbf{x})} + \sum_{i=1}^{l} \mathbf{S}_{L_i}^{Y(s,\mathbf{x})} - \sum_{i=1}^{c} \mathbf{S}_{C_i}^{Y(s,\mathbf{x})} = -1 \tag{4.92}$$

Finally, the transfer functions (voltage or current) do not depend on the scaling operation, i.e., if $F(s, \mathbf{x}) = T(s, \mathbf{x})$, hence,

$$T(s, \lambda\mathbf{x}) = T(s, \mathbf{x}) \tag{4.93}$$

which is identical to Equation 4.85, with $k = 0$. Then,

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{T(s,\mathbf{x})} + \sum_{i=1}^{l} \mathbf{S}_{L_i}^{T(s,\mathbf{x})} - \sum_{i=1}^{c} \mathbf{S}_{C_i}^{T(s,\mathbf{x})} = 0 \tag{4.94}$$

Additional sensitivity invariants can be obtained using the relation $\mathbf{S}_{x_i}^{F(s,\mathbf{x})} = -\mathbf{S}_{1/x_i}^{Z(s,\mathbf{x})}$ and using $G_i = 1/R_i$ and $\Gamma_i = 1/L_i$.

Another group of sensitivity invariants is obtained using the frequency scaling operation. The following relationship is held for a filter function:

$$F(s, R_i, \lambda L_i, \lambda C_i) = F(\lambda s, R_i, L_i, C_i) \tag{4.95}$$

Taking the logarithm of both parts, then differentiating both sides with respect to λ and substituting $\lambda = 1$ in both sides gives

$$\sum_{i=1}^{l} \mathbf{S}_{L_i}^{F(s,\mathbf{x})} + \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(s,\mathbf{x})} = \mathbf{S}_{s}^{F(s,\mathbf{x})} \tag{4.96}$$

Substituting $s = j\omega$ in Equation 4.96 and dividing the real and imaginary parts, one obtains

$$\mathrm{Re} \sum_{i=1}^{l} \mathbf{S}_{L_i}^{F(j\omega,\mathbf{x})} + \mathrm{Re} \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(j\omega,\mathbf{x})} = \omega \frac{\partial \ln |T(\omega)|}{\partial \omega} = -\frac{\partial \alpha(\omega)}{\partial \omega} \tag{4.97}$$

and

$$\mathrm{Im} \sum_{i=1}^{l} \mathbf{S}_{L_i}^{F(j\omega,\mathbf{x})} + \mathrm{Im} \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(j\omega,\mathbf{x})} = \omega \frac{\partial \arg T(\omega)}{\partial \omega} \tag{4.98}$$

The results (Equations 4.97 and 4.98) show that in an *RLC* filter, when all inductors and capacitors (but not resistors) are subjected to the same relative change, then the resulting change in the magnitude

characteristic does not depend on the circuit realization and is determined by the slope of the magnitude characteristic at a chosen frequency. A similar statement is valid for the phase characteristic.

The sensitivity invariants for passive *RC* circuits can be obtained from the corresponding invariants for passive *RLC* circuits omitting the terms for sensitivities to the inductor variations. The results can be summarized the following way. For a passive *RC* circuit

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{F(s,\mathbf{x})} - \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(s,\mathbf{x})} = k \tag{4.99}$$

where $k = 1$ if $F(s, \mathbf{x})$ is an input impedance or transimpedance function, then $k = 0$ if $F(s, \mathbf{x})$ is a voltage- or current-transfer function, and $k = -1$ if $F(s, \mathbf{x})$ is an input admittance or transconductance function. Application of the frequency scaling gives the result

$$\sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(s,\mathbf{x})} = \mathbf{S}_{s}^{F(s,\mathbf{x})} \tag{4.100}$$

and combination of Equations 4.99 and 4.100 gives

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{F(s,\mathbf{x})} = \mathbf{S}_{s}^{F(s,\mathbf{x})} + k \tag{4.101}$$

and

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{F(s,\mathbf{x})} + \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(s,\mathbf{x})} = 2\mathbf{S}_{s}^{F(s,\mathbf{x})} + k \tag{4.102}$$

Considering real and imaginary parts of Equations 4.100 through 4.102, one can obtain the results that determine the limitations imposed on the sensitivity sums by the function $F(j\omega, \mathbf{x})$ when resistors and/or capacitors are subjected to the same relative change.

Finally, if the filter is not passive, then the vector of parameters $\mathbf{x} = \begin{bmatrix} R_1 R_2 \cdots R_r L_1 L_2 \cdots L_l C_1 C_2 \cdots C_c R_{T1} R_{T2} \cdots R_{Ta} \\ G_{T1} G_{T2} \cdots G_{Tb} A_{v1} A_{v2} \cdots A_{vp} A_{i1} A_{i2} \cdots A_{iq} \end{bmatrix}$ includes the components of transresistors R_{Tk}, transconductances G_{Tk}, voltage amplifiers A_{vk}, and current amplifiers A_{ik}. Applying the impedance scaling one can obtain the sensitivity invariant

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}^{F(s,\mathbf{x})} + \sum_{i=1}^{l} \mathbf{S}_{L_i}^{F(s,\mathbf{x})} - \sum_{i=1}^{c} \mathbf{S}_{C_i}^{F(s,\mathbf{x})} + \sum_{i=1}^{a} \mathbf{S}_{R_{Ti}}^{F(s,\mathbf{x})} - \sum_{i=1}^{b} \mathbf{S}_{G_{Ti}}^{F(s,\mathbf{x})} = k \tag{4.103}$$

where

$k = 1$ if $F(s, \mathbf{x})$ is an impedance function
$k = 0$ if $F(s, \mathbf{x})$ is a transfer function
$k = -1$ if $F(s, \mathbf{x})$ is an admittance function

The frequency scaling will give the same result as Equation 4.96.

The pole (or zero) sensitivities are also related by some invariant relationships. Indeed, the impedance scaling provides the result

$$p_k\left(\lambda R_i, \lambda L_i, \frac{C_i}{\lambda}\right) = p_k(R_i, L_i, C_i) \tag{4.104}$$

Taking the derivative of both sides of Equation 4.104 with respect to λ and substituting $\lambda = 1$, one obtains that for an arbitrary *RLC* circuit

$$\sum_{i=1}^{r} \mathbf{S}_{R_i}(p_k) + \sum_{i=1}^{l} \mathbf{S}_{L_i}(p_k) - \sum_{i=1}^{c} \mathbf{S}_{C_i}(p_k) = 0 \tag{4.105}$$

This is the relationship between semirelative sensitivities. If $p_k \neq 0$ one can divide both sides of Equation 4.105 by p_k and obtain similar invariants for relative sensitivities. The frequency scaling gives

$$p_k\left(R_i, \frac{L_i}{\lambda}, \frac{C_i}{\lambda}\right) = \lambda p_k(R_i, L_i, C_i) \tag{4.106}$$

and from Equation 4.106 one obtains, for relative sensitivities only, that

$$\sum_{i=1}^{l} \mathbf{S}_{L_i}^{p_k} + \sum_{i=1}^{c} \mathbf{S}_{C_i}^{p_k} = -1 \tag{4.107}$$

The pole sensitivity invariants for passive *RC* circuits are obtained from Equations 4.106 and 4.107, omitting the terms corresponding to the inductor sensitivities.

4.9 Sensitivity Bounds

For some classes of filters, the worst-case magnitude sensitivity index may be shown to have a lower bound [8]. Such a bound, for example, exists for filters whose passive elements are limited to resistors, capacitors, and ideal transformers, and whose active elements are limited to gyrators characterized by two gyration resistances (realized as a series connection of transresistance amplifiers and considered as different for sensitivity calculations), current-controlled current sources (CCCSs), voltage-controlled voltage sources (VCVSs), voltage-controlled current sources (VCCSs), and current-controlled voltage sources (CCVSs). Using the sensitivity invariants, it is easy to show that for such a class of networks for any dimensionless transfer function $T(s)$

$$\sum_{i=1}^{n} \mathbf{S}_{x_i}^{T(s)} = 2\mathbf{S}_s^{T(s)} \tag{4.108}$$

where the x_i are taken to include only the passive elements of resistors and capacitors and the active elements of CCVSs and gyrators (if the gyrators are realized as parallel connection of transconductance amplifiers, the corresponding terms should be taken with the negative sign). Substituting $s = j\omega$ in Equation 4.108 and equating real parts, one obtains that

$$\sum_{i=1}^{n} \mathbf{S}_{x_i}^{|T(j\omega)|} = 2\mathbf{S}_{j\omega}^{|T(j\omega)|} \tag{4.109}$$

Applying Equation 4.54 to the above equations, one obtains for the worst-case magnitude sensitivity index

$$\nu(T) - \sum_{i=1}^{n} \left|\mathbf{S}_{x_i}^{|T(j\omega)|}\right| \geq \left|\sum_{i=1}^{n} \mathbf{S}^{|T(j\omega)|}\right| - \left|2\mathbf{S}^{|T(j\omega)|}\right| \tag{4.110}$$

Taking the first and the last terms of this expression, one may define a lower bound LBν(T) of the worst-case magnitude sensitivity index as

$$\mathrm{LB}\nu(T) = \left|2\mathbf{S}_{j\omega}^{|T(j\omega)|}\right| \tag{4.111}$$

This lower bound is a function only of the transfer function $T(s)$ and is independent on the particular synthesis technique (as soon as the above-mentioned restrictions are satisfied) used to realize this transfer function. A similar lower bound may be derived (taking the imaginary parts of Equation 4.108) for worst-case phase sensitivity; but it is impossible to find the design path by which one will arrive at the circuit realizing this minimal bound.

4.10 Remarks on the Sensitivity Applications

The components of active *RC* filters have inaccuracies and parasitic components that distort the filter characteristics. The most important imperfections are the following:

1. Values of the resistors and capacitors and the values of transconductances (the gyrators can be considered usually as parallel connection of two transconductance amplifiers) are different from their nominal values. The evaluation of these effects is done using the worst-case or (more frequently) quadratic multiparameter sensitivity index, and multiparameter sensitivity measures and the tolerances are chosen so that the contributions of the passive elements' variations in the sensitivity measure are equal.
2. Operational amplifiers have finite gain and this gain is frequency dependent. The effect of finite gain is evaluated using a semirelative sensitivity of the filter function with respect to variation $1/A$, where A is the operational amplifier gain. The semirelative sensitivity is

$$-\mathbf{S}^{F(s)}\left(\frac{1}{A}\right) = -\frac{\partial F(s)}{\left[F(s)\partial\left(\frac{1}{A}\right)\right]} = \left[\frac{A^2}{F(s)}\right]\left[\frac{\partial F(s)}{\partial A}\right] = A\mathbf{S}_A^{F(s)}$$

   which is called the gain-sensitivity product. In many cases, $\mathbf{S}_A^{F(s)} \to 0$ when $A \to \infty$, whereas $\mathbf{S}_{1/A}[F(s)]$ has a limit that is different from zero. The frequency dependence is difficult to take into consideration [1]. Only in case of cascade realization, as shown below, one can evaluate the effect of this frequency dependence using the sensitivity of Q-factor.
3. Temperature dependence and aging of the passive elements and operational amplifiers can be determined. The influence of temperature, aging, and other environmental factors on the values of the elements can be determined by a dependence of the probability distributions for these parameters. For example, if θ is temperature, one has to estimate $\mu_x(\theta)$ and $\sigma_x(\theta)$ before the calculation of sensitivity measures. Normally, at the nominal temperature θ_0 the average value $\mu_x(\theta_0) = 0$ and $\sigma_x(\theta_0)$ depends on the nominal precision of the elements. When the temperature changes, $\mu_x(\theta)$ increases or decreases depending on the temperature coefficients of the elements, whereas $\sigma_x(\theta)$ usually increases for any temperature variation.

4.11 Sensitivity Computations Using the Adjoint Network

The determination of the sensitivities defined in the previous sections may pose difficult computational problems. Finding the network function with the elements expressed in literal form is usually tedious and error prone and the difficulty of such a determination increases rapidly with the number of elements. Calculating the partial derivatives, which is the most important part of the sensitivity computation, provides additional tedium and increases the possibility of error still more. Thus, in general, it is advantageous to use digital computer methods to compute sensitivities. The most obvious method for doing this is to use one of

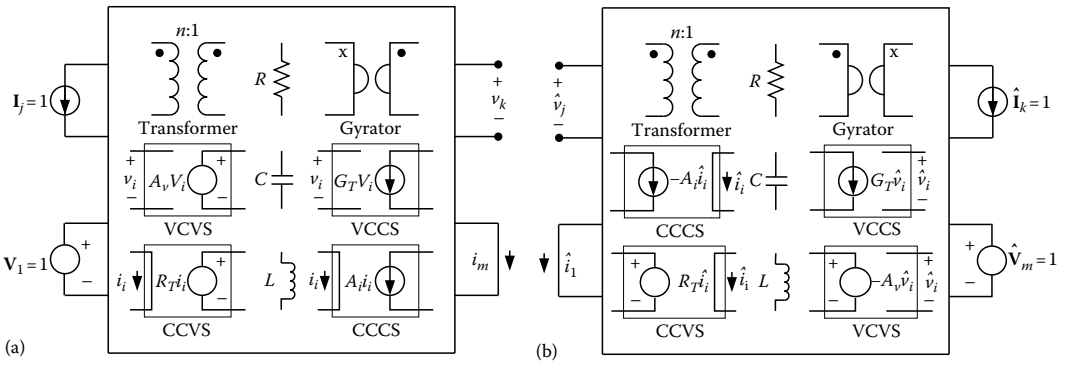


FIGURE 4.1 The components of a network (a) and its adjoint (b).

the many available computer-aided design programs (e.g., SPICE) to make an analysis of the network with nominal element values, and then repeat the analysis after having perturbed the value of one of the elements. This is not a desirable procedure, since it requires a large number of analyses. It can be justified if the network has some *a priori* known critical elements for which the analyses should be done.

The crucial part of sensitivity computation is, as was mentioned above, the calculation of network function derivatives with respect to element variations. To simplify this part of the calculation, the concept of adjoint network [9] is used. This method requires only two analyses to provide all the sensitivities of a given network immittance function.

If $\mathbf{N}$ and $\widehat{\mathbf{N}}$ are linear time invariant networks, then they are said to be adjoint (to each other) if the following hold. The two networks have the same topology and ordering of branches; thus their incidence matrices [10] are equal, namely $\mathbf{A} = \widehat{\mathbf{A}}$. If excitation with the unit current (unit voltage) at an arbitrary port j (port l) of network $\mathbf{N}$ yields a voltage (current) at an arbitrary port k (port m) of $\mathbf{N}$, excitation with the unit current (unit voltage) at port k (port m) of network $\widehat{\mathbf{N}}$ will yield the same voltage (current) as the above in port j (port l) of $\widehat{\mathbf{N}}$ (see Figure 4.1).

Figure 4.1 also shows how the adjoint network should be constructed, as follows. (a) All resistance capacitive and inductance branches and transformers in $\mathbf{N}$ are associated, respectively, with resistance, capacitive, and inductance branches and transformers in $\widehat{\mathbf{N}}$. (b) All gyrators in $\mathbf{N}$ with gyration resistance r become gyrators in $\widehat{\mathbf{N}}$ with gyration resistance $-r$. (c) All VCVSs in $\mathbf{N}$ become CCCSs in $\widehat{\mathbf{N}}$ with controlling and controlled branches reversing roles, and with the voltage amplification factor A_v becoming the current amplification factor $-A_i$. (d) All CCCSs in $\mathbf{N}$ become VCVSs in $\widehat{\mathbf{N}}$ with controlling and controlled branches reversing roles, and with the current amplification factor A_i becoming the voltage amplification factor $-A_v$. (e) All VCCCs and CCVSs have their controlling and controlled branches in $\mathbf{N}$ reversed in $\widehat{\mathbf{N}}$.

Thus, the Tellegen theorem [9] applies to the branch voltage and current variables of these two networks. If we let $\mathbf{V}$ be the vector of branch voltages and $\mathbf{I}$ be the vector of branch currents (using capital letters implies that the quantities are functions of the complex variable s), then

$$\mathbf{V}^{\mathrm{t}}\hat{\mathbf{I}} = \hat{\mathbf{I}}^{\mathrm{t}}\mathbf{V} = \widehat{\mathbf{V}}^{\mathrm{t}}\mathbf{I} = \mathbf{I}^{\mathrm{t}}\mathbf{V} = \mathbf{0} \tag{4.112}$$

If in both circuits all independent sources have been removed to form n external ports (as illustrated in Figure 4.2), then one can divide the variables in both circuits into two groups so that

$$\begin{aligned} \mathbf{I}^{\mathrm{t}} &= \left[\mathbf{I}_{\mathrm{p}}^{\mathrm{t}}\mathbf{I}_{\mathrm{b}}^{\mathrm{t}}\right] \quad \mathbf{V}^{\mathrm{t}} = \left[\mathbf{V}_{\mathrm{p}}^{\mathrm{t}}\mathbf{V}_{\mathrm{b}}^{\mathrm{t}}\right] \\ \hat{\mathbf{I}}^{\mathrm{t}} &= \left[\hat{\mathbf{I}}_{\mathrm{p}}^{\mathrm{t}}\hat{\mathbf{I}}_{\mathrm{b}}^{\mathrm{t}}\right] \quad \widehat{\mathbf{V}}^{\mathrm{t}} = \left[\widehat{\mathbf{V}}_{\mathrm{p}}^{\mathrm{t}}\widehat{\mathbf{V}}_{\mathrm{b}}^{\mathrm{t}}\right] \end{aligned} \tag{4.113}$$

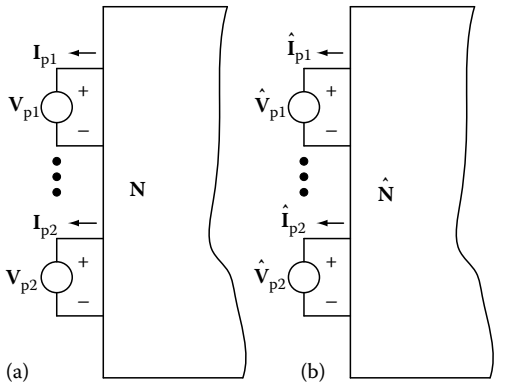


FIGURE 4.2 Separation of port variables in the network (a) and its adjoint (b).

The first are the vector of port variables $\mathbf{V}_p$ and $\mathbf{I}_p$ in $\mathbf{N}$ and $\hat{\mathbf{V}}_p$ and $\hat{\mathbf{I}}_p$, correspondingly, in $\hat{\mathbf{N}}$. These variables will define n-port open-circuit impedance matrices $\mathbf{Z}_{oc}$ and $\hat{\mathbf{Z}}_{oc}$ or n-port short-circuit admittance matrices $\mathbf{Y}_{sc}$ and $\hat{\mathbf{Y}}_{sc}$ via the relationships

$$\begin{aligned} \mathbf{V}_p &= -\mathbf{Z}_{oc}\mathbf{I}_p \quad \hat{\mathbf{V}}_p = -\hat{\mathbf{Z}}_{oc}\hat{\mathbf{I}}_p \\ \mathbf{I}_p &= -\mathbf{Y}_{sc}\mathbf{V}_p \quad \hat{\mathbf{I}}_p = -\hat{\mathbf{Y}}_{sc}\hat{\mathbf{V}}_p \end{aligned} \tag{4.114}$$

Then, the rest of the variables will be nonport variables (including the variables of dependent source branches) $\mathbf{V}_b$ and $\mathbf{I}_b$ for $\mathbf{N}$ and $\hat{\mathbf{V}}_b$ and $\hat{\mathbf{I}}_b$ for $\hat{\mathbf{N}}$. These variables may define branch impedance matrices $\mathbf{Z}_b$ and $\hat{\mathbf{Z}}_b$ and branch admittance matrices $\mathbf{Y}_b$ and $\hat{\mathbf{Y}}_b$ by the relationships

$$\begin{aligned} \mathbf{V}_b &= \mathbf{Z}_b\mathbf{I}_b \quad \hat{\mathbf{V}}_b = \hat{\mathbf{Z}}_b\hat{\mathbf{I}}_b \\ \mathbf{I}_b &= \mathbf{Y}_b\mathbf{V}_b \quad \hat{\mathbf{I}}_b = \hat{\mathbf{Y}}_b\hat{\mathbf{V}}_b \end{aligned} \tag{4.115}$$

If the branch impedance and branch admittance matrices do not exist, a hybrid matrix may be used to relate the branch variables. For $\mathbf{N}$, this may be put in the form

$$\begin{bmatrix} \mathbf{V}_{b1} \\ \mathbf{I}_{b2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{b1} \\ \mathbf{V}_{b2} \end{bmatrix} \tag{4.116}$$

Similarly, for $\hat{\mathbf{N}}$ one may write

$$\begin{bmatrix} \hat{\mathbf{V}}_{b1} \\ \hat{\mathbf{I}}_{b2} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{H}}_{11} & \hat{\mathbf{H}}_{12} \\ \hat{\mathbf{H}}_{21} & \hat{\mathbf{H}}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{I}}_{b1} \\ \hat{\mathbf{V}}_{b2} \end{bmatrix} \tag{4.117}$$

For the adjoint networks, the branch impedance and branch admittance matrices (if they exist) are transposed, namely

$$\mathbf{Z}_b^t = \hat{\mathbf{Z}}_b \quad \mathbf{Y}_b^t = \hat{\mathbf{Y}}_b \tag{4.118}$$

and, if a hybrid representation is used, the matrices are connected by the relationship

$$\begin{bmatrix} \mathbf{H}_{11}^t & -\mathbf{H}_{21}^t \\ -\mathbf{H}_{12}^t & \mathbf{H}_{22}^t \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{H}}_{11} & \hat{\mathbf{H}}_{12} \\ \hat{\mathbf{H}}_{21} & \hat{\mathbf{H}}_{22} \end{bmatrix} \tag{4.119}$$

As a result of these relationships, if no controlled sources are present in two networks, they are identical. In the general case it may be shown that

$$\mathbf{Z}_{oc}^t = \hat{\mathbf{Z}}_{oc} \quad \mathbf{Y}_{sc}^t = \hat{\mathbf{Y}}_{sc} \tag{4.120}$$

The application of adjoint circuits for sensitivity calculations requires that, first of all, using the port variables as independent ones, one finds the branch variables, and this is done for both circuits. Assume,

for example, that the branch impedance matrices exist. Then, using unity-valued excitation currents as the components of $\mathbf{I}_p$ and $\hat{\mathbf{I}}_p$ one has, first, to find $\mathbf{I}_b$ and $\hat{\mathbf{I}}_b$. Now, in the original network let the elements be perturbed. The resulting vector of currents may thus be written as $\mathbf{I} + \Delta\mathbf{I}$ and the resulting vector of voltages as $\mathbf{V} + \Delta\mathbf{V}$. From Kirchhoff's current law, we have $\mathbf{A}(\mathbf{I} + \Delta\mathbf{I}) = \mathbf{0}$, and since $\mathbf{AI} = \mathbf{0}$ one also has $\mathbf{A}\Delta\mathbf{I} = \mathbf{0}$. Thus, $\Delta\mathbf{I} = \Delta\mathbf{I}_p + \Delta\mathbf{V}_b$ may be substituted in any of the relations in Equation 4.112. By similar reasoning one can conclude that it is possible as well to substitute the perturbation vector $\Delta\mathbf{V} = \Delta\mathbf{V}_p + \Delta\mathbf{V}_b$ instead of $\mathbf{V}$ in these relations. Making these substitutions, one obtains

$$\begin{aligned} \hat{\mathbf{V}}_p^t \Delta\mathbf{I}_p + \hat{\mathbf{V}}_b^t \Delta\mathbf{I}_p &= 0 \\ \mathbf{I}_p^t \Delta\mathbf{V}_p + \hat{\mathbf{I}}_b^t \Delta\mathbf{V}_b &= 0 \end{aligned} \tag{4.121}$$

Subtracting these two equations, one has

$$\hat{\mathbf{V}}_p^t \Delta\mathbf{I}_p - \mathbf{I}_p^t \Delta\mathbf{V}_p + \hat{\mathbf{V}}_b^t \Delta\mathbf{I}_p - \hat{\mathbf{I}}_b^t \Delta\mathbf{V}_b = 0 \tag{4.122}$$

To a first-order approximation, we have

$$\begin{aligned} \Delta\mathbf{V}_p &= -\Delta(\mathbf{Z}_{oc}\mathbf{I}_p) \approx -\Delta\mathbf{Z}_{oc}\mathbf{I}_p - \mathbf{Z}_{oc}\Delta\mathbf{I}_p \\ \Delta\mathbf{V}_b &= -\Delta(\mathbf{Z}_b\mathbf{I}_b) \approx -\Delta\mathbf{Z}_b\mathbf{I}_b - \mathbf{Z}_b\Delta\mathbf{I}_b \end{aligned} \tag{4.123}$$

Substituting Equation 4.123 into Equation 4.122 and taking into consideration that $\hat{\mathbf{V}}_p^t = -\hat{\mathbf{I}}_p^t \hat{\mathbf{Z}}_{oc}^t = -\hat{\mathbf{I}}_p^t \mathbf{Z}_{oc}$ and that $\hat{\mathbf{V}}_b^t = \hat{\mathbf{I}}_b^t \mathbf{Z}_b$, one can simplify the result (Equation 4.122) to

$$\hat{\mathbf{I}}_p^t \Delta\mathbf{Z}_{oc}\mathbf{I}_p = \hat{\mathbf{I}}_b^t \Delta\mathbf{Z}_b\mathbf{I}_b \tag{4.124}$$

Equation 4.124 clearly shows that if all currents in the original network and its adjoint are known, one can easily calculate the absolute sensitivities $\mathbf{S}(\mathbf{Z}_{ij}, \mathbf{Z}_b) \approx \Delta\mathbf{Z}_{ij}/\Delta\mathbf{Z}_b$, which can be used for calculation of the corresponding relative sensitivities. Here, $\mathbf{Z}_{ij}$ is an element of the n-port open circuit impedance matrix. Then, if necessary, these sensitivities can be used for evaluation of the transfer function sensitivity or the sensitivities of other functions derived via the n-port open circuit impedance matrix.

Usually, the transfer function calculation can be easily reduced to the calculation of a particular $\mathbf{Z}_{ij}$. In this case one can choose $\hat{I}_j = 1, \hat{I}_k = 0$ (for all $k \neq j$) as an excitation in the adjoint network. Then, Equation 4.124 becomes

$$\Delta\mathbf{Z}_{ij} = \hat{\mathbf{I}}_b^t \Delta\mathbf{Z}_b\mathbf{I}_b \tag{4.125}$$

where $\mathbf{I}_b$ and $\hat{\mathbf{I}}_b$ are the branch currents in $\mathbf{N}$ and $\hat{\mathbf{N}}$ corresponding to the indicated excitation. The relations for other types of matrices are obtained in the same manner.

4.12 General Methods of Reducing Sensitivity

It is very desirable from the start of the realization procedure to concentrate on circuits that give lower sensitivity in comparison to other circuits. The practice of active filter realization allows formulation of some general suggestions ensuring that filter realizations will have low sensitivities to component variations.

It is possible to transfer the problem of lower sensitivity at the approximation stage, i.e., before any realization. It is obvious that a low-order transfer function $T(s)$ that just satisfies the specifications will require tighter tolerances in comparison to a higher order transfer function that easily satisfies the specifications. Figure 4.3 shows an example of such an approach for a low-pass filter. Hence, increasing the order of approximation and introducing a redundancy one achieves a set of wider element tolerances.

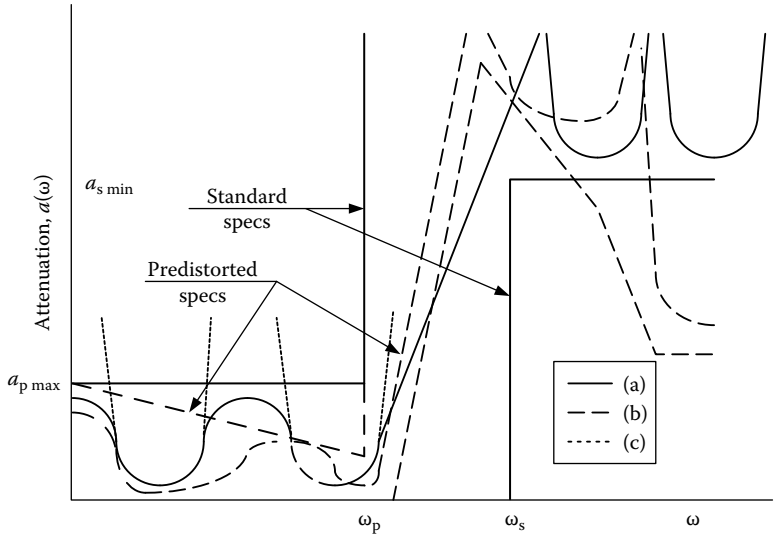


FIGURE 4.3 Attenuation requirements, their predistortion, and attenuation of different realizations: (a) attenuation of a standard circuit of a higher order; (b) attenuation of the nonstandard circuit; and (c) passband attenuation in the stages of cascade realization.

Usually the most critical region where it is difficult to satisfy the specifications is the edge of the passband. Two approaches can be used to find the function that will have less sensitivity in this frequency region. One way is to introduce predistortion in the transfer function specifications. This is also shown in Figure 4.3. The transfer function should satisfy the predistorted (tapered) specifications. It can be obtained directly if the numerical packages solving the approximation problem are available. One can also take a standard table higher order transfer function satisfying the modified specifications and then modify it to more uniformly use the tapered specifications (this allows the increase of the component tolerances even more).

Another way [11] is to preserve the initial transfer function specifications and to use transfer functions with a limited value of the maximum Q of the transfer function poles. In Ref. [11] one can find such transfer functions corresponding to the Cauer approximation. The nonstandard LC circuits corresponding to these approaches cannot be tabulated and simulated; hence, neither of them is widely used. In addition, they imply the cascaded (building-block) realization that intrinsically has worse sensitivity than the realizations using simulation of doubly terminated lossless matched filters.

4.13 Cascaded Realization of Active Filters

The cascaded (building-block) realization is based on the assumption that the transfer function will have low sensitivity if the realization provides tight control of the transfer function poles and zeros. The relationship between the element value and the transfer function poles and zeros can be established relatively easily if the transfer function is not more complicated than biquadratic (i.e., the ratio of two second-order polynomials). It is difficult (or even impossible) to establish such correspondence if, say, the denominator polynomial degree is higher than two. For a high-degree polynomial, a small variation in the polynomial coefficient can result in a large or undesirable migration of the root (it can move to the right-half plane). This justifies the cascaded approach: one hopes to have a low sensitivity of the transfer function under realization if one chooses a method allowing tight control of the poles' and zeros' locations; hence, cascade realization with $T(s) = T_1(s)T_2(s)\cdots T_i(s)\cdots T_k(s)$. Also, if one chooses for

realization of each function $T_i(s)$ the method of lowest sensitivity (discussed below), then it will be possible to obtain the largest element tolerances.

If the realization by cascade connection is chosen it is still a choice of optimum factorization of the transfer function $T(s)$ into low-order factors. This optimum depends on the filter application, the chosen method of the factor realization, and the transfer function itself. It is recommended [1] that, for realization of lower sensitivity, the poles and zeros in the partial functions $T_i(s)$ are located as far apart as possible. This statement is not always true and such a choice of poles and zeros in $T_i(s)$ is in contradiction with the requirement of high dynamic range of the stage. In general, CAD methods should be used.

The methods that are most popular for the realization of the partial transfer functions are mostly limited by the filters providing the output voltage at the output of an operational amplifier. The state–space relations satisfy this requirement and provide the direct realizations of the polynomial coefficients. It is not occasionally that such an approach is used by a series of manufacturers. This is not the best method from the sensitivity point of view; the methods of realization using gyrators usually give better results [3]. But the cascade realization of gyrator filters require buffers between the blocks, and is better to use this approach if the filter is not realized in cascade form. Other realization methods [1] are also occasionally used, mostly because of their simplicity.

If the transfer function is realized in a cascade form and $T(s) = T_1(s)T_2(s)\cdots T_i(s)\cdots T_k(s)$, the element x is located only in one stage. If this is the stage realizing $T_i(s)$, then

$$\mathbf{S}_x^{T(s)} = \mathbf{S}_x^{T_i(s)} \tag{4.126}$$

Assume that this $T_i(s)$ has the form

$$T_i(s) = K\frac{s^2 + \left(\frac{\omega_z}{Q_z}\right) + \omega_z^2}{s^2 + \left(\frac{\omega_p}{Q_p}\right) + \omega_p^2} \tag{4.127}$$

Then one can write

$$\mathbf{S}_x^{T_i(s)} = \mathbf{S}_K^{T_i(s)}\mathbf{S}_x^K + \mathbf{S}_{\omega_z}^{T_i(s)}\mathbf{S}_x^{\omega_z} + \mathbf{S}_{1/Q_z}^{T_i(s)}\mathbf{S}_x^{1/Q_z} + \mathbf{S}_{\omega_p}^{T_i(s)}\mathbf{S}_x^{\omega_p} + \mathbf{S}_{1/Q_p}^{T_i(s)}\mathbf{S}_x^{1/Q_p} \tag{4.128}$$

The second multiplier in each term of this sum depends on the stage realization method. In the first term $\mathbf{S}_K^{T_i(s)} = 1$, the first multipliers in other terms depend on Q-factors of zeros and poles. It is enough to consider the influence of the terms related to the poles. One can notice that $\mathbf{S}_{1/Q_p}^{T_i(s)}\mathbf{S}_x^{1/Q_p} = \mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_p}\right]\mathbf{S}_x\left(\frac{1}{Q_p}, x\right)$, then, calculating $\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_p}\right]$ (the calculation of the semirelative sensitivity is done for convenience of graphic representation) and $\mathbf{S}_{\omega_p}^{T_i(s)}$, one obtains

$$\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_p}\right] = \frac{1}{T_i(s)}\frac{\partial T_i(s)}{\partial\left(\frac{1}{Q_p}\right)} = -\frac{1}{\left(\frac{s}{\omega_p}\right) + \left(\frac{\omega_p}{s}\right) + \left(\frac{1}{Q_p}\right)} \tag{4.129}$$

and

$$\mathbf{S}_{\omega_p}^{T_i(s)} = \frac{\omega_p}{T_i(s)}\frac{\partial T_i(s)}{\partial \omega_p} = -\frac{\left(\frac{1}{Q_p}\right) + 2\left(\frac{\omega_p}{s}\right)}{\left(\frac{s}{\omega_p}\right) + \left(\frac{\omega_p}{s}\right) + \left(\frac{1}{Q_p}\right)} \tag{4.130}$$

Introducing the normalized frequency $\Omega = \left(\frac{\omega}{\omega_p} - \frac{\omega_p}{\omega}\right)$, one can find that

$$\mathrm{Re}\,\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_\mathrm{p}}\right] = -\frac{\frac{1}{Q_\mathrm{p}}}{\Omega^2 + \left(\frac{1}{Q_\mathrm{p}}\right)^2} \tag{4.131}$$

$$\mathrm{Im}\,\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_\mathrm{p}}\right] = -\frac{\Omega}{\Omega^2 + \left(\frac{1}{Q_\mathrm{p}}\right)^2} \tag{4.132}$$

$$\mathrm{Re}\,\mathbf{S}^{T_i(s)}_{\omega_\mathrm{p}} = \frac{\Omega\sqrt{\Omega^2+4} - \Omega^2 - \left(\frac{1}{Q_\mathrm{p}}\right)^2}{\Omega^2 + \left(\frac{1}{Q_\mathrm{p}}\right)^2} \tag{4.133}$$

$$\mathrm{Im}\,\mathbf{S}^{T_i(s)}_{\omega_\mathrm{p}} = \frac{\left(\frac{1}{Q_\mathrm{p}}\right)\sqrt{\Omega^2+4}}{\Omega^2 + \left(\frac{1}{Q_\mathrm{p}}\right)^2} \tag{4.134}$$

Figure 4.4 shows the graphs of these four functions. They allow the following conclusions [4]. The functions reach high values in the vicinity of $\Omega = 0$, i.e., when $\omega \approx \omega_\mathrm{p}$. This means that in the filter

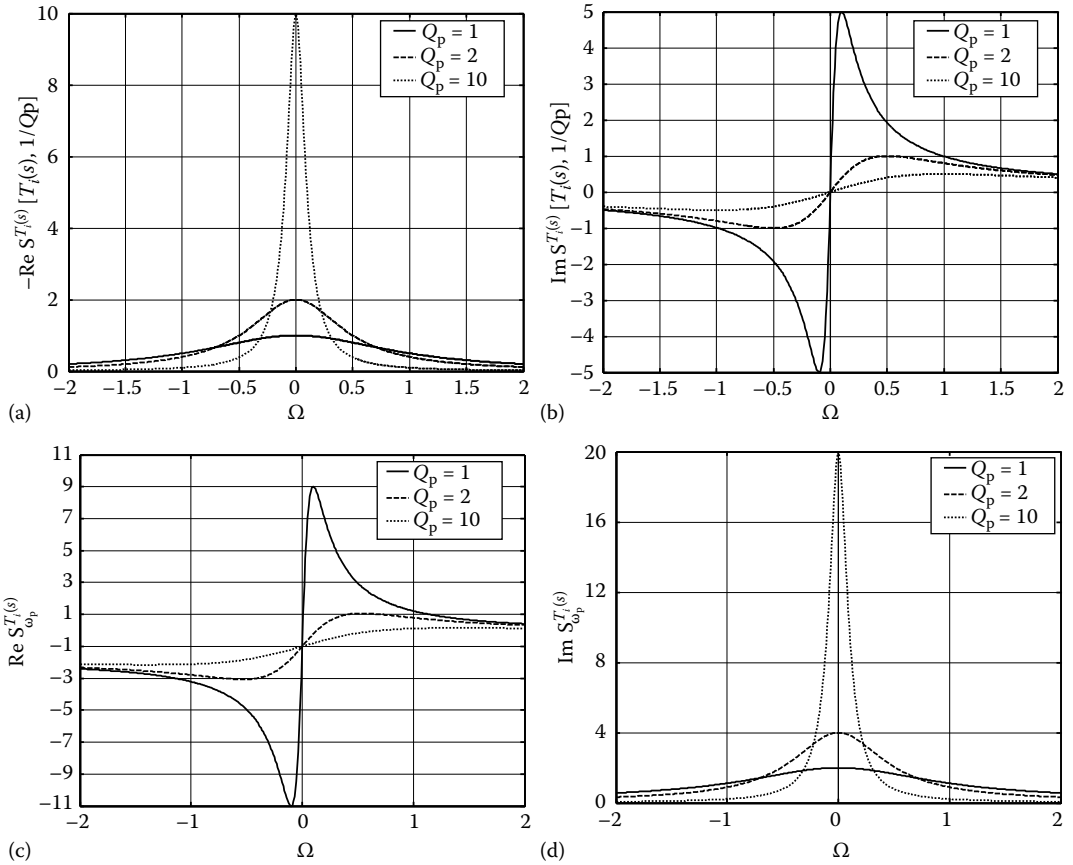


FIGURE 4.4 Stage sensitivities: (a) real and (b) imaginary parts of the *Q*-factor sensitivity; (c) real and (d) imaginary parts of the pole frequency sensitivity.

passband, especially if the poles and zeros are sufficiently divided (which is the condition of optimal cascading), one can neglect the contribution of zeros in modification of the transfer function $T_i(s)$. When Q_p becomes higher, this vicinity of $\omega \approx \omega_p$ with a rapid change of the sensitivity functions becomes relatively smaller in the case of $\mathrm{Re}\,\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_p}\right]$ (Figure 4.4a) and $\mathrm{Im}\,\mathbf{S}^{T_i(s)}_{\omega_p}$ (Figure 4.4d) and not so small in the case of $\mathrm{Im}\,\mathbf{S}^{T_i(s)}\left[T_i(s), \frac{1}{Q_p}\right]$ (Figure 4.4b) and $\mathrm{Re}\,\mathbf{S}^{T_i(s)}_{\omega_p}$ (Figure 4.4c). Normally, the second multipliers in the terms of the sum in Equation 4.128 are all real; this means that the function $\mathrm{Re}\,\mathbf{S}^{T_i(s)}_{\omega_p}$ is the most important one in estimation of the sensitivity to variations of passive elements. In many realization methods [1,3] one obtains that $\omega_p \propto (R_1R_2C_1C_2)^{-1/2}$. This implies that $\mathbf{S}^{\omega_p}_{R_1} = \mathbf{S}^{\omega_p}_{R_2} = \mathbf{S}^{\omega_p}_{C_1} = \mathbf{S}^{\omega_p}_{C_2} = -1/2$. Thus, finally, the function $\mathrm{Re}\,\mathbf{S}^{T_i(s)}_{\omega_p}$ (which can be called the main passive sensitivity term) will determine the maximum realizable Q_p for given tolerances of passive elements (or for the elements that are simulated as passive elements).

If a stage is realized using, for example, a state–space approach, it includes operational amplifiers. The stage is usually designed assuming ideal operational amplifiers, then the realization errors are analyzed considering that the operational amplifiers can be described by a model

$$A(s) = \frac{A_0}{1 + \left(\frac{s}{\omega_0}\right)} = \frac{\mathrm{GBW}}{s + \omega_0} \tag{4.135}$$

where

A_0 is the dc gain, is the amplifier bandwidth
$\mathrm{GBW} = A_0\omega_0$ is the gain-bandwidth product

If the stage transfer function is derived anew, with amplifiers described by the model equation (Equation 4.135), then $T_i(s)$ will no longer be a biquadratic. It will be a ratio of two higher degree polynomials, and the error analysis becomes very complicated [1]. To do an approximate analysis, one can pretend that the amplifier gain is simply a real constant A. Then, the transfer function $T_i(s)$ will preserve its biquadratic appearance, and the term $1/Q_p$ will be possible to represent as

$$\frac{1}{Q_p} = \frac{1}{Q} + \frac{k}{A} \tag{4.136}$$

The first term in Equation 4.136 is determined by the ratio of passive elements, the second term (k is the design constant) can be considered as an *absolute* change $\Delta(1/Q_p)$, which on the $j\omega$ axis becomes

$$\Delta\left(\frac{1}{Q_p}\right) \approx \frac{k}{A(j\omega)} = \frac{1}{A_0} + j\frac{k\omega}{\mathrm{GBW}} \approx j\frac{k\omega}{\mathrm{GBW}} \tag{4.137}$$

Then, in calculation of $\delta|T_i| = \mathrm{Re}\,\mathbf{S}^{T_i(s)}_K$, the function $\mathrm{Im}\,\mathbf{S}^{T_i(s)}_{\omega_p}$ becomes important (it can be called the main active sensitivity term) and the product $\left[\mathrm{Im}\,\mathbf{S}^{T_i(s)}_{\omega_p}\left(\frac{k\omega}{\mathrm{GBW}}\right)\right]$ allows evaluation of the limitations on the Q-factor caused by the stage operational amplifiers.

The relationships given next are useful for any arbitrary realization, but pertain more to the cascade realization, where one can better control the pole's location when the circuit parameters are changing. If the filter transfer function is represented as

$$T(s) = \frac{a_m\Pi_{i=1}^{m}(s - z_i)}{d_n\Pi_{i=1}^{n}(s - p_i)} \tag{4.138}$$

then taking the logarithm of Equation 4.138 and its derivatives gives

$$\frac{\mathrm{d}T(s)}{T(s)} = \frac{\mathrm{d}(a_m/d_n)}{a_m/d_n} - \sum_{i=1}^{m} \frac{\mathrm{d}z_i}{s - z_i} + \sum_{i=1}^{n} \frac{\mathrm{d}p_i}{s - p_i} \tag{4.139}$$

Multiplying both sides of Equation 4.139 by x and expressing the differentials via partial derivatives one obtains

$$\mathbf{S}_x^{T(s)} = \mathbf{S}_x^{a_m/d_n} - \sum_{i=1}^{m} \frac{\mathbf{S}_x(z_i)}{s - z_i} + \sum_{i=1}^{n} \frac{\mathbf{S}_x(p_i)}{s - p_i} \tag{4.140}$$

In the vicinity of the pole $p_i = -\sigma_i + j\omega_i$, the sensitivity is determined by the term $\mathbf{S}_x(p_i)/(s - p_i)$. Besides $\mathbf{S}_x(p_i) = -\mathbf{S}_x(\sigma_i) + j\mathbf{S}_x(\omega_i)$ and on the $j\omega$ axis in this region one has

$$\frac{\mathbf{S}_x(p_i)}{j\omega - p_i} = -\frac{\mathbf{S}_x(\sigma_i)\sigma_i - \mathbf{S}_x(\omega_i)(\omega - \omega_i)}{\sigma_i^2 + (\omega - \omega_i)^2} + j\frac{\mathbf{S}_x(\omega_i)\sigma_i + \mathbf{S}_x(\sigma_i)(\omega - \omega_i)}{\sigma_i^2 + (\omega - \omega_i)^2} \tag{4.141}$$

Hence, when $\omega = \omega_i$

$$\mathbf{S}_x^{|T(j\omega)|} \approx -\frac{\mathbf{S}_x(\sigma_i)\sigma_i - \mathbf{S}_x(\omega_i)(\omega - \omega_i)}{\sigma_i^2 + (\omega - \omega_i)^2} \tag{4.142}$$

and

$$\mathbf{S}_x[\arg T(j\omega)] \approx \frac{\mathbf{S}_x(\omega_i)\sigma_i + \mathbf{S}_x(\sigma_i)(\omega - \omega_i)}{\sigma_i^2 + (\omega - \omega_i)^2} \tag{4.143}$$

Usually Equations 4.142 and 4.143 are considered at the point $\omega - \omega_i$, where $\mathbf{S}_x^{|T(j\omega)|} = -\mathbf{S}_x(\sigma_i)/\sigma_i$ and $\mathbf{S}_x[\arg T(j\omega)] = \mathbf{S}_x(\omega_i)/\sigma_i$. The frequent conclusion that follows is that the pole's movement toward the $j\omega$ axis is more dangerous (it introduces transfer function magnitude change) than the movement parallel to the $j\omega$ axis. But it is not difficult to see that in the immediate vicinity of this point, at $\omega = \omega_i \pm \sigma_i$, one has $\mathbf{S}_x^{|T(j\omega)|} = [-\mathbf{S}_x(\sigma_i) \pm \mathbf{S}_x(\omega_i)]/(2\sigma_i)$ and $\mathbf{S}_x[\arg T(j\omega)] = [\mathbf{S}_x(\omega_i) \pm j\mathbf{S}_x(\sigma_i)]/(2\sigma_i)$; i.e., one has to reduce both components of the pole movement. If care is taken to get $\mathbf{S}_x(\sigma_i) = 0$, then, indeed, $\mathbf{S}_x^{|T(j\omega)|} = 0$ at $\omega = \omega_i$, but at $\omega = \omega_i + \sigma_i$ (closer to the edge of the passband) $\mathbf{S}_x^{|T(j\omega)|} = [\mathbf{S}_x(\omega_i)/\sigma_i]$ and this can result in an essential $\delta|T(j\omega)|$.

Finally, some additional relationships between different sensitivities can be obtained from the definition of $Q_\mathrm{p} = \omega_\mathrm{p}/(2\sigma_i) = \left[\sqrt{\sigma_i^2 + \omega_i^2}\right]/(2\sigma_i) \approx \omega_i/(2\sigma_i)$. One can find that

$$\mathbf{S}_x(\omega_\mathrm{p}) = \frac{\mathbf{S}_x(\sigma_i)}{2Q_\mathrm{p}} + \sqrt{1 - \left(\frac{1}{4Q_\mathrm{p}^2}\right)}\mathbf{S}_x(\omega_i) \tag{4.144}$$

and

$$\mathbf{S}_x(Q_\mathrm{p}) = \frac{\sqrt{4Q_\mathrm{p}^2 - 1}\mathbf{S}_x(\omega_i) - (4Q_\mathrm{p}^2 - 1)\mathbf{S}_x(\sigma_i)}{\omega_\mathrm{p}} \tag{4.145}$$

for semirelative sensitivities. From this basic definition of the Q-factor, one can also derive the relationships

$$\mathbf{S}_x^{Q_p} = \frac{\mathbf{S}_x^{\sigma_i}}{4Q_p^2} + \sqrt{1 - \left(\frac{1}{4Q_p^2}\right)}\,\mathbf{S}_x^{\omega_i} \tag{4.146}$$

and

$$\mathbf{S}_x^{Q_p} \approx \mathbf{S}_x^{\omega_i} - \mathbf{S}_x^{\sigma_i} \tag{4.147}$$

involving relative sensitivities. Another group of results can be obtained considering relative sensitivity of the pole $p_i = -\sigma_i + j\omega_i$. For example, one can find that

$$\mathbf{S}_x^{Q_p} = -\sqrt{4Q_p^2 - 1}\,\mathrm{Im}\,\mathbf{S}_x^{p_i} = \frac{x}{\omega_p^2}\left(\frac{\omega_i}{\sigma_i}\right)\left(\sigma_i\frac{\partial\omega_i}{\partial x} - \omega_i\frac{\partial\sigma_i}{\partial x}\right) \tag{4.148}$$

If $\dfrac{\partial\omega_i}{\partial x} = 0$ and $\dfrac{\partial\sigma_i}{\partial x} =$ constant, then

$$\mathbf{S}_x^{Q_p} \approx kQ_p \tag{4.149}$$

which shows that in this case, $\mathbf{S}_x^{Q_p}$ increases proportionally to the Q-factor independently of the cause of this high sensitivity.

4.14 Simulation of Doubly Terminated Matched Lossless Filters

By cascading the first- and second-order sections (occasionally a third-order section is realized in odd-order filters instead of the cascade connection of a first- and a second-order section) any high-order transfer function $T(s)$ can be realized. In practice, however, the resulting circuit is difficult to fabricate for high-order and/or highly selective filters. The transfer function of such filters usually contains a pair of complex–conjugate poles very close to the $j\omega$ axis. The sensitivity of this section that realizes high-Q poles is high and the element tolerances for this section can be very tight. The section can be unacceptable for fabrication.

For filters that have such high-Q transfer function poles, other design techniques are often used. The most successful and widely used of these alternative strategies are based on simulating the low-sensitivity transfer function of a doubly-terminated lossless (reactance) two-port.

Assume that the two-port shown in Figure 4.5 is lossless and the transfer function $T(s) = V_2(s)/E(s)$ is realized. Considering power relations, one can show [12] that for steady-state sinusoidal operation the equation

$$|\rho(j\omega)|^2 + \frac{4R_1}{R_2}|T(j\omega)|^2 = 1 \tag{4.150}$$

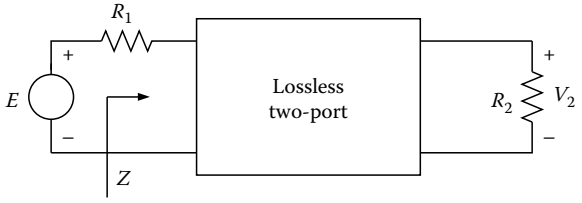


FIGURE 4.5 Lossless two-port with resistive loadings.

is valid for this circuit. Here, $\rho(j\omega) = [R_1 - Z(j\omega)]/[R_1 + Z(j\omega)]$ is the reflection coefficient and $Z(j\omega)$ is the input impedance of the loaded two-port. In many cases, the filter requirements are formulated for the transducer function:

$$H(s) = \sqrt{\frac{R_2}{4R_1}}\frac{E(s)}{V_2(s)} = \sqrt{\frac{R_2}{4R_1}}\frac{1}{T(s)} \tag{4.151}$$

For this function

$$\ln H(j\omega) = \alpha(\omega) + j\varphi(\omega) = -\ln|T(j\omega)| + \ln[\sqrt{(R_2/4R_1)}] - j\arg T(j\omega) \tag{4.152}$$

Here, $\alpha(\omega)$ is attenuation (it is different from the previously used only by the value of $\ln[\sqrt{(R_2/4R_1)}]$) and $\varphi(\omega) = -\beta(\omega) = -\arg T(j\omega)$ is phase. The impedance $Z(j\omega)$ satisfies the condition $\mathrm{Re}\, Z(j\omega) \geq 0$ (as for any passive circuit input impedance), which means that $|\rho(j\omega)|^2 < 1$. Then, as it follows from Equation 4.150 $|H(j\omega)|^2 \geq 1$ and $\alpha(\omega) \geq 0$.

When a filter is designed using $\alpha(\omega)$, the attenuation is optimized so that it is zero in one or more passband points (Figure 4.3 shows, for example, the attenuation characteristics with two zeros in the passband). Then the attenuation partial derivative with respect to the value of each of the two-port elements is equal to zero at the attenuation zeros. Indeed, let x_i be any element of the two-port and $\partial\alpha/\partial x_i$ be the partial derivative of $\alpha(\omega, x_i)$ with respect to that element. Suppose that x_i does not have its nominal value and differs from it by a small Δx_i variation. Expanding $\alpha(\omega, x_i)$ in the Taylor series one obtains that

$$\alpha(\omega, x_i + \Delta x_i) \cong \alpha(\omega, x_i) + \Delta x_i \frac{\partial\alpha(\omega, x_i)}{\partial x_i} \tag{4.153}$$

If ω_k is an attenuation zero, then $\alpha(\omega_k, x_i) = 0$; but, as was mentioned before, $\alpha(\omega) \geq 0$, and from Equation 4.153, one obtains that at the point $\omega = \omega_k$ one has

$$\Delta x_i \frac{\partial\alpha(\omega_k, x_i)}{\partial x_i} \geq 0 \tag{4.154}$$

Now, the variation Δx_i was of unspecified sign. Therefore, Equation 4.154 can only be satisfied with equality sign, which means that $\frac{\partial\alpha(\omega_k, x_i)}{\partial x_i} = 0$.

This result is called the *Fettweis-Orchard theorem* [4] and it explains why the preference is always given to the filter realized as a nondissipative two-port between resistive terminations or to the simulation of such a filter if the filter should be realized as an active circuit. First, considering the real parts of Equation 4.152, one obtains that

$$\frac{\partial\alpha(\omega, x_i)}{\partial x_i} = \mathbf{S}_{x_i}[\alpha(\omega)] = -\frac{x_i}{|T(j\omega)|}\frac{\partial|T(j\omega)|}{\partial x_i} = -\mathbf{S}_{x_i}^{|T(j\omega)|} \tag{4.155}$$

when x_i is any element inside the two-port. Hence, the points where $\alpha(\omega) = 0$ and, simultaneously, $\frac{\partial\alpha(\omega, x_i)}{\partial x_i} = 0$ are the points of zero sensitivity not only for attenuation but for transfer function magnitude as well (as a result of Equation 4.155 and this discussion, one cannot use at these points the relationships $d|T(j\omega)| \approx \mathbf{S}_{x_i}^{|T(j\omega)|}\delta x_i$; the relative change $\delta|T(j\omega)|$ is always negative and different from zero with δx_i of unspecified sign). Moreover, if α is small, $\partial\alpha/\partial x_i$ will also remain small [13], which means that the sensitivities are small in the whole passband. If $x_i = R_1$ or $x_i = R_2$, one obtains from Equation 4.152 that

$$R_1 \frac{\partial \alpha(\omega, R_1)}{\partial R_1} = -\mathbf{S}_{R_1}^{|T(j\omega)|} - \frac{1}{2} \tag{4.156}$$

and

$$R_2 \frac{\partial \alpha(\omega, R_2)}{\partial R_2} = -\mathbf{S}_{R_2}^{|T(j\omega)|} + \frac{1}{2} \tag{4.157}$$

The derivatives $\frac{\partial \alpha}{\partial R_i} (i = 1, 2)$ are also zero at the points where $\alpha = 0$ and they are small when α remains small [13]. This means that in the passband $\mathbf{S}_{R_1}^{|T(j\omega)|} \approx -1/2$ and $\mathbf{S}_{R_2}^{|T(j\omega)|} \approx 1/2$. Thus, $|T(j\omega)|$ will share the zero sensitivity of α with respect to all the elements inside the two-port, but due to the terms $\pm 1/2$ in Equations 4.156 and 4.157 a change either in R_1 or R_2 will produce a frequency-independent shift (which can usually be tolerated) in $|T(j\omega)|$ in addition to the small effects proportional to $\frac{\partial \alpha}{\partial R_i}$. This is the basis of the low sensitivity of conventional *LC*-ladder filters and of those active, switched-capacitor, or digital filters that are based on *LC* filter model. This is valid with the condition that the transfer functions of the active filter and *LC* prototype are the same and the parameters of the two filters enter their respective transfer functions the same way.

The Fettweis-Orchard theorem explains why the filtering characteristics sought are those with the maximum number of attenuation zeros (for a given order of the transfer function $T(s)$). It also helps to understand why it is difficult to design a filter that simultaneously meets the requirements of α and $\varphi(\omega)$ (or to $|T(j\omega)|$ and $\beta(\omega)$); the degrees of freedom used for optimizing $\varphi(\omega)$ will not be available to attain the maximum number of attenuation zeros. It also explains why a cascade realization is more sensitive than the realization based on *LC* lossless model. Indeed, assume that, say, one of the characteristics of Figure 4.3 is realized by two cascaded sections (with the attenuation of each section shown by the dash-and-dotted line) with each actual section realized in doubly terminated matched lossless form. Each such section of the cascaded filter will be matched at one frequency and the sensitivities to the elements that are in unmatched sections will be different from zero. In addition, the attenuation ripple in each section is usually much larger than the total ripple, and the derivative $\frac{\partial \alpha}{\partial x_i}$, which is, in the first approximation, proportional to the attenuation ripple, will not be small. Indeed, practice shows [4] that there is, in fact, a substantial increase in sensitivity in the factored realization.

4.15 Sensitivity of Active *RC* Filters

The required component tolerances are very important factors determining the cost of filters. They are especially important with integrated realizations (where the tolerances are usually higher than in discrete technology). Also, the active filter realizations commonly require tighter tolerances than *LC* realizations. Yet two classes of active *RC* filters have tolerances comparable with those of passive *LC* filters. These are analog-computer and gyrator filters that simulate doubly terminated passive *LC* filters. The tolerance comparison [4] shows the tolerance advantages (sometimes by an order of magnitude) of the doubly terminated lossless structure as compared to any cascade realization. These are the only methods that are now used [14] for high-order high-*Q* sharp cutoff filters with tight tolerances. For less demanding requirements, cascaded realizations could be used. The main advantages that are put forth in this case are the ease of design and simplicity of tuning. But even here the tolerance comparison [4] shows that the stages have better tolerances if they are realized using gyrators and computer simulation methods.

4.16 Errors in Sensitivity Comparisons

In conclusion we briefly outline some common errors in sensitivity comparison. More detailed treatment can be found in Ref. [4].

1. Calculating the wrong sensitivities. The calculated sensitivities should have as close a relation to the filter specification as possible. In general, for a filter specified in the frequency domain, the sensitivities of amplitude and phase should be calculated along the $j\omega$ axis. Sensitivities of poles, zeros, Q's, resonant frequencies, etc. should be carefully interpreted in the context of their connection with amplitude and phase sensitivities.
2. Sensitivities of optimized designs. The optimization should use a criterion as closely related as possible to filter specifications. The use of a criterion that is not closely related to the filter specifications (e.g., pole sensitivity) can lead to valid conclusions if the filters being compared differ by an order of magnitude in sensitivity [4]. A sensitivity comparison is valid only if all the circuits have been optimized using the criterion on which they will be compared. The optimized circuit should not be compared with a nonoptimized one. Another error is to optimize one part of the transfer function (usually the denominator) and forgetting about the modifying effect of the numerator.
3. Comparing the incomparable. A frequent error occurs when comparing sensitivities with respect to different types of elements. In general, different types of elements can be realized with different tolerances, and the comparison is valid only if sensitivities are weighted proportionally. Besides, there are basic differences in variability between circuit parameters with physical dimensions and those without. The latter are often determined in the circuit as the ratio of dimensional quantities (as a result, the tolerance of the ratio will be about double the tolerances of the two-dimensional quantities determining them). In integrated technologies the dimensioned quantities usually have worse tolerances but better matching and tracking ability, especially with temperature. Hence, any conclusion involving sensitivities to different types of components, in addition, is technologically dependent.
4. Correlations between component values. The correlations between components are neglected when they are essential (this is usually done for simplification of the statistical analysis). From the other side, an unwarranted correlation is introduced when it does not exist. This is a frequent case where the realization involves cancellation of terms that are equal when the elements have their nominal values (e.g., a cancellation of a pole of one section of a filter by a zero of another section, cancellation of a positive conductance by a negative conductance).
5. Incomplete analysis. Very often, only sensitivities to variations of a single component (usually an amplifier gain) are considered. This is satisfactory only if it is the most critical component, which is seldom the case. Another form of incomplete analysis is to calculate only one coordinate of a complex sensitivity measure ($\mathbf{S}_x^Q$ is calculated while $\mathbf{S}_x^{\omega_0}$ is ignored). Also, frequency-dependent sensitivities are calculated and compared at one discrete frequency instead of being calculated in frequency intervals.
6. First-order differential sensitivities are the most commonly calculated. But the fact is that $\frac{\partial y}{\partial x} = 0$ implies that the variation of y with x is quadratic at the point considered. A consequence of this is that zero sensitivities do not imply infinitely wide tolerances for the components in question. Similarly, infinite sensitivities do not imply infinitely narrow tolerances. Infinite values arise if the nominal value of y is zero, and the finite variations of x will almost always give finite variations of y.

References

1. L.P. Huelsman and P.E. Allen, *Introduction to the Theory and Design of Active Filters*, New York: McGraw-Hill, 1980.
2. K. Geher, *Theory of Network Tolerances*, Budapest, Hungary: Akademiai Kiado, 1971.
3. W.E. Heinlein and W.H. Holmes, *Active Filters for Integrated Circuits*, London: Prentice Hall, 1974.
4. M. Hasler and J. Neirynck, *Electric Filters*, Dedham, MA: Artech House, 1986.
5. S.K. Mitra, Ed., *Active Inductorless Filters*, New York: IEEE, 1971.

6. L.P. Huelsman, Ed., Active RC-filters: Theory and application, in *Benchmark Papers in Electrical Engineering and Computer Science*, vol. 15, Stroudsburg, PA: Dowden, Hutchinson and Ross, Inc., 1976.
7. A.F. Schwarz, *Computer-Aided Design of Microelectronic Circuits and Systems*, vol. 1. Orlando, FL: Academic, 1987.
8. M.L. Blostein, Some bounds on the sensitivity in RLC networks, in *Proc. 1st Allerton Conf. Circuits Syst. Theory*, Monticello, IL, November 15–17, 1963, pp. 488–501.
9. R.K. Brayton and R. Spence, Sensitivity and optimization, in *Computer-Aided Design of Electronic Circuits*, vol. 2, Amsterdam: Elsevier, 1980.
10. C. Desoer and E.S. Kuh, *Basic Circuit Theory*, New York: McGraw-Hill, 1969.
11. M. Biey and A. Premoli, *Cauer and MCPER Functions for Low-Q Filter Design*, St. Saphorin, Switzerland: Georgi, 1980.
12. N. Balabanian and T.A. Bickart, *Electrical Network Theory*, New York: Wiley, 1969.
13. H.J. Orchard, Loss sensitivities in singly and doubly terminated filters, *IEEE Trans. Circuits Syst.*, CAS-26, 293–297, 1979.
14. B. Nauta, *Analog CMOS Filters for Very High Frequencies*, Boston, MA: Kluwer Academic, 1993.

5

Passive Immittances and Positive-Real Functions

Wai-Kai Chen
University of Illinois at Chicago

References **5**-8

In this chapter on passive filters, we deal with the design of one-port networks composed exclusively of passive elements such as resistors R, inductors L, capacitors C, and coupled inductors M. The one-ports are specified by their driving-point *immittances*, *impedances*, or *admittances*. Our basic problem is that given an immittance function, is it possible to find a one-port composed only of R, L, C, and M elements called the *RLCM one-port network* that realizes the given immittance function? This is known as the realizability problem, and its complete solution was first given by Brune [1].

Consider a linear RLCM one-port network of Figure 5.1 excited by a voltage source $V_1(s)$. For our purposes, we assume that there are b branches and the branch corresponding to the voltage source $V_1(s)$ is numbered branch 1 and all other branches are numbered from 2 to b. The Laplace transformed Kirchhoff current law equation can be written as

$$\mathbf{A}\mathbf{I}(s) = \mathbf{0} \tag{5.1}$$

where

A is the basis incidence matrix
I(s) is the branch-current vector of the network

If $\mathbf{V}_n(s)$ is the nodal voltage vector, then the branch-voltage vector $\mathbf{V}(s)$ can be expressed in terms of $\mathbf{V}_n(s)$ by

$$\mathbf{V}(s) = \mathbf{A}'\mathbf{V}_n(s) \tag{5.2}$$

where the prime denotes the matrix transpose. Taking the complex conjugate of Equation 5.1 in conjunction with Equation 5.2 gives

$$\mathbf{V}'(s)\bar{\mathbf{I}}(s) = \mathbf{V}_n'(s)\mathbf{A}\bar{\mathbf{I}}(s) = \mathbf{V}_n'(s)\mathbf{0} = 0 \tag{5.3}$$

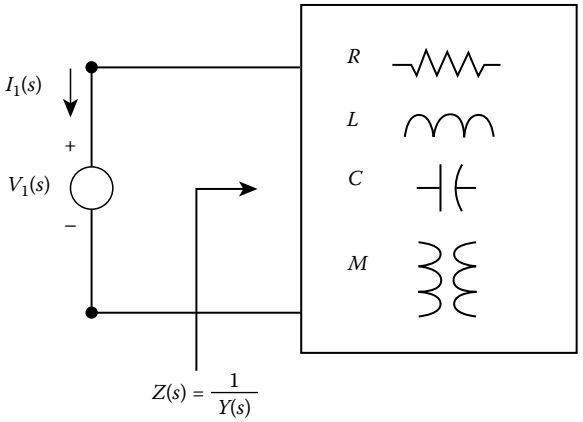


FIGURE 5.1 General linear *RLCM* one-port network.

or

$$\sum_{k=1}^{b} V_k(s)\bar{I}_k(s) = 0 \tag{5.4}$$

where $V_k(s)$ and $I_k(s)$ are the branch voltages and the branch currents, respectively.

From Figure 5.1, the driving-point impedance of the one-port is defined to be the ratio of $V_1(s)$ to $-I_1(s)$, or

$$Z(s) \equiv \frac{V_1(s)}{-I_1(s)} = \frac{V_1(s)\bar{I}_1(s)}{-I_1(s)\bar{I}_1(s)} = -\frac{V_1(s)\bar{I}_1(s)}{|I_1(s)|^2} \tag{5.5}$$

Equation 5.4 can be rewritten as

$$-V_1(s)\bar{I}_1(s) = \sum_{k=2}^{b} V_k(s)\bar{I}_k(s) \tag{5.6}$$

Substituting this in Equation 5.5 yields

$$Z(s) = \frac{1}{|I_1(s)|^2}\sum_{k=2}^{b} V_k(s)\bar{I}_k(s) \tag{5.7}$$

Likewise, the dual relation of the input admittance

$$Y(s) \equiv \frac{-I_1(s)}{V_1(s)} = \frac{1}{|V_1(s)|^2}\sum_{k=2}^{b} \overline{V}_k(s)I_k(s) \tag{5.8}$$

holds.

We know consider individual types of elements inside the one-port. For a resistive branch k of resistance R_k, we have

$$V_k(s) = R_k I_k(s) \tag{5.9}$$

For a capacitive branch of capacitance C,

$$V_k(s) = \frac{1}{sC_k} I_k(s) \tag{5.10}$$

Finally, for an inductive branch of self-inductance L_k and mutual inductances M_{kj},

$$V_k(s) = sL_kI_k(s) + \sum_{\text{all } j, j\neq k} sM_{kj}I_j(s) \tag{5.11}$$

Substituting these in Equation 5.7 and grouping the summation as sums of all resistors R, all capacitors C, and all inductors LM, we obtain

$$\begin{aligned} Z(s) &= \frac{1}{|I_1(s)|^2}\left[\sum_R R_k|I_k(s)|^2 + \sum_C \frac{1}{sC_k}|I_k(s)|^2 + \sum_{LM}\left(sL_k|I_k(s)|^2 + \sum_{\text{all } j, j\neq k} sM_{kj}I_j(s)\bar{I}_k(s)\right)\right] \\ &= \frac{1}{|I_1(s)|^2}\left[F_0(s) + \frac{1}{s}V_0(s) + sM_0(s)\right] \end{aligned} \tag{5.12}$$

where

$$F_0(s) \equiv \sum_R R_k|I_k(s)|^2 \geq 0 \tag{5.13a}$$

$$V_0(s) \equiv \sum_C \frac{1}{C_k}|I_k(s)|^2 \geq 0 \tag{5.13b}$$

$$M_0(s) \equiv \sum_{LM}\left(L_k|I_k(s)|^2 + \sum_{\text{all } j, j\neq k} M_{kj}I_j(s)\bar{I}_k(s)\right) \tag{5.13c}$$

These quantities are closely related to the average power and stored energies of the one-port under steady-state sinusoidal conditions. The average power dissipated in the resistors is

$$P_{\text{ave}} = \frac{1}{2}\sum_R R_k|I_k(j\omega)|^2 = \frac{1}{2}F_0(j\omega) \tag{5.14}$$

showing that $F_0(j\omega)$ represents twice the average power dissipated in the resistors of the one-port. The average electric energy stored in the capacitors is

$$E_C = \frac{1}{4\omega^2}\sum_C \frac{1}{C_k}|I_k(j\omega)|^2 = \frac{1}{4\omega^2}V_0(j\omega) \tag{5.15}$$

Thus, $V_0(j\omega)$ denotes $4\omega^2$ times the average electric energy stored in the capacitors. Similarly, the average magnetic energy stored in the inductors is

$$E_M = \frac{1}{4}\sum_{LM}\left[L_k|I_k(j\omega)|^2 + \sum_{\text{all } q, q\neq k} M_{kq}I_q(j\omega)\bar{I}_k(j\omega)\right] = \frac{1}{4}M_0(j\omega) \tag{5.16}$$

indicating that $M_0(j\omega)$ represents four times the average magnetic energy stored in the inductors.

Therefore, all the three quantities $F_0(j\omega)$, $V_0(j\omega)$, and $M_0(j\omega)$ are real and nonnegative, and Equation 5.12 can be rewritten as

$$Z(s) = \frac{1}{|I_1(s)|^2}\left(F_0 + \frac{1}{s}V_0 + sM_0\right) \tag{5.17}$$

Likewise, the dual result for $Y(s)$ is found to be

$$Y(s) = \frac{1}{|V_1(s)|^2}\left(F_0 + \frac{1}{\bar{s}}V_0 + \bar{s}M_0\right) \tag{5.18}$$

Now, we set $s = \sigma + j\omega$ and compute the real part and imaginary part of $Z(s)$ and obtain

$$\text{Re } Z(s) = \frac{1}{|I_1(s)|^2}\left(F_0 + \frac{\sigma}{\sigma^2+\omega^2}V_0 + \sigma M_0\right) \tag{5.19}$$

$$\text{Im } Z(s) = \frac{\omega}{|I_1(s)|^2}\left(M_0 - \frac{1}{\sigma^2+\omega^2}V_0\right) \tag{5.20}$$

where

Re stands for ''real part of''
Im for ''imaginary part of''

These equations are valid irrespective of the value of s, except at the zeros of $I_1(s)$. They are extremely important in that many analytic properties of passive impedances can be obtained from them. The following is one of such consequences:

THEOREM 5.1

If $Z(s)$ is the driving-point impedance of a linear, passive, lumped, reciprocal, and time-invariant one-port network N, then

1. *Whenever* $\sigma \geq 0$, Re $Z(s) \geq 0$.
2. *If N contains no resistors, then*
   $\sigma > 0$ *implies* Re $Z(s) > 0$
   $\sigma = 0$ *implies* Re $Z(s) = 0$
   $\sigma < 0$ *implies* Re $Z(s) < 0$
3. *If N contains no capacitors, then*
   $\omega > 0$ *implies* Im $Z(s) > 0$
   $\omega = 0$ *implies* Im $Z(s) = 0$
   $\omega < 0$ *implies* Im $Z(s) < 0$
4. *If N contains no self- and mutual-inductors, then*
   $\omega > 0$ *implies* Im $Z(s) < 0$
   $\omega = 0$ *implies* Im $Z(s) = 0$
   $\omega < 0$ *implies* Im $Z(s) > 0$

Similar results can be stated for the admittance function $Y(s)$ simply by replacing $Z(s)$, Re $Z(s)$, *and* Im $Z(s)$ *by $Y(s)$,* Re $Y(s)$, *and* Im $Y(s)$, *respectively.*

The theorem states that the driving-point impedance $Z(s)$ of a passive LMC, RLM, or RC one-port network maps different regions of the complex-frequency s-plane into various regions of the Z-plane. Now, we assert that the driving-point immittance of a passive one-port is a positive-real function, and every positive-real function can be realized as the input immittance of an RLCM one-port network.

Definition 5.1: *Positive-real function. A positive-real function $F(s)$,* abbreviated as a PR function, is an analytic function of the complex variable $s = \sigma + j\omega$ satisfying the following three conditions:

1. $F(s)$ is analytic in the open RHS (*right-half* of the s-plane), i.e., $\sigma > 0$.
2. $F(\overline{s}) = \overline{F}(s)$ for all s in the open RHS.
3. Re $F(s) \geq 0$ whenever Re $s \geq 0$.

The concept of a positive-real function, as well as many of its properties, is credited to Otto Brune [1]. Our objective is to show that positive realness is a necessary and sufficient condition for a passive one-port immittance. The above definition holds for both rational and transcendental functions. A rational function is defined as a ratio of two polynomials. Network functions associated with any linear lumped system, with which we deal exclusively in this section, are rational. In the case of rational functions, not all three conditions in the definition are independent. For example, the analyticity requirement is implied by the other two. The second condition is equivalent to stating the $F(s)$ is real when s is real, and for a rational $F(s)$ it is always satisfied if all the coefficients of the polynomial are real.

Some important properties of a positive-real function can be stated as follows:

1. If $F_1(s)$ and $F_2(s)$ are positive real, so is $F_1[F_2(s)]$.
2. If $F(s)$ is positive real, so are $1/F(s)$ and $F(1/s)$.
3. A positive-real function is devoid of poles and zeros in the open RHS.
4. If a positive-real function has any poles or zeros on the $j\omega$-axis (0 and ∞ included), such poles and zeros must be simple. At a simple pole on the $j\omega$-axis, the residue is real positive.

Property 1 states that a positive-real function of a positive-real function is itself positive real, and property 2 shows that the reciprocal of a positive-real function is positive real. The real significance of the positive-real functions is its use in the characterization of the passive one-port immittances. This characterization is one of the most penetrating results in network theory, and is stated as

THEOREM 5.2

A real rational function is the driving-point immittance of a linear, passive, lumped, reciprocal, and time-invariant one-port network if and only if it is positive real.

The necessity of the theorem follows directly from Equation 5.19. The sufficiency was first established by Brune in 1930 by showing that any given positive-real rational function can be realized as the input immittance of a passive one-port network using only the passive elements such as resistors, capacitors, and self- and mutual inductors. A formal constructive proof will be presented in the following section.

Example 5.1

Consider the passive one-port of Figure 5.2, the driving-point impedance of which is found to be

$$Z(s) = \frac{3s^2 + s + 2}{2s^2 + s + 3} \tag{5.21}$$

To verify that the function $Z(s)$ is positive real, we compute its real part by substituting $s = \sigma + j\omega$ and obtain

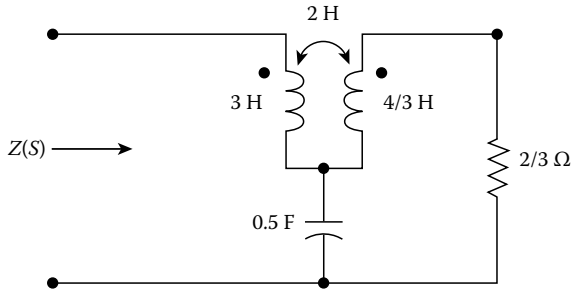


FIGURE 5.2 A passive one-port network.

$$\begin{aligned}\text{Re } Z(s) = \text{Re } Z(\sigma + j\omega) &= \text{Re}\frac{3(\sigma + j\omega)^2 + (\sigma + j\omega) + 2}{2(\sigma + j\omega)^2 + (\sigma + j\omega) + 3} \\ &= \frac{6(\omega^2 - 1)^2 + (12\omega^2\sigma + 5\omega^2 + 6\sigma^3 + 5\sigma^2 + 14\sigma + 5)\sigma}{(2\sigma^2 - 2\omega^2 + \sigma + 3)^2 + \omega^2(4\sigma + 1)^2} \geq 0, \quad \sigma \geq 0 \end{aligned} \tag{5.22}$$

This, in conjunction with the facts that $Z(s)$ is analytic in the open RHS and that all the coefficients of $Z(s)$ are real, shows that $Z(s)$ is positive real.

Observe that if the function $Z(s)$ is of high order, the task of ascertaining its positive realness is difficult if condition 3 of Definition 5.1 is employed for checking. Hence, it is desirable to have alternate but much simpler conditions for testing. For this reason, we introduce the following equivalent conditions that are relatively easy to apply:

THEOREM 5.3

A rational function $F(s)$ is positive real if and only if the following conditions are satisfied:

1. *$F(s)$ is real when s is real.*
2. *$F(s)$ has no poles in the open RHS.*
3. *Poles of $F(s)$ on the $j\omega$-axis, if they exist, are simple, and residues evaluated at these poles are real and positive.*
4. *Re $F(j\omega) \geq 0$ for all ω, except at the poles.*

PROOF: From the definition of a PR function, we see immediately that all the conditions are necessary. To prove sufficiency, we expand $F(s)$ in a partial fraction as

$$\begin{aligned} F(s) &= \left[k_\infty s + \frac{k_0}{s} + \sum_x \left(\frac{k_x}{s + j\omega_x} + \frac{k_x}{s - j\omega_x}\right)\right] + F_1(s) \\ &= \left(k_\infty s + \frac{k_0}{s} + \sum_x \frac{2k_x s}{s^2 + \omega_x^2}\right) + F_1(s) \end{aligned} \tag{5.23}$$

where k_∞, k_0, and k_x are residues evaluated at the $j\omega$-axis poles $j\infty$, 0, and $\pm j\omega_x$, respectively, and are real and positive. $F_1(s)$ is the function formed by the terms corresponding to the open LHS (left-half of the s-plane) poles of $F(s)$, and therefore is analytic in the RHS and the entire $j\omega$-axis including the point at infinity. For such a function, the minimum value of the real part throughout the region where the function is analytic lies on the boundary, namely, the $j\omega$-axis. (See, for example, Churchill [2]. This shows

that the minimum value of Re $F_1(s)$ for all Re $s \geq 0$ occurs on the $j\omega$-axis; but according to Equation 5.23, this value is nonnegative:

$$\text{Re } F_1(j\omega) = \text{Re } F(j\omega) \geq 0 \tag{5.24}$$

Thus, the real part of $F_1(s)$ is nonnegative everywhere in the closed RHS or

$$\text{Re } F_1(s) \geq 0 \quad \text{for Re } s \geq 0 \tag{5.25}$$

This, together with the fact that $F_1(s)$ is real whenever s is real, shows that $F_1(s)$ is positive real.

Since each term inside the parentheses of Equation 5.23 is positive real, and since the sum of two or more positive-real functions is positive real, $F(s)$ is positive real. This completes the proof of the theorem.

In testing for positive realness, we may eliminate some functions from consideration by inspection because they violate certain simple necessary conditions. For example, a function cannot be PR if it has a pole or zero in the open RHS. Another simple test is that the highest powers of s in numerator and denominator not differ by more than unity, because a PR function can have at most a simple pole or zero at the origin or infinity, both of which lie on the $j\omega$-axis.

A Hurwitz polynomial is a polynomial devoid of zeros in the open RHS. Thus, it may have zeros on the $j\omega$-axis. To distinguish such a polynomial from the one that has zeros neither in the open RHS nor on the $j\omega$-axis, the latter is referred to as a strictly Hurwitz polynomial. For computational purposes, Theorem 5.3 can be reformulated and put in a much more convenient form.

THEOREM 5.4

A rational function represented in the form

$$F(s) = \frac{P(s)}{Q(s)} = \frac{m_1(s) + n_1(s)}{m_2(s) + n_2(s)} \tag{5.26}$$

where $m_1(s)$, $m_2(s)$, and $n_1(s)$, $n_2(s)$ are the even and odd parts of the polynomials $P(s)$ and $Q(s)$, respectively, is positive real if and only if the following conditions are satisfied:

1. *$F(s)$ is real when s is real.*
2. *$P(s) + Q(s)$ is strictly Hurwitz.*
3. *$m_1(j\omega)m_2(j\omega) - n_1(j\omega)n_2(j\omega) \geq 0$ for all ω.*

A real polynomial is strictly Hurwitz if and only if the continued-fraction expansion of the ratio of the even part to the odd part or the odd part to the even part of the polynomial yields only real and positive coefficients, and does not terminate prematurely. For $P(s) + Q(s)$ to be strictly Hurwitz, it is necessary and sufficient that the continued-fraction expansion

$$\left[\frac{m_1(s) + m_2(s)}{n_1(s) + n_2(s)}\right]^{\pm 1} = \alpha_1 s + \cfrac{1}{\alpha_2 s + \cfrac{1}{\ddots + \cfrac{1}{\alpha_k s}}} \tag{5.27}$$

yields only real and positive α's, and does not terminate prematurely, i.e., k must equal the degree $m_1(s) + m_2(s)$ or $n_1(s) + n_2(s)$, whichever is larger. It can be shown that the third condition of the

theorem is satisfied if and only if its left-hand-side polynomial does not have real positive roots of odd multiplicity. This may be determined by factoring it or by the use of the Sturm's theorem, which can be found in most texts on elementary theory of equations. We illustrate the above procedure by the following examples.

Example 5.2

Test the following function to see if it is PR:

$$F(s) = \frac{2s^4 + 4s^3 + 5s^2 + 5s + 2}{s^3 + s^2 + s + 1} \tag{5.28}$$

For illustrative purposes, we follow the three steps outlined in the theorem, as follows:

$$F(s) = \frac{2s^4 + 4s^3 + 5s^2 + 5s + 2}{s^3 + s^2 + s + 1} = \frac{P(s)}{Q(s)} = \frac{m_1(s) + n_1(s)}{m_2(s) + n_2(s)} \tag{5.29}$$

where

$$m_1(s) = 2s^4 + 5s^2 + 2, \quad n_1(s) = 4s^3 + 5s \tag{5.30a}$$

$$m_2(s) = s^2 + 1, \quad n_2(s) = s^3 + s \tag{5.30b}$$

Condition 1 is clearly satisfied. To test condition 2, we perform the Hurwitz test, which gives

$$\frac{m_1(s) + m_2(s)}{n_1(s) + n_2(s)} = \frac{2s^4 + 6s^2 + 3}{5s^3 + 6s} = \frac{2}{5}s + \cfrac{1}{\frac{25}{18}s + \cfrac{1}{\frac{324}{165}s + \cfrac{1}{\frac{33}{54}s}}} \tag{5.31}$$

Since all the coefficients are real and positive and since the continued-fraction expansion does not terminate prematurely, the polynomial $P(s) + Q(s)$ is strictly Hurwitz. Thus, condition 2 is satisfied.

To test condition 3, we compute

$$\begin{aligned} m_1(j\omega)m_2(j\omega) - n_1(j\omega)n_2(j\omega) &= 2\omega^6 - 2\omega^4 - 2\omega^2 + 2 \\ &= 2\left(\omega^2 + 1\right)\left(\omega^2 - 1\right)^2 \geq 0 \end{aligned} \tag{5.32}$$

which is nonnegative for all ω, or, equivalently, which does not possess any real positive roots of odd multiplicity. Therefore $F(s)$ is positive real.

References

1. O. Brune, Synthesis of a finite two-terminal network whose driving-point impedance is a prescribed function of frequency, *J. Math. Phys.,* 10, 191–236, 1931.
2. R.V. Churchill, *Introduction to Complex Variables and Applications,* New York: McGraw-Hill, 1960.

6

Passive Cascade Synthesis

6.1 Introduction **6**-1
6.2 Type-E Section **6**-6
6.3 Richards Section **6**-10
6.4 Darlington Type-D Section **6**-11
References **6**-16

Wai-Kai Chen
University of Illinois at Chicago

6.1 Introduction

In this chapter, we demonstrate that any rational positive-real function can be realized as the input immittance of a passive one-port network terminated in a resistor, thereby also proving the sufficiency of Theorem 5.2.

Consider the even part

$$\text{Ev } Z(s) = r(s) = \frac{1}{2}[Z(s) + Z(-s)] \tag{6.1}$$

of a given rational positive-real impedance $Z(s)$. As in Equation 5.26, we first separate the numerator and denominator polynomials of $Z(s)$ into even and odd parts, and write

$$Z(s) = \frac{m_1 + n_1}{m_2 + n_2} \tag{6.2}$$

Then, we have

$$r(s) = \frac{m_1 m_2 - n_1 n_2}{m_2^2 + n_2^2} \tag{6.3}$$

showing that if s_0 is a zero or pole of $r(s)$, so is $-s_0$. Thus, the zeros and poles of $r(s)$ possess quadrantal symmetry with respect to both the real and imaginary axes. They may appear in pairs on the real axis, in pairs on the $j\omega$-axis, or in the form of sets of quadruplets in the complex-frequency plane. Furthermore, for a positive-real $Z(s)$, the $j\omega$-axis zeros of $r(j\omega)$ are required to be of even multiplicity in order that Re $Z(j\omega) = r(j\omega)$ never be negative.

Suppose that we can extract from $Z(s)$ a set of open-circuit impedance parameters $z_{ij}(s)$ characterizing a component two-port network, as depicted in Figure 6.1, which produces one pair of real axis zeros, one pair of $j\omega$-axis zeros, or one set of quadruplet of zeros of $r(s)$, and leaves a rational positive-real impedance $Z_1(s)$ of lower degree, the even part of which $r_1(s)$ is devoid of these zeros but contains all

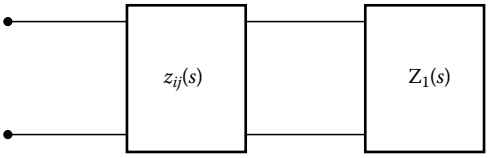


FIGURE 6.1 Two-port network terminated in $Z_1(s)$.

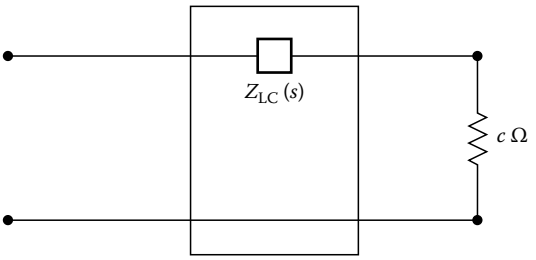


FIGURE 6.2 Two-port network terminated in a resistor.

other zeros of $r(s)$. After a finite q steps, we arrive at a rational positive-real impedance $Z_q(s)$, the even part $r_q(s)$ of which is devoid of zeros in the entire complex-frequency plane, meaning that its even part must be a nonnegative constant c:

$$r_q(s) = \frac{1}{2}\left[Z_q(s) + Z_q(-s)\right] = c \tag{6.4}$$

Therefore, $Z_q(s)$ is expressible as the sum of a reactance function* $Z_{LC}(s)$ and a resistance c:

$$Z_q(s) = Z_{LC}(s) + c \tag{6.5}$$

which can be realized as the input impedance of a lossless two-port network terminated in a c-ohm resistor, as shown in Figure 6.2.

To motivate our discussion, we first present a theorem credited to Richards [1,2], which is intimately tied up with the famous Bott–Duffin technique [3].

THEOREM 6.1

Let $Z(s)$ be a positive-real function that is neither of the form Ls nor $1/Cs$. Let k be an arbitrary positive-real constant. Then, the Richards function

$$W(s) = \frac{kZ(s) - sZ(k)}{kZ(k) - sZ(s)} \tag{6.6}$$

is also positive real.

The degree of a rational function is defined as the sum of the degrees of its relatively prime numerator and denominator polynomials. Thus, the Richards function $W(s)$ is also rational, the degree of which is

* A formal definition will be given in Chapter 7.

not greater than that of $Z(s)$. It was first pointed out by Richards that if k can be chosen so that the even part of $Z(s)$ vanishes at k, then the degree of $W(s)$ is at least two less than that of $Z(s)$. Let

$$s_0 = \sigma_0 + j\omega_0 \tag{6.7}$$

be a point in the closed RHS. Then, according to the preceding theorem, the function

$$\widehat{W}_1(s) = \frac{s_0 Z(s) - sZ(s_0)}{s_0 Z(s_0) - sZ(s)} \tag{6.8}$$

is positive real if s_0 is positive real; and the function

$$W_1(s) = Z(s_0)\widehat{W}_1(\bar{s}_0)\frac{\bar{s}_0\widehat{W}_1(s) - s\widehat{W}_1(\bar{s}_0)}{\bar{s}_0\widehat{W}_1(\bar{s}_0) - s\widehat{W}_1(s)} \tag{6.9}$$

is positive real if s_0 is a positive-real constant and $\widehat{W}_1(s)$ is a positive-real function. Substituting Equation 6.8 in Equation 6.9 yields

$$W_1(s) = \frac{D_1(s)Z(s) - B_1(s)}{-C_1(s)Z(s) + A_1(s)} \tag{6.10}$$

where

$$A_1(s) = q_4 s^2 + |s_0|^2 \tag{6.11a}$$

$$B_1(s) = q_2 s \tag{6.11b}$$

$$C_1(s) = q_3 s \tag{6.11c}$$

$$D_1(s) = q_1 s^2 + |s_0|^2 \tag{6.11d}$$

$$q_1 = \frac{R_0/\sigma_0 - X_0/\omega_0}{R_0/\sigma_0 + X_0/\omega_0} \tag{6.12a}$$

$$q_2 = \frac{2|Z_0|^2}{R_0/\sigma_0 + X_0/\omega_0} \tag{6.12b}$$

$$q_3 = \frac{2}{R_0/\sigma_0 - X_0/\omega_0} \tag{6.12c}$$

$$q_4 = \frac{R_0/\sigma_0 + X_0/\omega_0}{R_0/\sigma_0 - X_0/\omega_0} = \frac{1}{q_1} \tag{6.12d}$$

in which

$$Z(s_0) = R_0 + jX_0 \equiv Z_0 \tag{6.13}$$

In the case $\omega_0 = 0$, then X_0/ω_0 must be replaced by $Z'(\sigma_0)$:

$$\frac{X_0}{\omega_0} \to Z'(\sigma_0) = \left.\frac{dZ(s)}{ds}\right|_{s=\sigma_0} \tag{6.14a}$$

For $\sigma = 0$ and $R_0 = 0$, R_0/σ_0 is replaced by $X'(\omega_0)$:

$$\frac{R_0}{\sigma_0} \to X'(\omega_0) = \left.\frac{dZ(s)}{ds}\right|_{s=j\omega_0} \tag{6.14b}$$

Definition 6.1: *Index set*

For a given positive-real function $Z(s)$, let s_0 be any point in the open RHS or any finite nonzero point on the $j\omega$-axis where $Z(s)$ is analytic. Then, the set of four real numbers q_1, q_2, q_3, and q_4, as defined in Equations 6.12 through 6.14, is called the index set assigned to the point s_0 by the positive-real function $Z(s)$.

We illustrate this concept by the following example.

Example 6.1

Determine the index set assigned to the point $s_0 = 0.4551 + j1.099$ by the positive-real function

$$Z(s) = \frac{s^2 + s + 1}{s^2 + s + 2} \tag{6.15}$$

From definition and Equation 6.15, we have

$$s_0 = 0.4551 + j1.099 = \sigma_0 + j\omega_0 \tag{6.16a}$$

$$\begin{aligned} Z(s_0) &= Z(0.4551 + j1.099) \\ &= 0.7770 + j0.3218 = 0.8410e^{j22.5^\circ} \\ &= R_0 + jX_0 \equiv Z_0 \end{aligned} \tag{6.16b}$$

$$|Z_0|^2 = 0.7073 \tag{6.17}$$

obtaining from Equation 6.12

$$q_1 = 0.707, \quad q_2 = 0.707, \quad q_3 = 1.414, \quad q_4 = 1.414 \tag{6.18}$$

With these preliminaries, we now state the following theorem, which forms the cornerstone of the method of cascade synthesis of a rational positive-real impedance according to the Darlington theory [4].

THEOREM 6.2

Let $Z(s)$ be a positive-real function, which is neither of the form Ls nor $1/Cs$, L and C being real nonnegative constants. Let $s_0 = \sigma_0 + j\omega_0$ be a finite nonzero point in the closed RHS where $Z(s)$ is analytic, then the function

$$W_1(s) = \frac{D_1(s)Z(s) - B_1(s)}{-C_1(s)Z(s) + A_1(s)} \tag{6.19}$$

is positive real; where A_1, B_1, C_1, and D_1 are defined in Equation 6.11 and $\{q_1, q_2, q_3, q_4\}$ is the index set assigned to the point s_0 by $Z(s)$. Furthermore, $W_1(s)$ possesses the following attributes:

(1) *If $Z(s)$ is rational, $W_1(s)$ is rational, the degree of which is not greater than that of $Z(s)$, or*

$$\text{degree } W_1(s) \leq \text{degree } Z(s) \tag{6.20}$$

(2) *If $Z(s)$ rational and if s_0 is a zero of its even part $r(s)$, then*

$$\text{degree } W_1(s) \leq \text{degree } Z(s) - 4, \quad \omega_0 \neq 0 \tag{6.21a}$$

$$\text{degree } W_1(s) \leq \text{degree } Z(s) - 2, \quad \omega_0 = 0 \tag{6.21b}$$

(3) *If* $s_0 = \sigma_0 > 0$ *is a real zero of* $r(s)$ *of at least multiplicity* 2 *and if* $Z(s)$ *is rational, then*

$$\text{degree } W_1(s) \leq \text{degree } Z(s) - 4 \tag{6.22}$$

We remark that since $Z(s)$ is positive real, all the points in the open RHS are admissible. Any point on the $j\omega$-axis, exclusive of the origin and infinity, where $Z(s)$ is analytic is admissible as s_0.

We are now in a position to show that any positive-real function can be realized as the input impedance of a lossless one-port network terminated in a resistor. Our starting point is Equation 6.19, which after solving $Z(s)$ in terms of $W_1(s)$ yields

$$Z(s) = \frac{A_1(s)W_1(s) + B_1(s)}{C_1(s)W_1(s) + D_1(s)} \tag{6.23}$$

It can be shown that $Z(s)$ can be realized as the input impedance of a two-port network N_1, which is characterized by its transmission matrix

$$\mathbf{T}_1(s) = \begin{bmatrix} A_1(s) & B_1(s) \\ C_1(s) & D_1(s) \end{bmatrix} \tag{6.24}$$

terminated in $W_1(s)$, as depicted in Figure 6.3. To see this, we first compute the corresponding impedance matrix $\mathbf{Z}_1(s)$ of N_1 from $\mathbf{T}_1(s)$ and obtain

$$\mathbf{Z}_1(s) = \begin{bmatrix} z_{11}(s) & z_{12}(s) \\ z_{21}(s) & z_{22}(s) \end{bmatrix} = \frac{1}{C_1(s)} \begin{bmatrix} A_1(s) & A_1(s)D_1(s) - B_1(s)C_1(s) \\ 1 & D_1(s) \end{bmatrix} \tag{6.25}$$

The input impedance $Z_{11}(s)$ of N_1 with the output port terminating in $W_1(s)$ is found to be

$$Z_{11}(s) = z_{11}(s) - \frac{z_{12}(s)z_{21}(s)}{z_{22}(s) + W_1(s)} = \frac{A_1(s)W_1(s) + B_1(s)}{C_1(s)W_1(s) + D_1(s)} = Z(s) \tag{6.26}$$

The determinant of the transmission matrix $\mathbf{T}_1(s)$ is computed as

$$\det \mathbf{T}_1(s) = A_1(s)D_1(s) - B_1(s)C_1(s) = s^4 + 2\left(\omega_0^2 - \sigma_0^2\right)s^2 + |s_0|^4 \tag{6.27}$$

Observe that det $\mathbf{T}_1(s)$ depends only upon the point s_0 and not on $Z(s)$, and that the input impedance $Z_{11}(s)$ remains unaltered if each element of $\mathbf{T}_1(s)$ is multiplied or divided by a nonzero

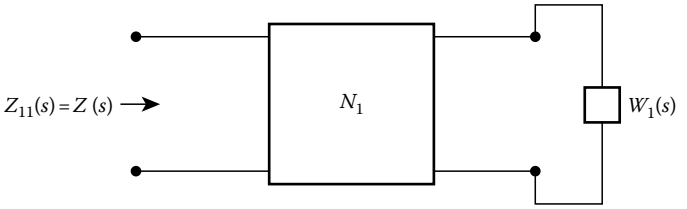


FIGURE 6.3 Two-port network N_1 terminated in the impedance $W_1(s)$.

finite quality. To complete the realization, we must now demonstrate that the two-port network N_1 is physically realizable.

6.2 Type-E Section

Consider the lossless nonreciprocal two-port network of Figure 6.4 known as the type-E section. Our objective is to show that this two-port realizes N_1. To this end, we first compute its impedance matrix $\mathbf{Z}_E(s)$ as

$$\mathbf{Z}_E(s) = \begin{bmatrix} L_1(s) + 1/Cs & Ms + 1/Cs + \zeta \\ Ms + 1/Cs - \zeta & L_2 s + 1/Cs \end{bmatrix} \tag{6.28}$$

where $M^2 = L_1 L_2$, the determinant of which is given by

$$\det\ \mathbf{Z}_E(s) = \frac{L_1 + L_2 - 2M + \zeta^2 C}{C} \tag{6.29}$$

a constant independent of s due to perfect coupling. From the impedance matrix $\mathbf{Z}_E(s)$, its corresponding transmission matrix $\mathbf{T}_E(s)$ is found to be

$$\mathbf{T}_E(s) = \frac{1}{MCs^2 - \zeta Cs + 1} \begin{bmatrix} L_1 Cs^2 + 1 & \left(L_1 + L_2 - 2M + \zeta^2 C\right)s \\ Cs & L_2 Cs^2 + 1 \end{bmatrix} \tag{6.30}$$

To show that the type-E section realizes N_1, we divide each element of $\mathbf{T}_1(s)$ of Equation 6.24 by $|s_0|^2$ $(MCs^2 - \zeta Cs + 1)$. This manipulation will not affect the input impedance $Z(s)$ but it will result in a transmission matrix having the form of $\mathbf{T}_E(s)$. Comparing this new matrix with Equation 6.30 in conjunction with Equation 6.11 yields the following identifications:

$$L_1 C = \frac{q_4}{|s_0|^2} \tag{6.31a}$$

$$L_1 + L_2 - 2M + \zeta^2 C = \frac{q_2}{|s_0|^2} \tag{6.31b}$$

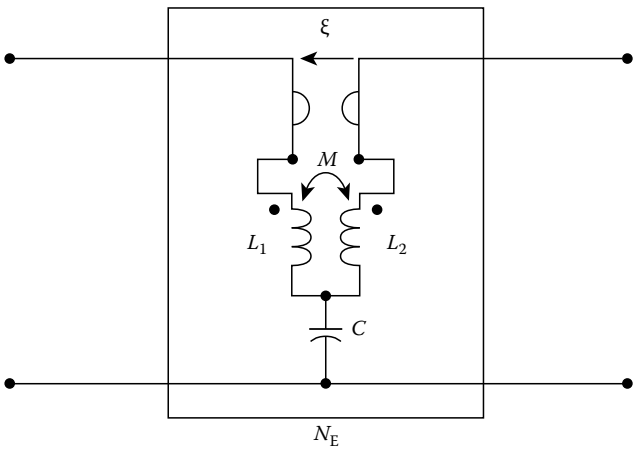


FIGURE 6.4 Type-E section.

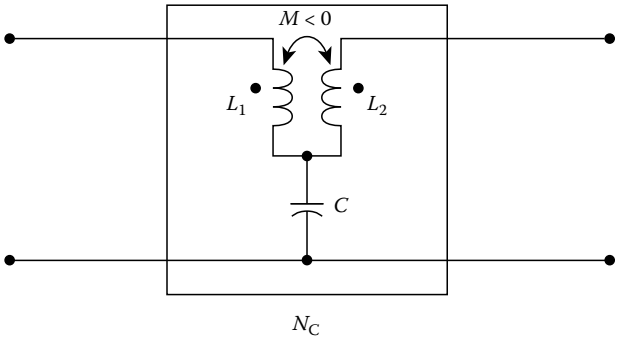


FIGURE 6.5 Darlington type-C section.

$$C = \frac{q_3}{|s_0|^2} \tag{6.31c}$$

$$L_2 C = \frac{q_1}{|s_0|^2} \tag{6.31d}$$

Solving these for the element values of the type-E section, we obtain

$$L_1 = \frac{q_4}{q_3} = \frac{1}{q_1 q_3}, \quad L_2 = \frac{q_1}{q_3} \tag{6.32a}$$

$$C = \frac{q_3}{|s_0|^2}, \quad M = \frac{1}{q_3}, \quad \zeta = \pm\frac{2\sigma_0}{q_3} \tag{6.32b}$$

Since the elements of the index set assigned to the point s_0 by $Z(s)$ are positive and finited for all admissible points s_0 in the closed RHS except at those admissible points $s_0 = j\omega_0$ on the $j\omega$-axis where $R_0 \neq 0$, all the elements in Equation 6.32 are physical. But at those admissible points $s_0 = j\omega_0$ where $R_0 \neq 0$, R_0/σ_0 becomes infinity and $q_1 = q_4 = 1$ and $q_2 = q_3 = 0$. Under this situation, $W_1(s) = Z(s)$ and the corresponding two-port network N_1 degenerates into a pair of wires.

Appealing to Theorem 6.2 shows that if s_0 is chosen to be a complex open RHS zero of the even part $r(s)$ of $Z(s)$, the type-E section is capable of extracting a set of quadrantal zeros of $r(s)$ and leads to at least a four-degree reduction. For zeros of $r(s)$ on the $j\omega$-axis or the σ-axis, the type-E section degenerates into other types of sections, as follows:

Case 1. $s_0 = \sigma_0 > 0$. Then, we have $\zeta = 0$ and

$$L_1 = \frac{q_4}{q_3} = \frac{1}{q_1 q_3}, \quad L_2 = \frac{q_1}{q_3}, \quad C = \frac{q_3}{\sigma_0^2}, \quad M = -\frac{1}{q_3} < 0 \tag{6.33}$$

The type-E section degenerates to the Darlington type-C section, as shown in Figure 6.5.

Case 2. $s_0 = j\omega_0$ and $R_0 = 0$. In this case, we replace R_0/σ_0 by $X'(\omega_0)$ and the gyrator in the type-E section can be avoided because $\zeta = 0$. The type-E section degenerates into the Brune section of Figure 6.6, the element values of which are given by

$$L_1 = \frac{q_4}{q_3} = \frac{1}{q_1 q_3} = \frac{\omega_0 X'(\omega_0) + X_0}{2\omega_0} \tag{6.34a}$$

$$L_2 = \frac{q_1}{q_3} = \frac{[\omega_0 X'(\omega_0) - X_0]^2}{2\omega_0[\omega_0 X'(\omega_0) + X_0]} \tag{6.34b}$$

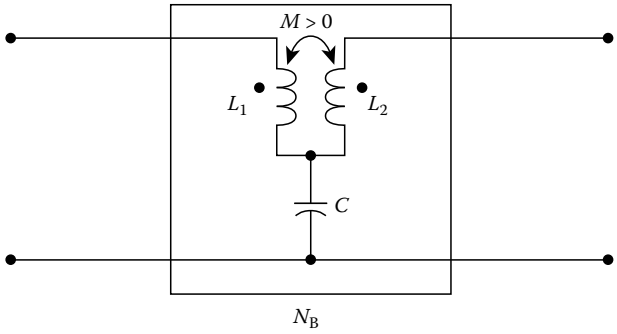


FIGURE 6.6 Brune section.

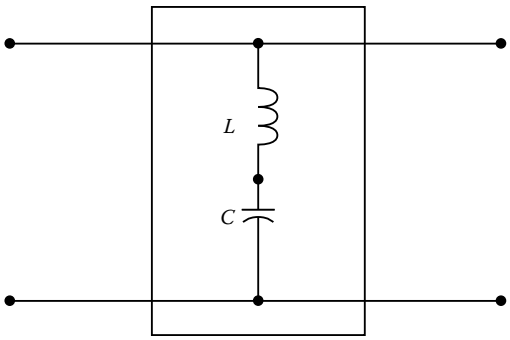


FIGURE 6.7 Degenerate Brune section.

$$C = \frac{q_3}{|s_0|^2} = \frac{2}{\omega_0[\omega_0 X'(\omega_0) - X_0]} \tag{6.34c}$$

$$M = \frac{1}{q_3} = \frac{\omega_0 X'(\omega_0) - X_0}{2\omega_0} > 0 \tag{6.34d}$$

In particular, if $X_0 = 0$ or $Z(j\omega_0) = 0$, the Brune section degenerates into the two-port network of Figure 6.7 with element values

$$L = \frac{1}{2}X'(\omega_0), \quad C = \frac{2}{\omega_0^2 X'(\omega_0)} \tag{6.35}$$

As ω_0 approaches zeros, this degenerate Brune section goes into the type-A section of Figure 6.8 with

$$L = \frac{1}{2}Z'(0) \tag{6.36}$$

When ω_0 approaches infinity, the degenerate Brune section collapses into the type-B section of Figure 6.9 with

$$\frac{2}{C} = \lim_{s\to\infty} sZ(s) \tag{6.37}$$

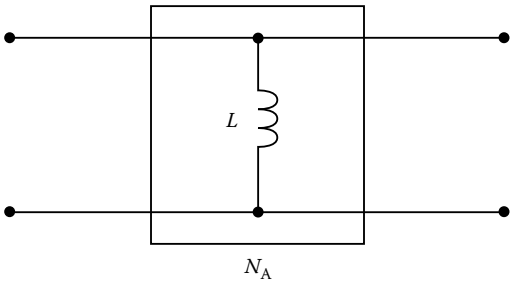


FIGURE 6.8 Type-A section.

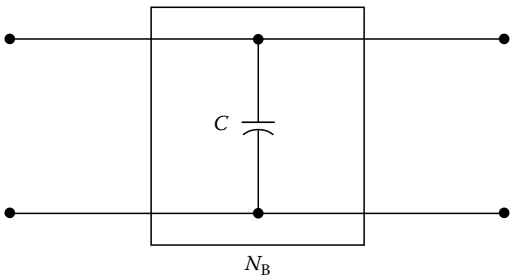


FIGURE 6.9 Type-B section.

Case 3. $s_0 = j\omega_0$ and $R_0 \neq 0$. In this case, R_0/σ_0 is infinity and

$$q_1 = q_4 = 1, \quad q_2 = q_3 = 0 \tag{6.38}$$

The type-E section degenerates into a pair of wires.

Therefore, the Brune section is capable of extracting any $j\omega$-axis zero of the even part of a positive-real impedance, and leads to at least a four-degree reduction if $j\omega_0$ is nonzero and finite, a two-degree reduction otherwise. The latter corresponds to the type-A or type-B section.

Example 6.2

Consider the positive-real impedance

$$Z(s) = \frac{8s^2 + 9s + 10}{2s^2 + 4s + 4} \tag{6.39}$$

The zeros of its even part $r(s)$ are found from the polynomial

$$m_1 m_2 - n_1 n_2 = 16\left(s^4 + s^2 + 2.5\right) \tag{6.40}$$

obtaining

$$s_0 = \sigma_0 + j\omega_0 = 0.735 + j1.020 \tag{6.41a}$$

$$Z(s_0) \equiv R_0 + jX_0 = 2.633 + j0.4279 = 2.667e^{j9.23} \equiv Z_0 \tag{6.41b}$$

The elements of the index set assigned to s_0 by $Z(s)$ are computed as

$$q_1 = 0.7904, \quad q_2 = 3.556, \quad q_3 = 0.6323, \quad q_4 = 1.265 \tag{6.42}$$

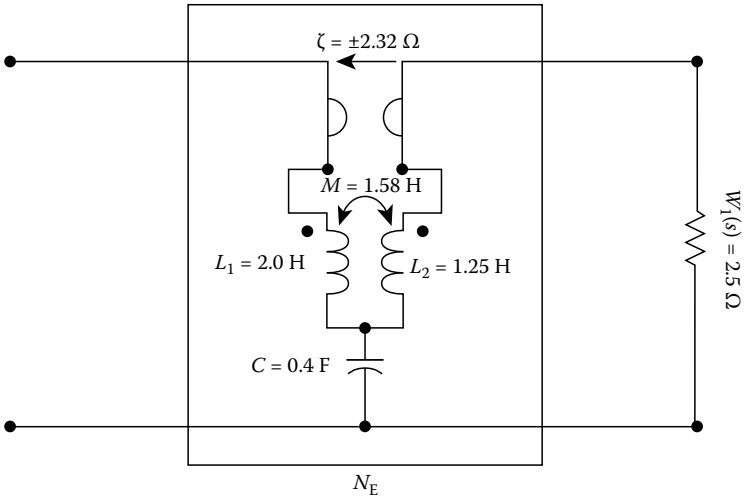


FIGURE 6.10 Realization of the impedance function of (6.39).

Substituting these in Equation 6.32 yields the element values of the type-E section as shown in Figure 6.10.

$$L_1 = 2 \text{ H}, \quad L_2 = 1.25 \text{ H}, \quad C = 0.40 \text{ F} \tag{6.43a}$$

$$M = 1.58 \text{ H}, \quad \zeta = \pm 2.32 \ \Omega \tag{6.43b}$$

The terminating impedance $W_1(s)$ is a resistance of value

$$W_1(s) = Z(0) = 2.5 \ \Omega \tag{6.44}$$

as shown in Figure 6.10.

6.3 Richards Section

In this part, we show that any positive real zero $s_0 = \sigma_0$ of the even part $r(s)$ of a positive-real impedance $Z(s)$, in addition to being realized by the reciprocal Darlington type-C section, can also be realized by a nonreciprocal section called the Richards section of Figure 6.11.

Let $Z(s)$ be a rational positive-real function. Then according to Theorem 6.1, for any positive real σ_0, the function

$$W_1(s) = Z(\sigma_0)\frac{\sigma_0 Z(s) - sZ(\sigma_0)}{\sigma_0 Z(\sigma_0) - sZ(s)} \tag{6.45}$$

is also rational and positive real, the degree of which is not greater than that of $Z(s)$. As pointed out by Richards [1], if σ_0 is a zero of $r(s)$, then

$$\text{degree } W_1(s) \leq \text{degree } Z(s) - 2 \tag{6.46}$$

Inverting Equation 6.45 for $Z(s)$ yields

$$Z(s) = \frac{\sigma_0 W_1(s) + sZ(\sigma_0)}{sW_1(s)/Z(\sigma_0) + \sigma_0} \tag{6.47}$$

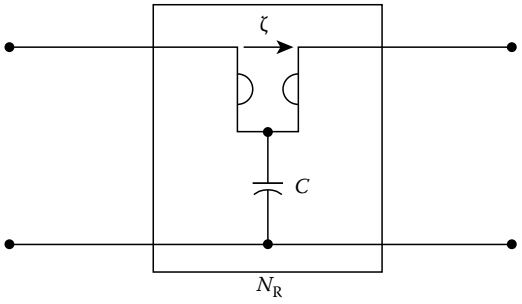


FIGURE 6.11 Richards section.

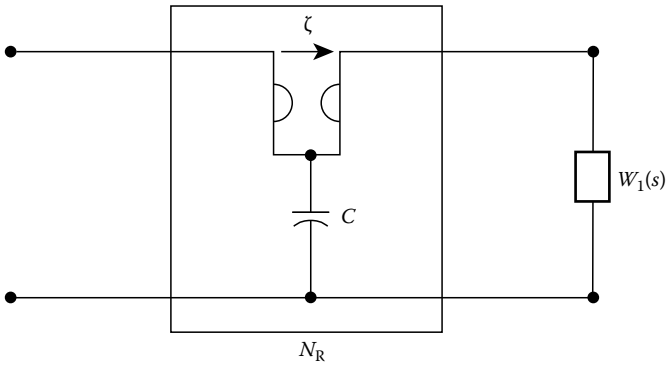


FIGURE 6.12 Realization of $Z(s)$ by Richards section.

This impedance can be realized by the Richards section terminated in the impedance $W_1(s)$ as indicated in Figure 6.12 with the element values

$$C = \frac{1}{\sigma_0 Z(\sigma_0)}, \quad \zeta = \pm Z(\sigma_0) \tag{6.48}$$

6.4 Darlington Type-D Section

In the foregoing, we have demonstrated that the lossless two-port network N_1 can be realized by the lossless nonreciprocal type-E section, which degenerates into the classical type-A, type-B, type-C, and the Brune sections when the even part zero s_0 of the positive-real impedance is restricted to the $j\omega$-axis or the positive σ-axis. In the present section, we show that N_1 can also be realized by a lossless reciprocal two-port network by the application of Theorem 6.1 twice.

Let s_0 be a zero of the even part $r(s)$ of a rational positive-real impedance $Z(s)$. By Theorem 6.2, the function $W_1(s)$ of Equation 6.19 is also rational positive real, and its degree is at least four or two less that that of $Z(s)$, depending on whether $\omega_0 \neq 0$ or $\omega_0 = 0$. Now, apply Theorem 6.2 to $W_1(s)$ at the same point s_0. Then, the function

$$W_2(s) = \frac{D_2(s)W_1(s) - B_2(S)}{-C_2(s)W_1(s) + A_2(s)} \tag{6.49}$$

is rational positive real, the degree of which cannot exceed that of $W_1(s)$, being at least two or four degrees less than that of $Z(s)$, where

$$A_2(s) = p_4 s^2 + |s_0|^2 \tag{6.50a}$$

$$B_2(s) = p_2 s \tag{6.50b}$$

$$C_2(s) = p_3 s \tag{6.50c}$$

$$D_2(s) = p_1 s^2 + |s_0|^2 \tag{6.50d}$$

and $\{p_1, p_2, p_3, p_4\}$ is the index set assigned to the point s_0 by the positive-real function $W_1(s)$. Solving for $W_1(s)$ in Equation 6.49 gives

$$W_1(s) = \frac{A_2(s)W_2(s) + B_2(s)}{C_2(s)W_2(s) + D_2(s)} \tag{6.51}$$

which can be realized as the input impedance of a two-port network N_2 characterized by the transmission matrix

$$\mathbf{T}_2(s) = \begin{bmatrix} A_2(s) & B_2(s) \\ C_2(s) & D_2(s) \end{bmatrix} \tag{6.52}$$

terminated in $W_2(s)$, as depicted in Figure 6.13.

Consider the cascade connection of the two-port N_1 of Figure 6.3 and N_2 of Figure 6.13 terminated in $W_2(s)$, as shown in Figure 6.14. The transmission matrix $\mathbf{T}(s)$ of the overall two-port network N is simply the product of the transmission matrices of the individual two-ports:

$$\mathbf{T}(s) = \mathbf{T}_1(s)\mathbf{T}_2(s) \tag{6.53}$$

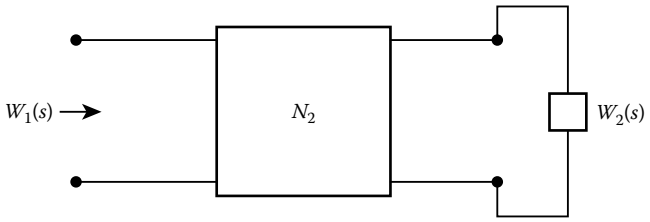


FIGURE 6.13 Realization of the impedance function $W_1(s)$.

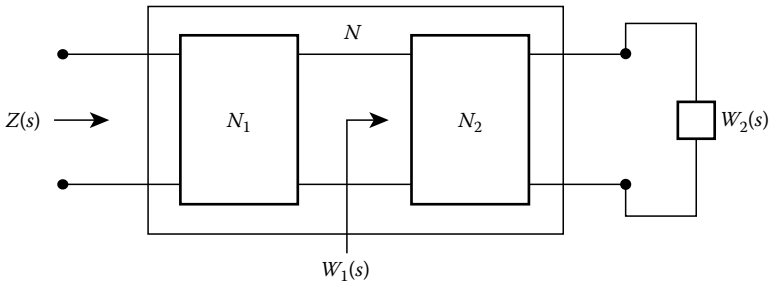


FIGURE 6.14 Cascade connection of two-port networks N_1 and N_2.

the determinant of which is found to be

$$\det\ \mathbf{T}(s) = [\det\ \mathbf{T}_1(s)][\det\ \mathbf{T}_2(s)] = \left[s^4 + 2\left(\omega_0^2 - \sigma_0^2\right)s^2 + |s_0|^4\right]^2 \tag{6.54}$$

Thus, when N is terminated in $W_2(s)$, the input impedance of Figure 6.14 is $Z(s)$. This impedance remains unaltered if each element of $\mathbf{T}_1(s)$ is divided by a nonzero finite quantity. For our purposes, we stipulate that the two-port N_1 be characterized by the transmission matrix

$$\widehat{\mathbf{T}}_1(s) = \frac{1}{\Delta(s)}\mathbf{T}_1(s) \tag{6.55}$$

where

$$\Delta(s) = s^4 + 2\left(\omega_0^2 - \sigma_0^2\right)s^2 + |s_0|^4 \tag{6.56}$$

Using this matrix $\widehat{\mathbf{T}}_1(s)$ for N_1, the transmission matrix $\widehat{\mathbf{T}}_1(s)$ of the overall two-port network N becomes

$$\begin{aligned}\widehat{\mathbf{T}}(s) &= \frac{1}{\Delta(s)}\mathbf{T}_1(s)\mathbf{T}_2(s) \equiv \frac{1}{\Delta(s)}\begin{bmatrix} A(s) & B(s) \\ C(s) & D(s)\end{bmatrix} \\ &= \frac{1}{\Delta(s)}\begin{bmatrix} p_4q_4s^4 + |s_0|^4 & (p_2q_4 + p_1q_2)s^3 \\ \quad +\left[(p_4 + q_4)|s_0|^2 + p_3q_2\right]s^2 & \quad +(p_2 + q_2)|s_0|^2 s \\ (p_4q_3 + p_3q_1)s^3 & p_1q_1s^4 + |s_0|^4 + \left[(p_1 + q_1)|s_0|^2\right. \\ \quad +(p_3 + q_3)|s_0|^2 s & \quad \left. + p_2q_3\right]s^2\end{bmatrix}\end{aligned} \tag{6.57}$$

The corresponding impedance matrix $\mathbf{Z}(s)$ of the overall two-port network is found to be

$$\mathbf{Z}(s) = \frac{1}{C(s)}\begin{bmatrix} A(s) & \Delta(s) \\ \Delta(s) & D(s)\end{bmatrix} \tag{6.58}$$

showing that N is reciprocal because $\mathbf{Z}(s)$ is symmetric.

Now consider the reciprocal lossless Darlington type-D section N_D of Figure 6.15 with two perfectly coupled transformers

$$L_1L_2 = M_1^2, \quad L_3L_4 = M_2^2 \tag{6.59}$$

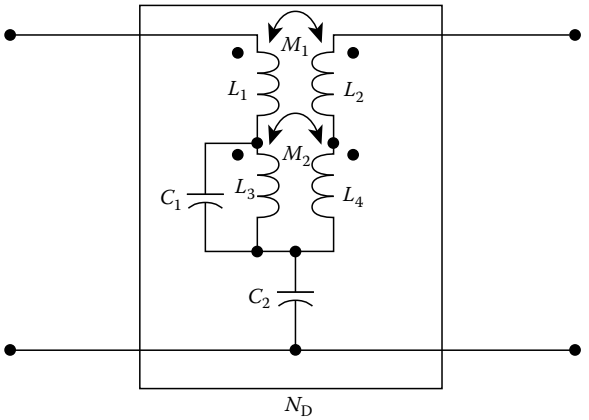


FIGURE 6.15 Darlington type-D section N_D.

The impedance matrix $\mathbf{Z}_{\mathrm{D}}(s)$ of N_{D} is found to be

$$\mathbf{Z}_{\mathrm{D}}(s) = \begin{bmatrix} L_1 s + \frac{1}{C_2 s} + \dfrac{s/C_1}{s^2+\omega_a^2} & M_1 s + \frac{1}{C_2 s} + \dfrac{\omega_a^2 M_2 s}{s^2+\omega_a^2} \\ M_1 s + \dfrac{1}{C_2 s} + \dfrac{\omega_a^2 M_2 s}{s^2+\omega_a^2} & L_2 s + \dfrac{1}{C_2 s} + \dfrac{\omega_a^2 L_4 s}{s^2+\omega_a^2} \end{bmatrix} \tag{6.60}$$

where $\omega_a^2 = 1/C_1 L_3$.

Setting $\mathbf{Z}_{\mathrm{D}}(s) = \mathbf{Z}(s)$ in conjunction with Equation 6.57 and after considerable algebraic manipulations, we can make the following identifications:

$$L_1 = \frac{p_4 q_4}{p_4 q_3 + p_3 q_1} \tag{6.61a}$$

$$L_2 = \frac{p_1 q_1}{p_4 q_3 + p_3 q_1} = \frac{M_1^2}{L_1} \tag{6.61b}$$

$$M_1 = \frac{1}{p_4 q_3 + p_3 q_1} = \sqrt{L_1 L_2} \tag{6.61c}$$

$$C_2 = \frac{p_3 + q_3}{|s_0|^2} \tag{6.61d}$$

$$\omega_a^2 = \omega_1^2 = \frac{|s_0|^2 (p_3 + q_3)}{p_4 q_3 + p_3 q_1} \tag{6.61e}$$

$$\begin{aligned} M_2 &= -\frac{\omega_1^4 - 2\left(\omega_0^2 - \sigma_0^2\right)\omega_1^2 + |s_0|^4}{\omega_1^4 (p_4 q_3 + p_3 q_1)} \\ &= -\frac{p_3^2 q_3^2 \left|\overline{W_1}(s_0) + Z(s_0) q_1\right|^2}{|s_0|^2 (p_4 q_3 + p_3 q_1)(p_3 + q_3)^2} \le 0 \end{aligned} \tag{6.61f}$$

$$L_4 = \frac{\left[(p_1 + q_1)|s_0|^2 + p_2 q_3\right]\omega_1^2 - p_1 q_1 \omega_1^4 - |s_0|^4}{\omega_1^4 (p_4 q_3 + p_3 q_1)} = -\frac{q_3 M_2}{p_3} \tag{6.61g}$$

$$L_3 = \frac{M_2^2}{L_4} = -\frac{p_3 M_2}{q_3} \tag{6.61h}$$

$$C_1 = \frac{1}{\omega_1^2 L_3} = -\frac{q_3}{\omega_1^2 p_3 M_2} \tag{6.61i}$$

Thus, all the element values except M_2 are nonnegative, and the lossless reciprocal Darlington type-D section is equivalent to the two type-E sections in cascade.

Example 6.3

Consider the positive-real impedance

$$Z(s) = \frac{6s^2 + 5s + 6}{2s^2 + 4s + 4} \tag{6.62}$$

the even part of which has a zero at

$$s_0 = \sigma_0 + j\omega_0 = 0.61139 + j1.02005 \tag{6.63}$$

The elements of the index set assigned to s_0 by $Z(s)$ are given by

$$q_1 = 0.70711, \quad q_2 = 1.76784, \quad q_3 = 0.94283, \quad q_4 = 1.41421 \tag{6.64}$$

The terminating impedance $W_1(s)$ is determined to be

$$W_1(s) = W_1(0) = Z(0) = 1.5\ \Omega \tag{6.65}$$

The elements of the index set assigned to the point s_0 by $W_1(s)$ are found to be

$$p_1 = 1, \quad p_2 = 1.83417, \quad p_3 = 0.81519, \quad p_4 = 1 \tag{6.66}$$

Substituting these in Equation 6.61 yields the desired element values of the type-D section, as follows:

$$L_1 = \frac{p_4 q_4}{p_4 q_3 + p_3 q_1} = 0.93086\ \text{H} \tag{6.67a}$$

$$L_2 = \frac{p_1 q_1}{p_4 q_3 + p_3 q_1} = 0.46543\ \text{H} \tag{6.67b}$$

$$M_1 = \frac{1}{p_3 q_3 + p_3 q_1} = 0.65822\ \text{H} \tag{6.67c}$$

$$C_2 = \frac{p_3 + q_3}{|s_0|^2} = 1.24303\ \text{F} \tag{6.67d}$$

$$\omega_1^2 = \frac{|s_0^2|(p_3 + q_3)}{p_4 q_3 + p_3 q_1} = 1.63656 \tag{6.67e}$$

$$M_2 = -\frac{\omega_1^4 - 2(\omega_0^2 - \sigma_0^2)\omega_1^2 + |s_0|^4}{\omega_1^4(p_4 q_3 + p_3 q_1)} = -0.61350\ \text{H} \tag{6.67f}$$

$$L_4 = -\frac{q_3 M_2}{p_3} = 0.70956\ \text{H} \tag{6.67g}$$

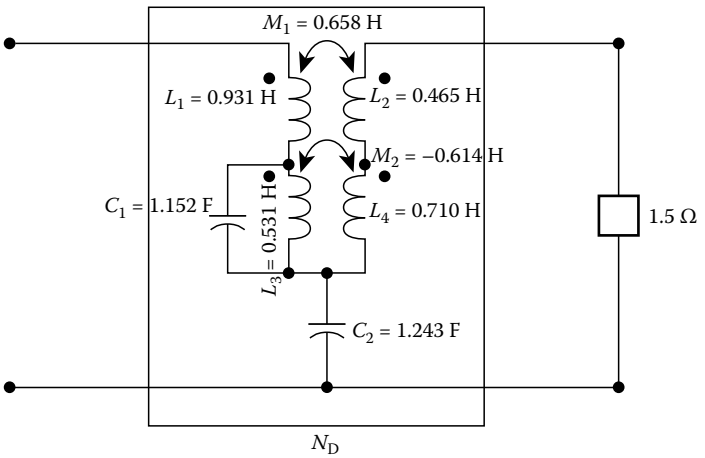


FIGURE 6.16 Darlington type-D section terminated in a resistor.

$$L_3 = \frac{M_2^2}{L_4} = -\frac{p_3 M_2}{q_3} = 0.53044 \text{ H} \tag{6.67h}$$

$$C_1 = \frac{1}{\omega_1^2 L_3} = -\frac{q_3}{\omega_1^2 p_3 M_2} = 1.15193 \text{ F} \tag{6.67i}$$

The complete network together with its termination is presented in Figure 6.16.

References

1. P. I. Richards, A special class of functions with positive real part in half-plane, *Duke Math. J*., 14, 777–786, 1947.
2. P. I. Richards, General impedance-function theory, *Quart. Appl. Math*., 6, 21–29, 1948.
3. R. Bott and R. J. Duffin, Impedance synthesis without the use of transformers, *J. Appl. Phys.,* 20, 816, 1949.
4. W. K. Chen, *Passive and Active Filters: Theory and Implementations,* New York: John Wiley & Sons, 1986.

7

Synthesis of LCM and RC One-Port Networks

7.1 Introduction 7-1
7.2 LCM One-Port Networks 7-1
7.3 RC One-Port Networks 7-6
References 7-11

Wai-Kai Chen
University of Illinois at Chicago

7.1 Introduction

In Chapter 6, we showed that any positive-real function can be realized as the input immittance of a passive one-port network, which is describable as a lossless two-port network terminated in a resistor. Therefore, insofar as the input immittance is concerned, any passive network is equivalent to one containing at most one resistor. In this section, we consider the synthesis of a one-port network composed only of self and mutual inductors and capacitors called the LCM one-port, or a one-port composed only of resistors and capacitors called the RC one-port.

7.2 LCM One-Port Networks

Consider the input impedance $Z(s)$ of an LCM one-port network written in the form

$$Z(s) = \frac{m_1 + n_1}{m_2 + n_2} \tag{7.1}$$

the even part of which is given by

$$r(s) = \frac{m_1 m_2 - n_1 n_2}{m_2^2 - n_2^2} \tag{7.2}$$

Since the one-port is lossless, we have

$$r(j\omega) = \text{Re } Z(j\omega) = 0 \quad \text{for all } \omega \tag{7.3}$$

To make Re $Z(j\omega) = 0$, there are three nontrivial ways: (1) $m_1 = 0$ and $n_2 = 0$, (2) $m_2 = 0$ and $n_1 = 0$, and (3) $m_1 m_2 - n_1 n_2 = 0$. The first possibility leads $Z(s)$ to n_1/m_2, the second to m_1/n_2. For the third possibility, we require that $m_1 m_2 = n_1 n_2$ or

$$(m_1 + n_1)m_2 = (m_2 + n_2)n_1 \tag{7.4}$$

which is equivalent to

$$Z(s) = \frac{m_1 + n_1}{m_2 + n_2} = \frac{n_1}{m_2} \tag{7.5}$$

Therefore, the driving-point immittance of a lossless network is always the quotient of even to odd or odd to even polynomials. Its zeros and poles must occur in quadrantal symmetry, being symmetric with respect to both axes. As a result, they are simple and purely imaginary from stability considerations, or $Z(s)$ can be explicitly written as

$$Z(s) = H\frac{\left(s^2 + \omega_{z1}^2\right)\left(s^2 + \omega_{z2}^2\right)\left(s^2 + \omega_{z3}^2\right)\cdots}{s\left(s^2 + \omega_{p1}^2\right)\left(s^2 + \omega_{p2}^2\right)\cdots} \tag{7.6}$$

where $\omega_{z1} \geq 0$. This equation can be expanded in partial fraction as

$$Z(s) = Hs + \frac{K_0}{s} + \sum_{i=1}^{n} \frac{2K_i s}{s^2 + \omega_i^2} \tag{7.7}$$

where $\omega_{pi} = \omega_i$, and the residues H, K_0, and K_i are all real and positive.

Substituting $s = j\omega$ and writing $Z(j\omega) = \text{Re}\, Z(j\omega) + j\,\text{Im}\, Z(j\omega)$ results in an odd function known as the reactance function $X(\omega)$:

$$X(\omega) = \text{Im}\ Z(j\omega) = H\omega - \frac{K_0}{\omega} + \sum_{i=1}^{n} \frac{2K_i\omega}{-\omega^2 + \omega_i^2} \tag{7.8}$$

Taking the derivatives on both sides yields

$$\frac{dX(\omega)}{d\omega} = H + \frac{K_0}{\omega^2} + \sum_{i=1}^{n} \frac{2K_i\left(\omega^2 + \omega_i^2\right)}{\left(-\omega^2 + \omega_i^2\right)^2} \tag{7.9}$$

Since every factor in this equation is positive for all positive and negative values of ω, we conclude that

$$\frac{dX(\omega)}{d\omega} > 0 \quad \text{for } -\infty < \omega < \infty \tag{7.10}$$

It states that the slope of the reactance function versus frequency curve is always positive, as depicted in Figure 7.1. Consequently, the poles and zeros of $Z(s)$ alternate along the $j\omega$-axis. This is known as the separation property for reactance function credited to Foster [1]. Because of this, the pole and zero frequencies of Equation 7.6 are related by

$$0 \leq \omega_{z1} < \omega_{p1} < \omega_{z2} < \omega_{p2} < \cdots \tag{7.11}$$

We now consider the realization of $Z(s)$. If each term on the right-hand side of Equation 7.7 can be identified as the input impedance of the LC one-port, the series connection of these one-ports would yield the desired realization. The first term is the impedance of an inductor of inductance H, and the second term corresponds to a capacitor of capacitance $1/K_0$. Each of the remaining term can be realized

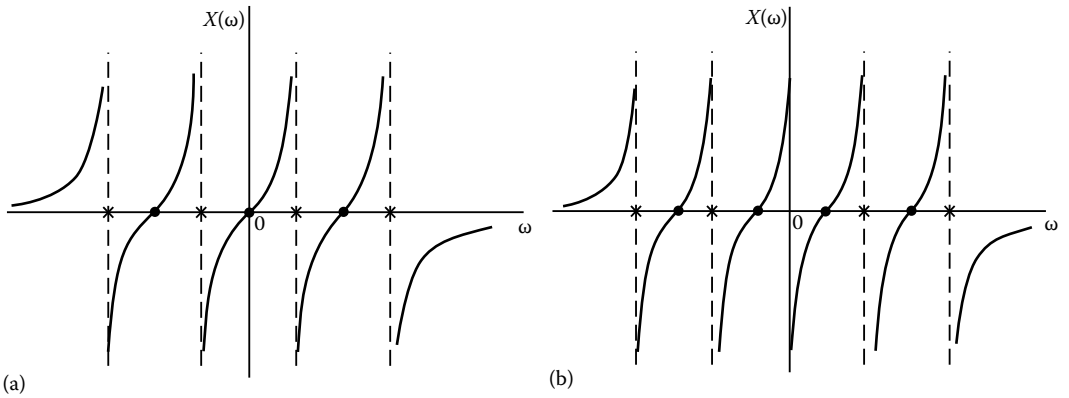


FIGURE 7.1 Plots of reactance function $X(\omega)$ versus ω. (a) The origin is a zero. (b) The origin is a pole.

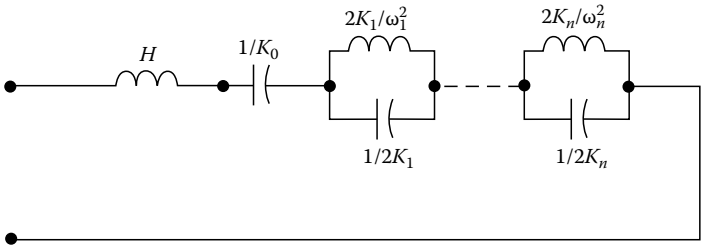


FIGURE 7.2 First Foster canonical form.

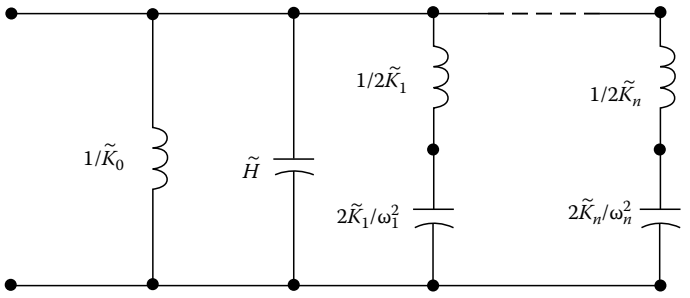


FIGURE 7.3 Second Foster canonical form.

as a parallel combination of an inductor of inductance $2K_i/\omega_i^2$ and a capacitor of capacitance $1/2K_i$. The resulting realization is shown in Figure 7.2 known as the first Foster canonical form. Likewise, if we consider the admittance function $Y(s) = 1/Z(s)$ and expanded it in partial fraction, we obtain

$$Y(s) = \tilde{H}s + \frac{\tilde{K}_0}{s} + \sum_{i=1}^{n} \frac{2\tilde{K}_i s}{s^2 + \omega_i^2} \tag{7.12}$$

which can be realized by the one-port of Figure 7.3 known as the second Foster canonical form. The term canonical form refers to a network containing the minimum number of elements to meet given specifications.

We summarize the preceding results by stating the following theorem:

THEOREM 7.1

A real rational function is the input immittance function of an LCM one-port network if and only if all of its zeros and poles are simple, lie on the jω-axis, and alternate with each other.

In addition to the two Foster canonical forms, there is another synthesis procedure, that gives rise to one-ports known as the Cauer canonical form [2]. Let us expand, $Z(s)$ in a continued fraction

$$Z(s) = \frac{m(s)}{n(s)} = L_1 s + \cfrac{1}{C_2 s + \cfrac{1}{L_3 s + \cfrac{1}{C_4 s + \cfrac{1}{\ddots}}}} \tag{7.13}$$

where $m(s)$ is assumed to be of higher degree than $n(s)$. Otherwise, we expand $Y(s) = 1/Z(s)$ instead of $Z(s)$. Equation 7.13 can be realized as the input impedance of the LC ladder network of Figure 7.4 and is known as the first Cauer canonical form.

Suppose now that we rearrange the numerator and denominator polynomials $m(s)$ and $n(s)$ in ascending order of s, and expand the resulting function in a continued fraction. Such an expansion yields

$$\begin{aligned} Z(s) = \frac{m(s)}{n(s)} &= \frac{a_0 + a_2 s^2 + \cdots + a_{k-2} s^{k-2} + a_k s^k}{b_1 s + b_3 s^3 + \cdots + b_{k-1} s^{k-1}} \\ &= \frac{1}{C_1 s} + \cfrac{1}{\cfrac{1}{L_2 s} + \cfrac{1}{\cfrac{1}{C_3 s} + \cfrac{1}{\cfrac{1}{L_4 s} + \cfrac{1}{\ddots}}}} \end{aligned} \tag{7.14}$$

which can be realized by the LC ladder of Figure 7.5 known as the second Cauer canonical form.

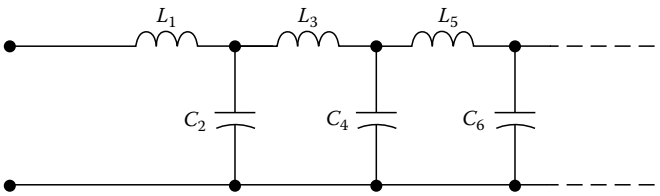


FIGURE 7.4 First Cauer canonical form.

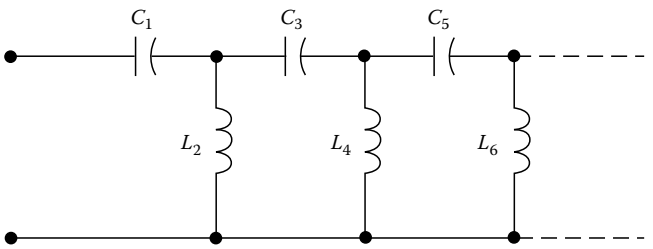


FIGURE 7.5 Second Cauer canonical form.

Example 7.1

Consider the reactance function

$$Z(s) = \frac{s(s^2+4)(s^2+36)}{(s^2+1)(s^2+25)(s^2+81)} \tag{7.15}$$

For the first Foster canonical form, we expand $Z(s)$ in a partial fraction

$$Z(s) = \frac{7s/128}{s^2+1} + \frac{11s/64}{s^2+25} + \frac{99s/128}{s^2+81} \tag{7.16}$$

and obtain the one-port network of Figure 7.6.

For the second Foster canonical form, we expand $Y(s) = 1/Z(s)$ in a partial fraction

$$Y(s) = s + \frac{225/16}{s} + \frac{4851s/128}{s^2+4} + \frac{1925s/128}{s^2+36} \tag{7.17}$$

and obtain the one-port network of Figure 7.7.

For the Cauer canonical form, we expand the function in a continued fraction

$$Z(s) = \cfrac{1}{s + \cfrac{1}{0.015s + \cfrac{1}{6.48s + \cfrac{1}{8.28 \times 10^{-3}s + \cfrac{1}{12.88s + \cfrac{1}{0.048s}}}}}} \tag{7.18}$$

and obtain the one-port network of Figure 7.8.

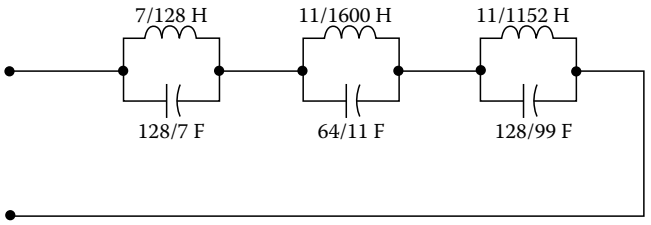


FIGURE 7.6 First Foster canonical form.

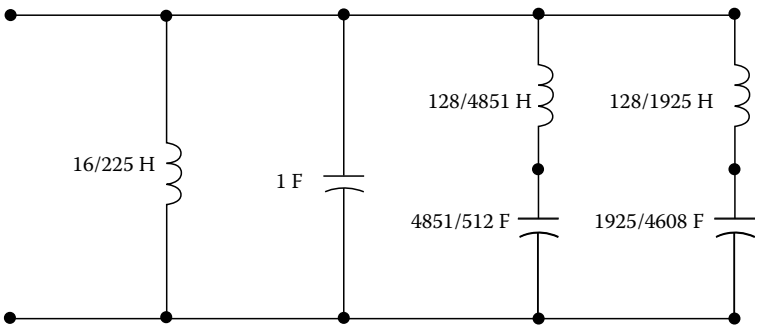


FIGURE 7.7 Second Foster canonical form.

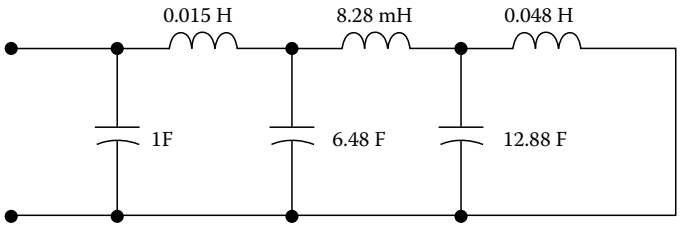


FIGURE 7.8 First Cauer canonical form.

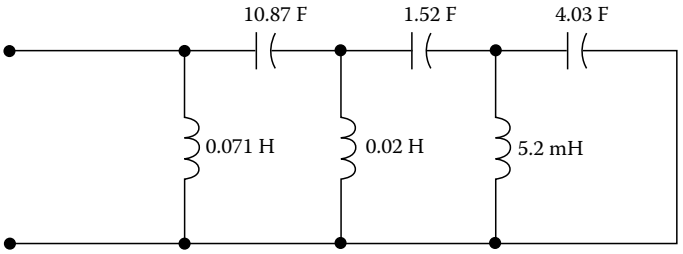


FIGURE 7.9 Second Cauer canonical form.

For the second Cauer canonical form, we rearrange the polynomials in ascending order of s, then expand the resulting function in a continued fraction, and obtain

$$Z(s) = \cfrac{1}{\cfrac{14.06}{s} + \cfrac{1}{\cfrac{0.092}{s} + \cfrac{1}{\cfrac{49.84}{s} + \cfrac{1}{\cfrac{0.66}{s} + \cfrac{1}{\cfrac{192.26}{s} + \cfrac{1}{\cfrac{0.248}{s}}}}}}} \tag{7.19}$$

The desired LC ladder is shown in Figure 7.9.

7.3 RC One-Port Networks

In this part, we exploit the properties of impedance functions of the RC one-ports from the known properties of the LCM one-ports of Section 7.2.

From a given RC one-port N_{RC}, we construct an LC one-port N_{LC} by replacing each resistor of resistance R_i by an inductor of inductance $L_i = R_i$. Suppose that we use loop analysis for both N_{RC} and N_{LC}, and choose the same set of loop currents. In addition, assume that the voltage source at the input port is traversed only by loop current ①. Then the input impedance $Z_{LC}(s)$ of N_{LC} is determined by the equation

$$Z_{LC}(s) = \frac{\tilde{\Delta}(s)}{\tilde{\Delta}_{11}(s)} \tag{7.20}$$

where

$\tilde{\Delta}$ is the loop determinant

$\tilde{\Delta}_{11}$ is the cofactor corresponding to loop current 1 in N_{LC}

Similarly, the input impedance $Z_{RC}(s)$ of N_{RC} can be written as

$$Z_{RC}(s) = \frac{\Delta(s)}{\Delta_{11}(s)} \tag{7.21}$$

where Δ is the loop determinant and Δ_{11} is the cofactor corresponding to loop current 1 in N_{RC}. It is not difficult to see that these loop determinants and cofactors are related by

$$\Delta(s) = \left.\frac{\tilde{\Delta}(p)}{p^r}\right|_{p^2=s} \tag{7.22a}$$

$$\Delta_{11}(s) = \left.\frac{\tilde{\Delta}_{11}(p)}{p^{r-1}}\right|_{p^2=s} \tag{7.22b}$$

where r is the order of the loop determinants $\tilde{\Delta}$ and Δ. Combining Equations 7.20 through 7.22 yields

$$Z_{RC}(s) = \left[\frac{1}{p} Z_{LC}(p)\right]_{p^2=s} \tag{7.23}$$

This relation allows us to deduce the properties of RC networks from those of the LC networks. Substituting Equations 7.6 and 7.7 in Equation 7.23, we obtain the general forms of the RC impedance function as

$$Z_{RC}(s) = H\frac{(s+\sigma_{z1})(s+\sigma_{z2})(s+\sigma_{z3})\cdots}{s(s+\sigma_{p1})(s+\sigma_{p2})\cdots} = H + \frac{K_0}{s} + \sum_{i=1}^{n}\frac{\hat{K}_i}{s+\sigma_i} \tag{7.24}$$

where $\sigma_{zj} = \omega_{zj}^2$, $\sigma_{pi} = \omega_{pi}^2$, $\sigma_i = \omega_i^2$ and $\hat{K}_i = 2K_i$, and from Equation 7.11

$$0 \le \sigma_{z1} < \sigma_{p1} < \sigma_{z2} < \sigma_{p2} < \cdots \tag{7.25}$$

Thus, the zeros and poles of an RC impedance alternate along the nonpositive real axis. This property turns out also to be sufficient to characterize the RC impedances.

THEOREM 7.2

A real rational function is the driving-point impedance of an RC one-port network if and only if all the poles and zeros are simple, lie on the nonpositive real axis, and alternate with each other, the first critical frequency (pole or zero) being a pole.

The slope of $Z_{RC}(\sigma)$ is found from Equation 7.24 to be

$$\frac{dZ_{RC}(\sigma)}{d\sigma} = -\frac{K_0}{\sigma^2} - \sum_{i=1}^{n}\frac{\hat{K}_i}{(\sigma+\sigma_i)^2} \tag{7.26}$$

which is negative for all values of σ, since K_0, and $\hat{k}_i$ are positive. Thus, we have

$$\frac{dZ_{RC}(\sigma)}{d\sigma} < 0 \tag{7.27}$$

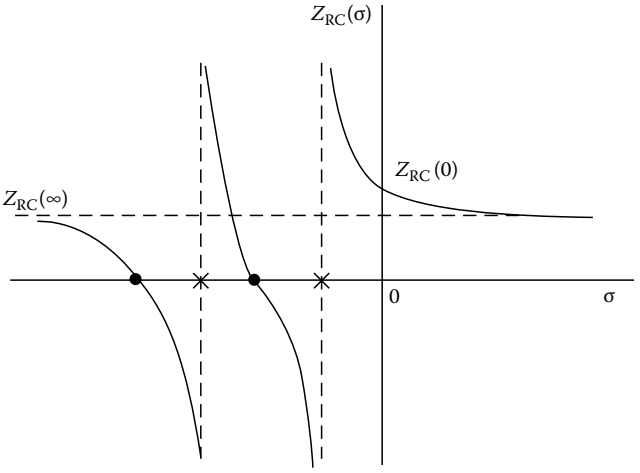


FIGURE 7.10 Plot of $Z_{RC}(\sigma)$ as a function of σ.

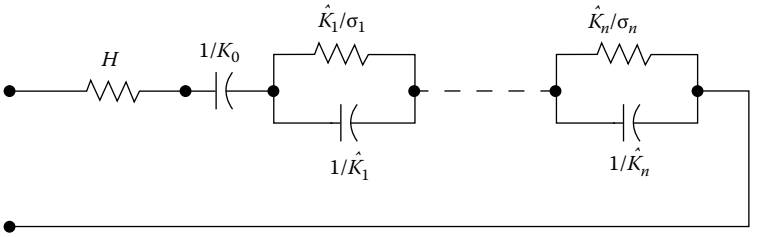


FIGURE 7.11 First Foster Canonical Form.

A plot of $Z_{RC}(\sigma)$ as a function of σ is shown in Figure 7.10. Since no poles and zeros exist along the positive real axis, we have

$$Z_{RC}(\infty) \leq Z_{RC}(0) \tag{7.28}$$

We now proceed to the realization of the RC one-port networks. Suppose that we are given $Z_{RC}(\sigma)$ as in Equation 7.24. By analogy to the LC case, this impedance can be realized by the one-port network of Figure 7.11 called the first Foster canonical form for the RC impedance.

To obtain the second Foster canonical form, we expand $Y_{RC}(s)/s$ in a partial fraction, where $Y_{RC}(s) = 1/Z_{RC}(s)$, and then multiply the resulting equation by s. The reason is that a direct partial-fraction expansion of $Y_{RC}(s)$ will result in negative residues. Proceeding in this way, we obtain

$$Y_{RC}(s) = K_0 + K_\infty s + \sum_{i=1}^{n} \frac{K_i s}{s + \sigma_i} \tag{7.29}$$

yielding the one-port network of Figure 7.12.

As before, in addition to the two Foster forms, RC ladder realizations are also possible. Following the LC case, we perform a continued-fraction expansion of $Z_{RC}(s)$ and obtain

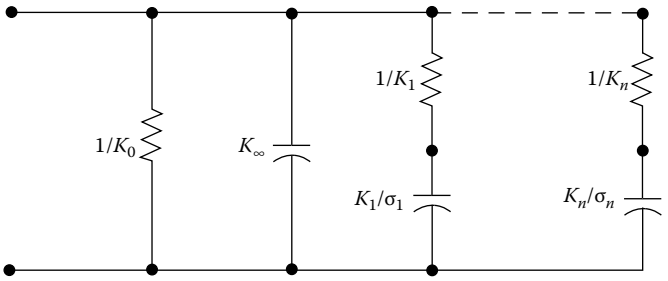


FIGURE 7.12 Second Foster canonical form.

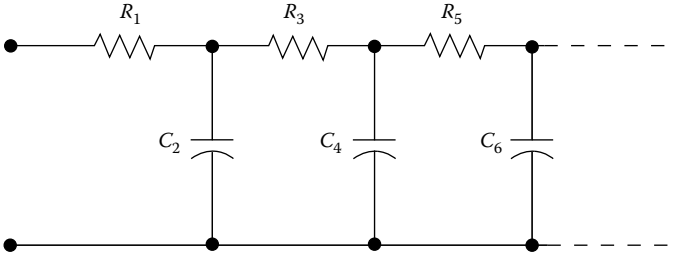


FIGURE 7.13 First Cauer canonical form.

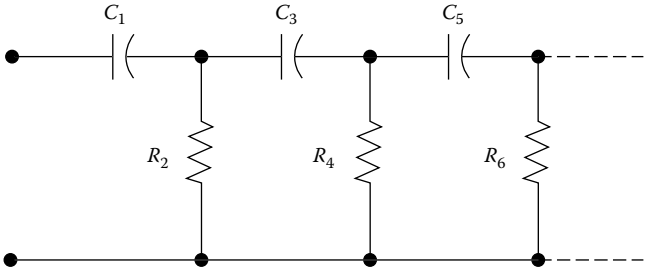


FIGURE 7.14 Second Cauer canonical form.

$$Z_{RC}(s) = R_1 + \cfrac{1}{C_2 s + \cfrac{1}{R_3 + \cfrac{1}{C_4 s + \cfrac{1}{\ddots}}}} \tag{7.30}$$

This expansion can be realized by the ladder network of Figure 7.13 known as the first Cauer canonical form for RC impedance. If we rearrange the terms of $Z_{RC}(s)$ so that the numerator and denominator polynomials appear in ascending order of s, the resulting continued-fraction expansion takes the general form

$$Z_{RC}(s) = \frac{1}{C_1 s} + \cfrac{1}{\cfrac{1}{R_2} + \cfrac{1}{\cfrac{1}{C_3 s} + \cfrac{1}{\cfrac{1}{R_4} + \cfrac{1}{\cfrac{1}{C_5 s} + \cfrac{1}{\ddots}}}}} \tag{7.31}$$

yielding the second Cauer canonical form of Figure 7.14.

Example 7.2

Consider the impedance function

$$Z(s) = \frac{s^2 + 12s + 35}{s^2 + 10s + 24} \tag{7.32}$$

To obtain the first Foster canonical form, we expand $Z(s)$ in partial fraction as

$$Z(s) = 1 + \frac{3/2}{s+4} + \frac{1/2}{s+6} \tag{7.33}$$

and obtain the one-port of Figure 7.15.

For the second Foster canonical form, the proper function to expand is $Y(s)/s$, yielding

$$Y(s) = \frac{24}{35} + \frac{s/10}{s+5} + \frac{3s/14}{s+7} \tag{7.34}$$

The corresponding realization is shown in Figure 7.16.

For the first Cauer canonical form, we expand $Z(s)$ in a continued fraction as

$$Z(s) = 1 + \cfrac{1}{\cfrac{s}{2} + \cfrac{1}{\cfrac{4}{9} + \cfrac{1}{\cfrac{27s}{2} + \cfrac{1}{\cfrac{1}{72}}}}} \tag{7.35}$$

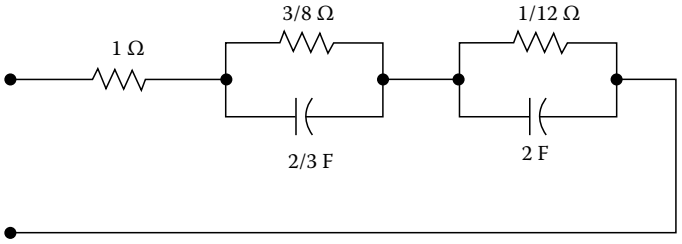


FIGURE 7.15 First Foster canonical form.

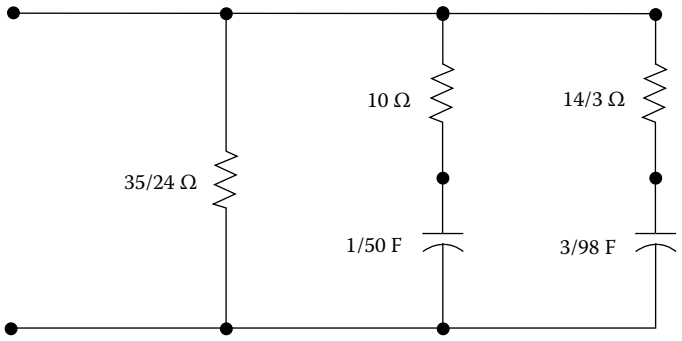


FIGURE 7.16 Second Foster canonical form.

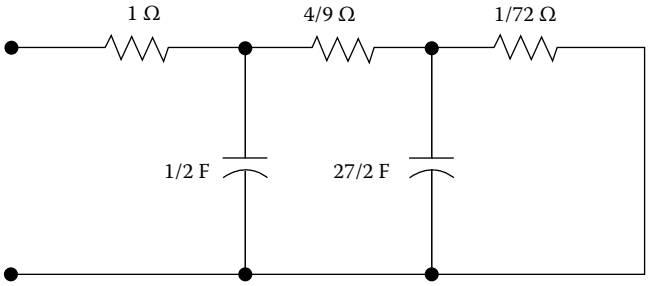


FIGURE 7.17 First Cauer canonical form.

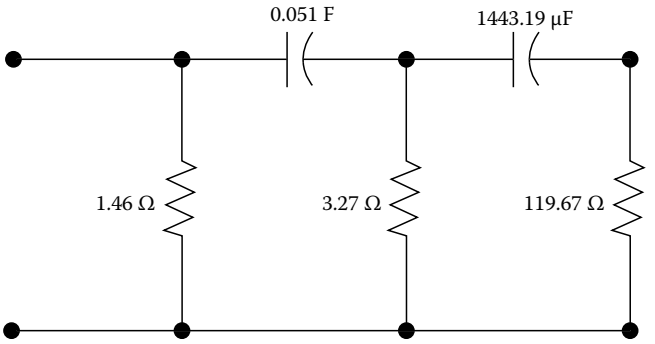


FIGURE 7.18 Second Cauer canonical form.

which can be realized by the ladder of Figure 7.17. To obtain the second Cauer canonical form, we rearrange the numerator and denominator polynomials in ascending order of s, and then expand in a continued fraction

$$Z(s) = \frac{35 + 12s + s^2}{24 + 10s + s^2} = \cfrac{1}{0.69 + \cfrac{1}{\cfrac{19.76}{s} + \cfrac{1}{0.306 + \cfrac{1}{\cfrac{692.91}{s} + \cfrac{1}{8.36 \times 10^{-3}}}}}} \tag{7.36}$$

yielding the ladder of Figure 7.18.

References

1. R. M. Foster, A reactance theorem, *Bell System Tech. J.*, 3, 259–267, 1924.
2. W. Cauer, *Synthesis of Linear Communication Networks,* New York: McGraw-Hill, Chapters 5 and 6, 1958.

8

Two-Part Synthesis by Ladder Development

8.1 Introduction **8**-1
8.2 LC Ladder **8**-2
8.3 RC Ladder **8**-7
8.4 Parallel or Series Ladders **8**-10
Reference **8**-16

Wai-Kai Chen
University of Illinois at Chicago

8.1 Introduction

In two-port synthesis, specifications are often given in terms of the transfer functions such as the transfer voltage ratio, transfer current ratio, transfer impedance, or transfer admittance. The actual realization, however, is accomplished by means of the y- or z-parameters. Figure 8.1 shows a two-port network driven by a voltage source with output terminating in an impedance $Z_2(s)$. It is straightforward to show that the transfer voltage ratio function $G_{12}(s)$ can be expressed in terms of its y-parameters $y_{ij}(s)$ or z-parameters $z_{ij}(s)$ by the equation

$$G_{12}(s) = \frac{V_2}{V_1} = \frac{-y_{21}}{y_{22} + Y_2} \tag{8.1}$$

where $Y_2(s) = 1/Z_2(s)$. When the output is open-circuited, Equation 8.1 becomes

$$G_{12}(s) = \frac{V_2}{V_1} = \frac{-y_{21}}{y_{22}} = \frac{z_{21}}{z_{11}} \tag{8.2}$$

Likewise, the transfer current ratio $\alpha_{12}(s)$ can be expressed as

$$\alpha_{12}(s) = -\frac{I_2}{I_1} = \frac{z_{21}}{z_{22} + Z_2} \tag{8.3}$$

The zeros of transmission of a two-port network are defined as the frequencies at which the two-port results in zero output for a finite input. They play an important role in ladder development. There are many ways of producing zeros of transmission. One possibility to prevent the input signal from reaching the output is by shorting together all transmission paths or by opening all transmission paths by means of a series or parallel resonance. Another possibility is that signals transmitted by different paths cancel at the output.

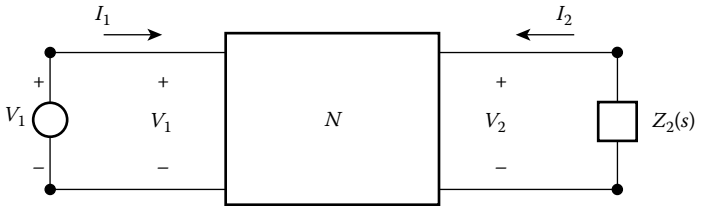


FIGURE 8.1 Terminated two-port network.

Observe from Equations 8.1 and 8.2 that zero output, $V_2 = 0$, implies a zero for each of these functions. Therefore, zeros of transmission are zeros of $-y_{21}$ or z_{21} provided y_{21} and y_{22} or z_{21} and z_{11} have the same poles. For the ladder network, the transmission can be interrupted only by the short circuit in a shunt arm or an open circuit in a series arm. The short circuit of a shunt arm corresponds to the pole frequencies of its admittances, whereas the open circuit of a series arm corresponds to the pole frequencies of its impedances. Therefore, the zeros of transmission of a ladder network can be identified directly with the zeros of the impedances of the shunt arms and the poles of the impedances of the series arms. For the LC ladder, all zeros of transmission lie on the $j\omega$-axis, and for the RC ladder they are on the nonpositive real axis of the complex-frequency s-plane.

8.2 LC Ladder

For LC ladders, the conditions imposed on $-y_{21}$ and y_{22} or z_{21} and z_{22} are that the driving-point functions y_{22} and z_{22} be positive real with poles and zeros interlaced on the $j\omega$-axis. The transfer functions $-y_{21}$ and z_{21}, assuming to have the same poles as y_{22} or z_{22}, must have all of its zeros on the $j\omega$-axis. However, these zeros need not be interlaced with the poles and they may not be simple. Our strategy in realization is that of carrying out the driving-point synthesis of y_{22} or z_{22}, using Cauer ladder development method, in such a way that the zeros of transmission are realized at the same time. The procedure consists of two steps: a zero-shifting step and a zero-producing step, as described below.

Zero shifting by partial removal. Consider an impedance of Equation 7.6, the partial-fraction expansion of which is given in Equation 7.7. The first term on the right-hand side of Equation 7.7 is due to the contribution of the pole at the infinity. If this term Hs is subtracted from $Z(s)$, the resulting function $Z(s) - Hs$ is devoid of the pole at the infinity. We say that the pole at infinity has been removed completely. Instead of complete removal of this pole, suppose that we subtract a fraction of the terms Hs from $Z(s)$ by introducing a constant k_p such that

$$Z_1(s) = Z(s) - k_p Hs, \quad k_p < 1 \tag{8.4}$$

We say that the pole at infinity has been partially removed or weakened. The function $Z_1(s)$ that results from the partial removal of the pole at infinity still possesses the pole at infinity. Since all the zeros of $Z_1(s)$ are again located on the $j\omega$-axis, these zeros are found by substituting $s = j\omega$ in Equation 8.4,

$$X_1(\omega) = X(\omega) - k_p H\omega \tag{8.5}$$

where $Z_1(j\omega) = jX_1(\omega)$. The zeros of $X_1(\omega)$ are values of ω satisfying the equation

$$X(\omega) = k_p H\omega \tag{8.6}$$

Solutions to this equation are found graphically from the intersections of the curves $X(\omega)$ and $k_p H\omega$, as depicted in Figure 8.2. Observe that all the zeros in the resulting function are shifted toward the pole

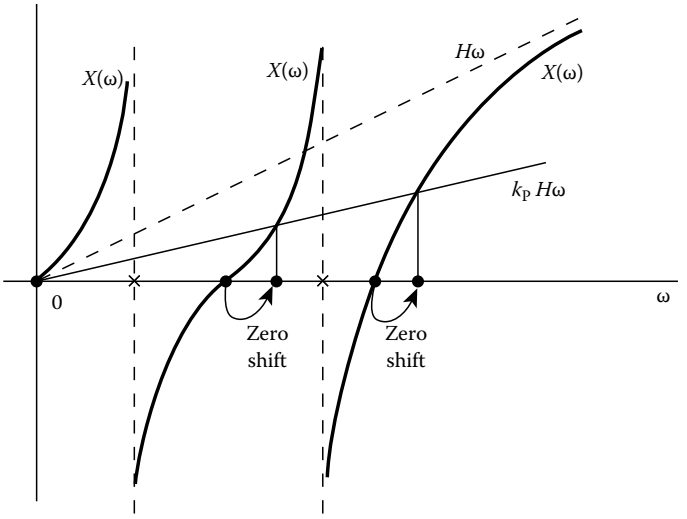


FIGURE 8.2 Zero shifting by weakening the pole at the infinity.

being weakened. The amount of shift of the zeros from their original positions depends on the value of k_p and the proximity of a zero to the pole being weakened.

We next consider the term K_0/s in Equation 7.7 due to the pole at the origin. The partial removal of this pole is equivalent to the operation

$$Z_2(s) = Z(s) - k_p \frac{K_0}{s} \tag{8.7}$$

As before, the zeros of $Z_2(s)$ are defined by the intersections of the curves $X(\omega)$ and $-k_p(K_0/\omega)$ with ω,

$$X(\omega) = -k_p \frac{K_0}{\omega} \tag{8.8}$$

as illustrated in Figure 8.3. Observe again that the zeros are shifted toward the pole being weakened, which is at the origin.

Finally, for the finite nonzero poles, the corresponding factors take the general form $2K_i s/(s^2 + \omega_i^2)$. The partial removal of this pair of complex conjugate poles results in the new function

$$Z_3(s) = Z(s) - k_p \frac{2K_i s}{s^2 + \omega_i^2}, \quad k_p < 1 \tag{8.9}$$

The zeros of this function are defined by the intersections of the plots of $X(\omega)$ and $-k_p(2K_i\omega/(\omega^2 - \omega_i^2))$ with ω,

$$X(\omega) = -k_p \frac{2K_i \omega}{\omega^2 - \omega_i^2}, \quad k_p < 1 \tag{8.10}$$

as illustrated in Figure 8.4.

Our conclusion is that the partial removal of a pole shifts the zeros toward that pole. The amount of shift depends on the value of k_p and the proximity of a zero to that pole, but in no case can a zero be shifted beyond an adjacent pole.

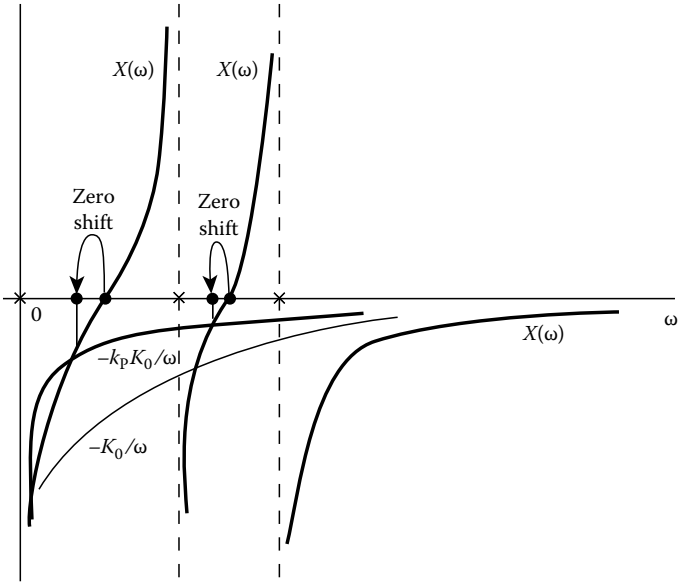


FIGURE 8.3 Zero shifting by weakening the pole at the origin.

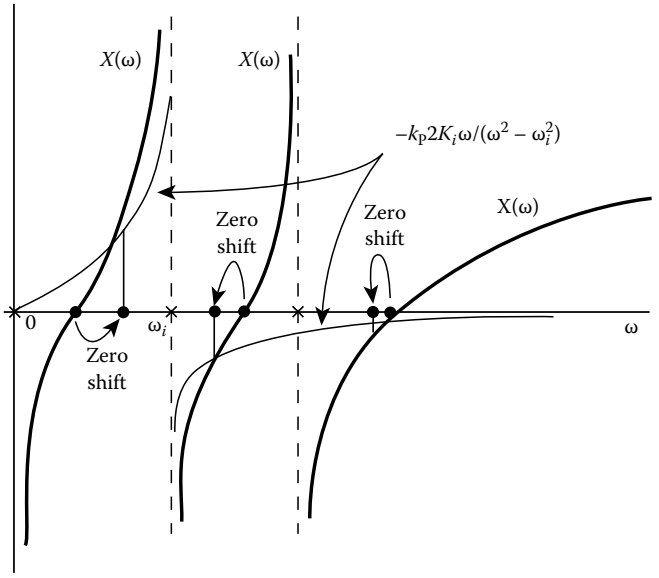


FIGURE 8.4 Zero shifting by weakening a finite nonzero pole.

Zero producing by complete pole removal. After a zero of transmission has been shifted to a desired location by the partial removal of an appropriate pole, the realization of this zero of transmission is accomplished by the complete removal of the pole of the reciprocal function corresponding to the shifted zero. For the LC ladder two-ports, a series combination of an inductor L and a capacitor C produces a zero at its resonant frequency $\omega = 1/\sqrt{LC}$ and this network is used in the shunt arm in the ladder to produce the desired zero of transmission. Likewise, the parallel connection of L and C yields

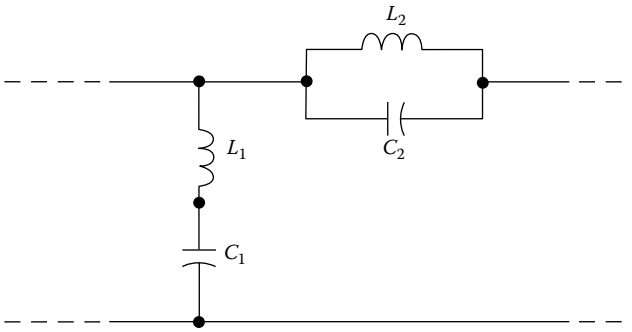


FIGURE 8.5 Zero producing sections in a ladder network.

an infinite impedance at its resonant frequency, and is used in the series arm of the ladder. They are shown in Figure 8.5.

Example 8.1

We wish to design a lossless two-port network terminated in a 100-Ω resistor to meet the specifications for the transfer voltage-ratio function

$$G_{12}(s) = \frac{V_2}{V_1} = K\frac{s^2+4}{s^3+4s^2+9s+4} \tag{8.11}$$

within a multiplicative constant.

Since magnitude scaling does not affect the voltage transfer ratio, without loss of generality, we first assume that the terminating resistor is 1 Ω. Equation 8.11 can be rewritten as

$$G_{12}(s) = K\frac{\dfrac{s^2+4}{s(s^2+9)}}{\dfrac{4(s^2+1)}{s(s^2+9)}+1} = \frac{-y_{21}}{y_{22}+Y_2} = \frac{-y_{21}}{y_{22}+1} \tag{8.12}$$

We can make the following identifications:

$$-y_{21}(s) = K\frac{s^2+4}{s(s^2+9)}, \quad y_{22}(s) = 4\frac{s^2+1}{s(s^2+9)} \tag{8.13}$$

Both functions have the same poles at $s=0$ and $s = \pm j3$ and the zeros of transmission of the ladder are located at $s=\pm j2$ and $s=\infty$. These poles and zeros are shown in Figure 8.6. To realize the zero of transmission at $s=\infty$, we remove the pole of $z_1(s) = 1/y_{22}(s)$ at $s=\infty$.

$$z_1(s) = \frac{1}{y_{22}(s)} = \frac{s(s^2+9)}{4(s^2+1)} = \frac{2s}{s^2+1} + \frac{s}{4} \tag{8.14}$$

After subtracting the term $s/4$ corresponding to an inductor of inductance 1/4 H shown in Figure 8.7 from $z_1(s)$, the remaining impedance $z_2(s)$ is found to be

$$z_2(s) = \frac{2s}{s^2+1} \tag{8.15}$$

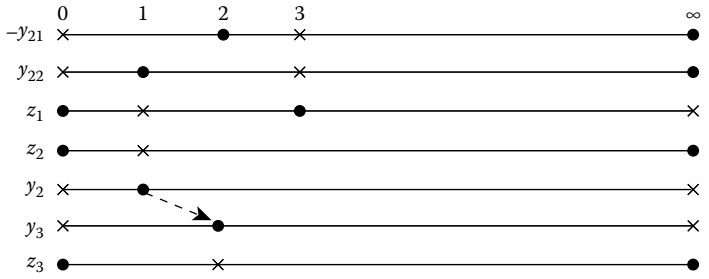


FIGURE 8.6 Poles and zeros of LC immittances.

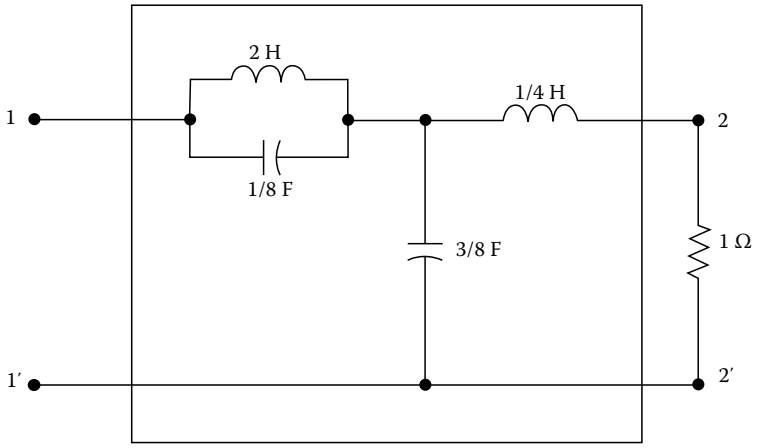


FIGURE 8.7 LC ladder realization of the transfer voltage ratio (Equation 8.11).

To realize the zero of transmission at $s = \pm j2$, we partially weaken the pole of the admittance

$$y_2(s) \equiv \frac{1}{z_2(s)} = \frac{s^2 + 1}{2s} \tag{8.16}$$

at s = ∞ in order to shift the zero at $s = \pm j$ to $s = \pm j2$, or

$$y_3(\pm j2) = y_2(s) - k_p \frac{1}{2} s \Big|_{s=\pm j2} = \frac{-4 + 1}{\pm j2 \times 2} \mp k_p \frac{1}{2} j2 = 0 \tag{8.17}$$

yielding $k_p = 3/4$. The new function becomes

$$y_3(s) = \frac{s^2 + 1}{2s} - \frac{3}{8} s = \frac{s^2 + 4}{8s} \tag{8.18}$$

after the removal of a shunt capacitor of capacitance 3/8 F, as shown in Figure 8.7. The factor $(s^2 + 4)$ in the numerator was anticipated because our objective was to produce a zero in the driving-point admittance $y_3(s)$ at $s = \pm j2$. To realize this zero, we consider the reciprocal function $z_3(s) = 1/y_3(s)$ by complete removal of its pole at $s = \pm j2$. This yields a parallel connection of $L = 2$ H and $C = 1/8$ F shown in Figure 8.7. The final realization is obtained by magnitude-scaling by a factor of 100. The realized constant K is found from the network to be $K = 1$.

8.3 RC Ladder

We now consider the realization of RC ladder with prescribed $-y_{21}(s)$ and $y_{22}(s)$ or $z_{21}(s)$ and $z_{22}(s)$. Following the LC case, the zero shifting for the RC driving-point functions is accomplished by one or any combination of the following three operations:

1. Partial removal of a constant $Z(\infty)$ from $Z(s)$
2. Partial removal of a constant $Y(0)$ from $Y(s)$
3. Partial removal of a pole from $Z(s)$ or $Y(s)$

The first operation permits a series resistance to be removed so that the resulting impedance is still positive real and possesses a desired zero of transmission:

$$Z_1(s) = Z(s) - k_p Z(\infty), \quad k_p \leq 1 \tag{8.19}$$

the zeros of which occur at those values of σ satisfying

$$Z(\sigma) = k_p Z(\infty), \quad k_p \leq 1 \tag{8.20}$$

Observe that from Figure 8.8 all zeros are shifted toward $s = \sigma = -\infty$ by the partial removal of $Z(\infty)$ because the slope is negative for the RC impedances.

The second operation corresponds to the removal of a shunt resistance, and the remaining admittance

$$Y_1(s) = Y(s) - k_p Y(0), \quad k_p \leq 1 \tag{8.21}$$

is still positive real, the zeros of which occur at those values of σ satisfying

$$Y(\sigma) = k_p Y(0), \quad k_p \leq 1 \tag{8.22}$$

Observe from Figure 8.9 that these zeros again are shifted toward $s = 0$ in relation to those of $Y(s)$.

Finally, the partial removal of a pole of $Z(s)$ results in a parallel connection of a resistor and a capacitor, and the remaining impedance becomes

$$Z_2(s) = Z(s) - k_p \frac{K_i}{s + \sigma_i}, \quad k_p < 1 \tag{8.23}$$

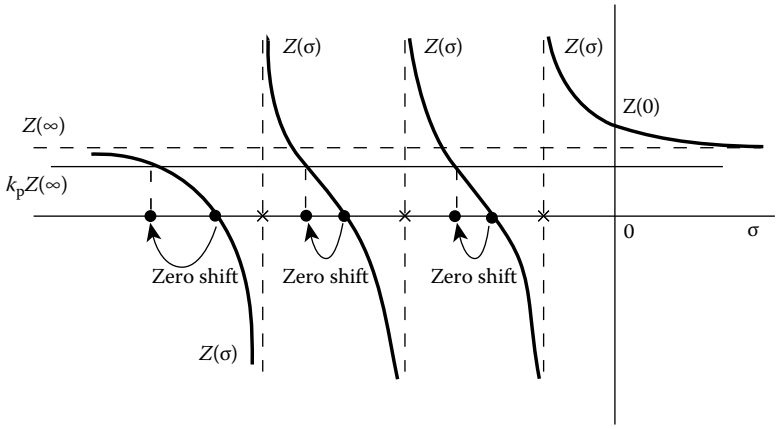


FIGURE 8.8 Zero shifting by the partial removal of $Z(\infty)$.

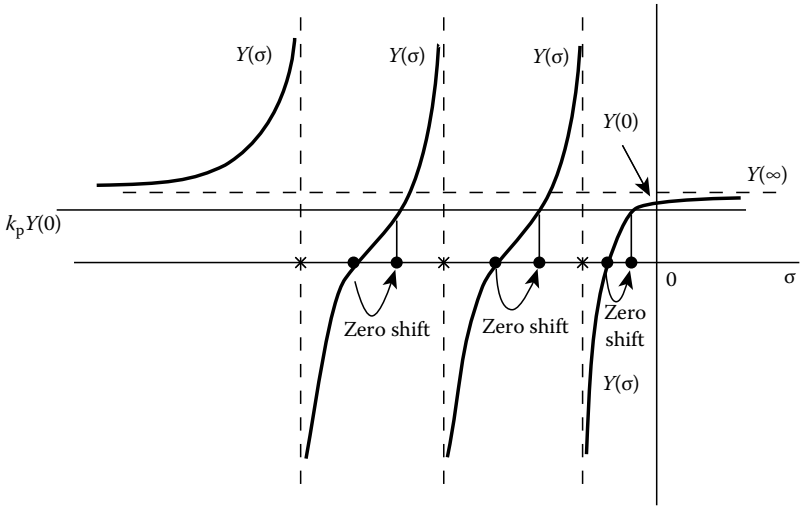


FIGURE 8.9 Zero shifting by partial removal of $Y(0)$.

the zeros of which occur at those values of σ satisfying

$$Z(\sigma) = k_p \frac{K_i}{\sigma + \sigma_i}, \quad k_p < 1 \tag{8.24}$$

As indicated in Figure 8.10, these zeros are again shifted toward the pole at $s = -\sigma_i$ being partially weakened.

Therefore, the partial removal of a constant or a pole shifts the zeros of the remaining function toward the quantity being weakened. The amount of shift depends on the value of k_p and the proximity of a zero to that quantity. Once a zero is shifted to an appropriate location, its realization is accomplished by the complete removal of the pole of the reciprocal function corresponding to the shifted zero.

Example 8.2

We wish to realize the transfer impedance

$$Z_{12}(s) = K \frac{s(s+1)}{2s^2 + 18s + 34} \tag{8.25}$$

of an RC two-port network terminated in a 1 Ω resistor.

From Equation 8.3, the transfer impedance can be written as

$$Z_{12}(s) = \frac{V_2}{I_1} = \frac{-I_2 \times 1}{I_1} = \frac{-I_2}{I_1} = \frac{z_{21}}{z_{22} + 1} = \frac{K \dfrac{s(s+1)}{(s+2)(s+5)}}{\dfrac{(s+3)(s+8)}{(s+2)(s+5)} + 1} \tag{8.26}$$

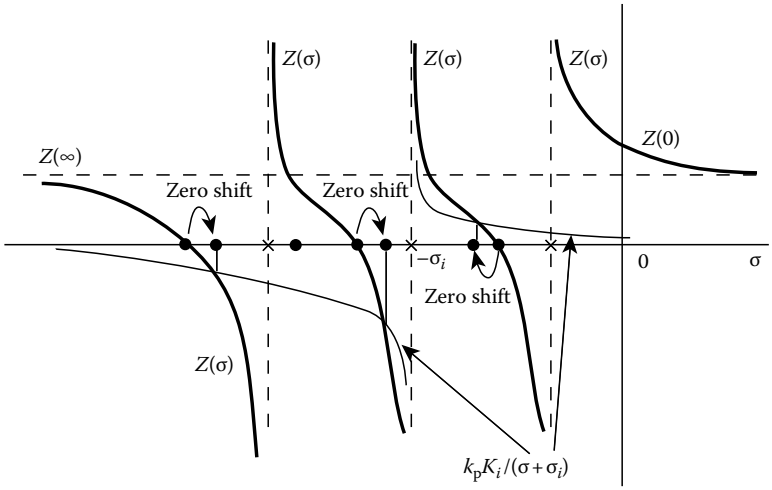


FIGURE 8.10 Zero shifting by partial removal of a pole of an RC impedance.

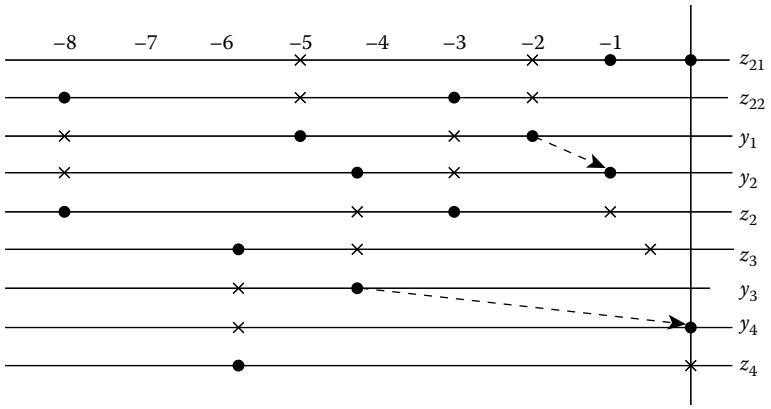


FIGURE 8.11 Zeros and poles of RC immittances.

identifying

$$z_{21}(s) = K\frac{s(s+1)}{(s+2)(s+5)}, \quad z_{22}(s) = \frac{(s+3)(s+8)}{(s+2)(s+5)} \tag{8.27}$$

The zeros of transmission are located at $s=0$ and $s=-1$. Suppose that we wish to realize these zeros in the order of -1 and 0. For this we consider the reciprocal function $y_1(s) = 1/z_{22}(s)$, the zeros. and poles of which are shown in Figure 8.11. Clearly, the zero at $s=-2$ can be shifted to $s=-1$ by partial removal of the constant $y_1(0) = 5/12$,

$$y_2(-1) = y_1(-1) - k_p y_1(0) = 0 \tag{8.28}$$

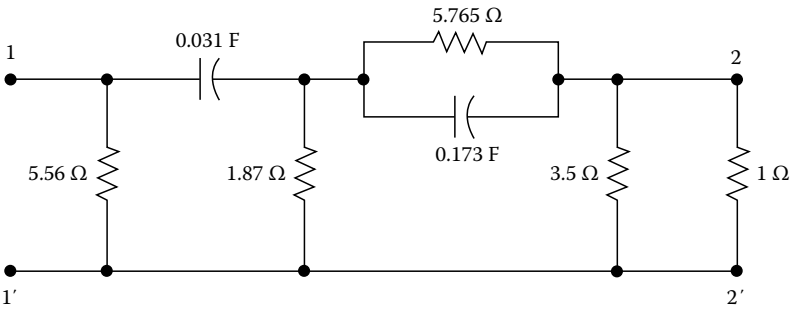


FIGURE 8.12 RC ladder realization of the transfer impedance $Z_{12}(s)$.

obtaining $k_p = 24/35$ and

$$y_2(s) = y_1(s) - \frac{24}{35}y_1(0) = \frac{(s+1)(5s+22)}{7(s^2+11s+24)} \tag{8.29}$$

This admittance has a zero at $s = -1$, as expected. The partially removed constant $k_p y_1(0)$ corresponds to a shunt resistor of resistance 3.5 Ω. To realize the zero of transmission at $s = -1$, we consider the reciprocal function $z_2(s) = 1/y_2(s)$ by the complete removal of its pole at $s = -1$. The remaining impedance $z_3(s)$ becomes

$$z_3(s) = z_2(s) - \frac{\frac{98}{17}}{s+1} = \frac{7}{5} \times \frac{s+\frac{100}{17}}{s+\frac{22}{5}} \tag{8.30}$$

The removed pole corresponds to a parallel connection of a resistor of resistance 98/17 Ω and a capacitor of capacitance 17/98 F, as illustrated in Figure 8.12.

For the zero of transmission at $s = 0$, we consider the reciprocal function $y_3(s) = 1/z_3(s)$. The zero of $y_3(s)$ at s = −22/5 can be shifted to 0 by partial removal of the constant $y_3(0) = 187/350$, or

$$y_4(0) = y_3(0) - k_p y_3(0) = 0 \tag{8.31}$$

yielding $k_p = 1$. The remaining admittance $y_4(s)$ becomes

$$y_4(s) = y_3(s) - y_3(0) = \frac{\frac{9}{50}s}{s+\frac{100}{17}} \tag{8.32}$$

showing a zero at the origin, as anticipated. This zero is realized by the complete removal of the pole at the origin of its reciprocal $z_4(s) = 1/y_4(s)$, yielding

$$z_5(s) = z_4(s) - \frac{32.68}{s} = 5.56 \tag{8.33}$$

The complete realization is presented in Figure 8.12, from which the constant K is found to be $K = 1$.

8.4 Parallel or Series Ladders

The zeros of transmission of the LC ladders are restricted to the $j\omega$-axis and those of the RC ladders to the negative real axis of the s-plane. For complex zeros of transmission such as those needed for certain phase-correction applications, they cannot be realized by a single LC or RC ladder because there is only a single transmission path from the input to the output. The use of parallel or series ladders, on the other

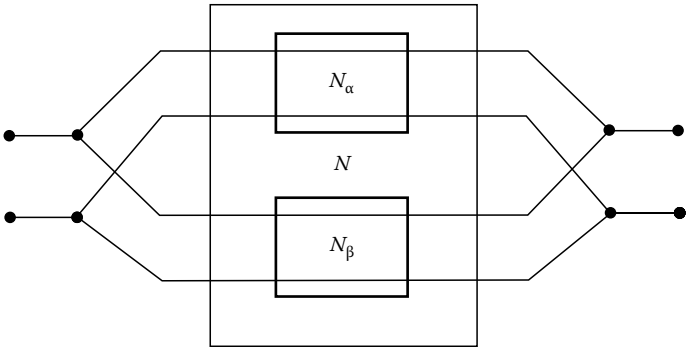


FIGURE 8.13 Parallel connection of two ladder networks.

hand, provides a conduit for multiple path signal transmission, so that the signals arriving at the output through the various paths may cancel one another, resulting in the zero output for a finite input. Therefore, they are capable of producing complex zeros of transmission. This structure was first suggested by Guillemin [1].

Figure 8.13 is the parallel connection of the ladder networks N_α and N_β. The y-parameters y_{ij} of the composite two-port N can be expressed in terms of those $y'_{ij\alpha}$ and $y'_{ij\beta}$ of the component two-ports N_α and N_β by the equation

$$y_{ij} = y'_{ij\alpha} + y'_{ij\beta}, \quad i,j = 1,2 \tag{8.34}$$

Thus, to realize $-y_{21}(s)$ and $y_{22}(s)$, we may separate them into pairs like $-y'_{21\alpha}$, $y'_{22\alpha}$ and $-y'_{21\beta}$, $y'_{22\beta}$ realize an individual pair as an LC or RC ladder. Then connect these individual ladders in parallel to realize $-y_{21}(s)$ and $y_{22}(s)$. In order for the procedure to succeed, we must resolve the following problem. Recall than in the Cauer development of LC and RC ladders, $-y_{21}(s)$ is realized only within the multiplicative constant k. Thus, the transfer admittances realized by the component two-ports actually will be $-k_\alpha y'_{21\alpha}$and $-k_\beta y'_{21\beta}$. The sum of these two functions will not result in the desired $-ky_{21}$ unless $k = k_\alpha = k_\beta$. To circumvent this difficulty, we introduce an additional degree of freedom by adjusting the admittance level of the α-ladder N_α by a factor b_α and the β-ladder N_β by b_β. Then the functions of the resulting realizations become

$$-y'_{21\alpha} = -b_\alpha k_\alpha y_{21\alpha}, \quad y'_{22\alpha} = b_\alpha y_{22\alpha} \tag{8.35a}$$

$$-y'_{21\beta} = -b_\beta k_\beta y_{21\beta}, \quad y'_{22\beta} = b_\beta y_{22\beta} \tag{8.35b}$$

where $y_{ij} = y_{ij\alpha} + y_{ij\beta}$, $i, j = 1, 2$. Substituting these in Equation 8.34 gives

$$y_{21} = b_\alpha k_\alpha y_{21\alpha} + b_\beta k_\beta y_{21\beta} \tag{8.36a}$$

$$y_{22} = b_\alpha y_{22\alpha} + b_\beta y_{22\beta} \tag{8.36b}$$

Our objective is to choose b_α and b_β to satisfy the above equations, once $k_\alpha$and k_β are known. One way to meet these requirements is to let $y'_{22\alpha}$ and $y'_{22\beta}$ have the same zeros and poles as y_{22} but different scale factors such that $y'_{22\alpha} = b_\alpha y_{22}$ and $y'_{22\beta} = b_\beta y_{22}$, obtaining from Equation 8.36b

$$b_\alpha + b_\beta = 1 \tag{8.37}$$

Since we can only realize y_{21} within a multiplicative constant, we replace y_{21} by ky_{21} in Equation 8.36a, and set

$$b_\alpha k_\alpha = b_\beta k_\beta = k \tag{8.38}$$

These two equations can be solved to yield the desired scale factors b_α and b_β.

In general, for m ladders in parallel, we require

$$b_1 + b_2 + \cdots + b_m = 1 \tag{8.39a}$$

$$b_1 k_1 = b_2 k_2 = \cdots = b_m k_m = k \tag{8.39b}$$

where

b_i is the admittance scale factor for the ith ladder

k_i is the realized multiplicative constant of the transfer admittance of the ith ladder

Once k_i are known, these m simultaneous equations can be solved for the m unknowns b_i, and the admittance level of each ladder can be scaled accordingly. The scaled ladders are connected in parallel to realize the $-y_{21}(s)$ specifications within the multiplicative constant k, and the $y_{22}(s)$ specifications exactly.

Similar results are obtained by using the z-parameters $z_{ij}(s)$ and the series connection of the component two-port networks, the details of which are omitted. However, a design example will be presented below.

Example 8.3

We wish to realize an RC two-port network to meet the following specifications:

$$-y_{21}(s) = k\frac{s^2 + 1}{(s + 5)(s + 9)} \tag{8.40a}$$

$$y_{22}(s) = \frac{(s + 3)(s + 7)}{(s + 5)(s + 9)} \tag{8.40b}$$

Since the zeros of transmission are located at $s = \pm j1$, they cannot be realized by a single RC ladder. For our purposes, we choose the y-parameters of the component two-ports as

$$-y_{21\alpha}(s) = k_\alpha \frac{s^2}{(s + 5)(s + 9)}, \quad y_{22\alpha}(s) = y_{22}(s) \tag{8.41a}$$

$$-y_{21\beta}(s) = \frac{k_\beta}{(s + 5)(s + 9)}, \quad y_{22\beta}(s) = y_{22}(s) \tag{8.41b}$$

For the α-ladder, since the zeros of transmission are all at the origin, they can be realized by the second Cauer canonical form shown in Figure 8.14 with $k_\alpha = 0.043$. For the β-ladder, since the zeros of transmission are all at the infinity, they can be realized by the first Cauer canonical form of Figure 8.15 with $k_\beta = 21$.

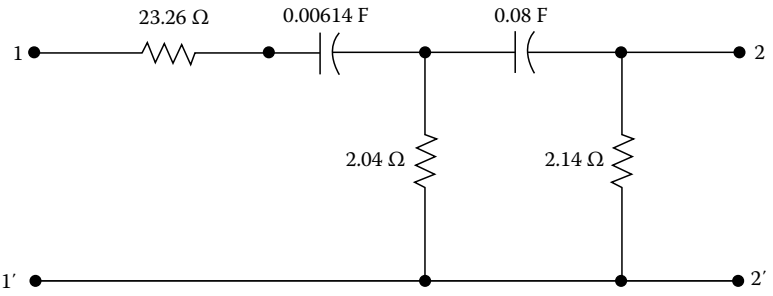


FIGURE 8.14 RC α-ladder realization.

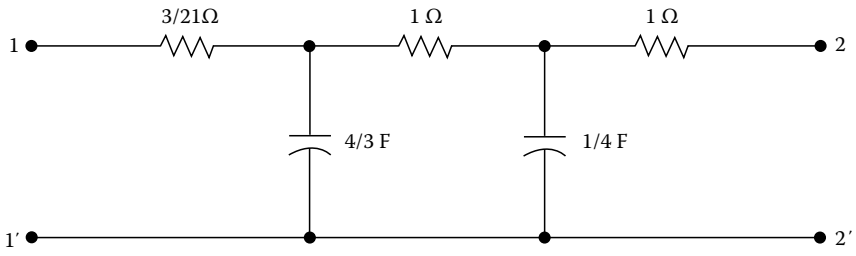


FIGURE 8.15 RC β-ladder realization.

Our next task is to adjust the admittance level of the individual ladders so that when they are connected in parallel, the $y_{22}(s)$ specifications are realized exactly, and the $-y_{21}(s)$ specifications are realized to within a multiplicative constant k. Appealing to Equations 8.37 and 8.38, we have

$$b_\alpha + b_\beta = 1 \tag{8.42a}$$

$$0.043 b_\alpha = 21 b_\beta = k \tag{8.42b}$$

yielding

$$b_\alpha = 0.998 \cong 1, \quad b_\beta = 2.043 \times 10^{-3} \tag{8.43}$$

We now adjust the admittance level of the β-ladder by a factor $b_\beta = 2.043 \times 10^{-3}$, leaving the α-ladder intact. The final realization is achieved by the parallel connection of the α-ladder and the resulting β-ladder shown in Figure 8.16. The realized multiplicative constant for the overall transfer admittance $-y_{21}(s)$ is found to be $k = 0.043$.

Example 8.4

We wish to realize the open-circuit transfer voltage ratio

$$G_{12}(s) = \frac{z_{21}(s)}{z_{11}(s)} = k\frac{s^2 - 2s + 10}{s^2 + 8s + 15} \tag{8.44}$$

by an RC two-port network.

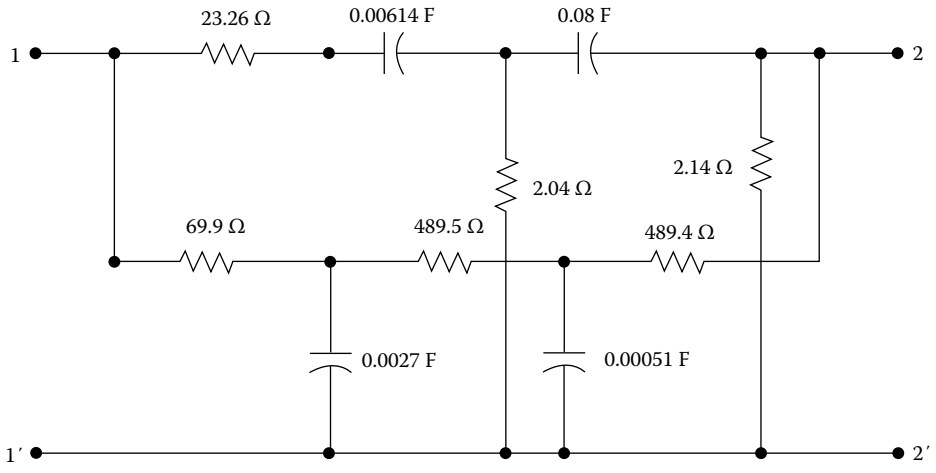


FIGURE 8.16 Parallel RC ladder realization of $-y_{21}(s)$ and $y_{22}(s)$.

First, we multiply the numerator and denominator of $G_{12}(s)$ by the factor $(s+2)$ to yield

$$G_{12}(s) = \frac{z_{21}(s)}{z_{11}(s)} = k\frac{s^3 + 6s + 20}{(s+2)(s+3)(s+5)} \tag{8.45}$$

and then divide by $(s+1)(s+2.5)(s+4)$ in order to make the following identifications:

$$z_{21}(s) = k\frac{s^3 + 6s + 20}{(s+1)(s+2.5)(s+4)} \tag{8.46a}$$

$$z_{11}(s) = \frac{(s+2)(s+3)(s+5)}{(s+1)(s+2.5)(s+4)} \tag{8.46b}$$

We next decompose the pairs into three pairs as follows:

$$z_{21\alpha}(s) = k_\alpha \frac{s^3}{(s+1)(s+2.5)(s+4)}, \quad z_{11\alpha}(s) = z_{11}(s) \tag{8.47a}$$

$$z_{21\beta}(s) = k_\beta \frac{6s}{(s+1)(s+2.5)(s+4)}, \quad z_{11\beta}(s) = z_{11}(s) \tag{8.47b}$$

$$z_{21\gamma}(s) = k_\gamma \frac{20}{(s+1)(s+2.5)(s+4)}, \quad z_{11\gamma}(s) = z_{11}(s) \tag{8.47c}$$

1. *The* α*-ladder*. Since all the zeros of transmission are located at the origin, it can be realized by the second Cauer canonical form of Figure 8.17 with $k_\alpha = 1$.
2. *The* β*-ladder*. Since one of the zeros of transmission is located at the origin, and two others at the infinity, the first half of the β-ladder can be realized as the second Cauer canonical form and the second half of the β-ladder by the first Cauer canonical forms as illustrated in Figure 8.18 with $k_\beta = 0.0117$.
3. *The* γ*-ladder*. Since all of its zeros of transmission are located at the infinity, it can be realized by the first Cauer canonical form of Figure 8.19 with $k_\gamma = 0.0145$.

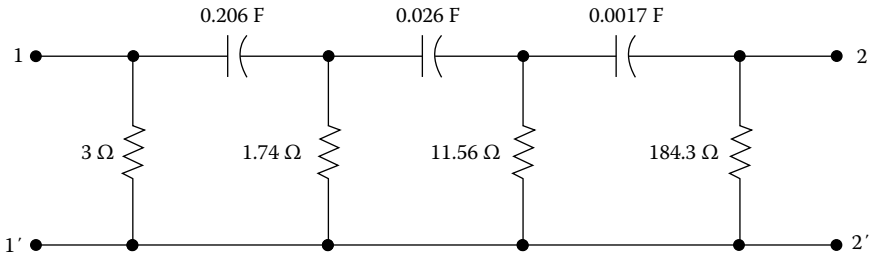


FIGURE 8.17 α-Ladder.

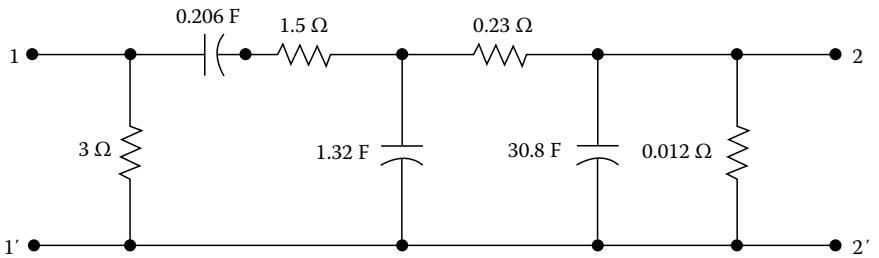


FIGURE 8.18 β-Ladder.

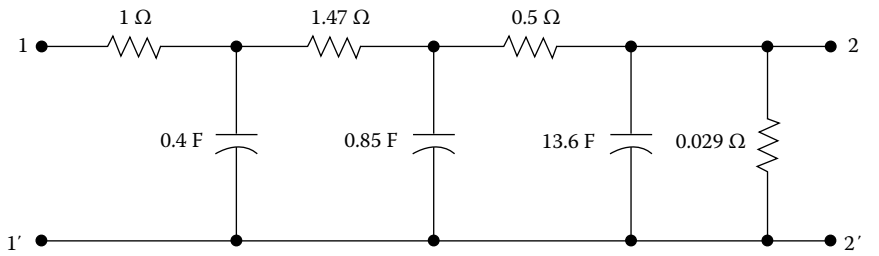


FIGURE 8.19 γ-Ladder.

We next adjust the admittance level of the ladders so that when they are connected in parallel, the desired specifications are realized. From Equation 8.39 we require that

$$b_\alpha + b_\beta + b_\gamma = 1 \tag{8.48a}$$

$$b_\alpha k_\alpha = b_\beta k_\beta = b_\gamma k_\gamma = k \tag{8.48b}$$

These equations can be solved to yield

$$b_\alpha = 0.0064, \quad b_\beta = 0.552, \quad b_\gamma = 0.4416 \tag{8.49}$$

with $k = 0.0064$. The final two-port network is shown in Figure 8.20.

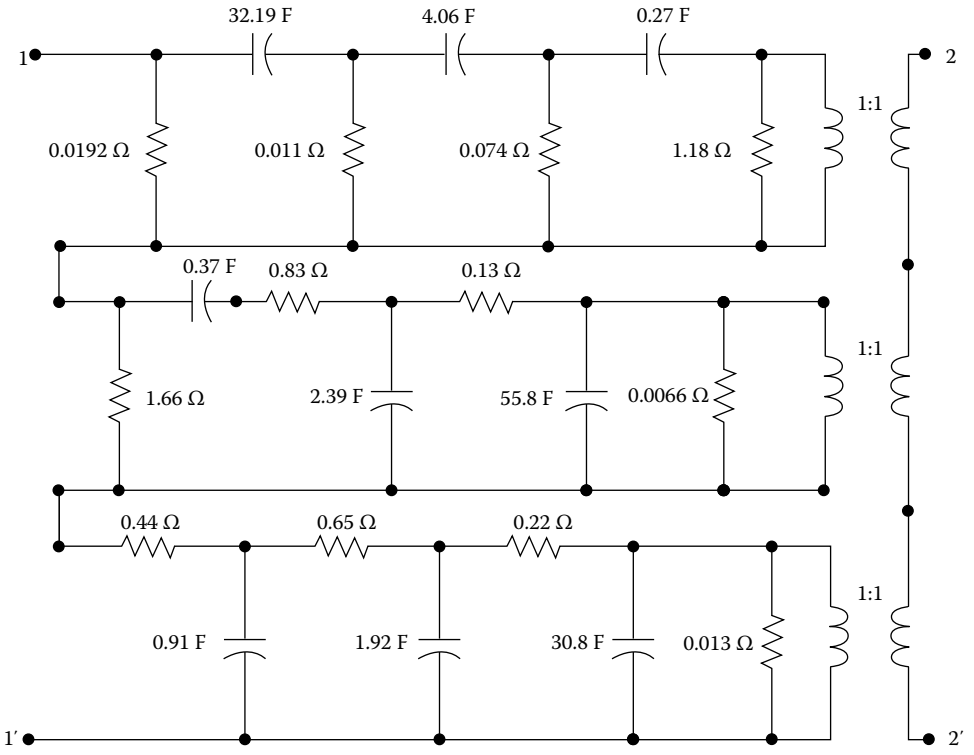


FIGURE 8.20 RC two-port realization of the open-circuit voltage ratio $G_{12}(s)$.

Reference

1. E. A. Guillemin, Synthesis of RC networks, *J. Math. Phys.*, 28, 22–42, 1949.

9

Design of Resistively Terminated Networks

9.1 Introduction **9**-1
9.2 Double-Terminated Butterworth Networks **9**-4
9.3 Double-Terminated Chebyshev Networks **9**-9
References **9**-13

Wai-Kai Chen
University of Illinois at Chicago

9.1 Introduction

In the design of communication systems, it is frequently required to synthesize a coupling network that will transform a given frequency-dependent load impedance into another specified one. We refer to this operation as impedance matching or equalization, and the resulting coupling network as a matching network or equalizer.

Refer to the network configuration of Figure 9.1 where the source is represented either by its Thévenin equivalent or by its Norton equivalent. Our objective here is to design a lossless two-port network or equalizer N, which when inserted between a resistive source and a resistive load will yield a preassigned transducer power-gain characteristic over the entire sinusoidal frequency spectrum. Explicit formulas for the design of Butterworth and Chebyshev LC ladder networks will be given. The more complicated situation where the load is frequency dependent will be discussed in Chapter 10.

In the networks of Figure 9.1, let $Z_{11}(s)$ and $Z_{22}(s)$ be the impedances looking into the input and output ports when the output and input ports are terminated in $z_2(s)$ and $z_1(s)$, respectively. The input and output reflection coefficients are defined by

$$\rho_{11}(s) = \frac{Z_{11}(s) - z_1(-s)}{Z_{11}(s) + z_1(s)} \tag{9.1a}$$

$$\rho_{22}(s) = \frac{Z_{22}(s) - z_2(-s)}{Z_{22}(s) + z_2(s)} \tag{9.1b}$$

respectively. We now demonstrate that the transducer power gain $G(\omega^2)$ defined as the ratio of average power delivered to the load to the maximum available average power at the source is given by

$$G(\omega^2) = 1 - |\rho_{11}(j\omega)|^2 \equiv |\rho_{21}(j\omega)|^2 \tag{9.2}$$

where $\rho_{21}(s)$ is known as the transmission coefficient. To prove this, we first compute

$$1 - \rho_{11}(s)\rho_{11}(-s) = \frac{4r_1(s)R_{11}(s)}{[Z_{11}(s) + z_1(s)\,][Z_{11}(-s) + z_1(-s)]} \tag{9.3}$$

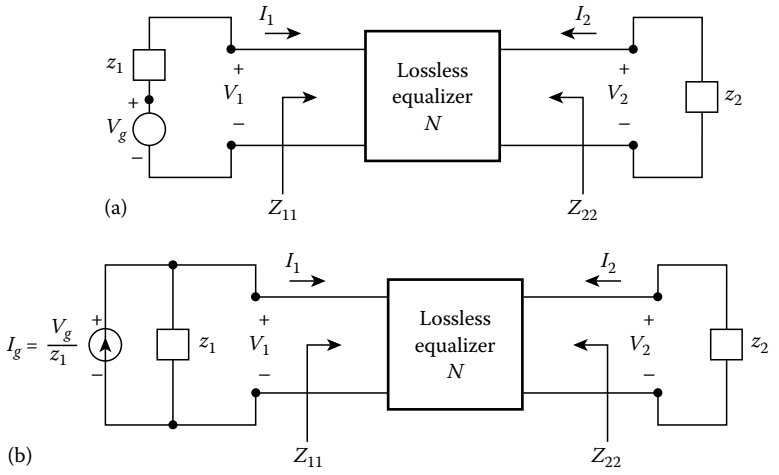


FIGURE 9.1 General broadband matching configuration. (a) The Thévenin equivalent. (b) The Norton equivalent.

where

$$r_1(s) = \text{Ev}\, z_1(s) = \frac{1}{2}[z_1(s) + z_1(-s)] \tag{9.4a}$$

$$R_{11}(s) = \text{Ev}\, Z_{11}(s) = \frac{1}{2}[Z_{11}(s) + Z_{11}(-s)] \tag{9.4b}$$

are the even parts of $z_1(s)$ and $Z_{11}(s)$, respectively. Thus, the impedance facing the voltage source V_g is given by

$$\frac{V_g(s)}{I_1(s)} = Z_{11}(s) + z_1(s) \tag{9.5}$$

On the $j\omega$-axis, Equation 9.3 reduces to

$$1 - |\rho_{11}(j\omega)|^2 = \frac{4r_1(j\omega)R_{11}(j\omega)}{|Z_{11}(j\omega) + z_1(j\omega)|^2} = \frac{|I_1(j\omega)|^2 R_{11}(j\omega)}{|V_g(j\omega)|^2/4r_1(j\omega)} \tag{9.6}$$

The power input to the network of Figure 9.1a under sinusoidal steady state is

$$P_{\text{in}} = |I_1(j\omega)|^2 R_{11}(j\omega) \tag{9.7}$$

while the power output to the load is

$$P_{\text{out}} = |I_2(j\omega)|^2 r_2(j\omega) \tag{9.8}$$

where

$$r_2(s) = \text{Ev}\, z_2(s) = \frac{1}{2}[z_2(s) + z_2(-s)] \tag{9.9}$$

is the even part of $z_2(s)$. Since the two-port N is lossless, the power input must be equal to power output or

$$|I_1(j\omega)|^2 R_{11}(j\omega) = |I_2(j\omega)|^2 r_2(j\omega) \tag{9.10}$$

The maximum average power that the source combination is capable of delivering to the network occurs when the input port is conjugately matched or $Z_{11}(j\omega) = -z_1(j\omega)$. Under this condition, the maximum available average power from the source combination is

$$P_{\text{ava}} = \frac{|V_g(j\omega)|^2}{4r_1(j\omega)} \tag{9.11}$$

Substituting Equations 9.8 and 9.11 in Equation 9.6 yields

$$1 - |\rho_{11}(j\omega)|^2 = \frac{|I_2(j\omega)|^2 r_2(j\omega)}{|V_g(j\omega)|^2/4r_1(j\omega)} = \frac{P_{\text{out}}}{P_{\text{ava}}} = G(\omega^2) \tag{9.12}$$

Using Equation 9.2 shows that

$$|\rho_{11}(j\omega)|^2 + |\rho_{21}(j\omega)|^2 = 1 \tag{9.13}$$

or

$$\begin{aligned} |\rho_{11}(j\omega)|^2 &= 1 - \frac{\text{average power to load}}{\text{average power available}} \\ &= \frac{\text{average power available} - \text{average power to load}}{\text{average power available}} \\ &= \frac{\text{“average reflected” power}}{\text{average power available}} \end{aligned} \tag{9.14}$$

$$|\rho_{21}(j\omega)|^2 = \frac{\text{average power to load}}{\text{average power available}} \tag{9.15}$$

Therefore, the magnitude squared of the reflection coefficient $|\rho_{11}(j\omega)|^2$ denotes the fraction of the maximum available average power that is reflected back to the source, and the magnitude squared of the transmission coefficient $|\rho_{21}(j\omega)|^2$ represents the fraction of the maximum available average power that is transmitted to the load from the source. In fact, their names are suggested by these interpretations. We remark that since the transducer power gain G is a function of ω^2, it is written as $G(\omega^2)$ to emphasize this.

We next express the transmission coefficient in terms of other specifications. From Equation 9.12, we have

$$|\rho_{21}[j\omega]|^2 = 4r_1(j\omega)r_2(j\omega)\left|\frac{I_2(j\omega)}{V_g(j\omega)}\right|^2 \tag{9.16}$$

Substituting $V_g(j\omega) = z_1(j\omega)I_g(j\omega)$ in Equation 9.16 gives

$$|\rho_{21}(j\omega)|^2 = \frac{4r_1(j\omega)r_2(j\omega)}{|z_1(j\omega)|^2}\left|\frac{I_2(j\omega)}{I_g(j\omega)}\right|^2 \tag{9.17}$$

In terms of the transfer voltage ratio and transfer impedance, we apply the relation $V_2(j\omega) = -I_2(j\omega)$ $z_2(j\omega)$ and obtain

$$|\rho_{21}(j\omega)|^2 = \frac{4r_1(j\omega)r_2(j\omega)}{|z_2(j\omega)|^2}\left|\frac{V_2(j\omega)}{V_g(j\omega)}\right|^2 \tag{9.18}$$

$$|\rho_{21}(j\omega)|^2 = \frac{4r_1(j\omega)r_2(j\omega)}{|z_1(j\omega)z_2(j\omega)|^2}\left|\frac{V_2(j\omega)}{I_g(j\omega)}\right|^2 \tag{9.19}$$

Similarly, we can derive a relation between the output reflection coefficient $\rho_{22}(j\omega)$ and the transmission coefficient $\rho_{12}(j\omega)$ magnitude squared as

$$|\rho_{12}(j\omega)|^2 \equiv 1 - |\rho_{22}(j\omega)|^2 \tag{9.20}$$

In fact, for the lossless reciprocal two-port network N we have

$$|\rho_{21}(j\omega)|^2 = |\rho_{12}(j\omega)|^2 = 1 - |\rho_{11}(j\omega)|^2 = 1 - |\rho_{22}(j\omega)|^2 \tag{9.21}$$

9.2 Double-Terminated Butterworth Networks

In this part, we show how to design a lossless two-port network operating between a resistive generator with internal resistance R_1 and a resistive load with resistance R_2 to yield the nth-order Butterworth transducer power-gain characteristic

$$G(\omega^2) = |\rho_{21}(j\omega)|^2 = \frac{K_n}{1 + (\omega/\omega_c)^{2n}} \tag{9.22}$$

Since for a passive network $G(\omega^2)$ is bounded between 0 and 1, the DC gain K_n is restricted by

$$0 \le K_n \le 1 \tag{9.23}$$

Substituting Equation 9.22 in Equation 9.12 yields the squared magnitude of the input reflection coefficient as

$$|\rho_{11}(j\omega)|^2 = 1 - G(\omega^2) = 1 - |\rho_{21}(j\omega)|^2 = \frac{1 - K_n + (\omega/\omega_c)^{2n}}{1 + (\omega/\omega_c)^{2n}} \tag{9.24}$$

or

$$\rho_{11}(j\omega)\rho_{11}(-j\omega) = \alpha^{2n}\frac{1 + (\omega/\alpha\omega_c)^{2n}}{1 + (\omega/\omega_c)^{2n}} \tag{9.25}$$

where

$$\alpha = (1 - K_n)^{1/2n} \tag{9.26}$$

Appealing to analytic continuation by substituting ω by $-js$ results in

$$\rho_{11}(s)\rho_{11}(-s) = \alpha^{2n}\frac{1+(-1)^n x^{2n}}{1+(-1)^n y^{2n}} \tag{9.27}$$

where

$$y = \frac{s}{\omega_c}, \quad x = \frac{y}{\alpha} \tag{9.28}$$

To obtain the input reflection coefficient $\rho_{11}(s)$ from $\rho_{11}(s)\ \rho_{11}(-s)$, we need to assign the zeros and poles of Equation 9.27. Since $\rho_{11}(s)$ is devoid of poles in the closed RHS, we must assign all the LHS poles to $\rho_{11}(s)$. The zeros of $\rho_{11}(s)$, however, may lie in the RHS, so that in general a number of different numerators are possible. For our purposes, we choose only the LHS zeros for $\rho_{11}(s)$. Define a minimum-phase reflection coefficient to be one that is devoid of zeros in the open RHS. Then, the minimum-phase solution of Equation 9.27 can be written as

$$\rho_{11}(s) = \pm\alpha^n \frac{q(x)}{q(y)} \tag{9.29}$$

where $q(x)$ is the Hurwitz polynomial with unity leading coefficient formed by the LHS roots of the equation $1+(-1)^n x^{2n} = 0$. From Equation 9.1a, the input impedance is found to be

$$Z_{11}(s) = R_1\frac{1+\rho_{11}(s)}{1-\rho_{11}(s)} \tag{9.30}$$

Combining this with Equation 9.29 yields

$$Z_{11}(s) = R_1\frac{q(y)\pm\alpha^n q(x)}{q(y)\mp\alpha^n q(x)} \tag{9.31}$$

If both R_1 and R_2 are specified, then the DC gain K_n cannot be chosen independently. In fact, by substituting $s=0$ in Equation 9.31 and assuming that $K_n \neq 0$ we obtain

$$\frac{R_2}{R_1} = \left(\frac{1+\alpha^n}{1-\alpha^n}\right)^{\pm 1} \tag{9.32}$$

where the $\pm$ signs are determined, respectively, according to $R_2 \geq R_1$ and $R_2 \leq R_1$. Therefore, if any two of the three quantities R_1, R_2, and K_n are specified, the third one is fixed.

We now show that the input impedance $Z_{11}(s)$ can be realized by an LC ladder terminated in a resistor. In fact, explicit formulas for their element values will be given, thereby reducing the design problem to simple arithmetic. Depending upon the choice of the plus and minus signs in Equation 9.32, two cases are distinguished.

Case 1. $\rho_{11}(0) \geq 0$. With the choice of the plus sign, the input impedance becomes

$$Z_{11}(s) = R_1\frac{q(y)+\alpha^n q(x)}{q(y)-\alpha^n q(x)} \tag{9.33}$$

which can be expanded in a continued fraction about infinity, as in the first Cauer canonical form, and results in an LC ladder terminated in a resistor:

$$Z_{11}(s) = L_1 s + \cfrac{1}{C_2 S + \cfrac{1}{L_3 s + \cfrac{1}{\ddots + \cfrac{1}{W}}}} \tag{9.34}$$

where W is a constant representing either a resistance or conductance. Depending upon whether n is odd or even, the LC ladder has the configuration of Figure 9.2. The element values can be computed by the following recurrence formulas:

$$L_1 = \frac{2R_1 \sin \pi/2n}{(1-\alpha)\omega_c} \tag{9.35}$$

$$L_{2m-1}C_{2m} = \frac{4 \sin \gamma_{4m-3} \sin \gamma_{4m-1}}{\omega_c^2(1 - 2\alpha \cos \gamma_{4m-2} + \alpha^2)} \tag{9.36a}$$

$$L_{2m+1}C_{2m} = \frac{4 \sin \gamma_{4m-1} \sin \gamma_{4m+1}}{\omega_c^2(1 - 2\alpha \cos \gamma_{4m} + \alpha^2)} \tag{9.36b}$$

for $m = 1, 2, \ldots, \lceil n/2 \rceil$, the largest integer not greater than $n/2$; where

$$\gamma_m = \frac{m\pi}{2n} \tag{9.37}$$

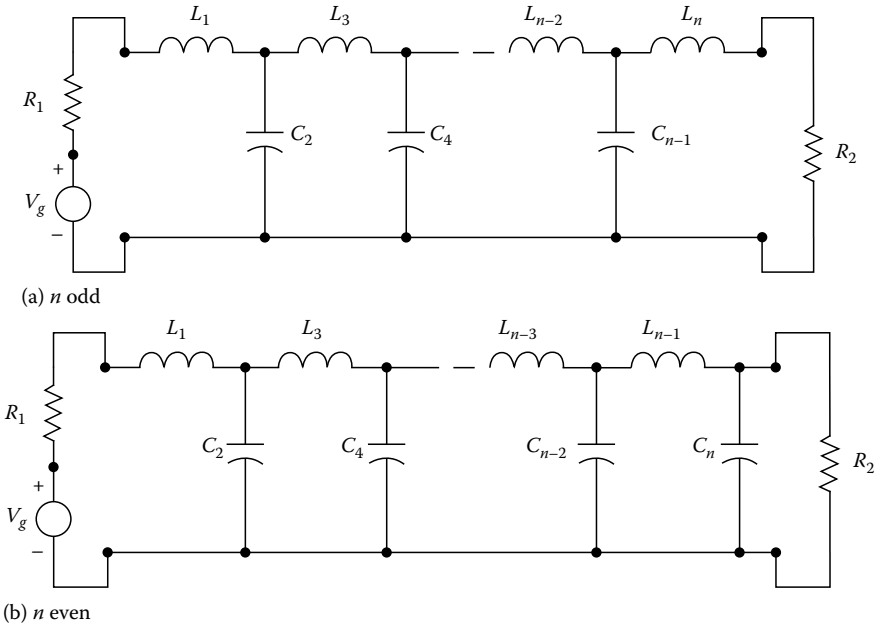


FIGURE 9.2 Butterworth LC ladder networks for $\rho_{11}(0) \geq 0$.

The values of the final elements can also be calculated directly by

$$L_n = \frac{2R_2 \sin \pi/2n}{(1+\alpha)\omega_c}, \quad n \text{ odd} \tag{9.38a}$$

$$C_n = \frac{2 \sin \pi/2n}{R_2(1+\alpha)\omega_c}, \quad n \text{ even} \tag{9.38b}$$

A complete derivation of these formulas was first given by Bossé [1]. Hence we can calculate the element values starting from either the first or the last element. When $R_1 = R_2$, formulas Equation 9.36 reduce to

$$L_{2m-1} = \frac{2R_1 \sin \gamma_{4m-3}}{\omega_c} \tag{9.39a}$$

$$C_{2m} = \frac{2 \sin \gamma_{4m-1}}{R_1 \omega_c} \tag{9.39b}$$

Example 9.1

Given

$$R_1 = 70\ \Omega, \quad R_2 = 200\ \Omega, \quad \omega_c = 10^5\ \text{rad/s}, \quad n = 4 \tag{9.40}$$

obtain a Butterworth LC ladder to meet these specifications.

Since $R_2 > R_1$, we choose the plus sign in Equation 9.32 and obtain $\rho_{11}(0) \geq 0$, $\alpha = 0.833$, and $\gamma_m = 22.5m$. Thus, from Equations 9.35 and 9.36 the element values are found to be

$$L_1 = \frac{2 \times 70 \sin 22.5^\circ}{(1-0.833) \times 10^5} = 3.2081\ \text{mH} \tag{9.41a}$$

$$C_2 = \frac{4 \sin 22.5^\circ \sin 67.5^\circ}{L_1(1.6939 - 1.666 \cos 45^\circ) \times 10^{10}} = 0.085456\ \mu\text{F} \tag{9.41b}$$

$$L_3 = \frac{4 \sin 67.5^\circ \sin 112.5^\circ}{C_2(1.6939 - 1.666 \cos 90^\circ) \times 10^{10}} = 2.3587\ \text{mH} \tag{9.41c}$$

$$C_4 = \frac{4 \sin 112.5^\circ \sin 157.5^\circ}{L_3(1.6939 - 1.666 \cos 135^\circ) \times 10^{10}} = 0.020877\ \mu\text{F} \tag{9.41d}$$

Alternatively, C_4 can be computed directly from Equation 9.39b as

$$C_4 = \frac{2 \sin 22.5^\circ}{200 \times (1+0.833) \times 10^5} = 0.020877\ \mu\text{F} \tag{9.42}$$

The ladder network together with its termination is presented in Figure 9.3. This network possesses the fourth-order Butterworth transducer power-gain response with a DC gain

$$K_4 = 1 - \alpha^8 = 0.7682 \tag{9.43}$$

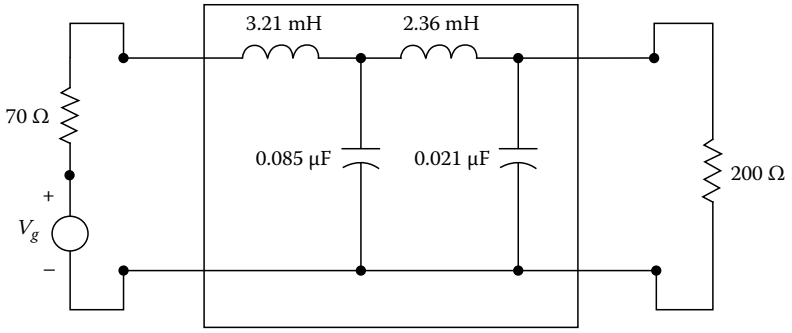


FIGURE 9.3 Fourth-order Butterworth LC ladder network.

Case 2. $\rho_{11}(0) < 0$. With the choice of the minus sign, the input impedance can be expanded in a continued fraction as

$$\frac{1}{Z_{11}(s)} = C_1 s + \cfrac{1}{L_2 s + \cfrac{1}{C_3 s + \cfrac{1}{\ddots + \cfrac{1}{W}}}} \tag{9.44}$$

which can be realized by the LC ladder networks of Figure 9.4, depending on whether W is even or odd, where W is the terminating resistance or conductance. Formulas for the element values are similar to

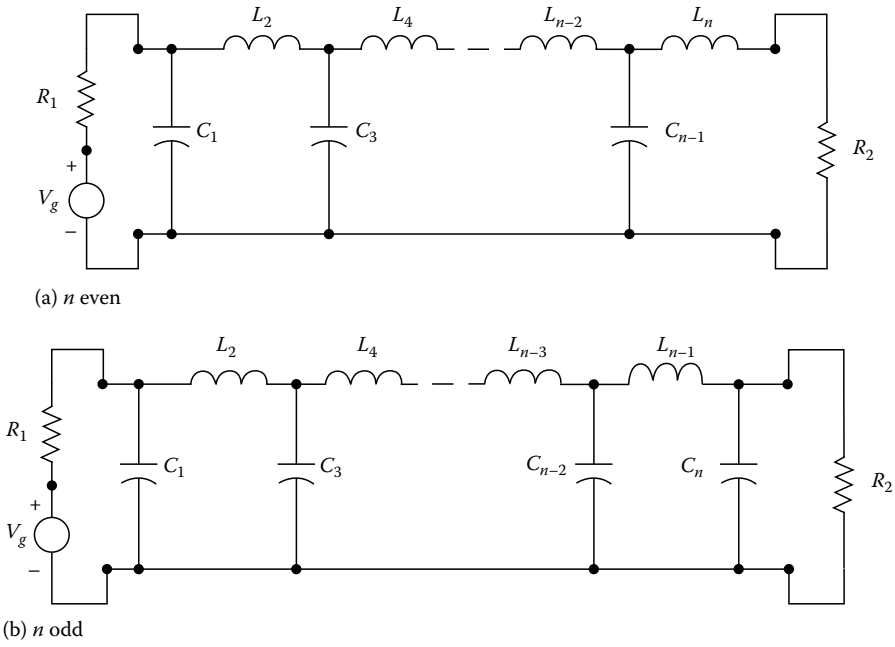


FIGURE 9.4 Butterworth LC ladder networks for $\rho_{11}(0) < 0$.

those given in Equations 9.35 through 9.39 except that the roles of C's and L's are interchanged and R_1 and R_2 are replaced by their reciprocals.

$$C_1 = \frac{2\sin\pi/2n}{R_1(1-\alpha)\omega_c} \tag{9.45a}$$

$$C_{2m-1}L_{2m} = \frac{4\sin\gamma_{4m-3}\sin\gamma_{4m-1}}{\omega_c^2(1-2\alpha\cos\gamma_{4m-2}+\alpha^2)} \tag{9.45b}$$

$$C_{2m+1}L_{2m} = \frac{4\sin\gamma_{4m-1}\sin\gamma_{4m+1}}{\omega_c^2(1-2\alpha\cos\gamma_{4m}+\alpha^2)} \tag{9.45c}$$

for $m = 1, 2, \ldots, \lceil n/2 \rceil$. The values of the final elements can also be calculated directly by

$$C_n = \frac{2\sin\pi/2n}{R_2(1+\alpha)\omega_c}, \quad n \text{ odd} \tag{9.46a}$$

$$L_n = \frac{2R_2\ \sin\pi/2n}{(1+\alpha)\omega_c}, \quad n \text{ even} \tag{9.46b}$$

9.3 Double-Terminated Chebyshev Networks

Now, we consider the problem of synthesizing an LC ladder which when connected between a resistive source of internal resistance R_1 and a resistive load of resistance R_2 will yield a preassigned Chebyshev transducer power-gain characteristic

$$G(\omega^2) = |\rho_{21}(j\omega)|^2 = \frac{K_n}{1+\varepsilon^2 C_n^2(\omega/\omega_c)} \tag{9.47}$$

with K_n bounded between 0 and 1. Following Equation 9.24, the squared magnitude of the input reflection coefficient can be written as

$$|\rho_{11}(j\omega)|^2 = 1 - G(\omega^2) = 1 - |\rho_{21}(j\omega)|^2 = \frac{1-K_n+\varepsilon^2 C_n^2(\omega/\omega_c)}{1+\varepsilon^2 C_n^2(\omega/\omega_c)} \tag{9.48}$$

Appealing to analytic continuation, we obtain

$$\rho_{11}(s)\rho_{11}(-s) = (1-K_n)\frac{1+\hat{\varepsilon}^2 C_n^2(-jy)}{1+\varepsilon^2 C_n^2(-jy)} \tag{9.49}$$

where

$$\hat{\varepsilon} = \frac{\varepsilon}{\sqrt{1-K_n}} \tag{9.50}$$

As in the Butterworth case, we assign LHS poles to $\rho_{11}(s)$ and the minimum-phase solution of Equation 9.49 becomes

$$\rho_{11}(s) = \pm\frac{\hat{p}(y)}{p(y)} \tag{9.51}$$

where $p(y)$ and $\hat{p}(y)$ are the Hurwitz polynomials with unity leading coefficient formed by the LHS roots of the equations $1 + \varepsilon^2 C_n^2(-jy) = 0$ and $1 + \hat{\varepsilon}^2 C_n^2(-jy) = 0$, respectively. From Equation 9.1a, the input impedance of the LC ladder when the output port is terminated in R_2 is found to be

$$Z_{11}(s) = R_1 \frac{p(y) \pm \hat{p}(y)}{p(y) \mp \hat{p}(y)} \tag{9.52}$$

A relationship among the quantities R_1, R_2, and K_n is given by

$$\frac{R_2}{R_1} = \left(\frac{1 + \sqrt{1 - K_n}}{1 - \sqrt{1 - K_n}}\right)^{\pm 1}, \quad n \text{ odd} \tag{9.53a}$$

$$= \left(\frac{\sqrt{1 + \varepsilon^2} + \sqrt{1 + \varepsilon^2 - K_n}}{\sqrt{1 + \varepsilon^2} - \sqrt{1 + \varepsilon^2 - K_n}}\right)^{\pm 1}, \quad n \text{ even} \tag{9.53b}$$

where the $\pm$ signs are determined, respectively, according to $R_2 \geq R_1$ and $R_2 \leq R_1$. Therefore, if n is odd and the DC gain is specified, the ratio of the terminating resistances is fixed by Equation 9.53a. On the other hand, if n is even and the peak-to-peak ripple in the passband and K_n or the DC gain is specified, the ratio of the resistances is given by Equation 9.53b.

We now show that the input impedance $Z_{11}(s)$ can be realized by an LC ladder terminated in a resistor. Again, explicit formulas for their element values will be given, thereby reducing the design problem to simple arithmetic. Depending upon the choice of the plus and minus signs in Equation 9.51, two cases are distinguished.

Case 1. $\rho_{11}(0) \geq 0$. With the choice of the plus sign, the input impedance becomes

$$Z_{11}(s) = R_1 \frac{p(y) + \hat{p}(y)}{p(y) - \hat{p}(y)} \tag{9.54}$$

which can be expanded in a continued fraction as in Equation 9.34. Depending on whether n is odd or even, the corresponding LC ladder network has the configurations of Figure 9.2. The element values can be computed by the following recurrence formulas:

$$L_1 = \frac{2R_1 \sin \pi/2n}{(\sinh a - \sinh \hat{a})\dot{\omega}_c} \tag{9.55}$$

$$L_{2m-1}C_{2m} = \frac{4 \sin \gamma_{4m-3} \sin \gamma_{4m-1}}{\omega_c^2 f_{2m-1}(\sinh a, \sinh \hat{a})} \tag{9.56a}$$

$$L_{2m+1}C_{2m} = \frac{4 \sin \gamma_{4m-1} \sin \gamma_{4m+1}}{\omega_c^2 f_{2m}(\sinh a, \sinh \hat{a})} \tag{9.56b}$$

for $m = 1, 2, \ldots, \lceil n/2 \rceil$, where

$$\gamma_m = \frac{m\pi}{2n} \tag{9.57a}$$

$$\hat{a} = \frac{1}{n} \sinh^{-1}\left(\frac{\sqrt{1 - K_n}}{\varepsilon}\right) \tag{9.57b}$$

$$f_m(u, v) = u^2 + v^2 + \sin^2 \gamma_{2m} - 2uv \cos \gamma_{2m} \tag{9.57c}$$

In addition, the values of the last elements can also be computed directly by the equations

$$L_n = \frac{2R_2 \sin \pi/2n}{(\sinh a + \sinh \hat{a})\omega_c}, \quad n \text{ odd} \tag{9.58a}$$

$$C_n = \frac{2 \sin \pi/2n}{R_2(\sinh a + \sinh \hat{a})\omega_c}, \quad n \text{ even} \tag{9.58b}$$

A formal proof of these formulas was first given by Takahasi [2]. Hence, we can calculate the element values starting from either the first or the last element.

Example 9.2

Given

$$R_1 = 150\ \Omega, \quad R_2 = 470\ \Omega, \quad \omega_c = 10^8\pi \text{ rad/s}, \quad n = 4 \tag{9.59}$$

find a Chebyshev LC ladder network to meet these specifications with peak-to-peak ripple in the passband not exceeding 1.5 dB.

Since R_1 and R_2 are both specified, the minimum passband gain $G_{\min}$ is fixed by Equation 9.53b as

$$G_{\min} \equiv \frac{K_n}{1+\varepsilon^2} = 1 - \left(\frac{\dfrac{470}{150} - 1}{\dfrac{470}{150} + 1}\right)^2 = 0.7336 \tag{9.60}$$

For the 1.5 dB ripple in the passband, the corresponding ripple factor is given by

$$\varepsilon = \sqrt{10^{0.15} - 1} = 0.64229 \tag{9.61}$$

obtaining $K_4 = 1.036$, which is too large for the network to be physically realizable. Thus, let $K_4 = 1$, the maximum permissible value, and the corresponding ripple factor becomes

$$\varepsilon = \sqrt{\frac{1}{G_{\min}} - 1} = \sqrt{\frac{1}{0.7336} - 1} = 0.6026 \quad \text{or} \quad 1.345 \text{ dB} \leq 1.5 \text{ dB} \tag{9.62}$$

We next compute the quantities

$$\gamma_m = \frac{m\pi}{2n} = 22.5m \tag{9.63a}$$

$$a = \frac{1}{4} \sinh^{-1} \frac{1}{0.6026} = 0.32, \quad \hat{a} = 0 \tag{9.63b}$$

$$f_m(\sinh\ 0.32,\ 0) = \sinh^2 0.32 + \sin^2 \gamma_{2m} = 0.1059 + \sin^2 \gamma_{2m} \tag{9.63c}$$

Appealing to formulas Equations 9.55 and 9.56, the element values are calculated as follows:

$$L_1 = \frac{2 \times 150\ \sin 22.5^\circ}{(\sinh\ 0.32 - \sinh\ 0) \times 10^8\pi} = 1.123\ \mu\text{H} \tag{9.64a}$$

$$C_2 = \frac{4 \sin 22.5^\circ \sin 67.5^\circ}{L_1 10^{16} \pi^2 \left(\sinh^2 0.32 + \sin^2 45^\circ\right)} = 21.062 \text{ pF} \tag{9.64b}$$

$$L_3 = \frac{4 \sin 67.5^\circ \sin 112.5^\circ}{C_2 10^{16} \pi^2 \left(\sinh^2 0.32 + \sin^2 90^\circ\right)} = 1.485 \ \mu\text{H} \tag{9.64c}$$

$$C_4 = \frac{4 \sin 112.5^\circ \sin 157.5^\circ}{L_3 10^{16} \pi^2 \left(\sinh^2 0.32 + \sin^2 135^\circ\right)} = 15.924 \text{ pF} \tag{9.64d}$$

Alternatively, the last capacitance can also be computed directly from Equation 9.58b as

$$C_4 = \frac{2 \sin \ 22.5^\circ}{470 \times (\sinh 0.32 + \sinh 0) \times 10^8 \pi} = 15.925 \text{ pF} \tag{9.65}$$

The LC ladder together with its terminations is presented in Figure 9.5.

Case 2. $\rho_{11}(0) < 0$. With the choice of the minus sign in Equation 9.51, the input impedance, aside from the constant R_1, becomes the reciprocal of Equation 9.54

$$Z_{11}(s) = R_1 \frac{p(y) - \hat{p}(y)}{p(y) + \hat{p}(y)} \tag{9.66}$$

and can be expanded in a continued fraction as that shown in Equation 9.44. Depending on whether n is even or odd, the LC ladder network has the configurations of Figure 9.4. Formulas for the element values are similar to those given in Equations 9.55 through 9.58 except that the roles of C's and L's are interchanged and R_1 and R_2 are replaced by their reciprocals:

$$C_1 = \frac{2 \sin \pi/2n}{R_1 (\sinh a - \sinh \hat{a}) \omega_c} \tag{9.67a}$$

$$C_{2m-1} L_{2m} = \frac{4 \sin \gamma_{4m-3} \sin \gamma_{4m-1}}{\omega_c^2 f_{2m-1}(\sinh a, \ \sinh \hat{a})} \tag{9.67b}$$

$$C_{2m+1} L_{2m} = \frac{4 \sin \gamma_{4m-1} \sin \gamma_{4m+1}}{\omega_c^2 f_{2m}(\sinh a, \ \sinh \hat{a})} \tag{9.67c}$$

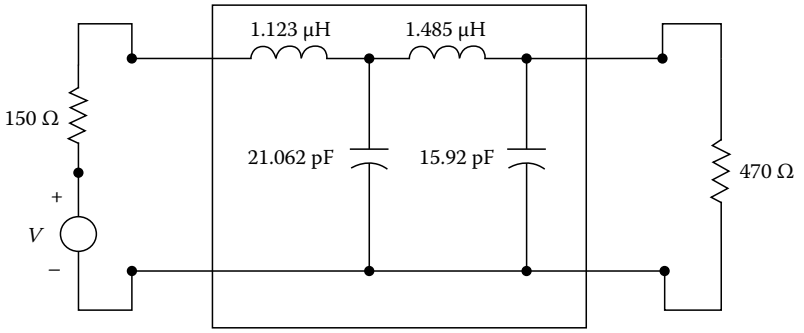


FIGURE 9.5 Fourth-order Chebyshev LC ladder network for $\rho_{11}(0) \geq 0$.

for $m = 1, 2, \ldots, \lceil n/2 \rceil$, where γ_m and $f_m(\sinh a, \sinh \hat{a})$ are defined in Equation 9.57. In addition, the values of the last elements can also be computed directly by the formulas

$$C_n = \frac{2 \sin \pi/2n}{R_2(\sinh a + \sinh \hat{a})\omega_c}, \quad n \text{ odd} \tag{9.68a}$$

$$L_n = \frac{2R_2 \sin \pi/2n}{(\sinh a + \sinh \hat{a})\omega_c}, \quad n \text{ even} \tag{9.68b}$$

References

1. G. Bossé, Siebketten ohne Dämpfungsschwankungen im Durchlassbereich (Potenzketten), *Frequenz*, 5, 279–284, 1951.
2. H. Takahasi, On the ladder-type filter network with Tchebysheff response, *J. Inst. Elec. Commun. Engrs. Japan*, 34, 65–74, 1951.

10

Design of Broadband Matching Networks

10.1 Introduction **10**-1
10.2 Basic Coefficient Constraints **10**-2
Basic Coefficient Constraints on $\rho(S)$
10.3 Design Procedure **10**-8
10.4 Explicit Formulas for the RLC Load **10**-14
Butterworth Networks • Chebyshev Networks
References **10**-30

Wai-Kai Chen
University of Illinois at Chicago

10.1 Introduction

Refer to the network configuration of Figure 10.1 where the source is represented either by its Thévenin equivalent or by its Norton equivalent. The load impedance $z_2(s)$ is assumed to be strictly passive over a frequency band of interest, because the matching problem cannot be meaningfully defined if the load is purely reactive. Our objective is to design an "optimum" lossless two-port network or equalizer N to match out the load impedance $z_2(s)$ to the resistive source impedance $z_1(s) = R_1$, and to achieve a preassigned transducer power-gain characteristic $G(\omega^2)$ over the entire sinusoidal frequency spectrum.

As stated in Chapter 9, the output reflection coefficient is given by

$$\rho_{22}(s) = \frac{Z_{22}(s) - z_2(-s)}{Z_{22}(s) + z_2(s)} \tag{10.1}$$

where $Z_{22}(s)$ is the impedance looking into the output port when the input port is terminated in the source resistance R_1. As shown in Chapter 9, the transducer power gain $G(\omega^2)$ is related to the transmission and reflection coefficients by the equation

$$G(\omega^2) = |\rho_{21}(j\omega)|^2 = |\rho_{12}(j\omega)|^2 = 1 - |\rho_{11}(j\omega)|^2 = 1 - |\rho_{22}(j\omega)|^2 \tag{10.2}$$

Recall that in computing $\rho_{11}(s)$ from $\rho_{11}(s)\rho_{11}(-s)$ we assign all of the LHS poles of $\rho_{11}(s)\rho_{11}(-s)$ to $\rho_{11}(s)$ because with resistive load $z_2(s) = R_2$, $\rho_{11}(s)$ is devoid of poles in the RHS. For the complex load, the poles of $\rho_{22}(s)$ include those of $z_2(-s)$, which may lie in the open RHS. As a result, the assignment of poles of $\rho_{22}(s)\rho_{22}(-s)$ is not unique. Furthermore, the nonanalyticity of $\rho_{22}(s)$ leaves much to be desired in terms of our ability to manipulate. For these reasons, we consider the normalized reflection coefficient defined by

$$\rho(s) = A(s)\rho_{22}(s) = A(s)\frac{Z_{22}(s) - z_2(-s)}{Z_{22}(s) + z_2(s)} \tag{10.3}$$

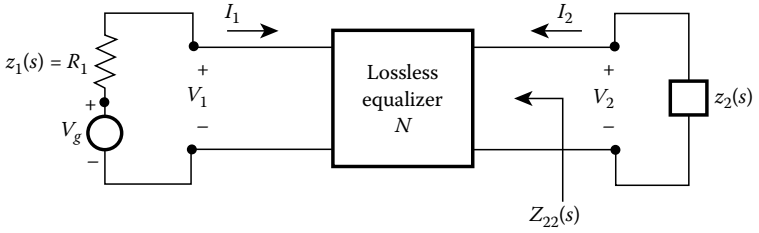


FIGURE 10.1 Schematic of broadband matching.

where

$$A(s) = \prod_{i=1}^{q} \frac{s - s_i}{s + s_i}, \quad \text{Re } s_i > 0 \tag{10.4}$$

is the real all-pass function defined by the open RHS poles s_i ($i = 1, 2, \ldots, q$) of $z_2(-s)$. An all-pass function is a function whose zeros are all located in the open RHS and whose poles are located at the LHS mirror image of the zeros. Therefore, it is analytic in the closed RHS and such that

$$A(s)A(-s) = 1 \tag{10.5}$$

On the $j\omega$-axis, the magnitude of $A(j\omega)$ is unity, being flat for all sinusoidal frequencies, and we have

$$|\rho(j\omega)| = |A(j\omega)\rho_{22}(j\omega)| = |\rho_{22}(j\omega)| \tag{10.6}$$

and Equation 10.2 becomes

$$G(\omega^2) = 1 - |\rho(j\omega)|^2 \tag{10.7}$$

This equation together with the normalized reflection coefficient $\rho(s)$ of Equation 10.3 forms the cornerstone of Youla's theory of broadband matching [1].

10.2 Basic Coefficient Constraints

In Chapter 8, we define the zeros of transmission for a terminate two-port network as the frequencies at which a zero output results for a finite input. We extend this concept by defining the zeros of transmission for a one-port impedance.

Definition 10.1: Zero of transmission. For a given impedance $z_2(s)$, a closed RHS zero of multiplicity k of the function

$$w(s) \equiv \frac{r_2(s)}{z_2(s)} \tag{10.8}$$

where $r_2(s)$ is the even part of $z_2(s)$, is called a zero of transmission of order k of $z_2(s)$.

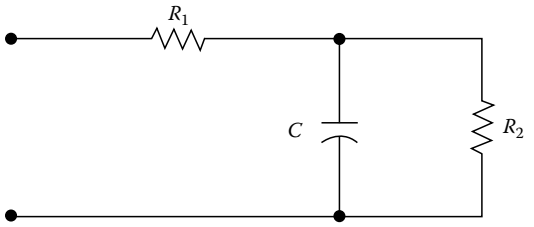


FIGURE 10.2 RC one-port network.

The reason for this name is that if we realize the impedance $z_2(s)$ as the input impedance of a lossless two-port network terminated in a $1-\Omega$ resistor, the magnitude squared of the transfer impedance function $Z_{12}(j\omega)$ between the $1-\Omega$ resistor and the input equals the real part of the input impedance,

$$|Z_{12}(j\omega)|^2 = \mathrm{Re}\, z_2(j\omega) = r_2(j\omega) \tag{10.9}$$

After appealing to analytic continuation by substituting ω by $-js$, the zeros of $r_2(s)$ are seen to be the zeros of transmission of the lossless two-port.

Consider, for example, the RC impedance $z_2(s)$ of Figure 10.2,

$$z_2(s) = R_1 + \frac{R_2}{R_2Cs+1} \tag{10.10}$$

the even part of which is given by

$$r_2(s) = \frac{1}{2}[z_2(s) + z_2(-s)] = \frac{R_1 + R_2 - R_1R_2^2C^2s^2}{1 - R_2^2C^2s^2} \tag{10.11}$$

obtaining

$$w(s) = \frac{r_2(s)}{z_2(s)} = \frac{R_2Cs^2 - (R_1+R_2)/R_1R_2C}{(R_2Cs-1)[s+(R_1+R_2)/R_1R_2C]} \tag{10.12}$$

Thus, the impedance $z_2(s)$ has a zero of transmission of order 1 located at

$$s = \sigma_0 = \frac{1}{R_2C}\sqrt{1+\frac{R_2}{R_1}} \tag{10.13}$$

For our purposes, the zeros of transmission are divided into four mutually exclusive classes.

Definition 10.2: *Classification of zeros of transmission*. Let $s_0 = \sigma_0 + j\omega_0$ be a zero of transmission of an impedance $z_2(s)$. Then s_0 belongs to one of the following four mutually exclusive classes depending on σ_0 and $z_2(s_0)$, as follows:

Class I: $\sigma_0 > 0$, which includes all the open RHS zeros of transmission.
Class II: $\sigma_0 = 0$ and $z_2(\omega_0) = 0$.

Class III: $\sigma_0 = 0$ and $0 < |z_2(j\omega 0)| < \infty$.
Class IV: $\sigma_0 = 0$ and $|z_2(j\omega 0)| = \infty$.

For the impedance $z_2(s)$ of Equation 10.10, its zero of transmission given in Equation 10.13 belongs to Class I of order 1. If $z_2(s)$ is the load of the network of Figure 10.1, it imposes the basic constraints on the normalized reflection coefficient $\rho(s)$. These constraints are important in that they are necessary and sufficient for $\rho(s)$ to be physically realizable, and are most conveniently formulated in terms of the coefficients of the Laurent series expansions of the following quantities about a zero of transmission $s_0 = \sigma_0 + j\omega_0$ of order k of $z_2(s)$:

$$\rho(s) = \rho_0 + \rho_1(s - s_0) + \rho_2(s - s_0)^2 + \cdots = \sum_{m=0}^{\infty} \rho_m (s - s_0)^m \tag{10.14a}$$

$$A(s) = A_0 + A_1(s - s_0) + A_2(s - s_0)^2 + \cdots = \sum_{m=0}^{\infty} A_m (s - s_0)^m \tag{10.14b}$$

$$F(s) \equiv 2r_2(s)A(s) = F_0 + F_1(s - s_0) + F_2(s - s_0)^2 + \cdots = \sum_{m=0}^{\infty} F_m (s - s_0)^m \tag{10.14c}$$

We remark that the expansions of the Laurent type can be found by any method because it is unique, and the resulting expansion is *the* Laurent series expansion. For the zero of transmission at infinity, the expansions take the form

$$\rho(s) = \rho_0 + \frac{\rho_1}{s} + \frac{\rho_2}{s^2} + \frac{\rho_3}{s^3} + \cdots = \sum_{m=0}^{\infty} \frac{\rho_m}{s^m} \tag{10.15a}$$

$$A(s) = A_0 + \frac{A_1}{s} + \frac{A_2}{s^2} + \frac{A_3}{s^3} + \cdots = \sum_{m=0}^{\infty} \frac{A_m}{s^m} \tag{10.15b}$$

$$F(s) = F_0 + \frac{F_1}{s} + \frac{F_2}{s^2} + \frac{F_3}{s^3} + \cdots = \sum_{m=0}^{\infty} \frac{F_m}{s^m} \tag{10.15c}$$

In fact, they can be obtained by means of the binomial expansion formula

$$(s + c)^n = s^n + ns^{n-1}c + \frac{n(n-1)}{2!}s^{n-2}c^2 + \cdots \tag{10.16}$$

which is valid for all values of n if $|s| > |c|$, and is valid only for nonnegative integers n if $|s| \leq |c|$.

Example 10.1

Assume that the network of Figure 10.1 is terminated in the passive impedance

$$z_2(s) = \frac{s}{s^2 + 2s + 1} \tag{10.17}$$

and possesses the transducer power-gain characteristic

$$G(\omega^2) = \frac{K\omega^2}{\omega^4 - \omega^2 + 1}, \quad 0 \leq K \leq 1 \tag{10.18}$$

We first compute the even part

$$r_2(s) = \frac{1}{2}[z_2(s) + z_2(-s)] = \frac{-2s^2}{(s^2 + 2s + 1)(s^2 - 2s + 1)} \tag{10.19}$$

of $z_2(s)$, and obtain the function

$$w(s) = \frac{r_2(s)}{z_2(s)} = \frac{-2s}{s^2 - 2s + 1} \tag{10.20}$$

showing that $z_2(s)$ possesses two Class II zeros of transmission at $s = 0$ and ∞. Since the pole of $z_2(-s)$ is located at $s = 1$ of order 2, the all-pass function $A(s)$ takes the form

$$A(s) = \frac{s^2 - 2s + 1}{s^2 + 2s + 1} \tag{10.21}$$

The other required functions are found to be

$$F(s) = 2A(s)r_2(s) = -\frac{4s^2}{(s^2 + 2s + 1)^2} \tag{10.22}$$

$$\boldsymbol{\rho}(s)\boldsymbol{\rho}(-s) = 1 - G(-s^2) = \frac{s^4 + (1 + K)s^2 + 1}{s^4 + s^2 + 1} \tag{10.23}$$

The minimum-phase solution of Equation 10.23 is determined as

$$\pm\hat{\boldsymbol{\rho}}(s) = \frac{s^2 + \sqrt{1 - K}s + 1}{s^2 + s + 1} \tag{10.24}$$

Now, we expand the functions $A(s)$, $F(s)$, and $\hat{\boldsymbol{\rho}}(s)$ in Laurent series about the zeros of transmission at the origin and at infinity, and obtain

$$A(s) = 1 - 4s + \cdots = 1 - \frac{4}{s} + \cdots \tag{10.25}$$

$$F(s) = 0 + 0 - 4s^2 + \cdots = 0 + 0 - \frac{4}{s^2} + \cdots \tag{10.26}$$

$$\pm\hat{\boldsymbol{\rho}}(s) = 1 + (\sqrt{1 - K} - 1)s + \cdots = 1 + \frac{\sqrt{1 - K} - 1}{s} + \cdots \tag{10.27}$$

In both expansions, we can make the following identifications:

$$A_0 = 1, \quad F_0 = 0, \quad \boldsymbol{\rho}_0 = 1 \tag{10.28a}$$

$$A_1 = -4, \quad F_1 = 0, \quad \boldsymbol{\rho}_1 = \sqrt{1 - K} - 1 \tag{10.28b}$$

$$F_2 = -4 \tag{10.28c}$$

10.2.1 Basic Coefficient Constraints on $\boldsymbol{\rho}(S)$

The basic constraints imposed on the normalized reflection coefficient $\boldsymbol{\rho}(s)$ by a load impedance $z_2(s)$ are most succinctly expressed in terms of the coefficients of the Laurent series expansions (Equation 10.14) of

the functions $\rho(s)$, $A(s)$, and $F(s)$ about each zero of transmission $s_0 = \sigma_0 + j\omega_0$. Depending on the classification of the zero of transmission, one of the following four sets of coefficient conditions must be satisfied:

Class I: For $x = 0, 1, 2, \ldots, k-1$

$$A_x = \rho_x \tag{10.29a}$$

Class II: $A_x = \rho_x$ for $x = 0, 1, 2, \ldots, k-1$, and

$$\frac{A_k - \rho_k}{F_{k+1}} \geq 0 \tag{10.29b}$$

Class III: $A_x = \rho_x$ for $x = 0, 1, 2, \ldots, k-2$, $k \geq 2$, and

$$\frac{A_{k-1} - \rho_{k-1}}{F_k} \geq 0 \tag{10.29c}$$

Class IV: $A_x = \rho_x$ for $x = 0, 1, 2, \ldots, k-1$, and

$$\frac{F_{k-1}}{A_k - \rho_k} \geq a_{-1}, \text{ the residue of } z_2(s) \text{ evaluated at the pole } s = j\omega_0 \tag{10.29d}$$

To determine the normalized reflection coefficient $\rho(s)$ from a preassigned transducer power-gain characteristic $G(\omega^2)$, we appeal to Equation 10.7 and analytic continuation by replacing ω by $-js$ and obtain

$$\rho(s)\rho(-s) = 1 - G(-s^2) \tag{10.30}$$

Since the zeros and poles of $\rho(s)\rho(-s)$ must appear in quadrantal symmetry, being symmetric with respect to both the real and imaginary axes of the s-plane, and since $\rho(s)$ is analytic in the closed RHS, the open LHS poles of $\rho(s)\rho(-s)$ belong to $\rho(s)$ whereas those in the open RHS belong to $\rho(-s)$. For a lumped system, $\rho(s)$ is devoid of poles on the $j\omega$-axis. For the zeros, no unique ways are available to assign them. The only requirement is that the complex–conjugate pair of zeros must be assigned together. However, if we specify that $\rho(s)$ be made a minimum-phase function, then all the open LHS zeros of $\rho(s)\rho(-s)$ are assigned to $\rho(s)$. The $j\omega$-axis zeros of $\rho(s)\rho(-s)$ are of even multiplicity, and thus they are divided equally between $\rho(s)$ and $\rho(-s)$. Therefore, $\rho(s)$ is uniquely determined by the zeros and poles of $\rho(s)\rho(-s)$ only if $\rho(s)$ is required to be minimum-phase.

Let $\hat{\rho}(s)$ be the minimum-phase solution of Equation 10.30. Then, any solution of the form

$$\rho(s) = \pm\eta(s)\hat{\rho}(s) \tag{10.31}$$

is admissible, where $\acute{\eta}(s)$ is an arbitrary real all-pass function possessing the property that

$$\eta(s)\eta(-s) = 1 \tag{10.32}$$

The significance of these coefficient constraints is that they are both necessary and sufficient for the physical realizability of $\rho(s)$, and is summarized in the following theorem. The proof of this result can be found in Ref. [2].

THEOREM 10.1

Given a strictly passive impedance $z_2(s)$, the function defined by the equation

$$Z_{22}(s) \equiv \frac{F(s)}{A(s) - \rho(s)} - z_2(s) \tag{10.33}$$

is positive real if and only if $|\rho(j\omega)| \leq 1$ for all ω and the coefficient conditions Equation 10.29 are satisfied.

The function $Z_{22}(s)$ defined in Equation 10.33 is actually the back-end impedance of a desired equalizer. To see this, we solve for $Z_{22}(s)$ in Equation 10.3 and obtain

$$Z_{22}(s) = \frac{A(s)[z_2(s) + z_2(-s)]}{A(s) - \rho(s)} - z_2(s) = \frac{F(s)}{A(s) - \rho(s)} - z_2(s) \tag{10.34}$$

which is guaranteed to be positive real by Theorem 10.1. This impedance can be realized as the input impedance of a lossless two-port network terminated in a resistor. The removal of this resistor gives the desired matching network. An ideal transformer may be needed at the input port to compensate for the actual level of the generator resistance R_1.

Example 10.2

Design a lossless matching network to equalize the load impedance

$$z_2(s) = \frac{s}{s^2 + 2s + 1} \tag{10.35}$$

to a resistive generator of internal resistance of 0.5 Ω and to achieve the transducer power-gain characteristic

$$G(\omega^2) = \frac{K\omega^2}{\omega^4 - \omega^2 + 1}, \quad 0 \leq K \leq 1 \tag{10.36}$$

From Example 10.1, the load possesses two Class II zeros of transmission of order 1 at $s = 0$ and $s = \infty$. The coefficients of the Laurent series expansions of the functions $A(s)$, $F(s)$, and $\rho(s)$ about the zeros of transmission at $s = 0$ and $s = \infty$ were computed in Example 10.1 as

$$A_0 = 1, \quad F_0 = 0, \quad \rho_0 = 1 \tag{10.37a}$$

$$A_1 = -4, \quad F_1 = 0, \quad \rho_1 = \sqrt{1 - K} - 1 \tag{10.37b}$$

$$F_2 = -4 \tag{10.37c}$$

The coefficient constraints (Equation 10.29b) for the Class II zeros of transmission of order 1 become

$$A_0 = \rho_0, \quad \frac{A_1 - \rho_1}{F_2} \geq 0 \tag{10.38}$$

Clearly, the first condition is always satisfied. To meet the second requirement, we set

$$\frac{-4 - \left(\sqrt{1 - K} - 1\right)}{-4} \geq 0 \tag{10.39}$$

or $0 \leq K \leq 1$, showing that the maximum realizable K is 1. For our purposes, set $K = 1$ and choose the plus sign in Equation 10.24. From Equation 10.34, the equalizer back-end impedance is computed as

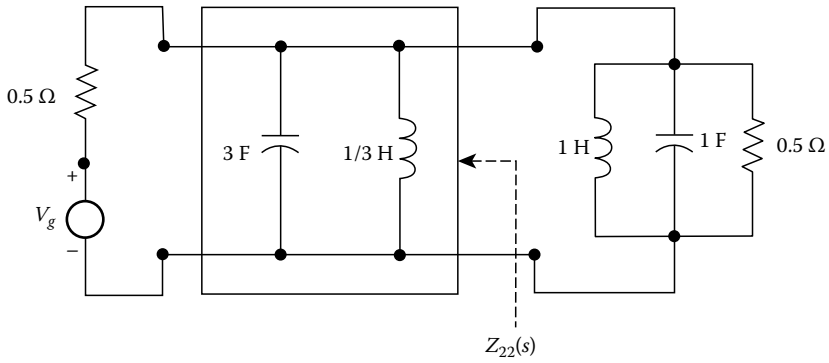


FIGURE 10.3 Equalizer having transducer power-gain characteristics Equation 10.36.

$$Z_{22}(s) = \frac{F(s)}{A(s) - \hat{\rho}(s)} - z_2(s) = \frac{\frac{-4s^2}{(s^2+2s+1)^2}}{\frac{s^2-2s+1}{s^2+2s+1} - \frac{s^2+1}{s^2+s+1}} - \frac{s}{s^2+2s+1}$$
$$= \frac{s}{3s^2+2s+3} \tag{10.40}$$

This impedance can be realized as the input impedance of the parallel connection of an inductor $L = 1/3$ H, a capacitor $C = 3$F, and a resistor $R = 0.5\ \Omega$. The resulting equalizer together with the load is presented in Figure 10.3.

10.3 Design Procedure

We now outline an eight-step procedure for the design of an optimum lossless matching network that equalizes a frequency-dependent load impedance $z_2(s)$ to a resistive generator of internal resistance R_1 and achieves a preassigned transducer power-gain characteristic $G(\omega^2)$ over the entire sinusoidal frequency spectrum.

Step 1. From a preassigned transducer power-gain characteristic $G(\omega^2)$, verify that $G(\omega^2)$ is an even rational real function and satisfies the inequality

$$0 \le G(\omega^2) \le 1 \quad \text{for all } \omega \tag{10.41}$$

The gain level is usually not specified to allow some flexibility.

Step 2. From a prescribed strictly passive load impedance $z_2(s)$, compute

$$r_2(s) \equiv \text{Ev}\, z_2(s) = \frac{1}{2}[z_2(s) + z_2(-s)] \tag{10.42}$$

$$A(s) = \prod_{i=1}^{q} \frac{s - s_i}{s + s_i}, \quad \text{Re}\, s_i > 0 \tag{10.43}$$

where s_i $(i = 1, 2, \ldots, q)$ are the open RHS poles of $z_2(-s)$, and

$$F(s) = 2A(s)r_2(s) \tag{10.44}$$

Step 3. Determine the locations and the orders of the zeros of transmission of $z_2(s)$, which are defined as the closed RHS zeros of the function

$$w(s) = \frac{r_2(s)}{z_2(s)} \tag{10.45}$$

and divide them into respective classes according to Definition 10.2.

Step 4. Perform the unique factorization of the function

$$\hat{\rho}(s)\hat{\rho}(-s) = 1 - G\left(-s^2\right) \tag{10.46}$$

in which the numerator of the minimum-phase solution $\hat{\rho}(s)$ is a Hurwitz polynomial and the denominator $\hat{\rho}(s)$ is a strictly Hurwitz polynomial.

Step 5. Obtain the Laurent series expansions of the functions $A(s)$, $F(s)$, and $\hat{\rho}(s)$ about each zero of transmission s_0 of $z_2(s)$, as follows:

$$A(s) = \sum_{m=0}^{\infty} A_m (s - s_0)^m \tag{10.47a}$$

$$F(s) = \sum_{m=0}^{\infty} F_m (s - s_0)^m \tag{10.47b}$$

$$\hat{\rho}(s) = \sum_{m=0}^{\infty} \rho_m (s - s_0)^m \tag{10.47c}$$

They may be obtained by any available methods.

Step 6. According to the classes of zeros of transmission, list the basic constraints (Equation 10.29) imposed on the coefficients of Equation 10.47. The gain level is ascertained from these constraints. If not all the constraints are satisfied, consider the more general solution

$$\rho(s) = \pm\eta(s)\hat{\rho}(s) \tag{10.48}$$

where $\acute{\eta}(s)$ is an arbitrary real all-pass function. Then repeat Step 5 for $\rho(s)$, starting with lower-order $\acute{\eta}(s)$. If the constraints still cannot be satisfied, modify the preassigned transducer power-gain characteristics $G(\omega^2)$. Otherwise, no match exists.

Step 7. Having successfully carried out Step 6, the equalizer back-end impedance is determined by the equation

$$Z_{22}(s) \equiv \frac{F(s)}{A(s) - \rho(s)} - z_2(s) \tag{10.49}$$

where $\rho(s)$ may be $\hat{\rho}(s)$ and $Z_{22}(s)$ is guaranteed to be positive real.

Step 8. Realize $Z_{22}(s)$ as the input impedance of a lossless two-port network terminated in a resistor. An ideal transformer may be required at the input port to compensate for the actual level of the generator resistance R_1. This completes the design of an equalizer.

Example 10.3

Design a lossless matching network to equalize the RLC load as shown in Figure 10.4 to a resistive generator and to achieve the fifth-order Butterworth transducer power-gain characteristic with a maximal DC gain. The cutoff frequency is 10^8 rad/s.

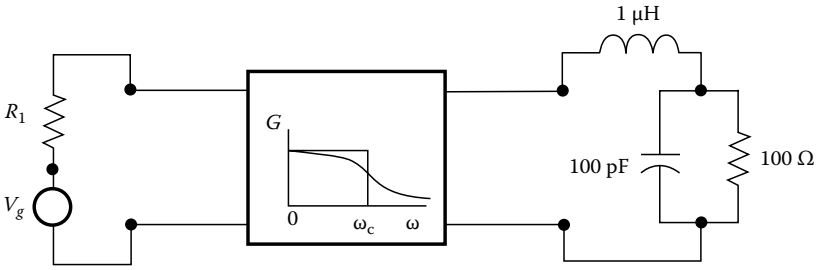


FIGURE 10.4 Broadband matching of an RLC load to a resistive generator.

To simplify the computation, we magnitude-scale the network by a factor of 10^{-2} and frequency-scale it by a factor of 10^{-8}. Thus, s denotes the normalized complex frequency and $\boldsymbol{\omega}$ the normalized real frequency. The load impedance becomes

$$z_2(s) = \frac{s^2 + s + 1}{s + 1} \tag{10.50}$$

We now follow the eight steps outlined below to design a lossless equalizer to meet the desired specifications.

Step 1. The fifth-order Butterworth transducer power-gain characteristic is given by

$$G(\boldsymbol{\omega}^2) = \frac{K_5}{1 + \boldsymbol{\omega}^{10}}, \quad 0 \le K_5 \le 1 \tag{10.51}$$

Our objective is to maximize the DC gain K_5.

Step 2. From the load impedance $z_2(s)$, we compute the functions

$$r_2(s) = \frac{1}{2}[z_2(s) + z_2(-s)] = \frac{1}{1 - s^2} \tag{10.52}$$

$$A(s) = \frac{s - 1}{s + 1} \tag{10.53}$$

where $s = 1$ is the open RHS pole of $z_2(-s)$, and

$$F(s) = 2A(s)r_2(s) = \frac{-2}{(s + 1)^2} \tag{10.54}$$

Step 3. The zeros of transmission $z_2(s)$ are defined by the closed RHS zero of the function.

$$w(s) = \frac{r_2(s)}{z_2(s)} = \frac{1}{(s^2 + s + 1)(1 - s)} \tag{10.55}$$

indicating that $s = \infty$ is a Class IV zero of transmission of order 3.

Step 4. Substituting Equation 10.51 in Equation 10.46 with $-js$ replacing $\boldsymbol{\omega}$ gives

$$\boldsymbol{\rho}(s)\boldsymbol{\rho}(-s) = 1 - G(-s^2) = 1 - \frac{K_5}{1 - s^{10}} = \boldsymbol{\alpha}^{10}\frac{1 - x^{10}}{1 - s^{10}} \tag{10.56}$$

where

$$\alpha = (1 - K_5)^{1/10}, \quad x = \frac{s}{\alpha} \tag{10.57}$$

The minimum-phase solution of Equation 10.56 is found to be

$$\hat{\rho}(s) = \frac{s^5 + 3.23607\alpha s^4 + 5.23607\alpha^2 s^3 + 5.23607\alpha^3 s^2 + 3.23607\alpha^4 s + \alpha^5}{s^5 + 3.23607s^4 + 5.23607s^3 + 5.23607s^2 + 3.23607s + 1} \tag{10.58}$$

The maximum attainable DC gain will be ascertained later from the coefficient conditions.

Step 5. The Laurent series expansions of the functions $A(s)$, $F(s)$, and $\hat{\rho}(s)$ about the zero of transmission $s_0 = \infty$ of $z_2(s)$ are obtained as follows:

$$A(s) = \frac{s-1}{s+1} = 1 - \frac{2}{s} + \frac{2}{s^2} - \frac{2}{s^3} + \cdots \tag{10.59a}$$

$$F(s) = \frac{-2}{(s+1)^2} = 0 + 0 - \frac{2}{s^2} + \frac{4}{s^3} + \cdots \tag{10.59b}$$

$$\hat{\rho}(s) = 1 + \frac{3.23607(\alpha - 1)}{s} + \frac{5.23607(\alpha - 1)^2}{s^2} + \frac{5.23607(\alpha^3 - 3.23607\alpha^2 + 3.23607\alpha - 1)}{s^3} + \cdots \tag{10.59c}$$

Step 6. For a Class IV zero of transmission of order 3, the coefficient conditions are, from Equation 10.29d with $k = 3$,

$$A_m = \rho_m, \quad m = 0, 1, 2 \tag{10.60a}$$

$$\frac{F_2}{A_3 - \rho_3} \geq a_{-1}(\infty) = 1 \tag{10.60b}$$

where $a_{-1}(\infty)$ is the residue of $z_2(s)$ evaluated at the pole $s = \infty$, which is also the zero of transmission of $z_2(s)$. Substituting the coefficients of Equation 10.59 in Equation 10.60 yields the constraints imposed on K_5 as

$$A_0 = 1 = \rho_0 \tag{10.61a}$$

$$A_1 = -2 = \rho_1 = 3.23607(\alpha - 1) \tag{10.61b}$$

$$A_2 = 2 = \rho_2 = 5.23607(\alpha - 1)^2 \tag{10.61c}$$

yielding $\alpha = 0.3819664$, and

$$\frac{F_2}{A_3 - p_3} = \frac{-2}{-2 - 5.23607(\alpha^3 - 3.23607\alpha^2 + 3.23607\alpha - 1)} \geq a_{-1}(\infty) = 1 \tag{10.62}$$

This inequality is satisfied for $\alpha = 0.3819664$. Hence, we choose $\alpha = 0.3819664$ and obtain from Equation 10.57 the maximum realizable DC gain K_5 as

$$K_5 = 1 - \alpha^{10} = 0.99993 \tag{10.63}$$

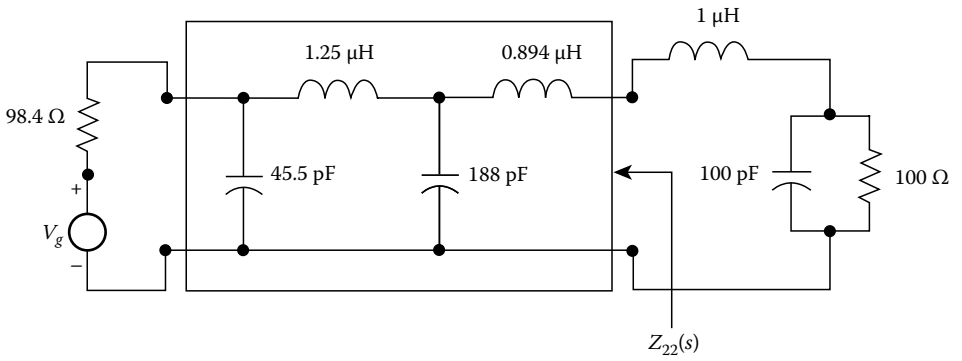


FIGURE 10.5 Fifth-order Butterworth broadband matching equalizer.

With this value of K_5, the minimum-phase reflection coefficient becomes

$$\hat{p}(s) = \frac{s^5 + 1.23607s^4 + 0.76393s^3 + 0.2918s^2 + 0.068884s + 0.0081307}{s^5 + 3.23607s^4 + 5.23607s^3 + 5.23607s^2 + 3.23607s + 1} \tag{10.64}$$

Step 7. The equalizer back-end impedance is determined by

$$\begin{aligned} Z_{22}(s) &\equiv \frac{F(s)}{A(s) - \hat{\rho}(s)} - z_2(s) = \frac{\frac{-2}{(s+1)^2}}{\frac{s-1}{s+1} - \hat{\rho}(s)} - \frac{s^2 + s + 1}{s + 1} \\ &= \frac{0.94427s^4 + 2.1115s^3 + 2.6312s^2 + 2.1591s + 0.9919}{1.0557s^3 + 2.3607s^2 + 2.3131s + 1.0081} \end{aligned} \tag{10.65}$$

Step 8. Expanding $Z_{22}(s)$ in a continued fraction results in

$$Z_{22}(s) = 0.894s + \cfrac{1}{1.88s + \cfrac{1}{1.25s + \cfrac{1}{0.455s + \cfrac{1}{0.984}}}} \tag{10.66}$$

which can be identified as an LC ladder network terminated in a resistor. Denormalizing the element values with regard to magnitude-scaling by a factor of 100 and frequency-scaling by a factor 10^8 gives the final design of the equalizer of Figure 10.5.

Example 10.4

Design a lossless equalizer to match the load

$$z_2(s) = \frac{s^2 + 9s + 8}{s^2 + 2s + 2} \tag{10.67}$$

to a resistive generator and to achieve the largest flat transducer power gain over the entire sinusoidal frequency spectrum.

Step 1. For truly-flat transducer power gain, let

$$G(\omega^2) = K, \quad 0 \le K \le 1 \tag{10.68}$$

Step 2. The following functions are computed from $z_2(s)$:

$$r_2(s)\frac{(s^2-4)^2}{(s^2+2s+2)(s^2-2s+2)} \tag{10.69a}$$

$$A(s)=\frac{s^2-2s+2}{s^2+2s+2} \tag{10.69b}$$

$$F(s)=\frac{2(s^4-8s^2+16)}{(s^2+2s+2)^2} \tag{10.69c}$$

Step 3. Since

$$w(s)=\frac{r_2(s)}{z_2(s)}=\frac{(s^2-4)^2}{(s^2-2s+2)(s^2+9s+8)} \tag{10.70}$$

the load impedance $z_2(s)$ possesses a Class I zero of transmission of order 2 at $s=2$.

Step 4. Substituting Equation 10.68 in Equation 10.46 yields

$$\boldsymbol{\rho}(s)\boldsymbol{\rho}(-s)=1-K \tag{10.71a}$$

the minimum-phase solution of which is found to be

$$\hat{\boldsymbol{\rho}}(s)=\pm\sqrt{1-K} \tag{10.71b}$$

Step 5. For a Class I zero of transmission of order 2, the coefficient conditions (Equation 10.29a) become

$$A(2)=\hat{\boldsymbol{\rho}}(2) \tag{10.72a}$$

$$A_1=\left.\frac{\mathrm{d}A(s)}{\mathrm{d}s}\right|_{s=2}=\boldsymbol{\rho}_1=\left.\frac{\mathrm{d}\boldsymbol{\rho}(s)}{\mathrm{d}s}\right|_{s=2} \tag{10.72b}$$

The Laurent series expansions of the functions $A(s)$, $F(s)$, and $\hat{\boldsymbol{\rho}}(s)$ about the zero of transmission at $s=2$ are not needed.

Step 6. Substituting Equations 10.69b and 10.71b in Equation 10.72 gives

$$A_0=A(2)=0.2=\boldsymbol{\rho}_0=\pm\sqrt{1-K} \tag{10.73a}$$

$$A_1=\left.\frac{\mathrm{d}A(s)}{\mathrm{d}s}\right|_{s=2}=0.08\neq\boldsymbol{\rho}_1=0 \tag{10.73b}$$

Since the coefficient conditions cannot all be satisfied without the insertion of a real all-pass function, let

$$\boldsymbol{\rho}(s)=\boldsymbol{\eta}(s)\hat{\boldsymbol{\rho}}(s)=\frac{s-\boldsymbol{\sigma}_1}{s+\boldsymbol{\sigma}_1}\hat{\boldsymbol{\rho}}(s) \tag{10.74}$$

Using this $\boldsymbol{\rho}$(s) in Equation 10.72 results in the new constraints.

$$A_0=0.2=\boldsymbol{\rho}_0=\pm\frac{2-\boldsymbol{\sigma}_1}{2+\boldsymbol{\sigma}_1}\sqrt{1-K} \tag{10.75a}$$

$$A_1=0.08=\boldsymbol{\rho}_1=\pm\frac{2\boldsymbol{\sigma}_1\sqrt{1-K}}{(2+\boldsymbol{\sigma}_1)^2} \tag{10.75b}$$

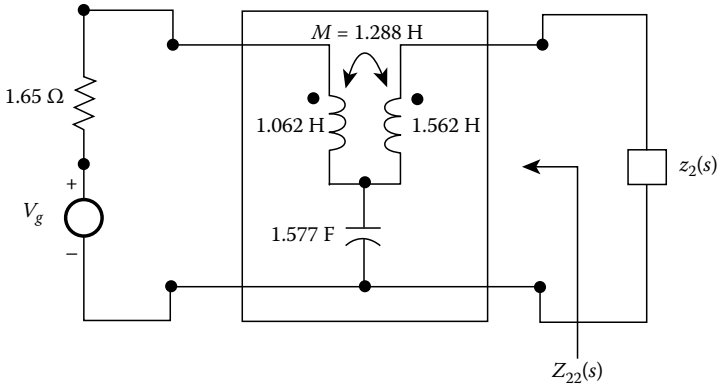


FIGURE 10.6 Lossless equalizer having a truly flat transducer power gain.

which can be combined to yield

$$\sigma_1^2 + 5\sigma_1 - 4 = 0 \tag{10.76}$$

obtaining $\sigma_1 = 0.70156$ or -5.7016. Choosing $\sigma_1 = 0.70156$ and the plus sign for $\rho(s)$, the maximum permissible flat transducer power gain is found to be

$$K_{\max} = 0.82684 \tag{10.77}$$

Step 7. The equalizer back-end impedance is determined as

$$Z_{22}(s) \equiv \frac{F(s)}{A(s) - \rho(s)} - z_2(s) = \frac{1.4161s + 0.8192}{0.58388s + 0.49675} \tag{10.78}$$

Step 8. The positive-real impedance $Z_{22}(s)$ can be realized as the input impedance of a lossless two-port network terminated in a resistor. The overall network is presented in Figure 10.6.

10.4 Explicit Formulas for the RLC Load

In many practical cases, the source can usually be represented by an ideal voltage source in series with a pure resistor, which may be the Thévenin equivalent of some other network, and the load is composed of the parallel combination of a resistor and a capacitor and then in series with an inductor, as shown in Figure 10.7, which may include the parasitic effects of a physical device. The problem is to match out this load and source over a preassigned frequency band to within a given tolerance, and to achieve a prescribed transducer power-gain characteristic $G(\omega^2)$. In the case that $G(\omega^2)$ is of Butterworth or Chebyshev type of response, explicit formulas for the design of such optimum matching networks for any RLC load of the type shown in Figure 10.7 are available, thereby avoiding the necessity of applying the coefficient constraints and solving the nonlinear equations for selecting the optimum design parameters. As a result, we reduce the design of these equalizers to simple arithmetic.

10.4.1 Butterworth Networks

Refer to Figure 10.7. We wish to match out the load impedance

$$z_2(s) = Ls + \frac{R}{RCs + 1} \tag{10.79}$$

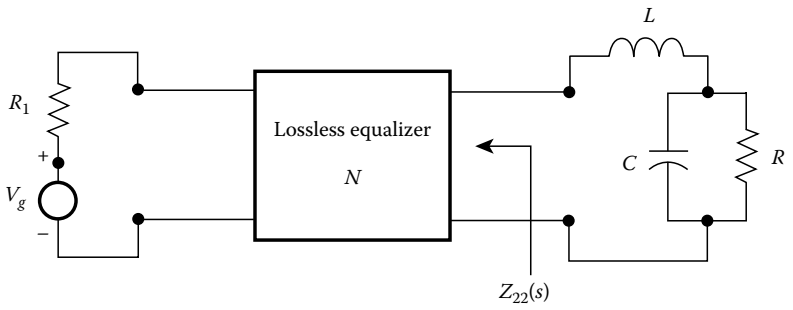


FIGURE 10.7 Broadband matching of an RLC load to a resistive source.

to a resistive generator and to achieve the *n*th-order Butterworth transducer power-gain characteristic

$$G(\omega^2) = \frac{K_n}{1 + (\omega/\omega_c)^{2n}}, \quad 0 \le K_n \le 1 \tag{10.80}$$

with maximum attainable DC gain K_n, where ω_c is the 3 dB bandwidth or the radian cutoff frequency.

The even part $r_2(s)$ of $z_2(s)$ is found to be

$$r_2(s) = \frac{-R}{R^2C^2s^2 - 1} \tag{10.81}$$

Since $z_2(-s)$ has an open RHS pole at $s = 1/RC$, the all-pass real function defined by this pole is given by

$$A(s) = \frac{s - 1/RC}{s + 1/RC} = \frac{RCs - 1}{RCs + 1} \tag{10.82}$$

yielding

$$F(s) = 2A(s)r_2(s) = \frac{-2R}{(RCs + 1)^2} \tag{10.83}$$

We next replace ω by $-js$ in Equation 10.80 and substitute the resulting equation in Equation 10.46 to obtain

$$\rho(s)\rho(-s) = \alpha^{2n}\frac{1 + (-1)^n x^{2n}}{1 + (-1)^n y^{2n}} \tag{10.84}$$

where

$$y = \frac{s}{\omega_c}, \quad x = \frac{y}{\alpha} \tag{10.85a}$$

$$\alpha = (1 - K_n)^{1/2n} \tag{10.85b}$$

As previously shown in Equation 9.29, the minimum-phase solution $\hat{\rho}(s)$ of Equation 10.84 is found to be

$$\hat{\rho}(s) = \alpha^n \frac{q(x)}{q(y)} \tag{10.86}$$

For our purposes, we consider the more general solution

$$\rho(s) = \pm\eta(s)\hat{\rho}(s) \tag{10.87}$$

where $\hat{\eta}(s)$ is an arbitrary first-order real all-pass function of the form

$$\eta(s) = \frac{s - \sigma_1}{s + \sigma_1}, \quad \sigma_1 \geq 0 \tag{10.88}$$

Since the load impedance $z_2(s)$ possesses a Class IV zero of transmission at infinity of order 3, the coefficient constraints become

$$A_m = \rho_m, \quad m = 0, 1, 2 \tag{10.89a}$$

$$L_a \equiv \frac{F_2}{A_3 - \rho_3} \geq L \tag{10.89b}$$

After substituting the coefficients F_2, A_3, and ρ_3 from the Laurent series expansions of $F(s)$, $A(s)$, and $\rho(s)$ in Equation 10.89b, Equation 10.89a can all be satisfied by requiring that the DC gain be

$$K_n = 1 - \left[1 - \frac{2(1 - RC\sigma_1)\sin\gamma_1}{RC\omega_c}\right]^{2n} \tag{10.90}$$

where γ_m is defined in Equation 9.37, and after considerable mathematical manipulations the constraint (Equation 10.89b) becomes

$$L_a = \frac{4R\sin\gamma_1\sin\gamma_3}{(1 - RC\sigma_1)\left[RC\omega_c^2(\alpha^2 - 2\alpha\cos\gamma_2 + 1) + 4\sigma_1\sin\gamma_1\sin\gamma_3\right]} \geq L \tag{10.91}$$

The details of these derivations can be found in Ref. [3]. Thus, with K_n as specified in Equation 10.90, a match is possible if and only if the series inductance L does not exceed a critical inductance L_a. To show that any RLC load can be matched, we must demonstrate that there exists a nonnegative real σ_1 such that L_a can be made at least as large as the given inductance L and satisfies the constraint (Equation 10.90) with $0 \leq K_n \leq 1$. To this end, four cases are distinguished. Let

$$L_{a1} = \frac{R^2C\omega_c\sin\gamma_3}{\left[(RC\omega_c - \sin\gamma_1)^2 + \cos^2\gamma_1\right]\omega_c\sin\gamma_1} > 0 \tag{10.92}$$

$$L_{a2} = \frac{8R\sin^2\gamma_1\sin\gamma_3}{\left[(RC\omega_c - \sin\gamma_3)^2 + (1 + 4\sin^2\gamma_1)\sin\gamma_1\sin\gamma_3\right]\omega_c} > 0 \tag{10.93}$$

Case 1. $RC\omega_c \geq 2\sin\gamma_1$ and $L_{a1} \geq L$. Under this situation, $\sigma_1 = 0$ and the maximum attainable K_n is given by Equation 10.90. The equalizer back-end impedance $Z_{22}(s)$ can be expanded in a continued fraction as

$$Z_{22}(s) = (L_{a1} - L)s + \cfrac{1}{C_2 s + \cfrac{1}{L_3 s + \cfrac{1}{\ddots + \frac{1}{W}}}} \tag{10.94}$$

where
W is a constant representing either a resistance or a conductance, and

$$L_1 = L_{a1} \tag{10.95a}$$

$$C_{2m}L_{2m-1} = \frac{4\sin\gamma_{4m-1}\sin\gamma_{4m+1}}{\omega_c^2(1-2\alpha\cos\gamma_{4m}+\alpha^2)}, \quad m \le \frac{1}{2}(n-1) \tag{10.95b}$$

$$C_{2m}L_{2m+1} = \frac{4\sin\gamma_{4m+1}\sin\gamma_{4m+3}}{\omega_c^2(1-2\alpha\cos\gamma_{4m+2}+\alpha^2)}, \quad m < \frac{1}{2}(n-1) \tag{10.95c}$$

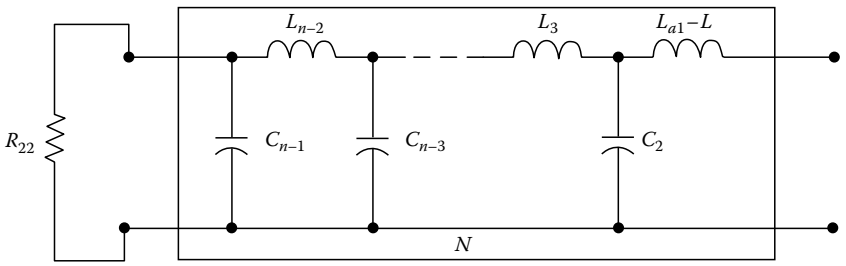
where $m = 1, 2, \ldots, \lceil \frac{1}{2}(n-1) \rceil, n > 1$. In addition, the final reactive element can also be computed directly by the formulas

$$C_{n-1} = \frac{2(1+\alpha^n)\sin\gamma_1}{R(1-\alpha^n)(1+\alpha)\omega_c}, \quad n \text{ odd} \tag{10.96a}$$

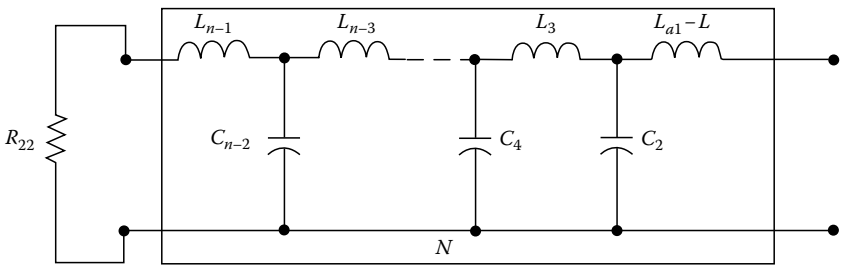
$$L_{n-1} = \frac{2R(1-\alpha^n)\sin\gamma_1}{(1+\alpha^n)(1+\alpha)\omega_c}, \quad n \text{ even} \tag{10.96b}$$

Equation 10.94 can be identified as an LC ladder terminated in a resistor, as depicted in Figure 10.8. The terminating resistance is determined by

$$R_{22} = R\frac{1-\alpha^n}{1+\alpha^n} \tag{10.97}$$



(a) n odd



(b) n even

FIGURE 10.8 nth-order Butterworth ladder network N.

Case 2. $RC\omega_c \geq 2\sin\gamma_1$ and $L_{a1} < L$. Under this situation, σ_1 is nonzero and can be determined by the formula

$$\sigma_1 = \frac{1}{RC}\left(1 + 2\sqrt{p}\sinh\frac{\varphi}{3} - \frac{2RC\omega_c\sin^2\gamma_1 + \sin\gamma_3}{3\sin\gamma_1}\right) \tag{10.98}$$

where

$$\mathrm{p} = \frac{(RC\omega_c - 2\sin\gamma_1)^2\sin\gamma_3}{9\sin\gamma_1} > 0 \tag{10.99a}$$

$$w = \frac{(2RC\omega_c\sin^2\gamma_1 + \sin\gamma_3)}{54\sin^3\gamma_1}\left[3(RC\omega_c - 2\sin\gamma_1)^2\sin\gamma_1\sin\gamma_3 + \left(2RC\omega_c\sin^2\gamma_1 + \sin\gamma_3\right)^2\right] - \frac{R^2C\sin\gamma_3}{2L\sin\gamma_1} \tag{10.99b}$$

$$\varphi = \sinh^{-1}\frac{w}{(\sqrt{p})^3} \tag{10.99c}$$

Using this value of σ_1, the DC gain K_n is computed by Equation 10.90.

Case 3. $RC\omega_c < 2\sin\gamma_1$ and $L_{a2} \geq L$. Then, we have

$$K_n = 1 \tag{10.100a}$$

$$\sigma_1 = \frac{1}{RC}\left(1 - \frac{RC\omega_c}{2\sin\gamma_1}\right) > 0 \tag{10.100b}$$

Case 4. $RC\omega_c < 2\sin\gamma_1$ and $L_{a2} < L$. Then the desired value of σ_1 can be computed by Equation 10.98. Using this value of σ_1, the DC gain K_n is computed by Equation 10.90.

Example 10.5

Let

$$R = 100\,\Omega, \quad C = 100\,\mathrm{pF}, \quad L = 0.5\,\mu\mathrm{F} \tag{10.101a}$$

$$n = 6, \quad \omega_c = 10^8\,\mathrm{rad/s} \tag{10.101b}$$

From Equation 10.92, we first compute

$$L_{a1} = \frac{100\sin 45^\circ}{[(1 - \sin 15^\circ)^2 + \cos^2 15^\circ] \times 10^8\sin 15^\circ} = 1.84304\,\mu\mathrm{H} \tag{10.102}$$

Since $L_{a1} > L$ and

$$RC\omega_c = 1 > 2\sin 15^\circ = 0.517638 \tag{10.103}$$

Case 1 applies and the matching network can be realized as an LC ladder terminating in a resistor as shown in Figure 10.8b. With $\sigma_1 = 0$, the maximum attainable DC gain K_6 is from Equation 10.90

$$K_6 = 1 - \left(1 - \frac{2\sin 15^\circ}{RC\omega_c}\right)^{12} = 0.999841 \tag{10.104}$$

giving from Equation 10.85b

$$\alpha = (1 - K_6)^{1/12} = 0.482362 \tag{10.105}$$

Applying Equation 10.95 yields the element values of the LC ladder network, as follows:

$$L_1 = L_{a1} = 1.84304\,\mu\text{H} \tag{10.106a}$$

$$C_2 = \frac{4\sin 45^\circ \sin 75^\circ}{1.84304 \times 10^{-6} \times 10^{16}(1 - 2 \times 0.482362\cos 60^\circ + 0.482362^2)} = 197.566\,\text{pF} \tag{10.106b}$$

$$L_3 = \frac{4\sin 75^\circ \sin 105^\circ}{197.566 \times 10^{-12} \times 10^{16}(1 - 2 \times 0.482362\cos 90^\circ + 0.482362^2)} = 1.53245\,\mu\text{H} \tag{10.106c}$$

$$C_4 = \frac{4\sin 105^\circ \sin 135^\circ}{1.53245 \times 10^{-6} \times 10^{16}(1 - 2 \times 0.482362\cos 120^\circ + 0.482362^2)} = 103.951\,\text{pF} \tag{10.106d}$$

$$L_5 = \frac{4\sin 135^\circ \sin 165^\circ}{103.951 \times 10^{-12} \times 10^{16}(1 - 2 \times 0.482362\cos 150^\circ + 0.482362^2)} = 0.34051\,\mu\text{H} \tag{10.106e}$$

The last reactive elements L_5 can also be calculated directly from Equation 10.96b as

$$L_5 = \frac{2 \times 100(1 - 0.482362^6)\sin 15^\circ}{(1 + 0.482362^6)(1 + 0.482362)10^8} = 0.34051\,\mu\text{H} \tag{10.107}$$

Finally, the terminating resistance is determined from Equation 10.97 as

$$R_{22} = 100\frac{1 - 0.482362^6}{1 + 0.482362^6} = 97.512\,\Omega \tag{10.108}$$

The matching network together with its terminations is presented in Figure 10.9. We remark that for computational accuracy we retain five significant figures in all the calculations. In practice, one or two significant digits are sufficient, as indicated in the figure.

Example 10.6

Let

$$R = 100\,\Omega, \quad C = 50\,\text{pF}, \quad L = 0.5\,\mu\text{F} \tag{10.109a}$$

$$n = 5, \quad \omega_c = 10^8\,\text{rad/s} \tag{10.109b}$$

Since

$$RC\omega_c = 0.5 < 2\sin 18^\circ = 0.618 \tag{10.110}$$

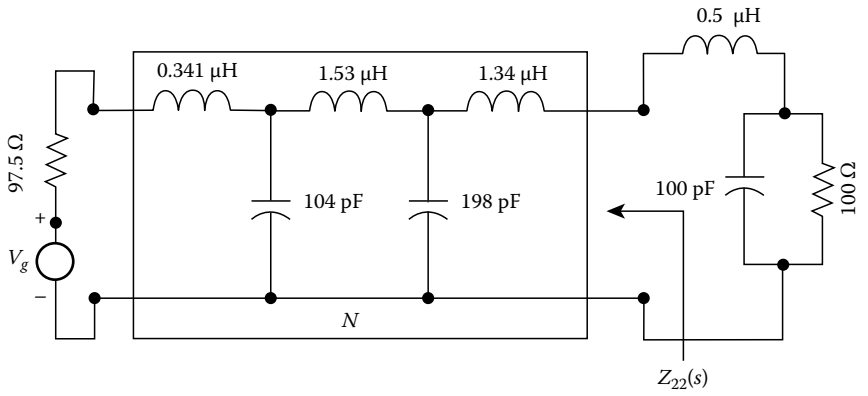


FIGURE 10.9 Sixth-order Butterworth matching network.

and from Equation 10.93,

$$L_{a2} = 1.401\ \mu\text{H} > L = 0.5\ \mu\text{H} \tag{10.111}$$

Case 3 applies, and we have $K_5 = 1$ and from Equation 10.100b

$$\sigma_1 = 0.381966 \times 10^8 \tag{10.112}$$

The normalized reflection coefficient is found to be

$$\rho(y) = \frac{(y - 0.381966)y^5}{(y + 0.381966)(y^5 + 3.23607y^4 + 5.23607y^3 + 5.23607y^2 + 3.23607y + 1)} \tag{10.113}$$

where $y = s/10^8$. Finally, we compute the equalizer back-end impedance as

$$\begin{aligned}
\frac{Z_{22}(s)}{100} &= \frac{F(y)}{A(y) - \rho(y)} - z_2(y) \\
&= \frac{\frac{-2}{(0.5y+1)^2}}{\frac{0.5y-1}{0.5y+1} - \rho(s)} - 0.5y - \frac{1}{0.5y + 1} \\
&= \frac{2.573y^5 + 4.1631y^4 + 5.177y^3 + 4.2136y^2 + 2.045y + 0.38197}{2.8541y^4 + 4.618y^3 + 4.118y^2 + 2.045y + 0.38197} \\
&= 0.9015y + \cfrac{1}{1.949y + \cfrac{1}{1.821y + \cfrac{1}{0.8002y + \cfrac{1}{\cfrac{1.005y + 0.3822}{0.9944y + 0.3822}}}}}
\end{aligned} \tag{10.114}$$

The final matching network together with its terminations is presented in Figure 10.10.

Example 10.7

Let

$$R = 100\ \Omega, \quad C = 50\ \text{pF}, \quad \text{L} = 3\ \mu\text{F} \tag{10.115a}$$

$$n = 4, \quad \omega_c = 10^8\ \text{rad/s} \tag{10.115b}$$

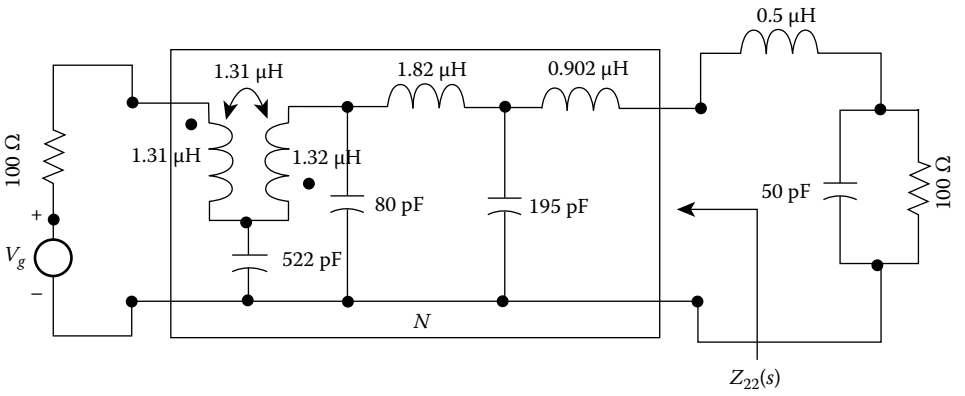


FIGURE 10.10 Fifth-order Butterworth matching network.

Since

$$RC\omega_c = 0.5 < 2\sin 22.5^\circ = 0.76537 \tag{10.116}$$

and from Equation 10.93

$$L_{a2} = 1.462\ \mu\text{H} < L = 3\ \mu\text{H} \tag{10.117}$$

Case 4 applies, and from Equation 10.98

$$\sigma_1 = 1.63129 \times 10^8 \tag{10.118}$$

where Equation 10.99

$$p = 0.0188898, \quad w = 0.2304, \quad \varphi = 5.17894 \tag{10.119}$$

From Equation 10.90, the maximum attainable DC gain is obtained as

$$K_4 = 1 - \left[1 - \frac{2(1 - 100 \times 50 \times 10^{-12} \times 1.63129 \times 10^8)\sin 22.5^\circ}{100 \times 50 \times 10^{-12} \times 10^8}\right]^8 = 0.929525 \tag{10.120}$$

giving from Equation 10.85b

$$\alpha = (1 - K_4)^{1/8} = 0.717802 \tag{10.121}$$

Finally, the normalized reflection coefficient $\rho(s)$ is obtained as

$$\rho(s) = \frac{(y - 1.63129)(y^4 + 1.87571\,y^3 + 1.75913\,y^2 + 0.966439\,y + 0.265471)}{(y + 1.63129)(y^4 + 2.61313\,y^3 + 3.41421\,y^2 + 2.61313\,y + 1)} \tag{10.122}$$

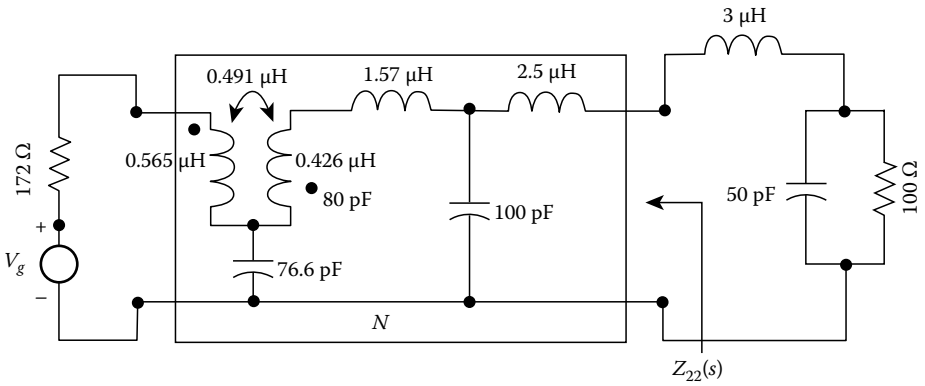


FIGURE 10.11 Fourth-order Butterworth matching network.

where $y = s/10^8$. Finally, we compute the equalizer back-end impedance as

$$\begin{aligned}\frac{Z_{22}(s)}{100} &= \frac{F(y)}{A(y)-\rho(y)} - z_2(y) = \frac{\frac{-2}{(0.5y+1)^2}}{\frac{0.5y-1}{0.5y+1}-\rho(s)} - 0.5y - \frac{1}{0.5y+1} \\ &= \frac{3.3333y^4 + 7.4814y^3 + 8.6261y^2 + 5.9748y + 2.0642}{1.3333y^3 + 2.9926y^2 + 2.9195y + 1.1982} \\ &= 2.5y + \cfrac{1}{1.0046y + \cfrac{1}{1.5691y + \cfrac{1}{\cfrac{1.0991y + 2.0642}{0.84591y + 1.1982}}}}\end{aligned} \tag{10.123}$$

The final matching network together with its terminations is presented in Figure 10.11.

10.4.2 Chebyshev Networks

Refer again to Figure 10.7. We wish to match out the load impedance

$$z_2(s) = Ls + \frac{R}{RCs+1} \tag{10.124}$$

to a resistive generator and to achieve the nth-order Chebyshev transducer power-gain characteristic

$$G(\omega^2) = \frac{K_n}{1+\varepsilon^2 C_n^2(\omega/\omega_c)}, \quad 0 \le K_n \le 1 \tag{10.125}$$

with maximum attainable constant K_n. Following Equation 10.84, we obtain

$$\rho(s)\rho(-s) = (1-K_n)\frac{1+\hat{\varepsilon}^2 C_n^2(-jy)}{1+\varepsilon^2 C_n^2(-jy)} \tag{10.126}$$

where $y = s/\omega_c$ and

$$\hat{\varepsilon} = \frac{\varepsilon}{\sqrt{1-K_n}} \tag{10.127}$$

As in Equation 10.86, let $\hat{\rho}(s)$ be the minimum-phase solution of Equation 10.126. For our purposes, we consider the more general solution

$$\rho(s) = \pm\eta(s)\hat{\rho}(s) \tag{10.128}$$

where $\acute{\eta}(s)$ is an arbitrary first-order real all-pass function of the form

$$\eta(s) = \frac{s - \sigma_1}{s + \sigma_1}, \quad \sigma_1 \geq 0 \tag{10.129}$$

Since the load impedance $z_2(s)$ possesses a Class IV zero of transmission at infinity of order 3, the coefficient constraints become

$$A_m = \rho_m, \quad m = 0, 1, 2 \tag{10.130a}$$

$$\frac{F_2}{A_3 - \rho_3} \geq L \tag{10.130b}$$

After substituting the coefficients F_2, A_3, and ρ_3 from the Laurent series expansions of $F(s)$, $A(s)$, and $\rho(s)$ in Equation 10.130a, they lead to the constraints on the constant K_n as

$$K_n = 1 - \varepsilon^2 \sinh^2 \left\{ n \sinh^{-1} \left[\sinh a - \frac{2(1 - RC\sigma_1)\sin\gamma_1}{RC\omega_c} \right] \right\} \tag{10.131}$$

where γ_m is defined in Equation 9.37, and

$$a = \frac{1}{n} \sinh^{-1} \frac{1}{\varepsilon} \tag{10.132}$$

To apply the constraint (Equation 10.130b), we rewrite it as

$$L_b \equiv \frac{F_2}{A_3 - \rho_3} \geq L \tag{10.133}$$

After substituting the coefficients F_2, A_3, and ρ_3 from the Laurent series expansions of $F(s)$, $A(s)$, and $\rho(s)$ in Equation 10.133 and after considerable mathematical manipulations, we obtain

$$L_b = \frac{4R \sin\gamma_1 \sin\gamma_3}{(1 - RC\sigma_1)\left[RC\omega_c^2 f_1(\sin a, \sinh\hat{a}) + 4\sigma_1 \sin\gamma_1 \sin\gamma_3\right]} \geq L \tag{10.134}$$

where

$$\hat{a} = \frac{1}{n} \sinh^{-1} \frac{1}{\hat{\varepsilon}} = \frac{1}{n} \sinh^{-1} \frac{\sqrt{1 - K_n}}{\varepsilon} \tag{10.135}$$

$$f_m(x,y) = x^2 + y^2 - 2xy\cos\gamma_{2m} + \sin^2\gamma_{2m}, \quad m = 1, 2, \ldots, \left\lceil \frac{1}{2} n \right\rceil \tag{10.136}$$

The details of these derivations can be found in Ref. [3]. Thus, with K_n as specified in Equation 10.131, a match is possible if and only if the series inductance L does not exceed a critical inductance L_b. To show that any RLC load can be matched, we must demonstrate that there exists a nonnegative real σ_1 such that L_b can be made at least as large as the given inductance L and satisfies the constraint (Equation 10.131) with $0 \le K_n \le 1$. To this end, four cases are distinguished. Let

$$L_{b1} = \frac{R^2 C\omega_c \sin\gamma_3}{\left[(1 - RC\omega_c \sinh a \sin\gamma_1)^2 + R^2C^2\omega_c^2 \cosh^2 a \cos^2\gamma_1\right]\omega_c \sin\gamma_1} > 0 \tag{10.137}$$

$$L_{b2} = \frac{8R \sin^2\gamma_1 \sin\gamma_3}{\left[(RC\omega_c \sinh a - \sin\gamma_3)^2 + (1 + 4\sin^2\gamma_1)\sin\gamma_1 \sin\gamma_3 + R^2C^2\omega_c^2 \sin^2\gamma_2\right]\omega_c \sinh a} > 0 \tag{10.138}$$

Observe that both L_{b1} and L_{b2} are positive.

Case 1. $RC\omega_c \sinh a \ge 2\sin\gamma_1$ and $L_{b1} \ge L$. Under this situation, $\sigma_1 = 0$ and the maximum attainable K_n is given by

$$K_n = 1 - \varepsilon^2 \sinh^2\left[n \sinh^{-1}\left(\sinh a - \frac{2\sin\gamma_1}{RC\omega_c}\right)\right] \tag{10.139}$$

The equalizer back-end impedance $Z_{22}(s)$ can be expanded in a continued fraction as in Equation 10.94 with L_{b1} replacing L_{a1} and realized by the LC ladders Figure 10.8 with

$$L_1 = L_{b1} \tag{10.140a}$$

$$C_{2m}L_{2m-1} = \frac{4\sin\gamma_{4m-1}\sin\gamma_{4m+1}}{\omega_c^2 f_{2m}(\sinh a, \sinh\hat{a})}, \quad m < \frac{1}{2}(n-1) \tag{10.140b}$$

$$C_{2m}L_{2m+1} = \frac{4\sin\gamma_{4m+1}\sin\gamma_{4m+3}}{\omega_c^2 f_{2m+1}(\sinh a, \sinh\hat{a})}, \quad m < \frac{1}{2}(n-1) \tag{10.140c}$$

where $m = 1, 2, \ldots, \left\lceil \frac{1}{2}(n-1)\right\rceil, n > 1$. In addition, the final reactive element can also be computed directly by the formulas

$$C_{n-1} = \frac{2(\sinh na + \sinh n\hat{a})\sin\gamma_1}{R\omega_c(\sinh a + \sinh\hat{a})(\sinh na - \sinh n\hat{a})}, \quad n \text{ odd} \tag{10.141a}$$

$$L_{n-1} = \frac{2R(\cosh na - \cosh n\hat{a})\sin\gamma_1}{\omega_c(\sinh a + \sinh\hat{a})(\cosh na + \cosh n\hat{a})}, \quad n \text{ even} \tag{10.141b}$$

The terminating resistance is determined by

$$R_{22} = R\frac{\sinh na - \sinh n\hat{a}}{\sinh na + \sinh n\hat{a}}, \quad n \text{ odd} \tag{10.142a}$$

$$R_{22} = R\frac{\cosh na - \cosh n\hat{a}}{\cosh na + \cosh n\hat{a}}, \quad n \text{ even} \tag{10.142b}$$

Case 2. $RC\omega_c \sinh a \ge 2\sin\gamma_1$ and $L_{b1} < L$. Under this situation, σ_1 is nonzero and can be determined by the formula

$$\sigma_1 = \frac{1}{RC}\left(1 + 2\sqrt{q}\sinh\frac{\varphi}{3} - \frac{2RC\omega_c \sin^2\gamma_1 \sinh a + \sin\gamma_3}{3\sin\gamma_1}\right) \tag{10.143}$$

where

$$q = \frac{(RC\omega_c \sinh a - 2\sin\gamma_1)^2 \sin\gamma_3 + 3R^2C^2\omega_c^2 \sin\gamma_1 \cos^2\gamma_1}{9\sin\gamma_1} > 0 \tag{10.144a}$$

$$\begin{aligned}\zeta = {} & \frac{(2RC\omega_c \sin^2\gamma_1 \sinh a + \sin\gamma_3)}{54\sin^3\gamma_1} \\ & \times \left[3(RC\omega_c \sinh a - 2\sin\gamma_1)^2 \sin\gamma_1 \sin\gamma_3 \right. \\ & \left. + \; 2.25R^2C^2\omega_c^2 \sin^2\gamma_2 + \left(2RC\omega_c \sin^2\gamma_1 \sinh a + \sin\gamma_3\right)^2\right] \\ & - \frac{R^2C\sin\gamma_3}{2L\sin\gamma_1}\end{aligned} \tag{10.144b}$$

$$\varphi = \sinh^{-1}\frac{\zeta}{\left(\sqrt{q}\right)^3} \tag{10.144c}$$

Using this value of σ_1, the constant K_n is computed by Equation 10.131.

Case 3. $RC\omega_c \sinh a < 2\sin\gamma_1$ and $L_{b2} \geq L$. Then, we have

$$K_n = 1 \tag{10.145a}$$

$$\sigma_1 = \frac{1}{RC}\left(1 - \frac{RC\omega_c \sinh a}{2\sin\gamma_1}\right) > 0 \tag{10.145b}$$

Case 4. $RC\omega_c \sinh a < 2\sin\gamma_1$ and $L_{b2} < L$. Then the desired value of σ_1 can be computed by formula given by Equation 10.143. Using this value of σ_1, the constant K_n is computed by Equation 10.131.

Example 10.8

Let

$$R = 100\,\Omega, \quad C = 500\,\text{pF}, \quad L = 0.5\,\mu\text{F} \tag{10.146a}$$

$$n = 6, \quad \varepsilon = 0.50885(1\text{ dB ripple}), \quad \omega_c = 10^8\text{ rad/s} \tag{10.146b}$$

From Equation 10.137, we first compute

$$\begin{aligned}L_{b1} &= \frac{500\sin 45^\circ}{\left[(1 - 5\sinh 0.237996\sin 15^\circ)^2 + 25\cosh^2 0.237996\cos^2 15^\circ\right]10^8\sin 15^\circ} \\ &= 0.54323\,\mu\text{H}\end{aligned} \tag{10.147}$$

Since $L_{b1} > L$ and

$$RC\omega_c \sinh a = 5\sinh 0.237996 = 1.20125 > 2\sin 15^\circ = 0.517638 \tag{10.148}$$

Case 1 applies and the matching network can be realized as an LC ladder terminating in a resistor as shown in Figure 10.8. With $\sigma_1 = 0$, the maximum attainable DC gain K_6 is from Equation 10.131

$$K_6 = 1 - 0.50885^2 \sinh^2\left[6\sinh^{-1}\left(\sinh 0.237996 - \frac{2\sin 15^\circ}{5}\right)\right] = 0.78462 \tag{10.149}$$

giving from Equations 10.127 and 10.135

$$\hat{\varepsilon} = \frac{0.50885}{\sqrt{1 - 0.78462}} = 1.0964 \tag{10.150a}$$

$$\hat{\alpha} = \frac{1}{6}\sinh^{-1}\frac{1}{1.0964} = 0.13630 \tag{10.150b}$$

Applying formulas given by Equation 10.140 yields the element values of the LC ladder network, as follows:

$$L_1 = L_{b1} = 0.54323\,\mu\text{H}, \quad C_2 = 634\,\text{pF}, \quad L_3 = 0.547\,\mu\text{H} \tag{10.151a}$$

$$C_4 = 581\,\text{pF}, \quad L_5 = 0.329\,\mu\text{H} \tag{10.151b}$$

The last reactive element L_5 can be calculated directly from Equation 10.141b as

$$L_5 = \frac{2 \times 100[\cosh(6 \times 0.237996) - \cosh(6 \times 0.13630)]\sin 15^\circ}{10^8(\sinh 0.237996 + \sinh 0.13630)[\cosh(6 \times 0.237996) + \cosh(6 \times 0.13630)]} = 0.328603\,\mu\text{H} \tag{10.152}$$

Finally, the terminating resistance is determined from Equation 10.142b as

$$R_{22} = 100\frac{\cosh(6 \times 0.237996) - \cosh(6 \times 0.13630)}{\cosh(6 \times 0.237996) + \cosh(6 \times 0.13630)} = 23.93062\,\Omega \tag{10.153}$$

The matching network together with its terminations is presented in Figure 10.12. We remark that for computational accuracy we retain five significant figures in all the calculations. In practice, one or two significant digits are sufficient, as indicated in the figure.

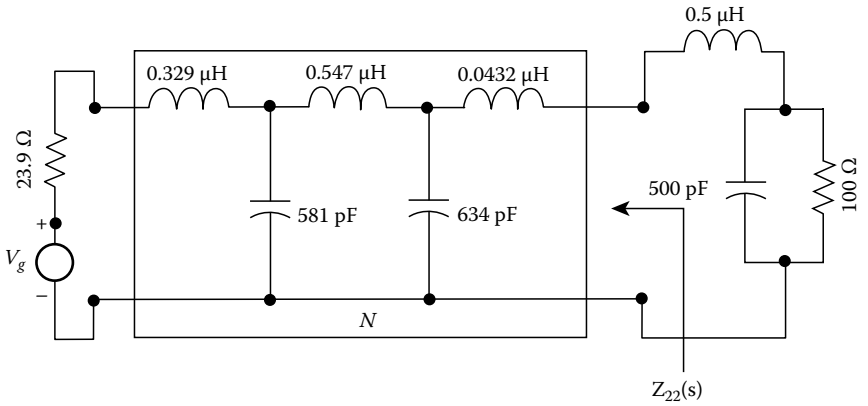


FIGURE 10.12 Sixth-order Chebyshev matching network.

Example 10.9

Let

$$R = 100\,\Omega, \quad C = 500\,\text{pF}, \quad L = 1\,\mu\text{H} \tag{10.154a}$$

$$n = 5, \quad \varepsilon = 0.50885\,(1\text{ dB ripple}), \quad \omega_c = 10^8\text{ rad/s} \tag{10.154b}$$

From Equation 10.137, we first compute

$$L_{b1} = 0.52755\,\mu\text{H} \tag{10.155}$$

Since $L_{b1} < L$ and

$$RC\omega_c \sinh a = 5 \sinh 0.28560 = 1.44747 > 2 \sin 18^\circ = 0.61803 \tag{10.156}$$

Case 2 applies. From Equation 10.144, we obtain

$$q = 7.73769, \quad \zeta = 7.84729, \quad \varphi = 0.356959 \tag{10.157}$$

Substituting these in Equation 10.143 gives

$$\sigma_1 = 0.0985305 \times 10^8 \tag{10.158}$$

From Equation 10.131, the maximum attainable constant K_5 is found to be

$$K_5 = 0.509206 \tag{10.159}$$

The rest of the calculations proceed exactly as in the previous example, and the details are omitted.

Example 10.10

Let

$$R = 100\,\Omega, \quad C = 100\,\text{pF}, \quad L = 0.5\,\mu\text{F} \tag{10.160a}$$

$$n = 5, \quad \varepsilon = 0.76478\,(2\text{ dB ripple}), \quad \omega_c = 10^8\text{ rad/s} \tag{10.160b}$$

We first compute

$$a = \frac{1}{5} \sinh^{-1} \frac{1}{\varepsilon} = 0.2166104 \tag{10.161}$$

and from Equation 10.138

$$L_{b2} = 2.72234\,\mu\text{H} \tag{10.162}$$

Since $L_{b2} \geq L$ and

$$RC\omega_c \sinh a = \sinh 0.2166104 = 0.218308 < 2 \sin 18^\circ = 0.618034 \tag{10.163}$$

Case 3 applies. Then $K_5 = 1$ and from Equation 10.145b

$$\sigma_1 = \frac{1}{10^{-8}}\left(1 - \frac{\sinh 0.2166104}{2\sin 18^\circ}\right) = 0.64677 \times 10^8 \tag{10.164}$$

For $K_5 = 1$, Equation 10.126 degenerates into

$$\rho(y)\rho(-y) = \frac{\varepsilon^2 C_5^2(-jy)}{1 + \varepsilon^2 C_5^2(-jy)} \tag{10.165}$$

where $y = s/10^8$, the minimum-phase solution of which is found to be

$$\hat{\rho}(y) = \frac{y^5 + 1.25y^3 + 0.3125y}{y^5 + 0.706461y^4 + 1.49954y^3 + 0.693477y^2 + 0.459349y + 0.0817225} \tag{10.166}$$

A more general solution is given by

$$\rho(y) = \frac{y - 0.64677 \times 10^8}{y + 0.64677 \times 10^8}\hat{\rho}(y) \tag{10.167}$$

Finally, we compute the equalizer back-end impedance as

$$\begin{aligned}
\frac{Z_{22}(y)}{100} &= \frac{F(y)}{A(y) - \rho(y)} - z_2(y) \\
&= \frac{\frac{-2}{(y+1)^2}}{\frac{y-1}{y+1} - \rho(x)} - 0.5y - \frac{1}{y+1} \\
&= \frac{1.63267y^5 + 0.576709y^4 + 2.15208y^3 + 0.533447y^2 + 0.554503y + 0.0528557}{0.734663y^4 + 0.259505y^3 + 0.639438y^2 + 0.123844y + 0.0528557} \\
&= 2.222y + \cfrac{1}{1.005y + \cfrac{1}{3.651y + \cfrac{1}{0.8204y + \cfrac{1}{\cfrac{0.2441y + 0.05286}{0.02736y + 0.05286}}}}}
\end{aligned} \tag{10.168}$$

The final matching network together with its terminations is presented in Figure 10.13.

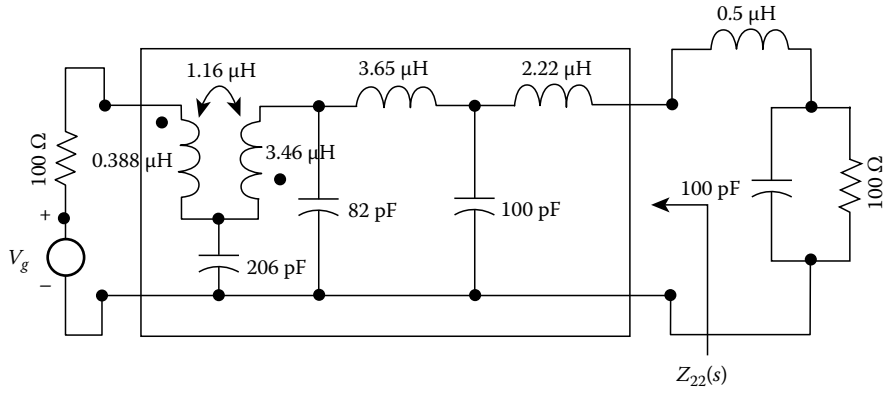


FIGURE 10.13 Fifth-order Chebyshev matching network.

Example 10.11

Let

$$R = 100\,\Omega, \quad C = 100\,\text{pF}, \quad L = 3\,\mu\text{F} \tag{10.169a}$$

$$n = 4, \quad \varepsilon = 0.76478\,(2\text{ dB ripple}), \quad \omega_c = 10^8\text{ rad/s} \tag{10.169b}$$

We first compute

$$a = \frac{1}{4}\sinh^{-1}\frac{1}{0.76478} = 0.27076 \tag{10.170}$$

and from Equation 10.138

$$L_{b2} = 2.66312\,\mu\text{H} \tag{10.171}$$

Since $L_{b2} < L$ and

$$RC\omega_c \sinh a = \sinh 0.27076 = 0.27408 < 2\sin 22.5^\circ = 0.765367 \tag{10.172}$$

Case 4 applies. From Equation 10.144, we obtain

$$q = 0.349261, \quad \zeta = 0.390434, \quad \varphi = 1.39406 \tag{10.173}$$

Substituting these in Equation 10.143 gives

$$\sigma_1 = 0.694564 \times 10^8 \tag{10.174}$$

obtaining from Equations 10.127 and 10.131

$$K_4 = 0.984668, \quad \hat{\varepsilon} = 6.17635 \tag{10.175}$$

From Equation 10.126,

$$\rho(y)\rho(-y) = (1 - K_4)\frac{1 + \hat{\varepsilon}^2 C_4^2(-jy)}{1 + \varepsilon^2 C_4^2(-jy)} \tag{10.176}$$

where $y = s/10^8$, the minimum-phase solution of which is found to be

$$\hat{\rho}(y) = \frac{y^4 + 0.105343\,y^3 + 1.00555\,y^2 + 0.0682693\,y + 0.126628}{y^4 + 0.716215\,y^3 + 1.25648\,y^2 + 0.516798\,y + 0.205765} \tag{10.177}$$

A more general solution is given by

$$\rho(y) = \frac{y - 0.694564 \times 10^8}{y + 0.694564 \times 10^8}\hat{\rho}(s) \tag{10.178}$$

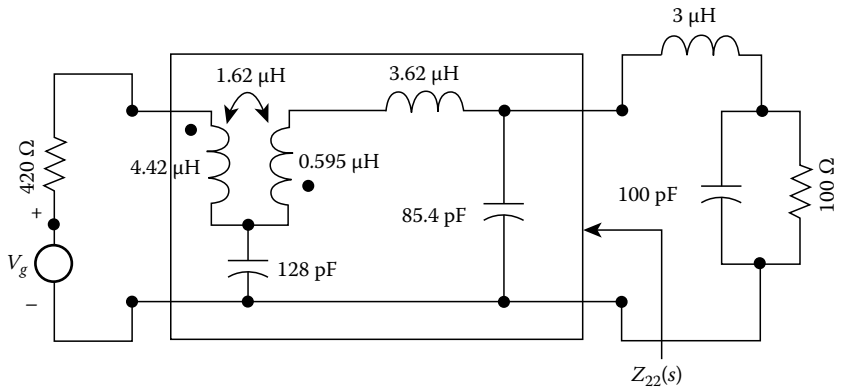


FIGURE 10.14 A fourth-order Chebyshev matching network.

Finally, we compute the equalizer back-end impedance as

$$\begin{aligned}\frac{Z_{22}(y)}{100} &= \frac{F(y)}{A(y)-\rho(y)} - z_2(y) = \frac{\frac{-2}{(y+1)^2}}{\frac{y-1}{y+1}-\rho(s)} - \frac{3y^2+3y+1}{y+1} \\ &= \frac{0.780486\,y^2+0.320604\,y+0.23087}{0.666665\,y^3+0.273851\,y^2+0.413057\,y+0.0549659} \\ &= \cfrac{1}{0.8542\,y+\cfrac{1}{3.616\,y+\cfrac{1}{\cfrac{0.1219\,y+0.2309}{0.2159\,y+0.05497}}}}\end{aligned} \tag{10.179}$$

The final matching network together with its terminations is presented in Figure 10.14.

References

1. D. C. Youla, A new theory of broad-band matching, *IEEE Trans. Circuit Theory*, CT-11, 30–50, 1964.
2. W. K. Chen, *Theory and Design of Broadband Matching Networks,* Cambridge, UK: Pergamon Press, 1976.
3. W. K. Chen, Explicit formulas for the synthesis of optimum broadband impedance-matching networks, *IEEE Trans. Circuits Syst*., CAS-24, 157–169, 1977.

II

Active Filters

Wai-Kai Chen
University of Illinois at Chicago

11 Low-Gain Active Filters *Phillip E. Allen, Benjamin J. Blalock, and Stephen W. Milam* **11**-1
Introduction • First- and Second-Order Transfer Functions • First-Order Filter Realizations • Second-Order Positive-Gain Filters • Second-Order Biquadratic Filters • Higher Order Filters • Influence of Nonidealities • Summary • References

12 Single-Amplifier Multiple-Feedback Filters *F. William Stephenson* **12**-1
Introduction • General Structure for Single-Amplifier Filters • All-Pole Realizations of the MFB Structure • MFB All-Pole Designs • Practical Considerations in the Design of MFB Filters • Modified Multiple-Loop Feedback Structure • Biquadratic MMFB Structure • Conclusions • References

13 Multiple-Amplifier Biquads *Norbert J. Fliege* **13**-1
Introduction • Biquads with Decoupled Time Constants • Dual-Amplifier Twin-*T* Biquads • GIC-Derived Dual-Amplifier Biquads • GIC-Derived Three-Amplifier Biquads • State-Variable-Based Biquads • All-Pass-Based Biquads • Summary • References

14 The Current Generalized Immittance Converter Biquads *Wasfy B. Mikhael* **14**-1
Introduction • Biquadratic Structure Using the Antoniou CGIC (2-OA CGIC Biquad) • Realization of the Most Commonly Used Biquad Functions Using the 2-OA CGIC • Stability during Activation and Sensitivity Using the 2-OA CGIC Biquad • Design and Tuning Procedure of the 2-OA CGIC Biquad • Design Examples Using the 2-OA CGIC Biquad • Practical High-Order Design Examples Using the 2-OA CGIC Biquad • Universal 2-OA CGIC Biquad • 3-OA CGIC Biquadratic Structure • Design and Tuning Procedure of the 3-OA CGIC Biquad • Practical Sixth-Order Elliptic BP Filter Design Using the 3-OA CGIC Biquad • Composite GIC Biquad: 2 C 2-OA CGIC • Summary • References

15 High-Order Filters *Rolf Schaumann* **15**-1
Introduction • Cascade Realizations • Multiple-Loop Feedback Realizations • Ladder Simulations • Summary • References

16 Continuous-Time Integrated Filters *Rolf Schaumann* **16**-1
Introduction • MOSFET-*C* Filters • g_m-*C* Filters • Tuning • Discussion and Conclusions • References

17 Switched-Capacitor Filters *Jose Silva-Martinez and Edgar Sánchez-Sinencio* **17**-1
Introduction • Basic Building Blocks • Effects of the Op-Amp Finite Parameters • Noise and Clock Feedthrough • Design Considerations for Low-Voltage Switched-Capacitor Circuits • Fully Differential Filters • Design Examples • References

11

Low-Gain Active Filters

11.1 Introduction **11**-1
11.2 First- and Second-Order Transfer Functions **11**-1
First-Order Transfer Functions • Second-Order Transfer Functions • Frequency Response (Magnitude and Phase) • Tuning Active Filters
11.3 First-Order Filter Realizations **11**-6
11.4 Second-Order Positive-Gain Filters **11**-7
11.5 Second-Order Biquadratic Filters **11**-11
Biquadratic Transfer Function • Biquad Implementations
11.6 Higher Order Filters **11**-16
11.7 Influence of Nonidealities **11**-22
Sensitivity Analysis • Gain-Bandwidth • Slew Rate • Noise
11.8 Summary **11**-32
References **11**-32

Phillip E. Allen
Georgia Institute of Technology

Benjamin J. Blalock
The University of Tennessee

Stephen W. Milam
RF Micro-Devices

11.1 Introduction

Active filters consist of only amplifiers, resistors, and capacitors. Complex roots are achieved by the use of feedback eliminating the need for inductors. The gain of the amplifier can be finite or infinite (an op-amp). This section describes active filters using low-gain or finite-gain amplifiers. Filter design equations and examples will be given along with the performance limits of low-gain amplifier filters.

11.2 First- and Second-Order Transfer Functions

Before discussing the characteristics and the synthesis of filters, it is important to understand the transfer functions of first- and second-order filters. Later we will explain the implementations of these filters and show how to construct higher order filters from first- and second-order sections. The transfer functions of most first- and second-order filters are examined in the following.

11.2.1 First-Order Transfer Functions

A standard form of the transfer function of a first-order low-pass filter is

$$T_{LP}(s) = \frac{T_{LP}(j0)\omega_o}{s + \omega_o} \tag{11.1}$$

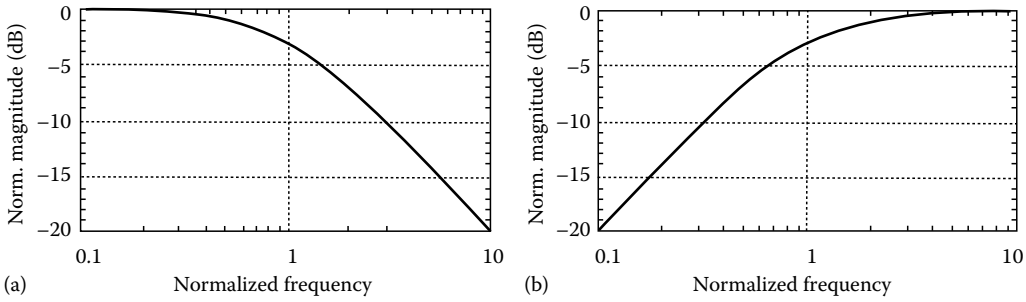


FIGURE 11.1 Normalized magnitude response of (a) first-order low-pass and (b) first-order high-pass filters.

where

$T_{LP}(j0)$ is the value of $T_{LP}(s)$ at dc
ω_o is the pole frequency

It is common practice to normalize both the magnitude and frequency. Normalizing Equation 11.1 yields

$$T_{LPn}(s_n) = \frac{T_{LP}(s_n\omega_o)}{|T_{LP}(j0)|} = \frac{1}{s_n + 1} \tag{11.2}$$

where $s_n = s/\omega_o$ and amplitude has been normalized as

$$T_{LPn}(s) = \frac{T_{LP}(s)}{|T_{LP}(j0)|} \tag{11.3}$$

The equivalent normalized forms of a first-order high-pass filter are

$$T_{HP}(s) = \frac{T_{HP}(j\infty)s}{s + \omega_o} \tag{11.4}$$

and

$$T_{HPn}(s_n) = \frac{T_{HP}(s_n\omega_o)}{|T_{HP}(j\infty)|} = \frac{s_n}{s_n + 1} \tag{11.5}$$

where $T_{HP}(j\infty) = T_{HP}(s)\,|$ at $\omega = \infty$. The normalized magnitude responses of these functions are shown in Figure 11.1.

11.2.2 Second-Order Transfer Functions

The standard form of a second-order low-pass filter is given as

$$T_{LP}(s) = \frac{T_{LP}(j0)\omega_o^2}{s^2 + \left(\frac{\omega_o}{Q}\right)s + \omega_o^2} \tag{11.6}$$

where

$T_{LP}(j0)$ is the value of $T_{LP}(s)$ at dc
ω_o is the pole frequency
Q is the pole Q or the pole quality factor

The damping factor ζ, which may be better known to the reader, is given as

$$\zeta = \frac{1}{2Q} \tag{11.7}$$

The poles of the transfer function of Equation 11.7 are

$$p_1, p_2 = \frac{-\omega_o}{2Q} \pm j\left(\frac{\omega_o}{2Q}\right)\sqrt{4Q^2 - 1} \tag{11.8}$$

Normalization of Equation 11.6 in both amplitude and frequency gives

$$T_{LPn}(s_n) = \frac{T_{LP}(s_n\omega_o)}{|T_{LP}(j0)|} = \frac{1}{s_n^2 + \frac{s_n}{Q} + 1} \tag{11.9}$$

where $s_n = s/\omega_o$. The standard second-order, high-pass and bandpass transfer functions are

$$T_{HP}(s) = \frac{T_{HP}(j\infty)s^2}{s^2 + \left(\frac{\omega_o}{Q}\right)s + \omega_o^2} \tag{11.10}$$

and

$$T_{BP}(s) = \frac{T_{BP}(j\omega_o)\left(\frac{\omega_o}{Q}\right)s}{s^2 + \left(\frac{\omega_o}{Q}\right)s + \omega_o^2} \tag{11.11}$$

where $T_{BP}(j\omega_o) = T_{BP}(s)$ at $s = j\omega = j\omega_o$. The poles of the second-order high-pass and bandpass transfer functions are given by Equation 11.8.

We can normalize these equations as we did for $T_{LP}(s)$ to get

$$T_{HPn}(s_n) = \frac{T_{HP}(s_n\omega_o)}{|T_{HP}(j\infty)|} = \frac{s_n^2}{s_n^2 + \frac{s_n}{Q} + 1} \tag{11.12}$$

$$T_{BPn}(s_n) = \frac{T_{BP}(s_n\omega_o)}{|T_{BP}(j\omega_o)|} = \frac{\frac{s_n}{Q}}{s_n^2 + \frac{s_n}{Q} + 1} \tag{11.13}$$

where

$$T_{HPn}(s) = \frac{T_{HP}(s)}{|T_{HP}(j\infty)|} \tag{11.14}$$

and

$$T_{BPn}(s) = \frac{T_{BP}(s)}{|T_{BP}(j\omega_o)|} \tag{11.15}$$

Two other types of second-order transfer function filters that we have not covered here are the bandstop and the all-pass. These transfer functions have the same poles as the previous ones. However, the zeros of the bandstop transfer function are on the $j\omega$ axis while the zeros of the all-pass transfer function are quadratically symmetric to the poles (they are mirror images of the poles in the right-half plane). Both of these transfer functions can be implemented by a second-order biquadratic transfer function whose transfer function is given as

$$T_{BQ}(s) = \frac{K\left[s^2 \pm \left(\frac{\omega_z}{Q_z}\right)s + \omega_z^2\right]}{s^2 + \left(\frac{\omega_p}{Q_p}\right)s + \omega_p^2} \tag{11.16}$$

where

K is a constant
ω_z is the zero frequency
Q_z the zero Q
ω_p is the pole frequency
Q_p the pole Q

11.2.3 Frequency Response (Magnitude and Phase)

The magnitude and phase response of the normalized second-order low-pass transfer function is shown in Figure 11.2, where Q is a parameter. In this figure we see that Q influences the frequency response near ω_o. If Q is greater than 0.707, then the normalized magnitude response has a peak value of

$$|T_n[\omega_n(\max)]| = \frac{Q}{\sqrt{1-(1/4Q^2)}} \tag{11.17}$$

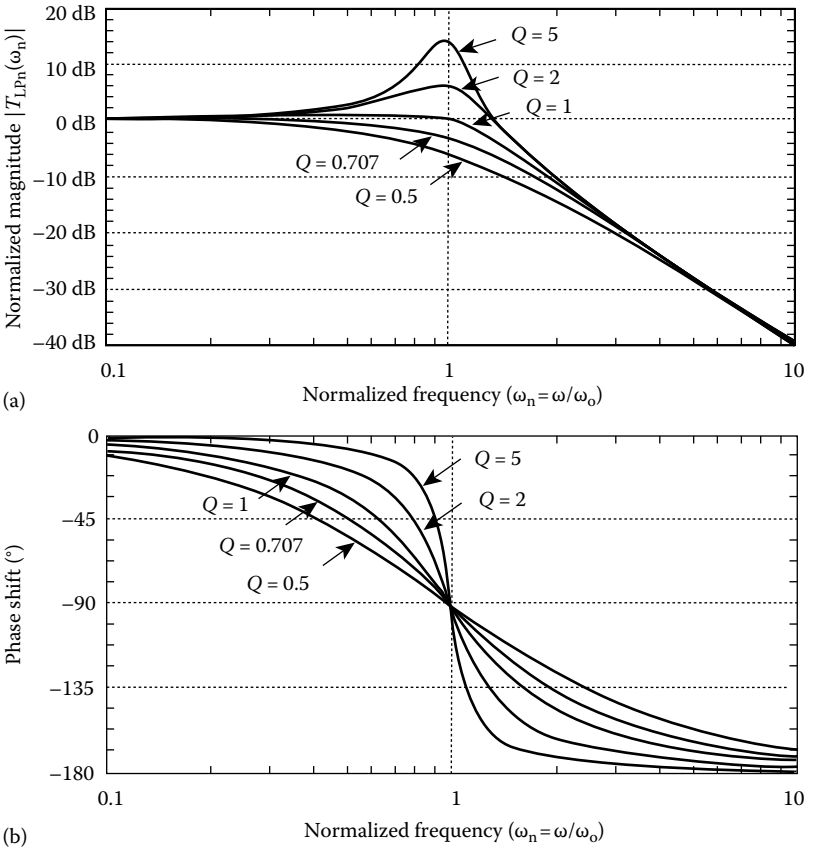


FIGURE 11.2 (a) Normalized magnitude and (b) phase response of the standard second-order low-pass transfer function with Q as a parameter.

at a frequency of

$$\omega_n(\max) = \sqrt{1 - \frac{1}{2Q^2}} \tag{11.18}$$

If the transfer function is multiplied by -1, the phase shift is shifted vertically by $\pm 180°$.

The magnitude and phase response of the normalized second-order high-pass transfer function is shown in Figure 11.3. For Q greater than 0.707 the peak value of the normalized magnitude response is as

$$\omega_n(\max) = \frac{1}{\sqrt{1 - \frac{1}{2Q^2}}} \tag{11.19}$$

The normalized frequency response of the standard second-order bandpass transfer function is shown in Figure 11.4. The slopes of the normalized magnitude curves at frequencies much greater or much less than ω_o are ± 20 dB/decade rather than ± 40 dB/decade of the second-order high and low-pass transfer functions. This difference is because one pole is causing the high-frequency roll-off while the other pole is causing the low-frequency roll-off. The peak of the magnitude occurs at $\omega = \omega_o$ or $\omega_n = 1$.

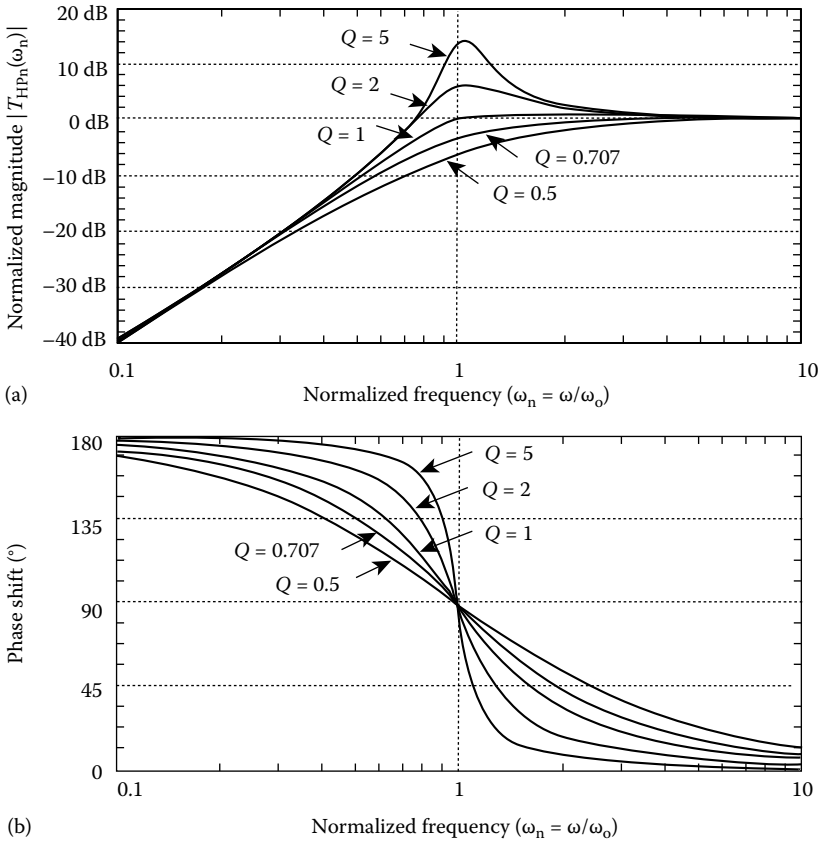


FIGURE 11.3 (a) Normalized magnitude and (b) phase response of the standard second-order high-pass transfer function with Q as a parameter.

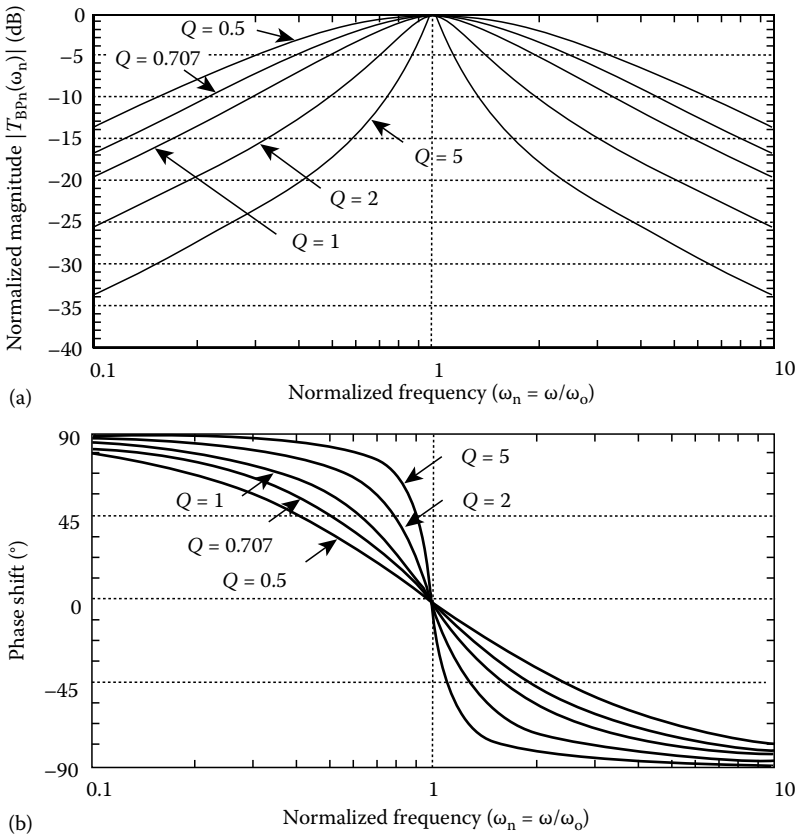


FIGURE 11.4 (a) Normalized magnitude and (b) phase response of the standard second-order bandpass transfer function with Q as a parameter.

11.2.4 Tuning Active Filters

A general tuning procedure for most second-order active filters is outlined below. This method is illustrated for adjusting the magnitude of the frequency response of a low-pass filter. The filter parameters are assumed to be the pole frequency f_o, the pole Q, and the gain $T(j0)$.

1. The component(s) which set(s) the parameter f_o is (are) tuned by adjusting the magnitude of the filter response to be $T(j0)/10$ or $T(j0)$ (dB) -20 dB at $10f_o$.
2. The component(s) that set(s) the parameter $T(j0)$ is (are) tuned by adjusting the magnitude to $T(j0)$ at $f_o/10$.
3. The component(s) that set(s) the parameter Q is (are) tuned by adjusting the magnitude of the peak (if there is one) to the value given by Figure 11.2. If there is no peaking, then adjust so that the magnitude at f_o is correct (i.e., -3 dB for $Q = 0.707$).

The tuning procedure should follow in the order of steps 1 through 3 and may be repeated if necessary. One could also use the phase shift to help in the tuning of the filter. The concept of the above tuning procedure is easily adaptable to other types of second-order filters.

11.3 First-Order Filter Realizations

A first-order filter has only one pole and zero. For stability, the pole must be on the negative real axis but the zero may be on the negative or positive real axis. A single-amplifier low-gain realization of a first-order low-pass filter is shown in Figure 11.5a. The transfer function for this filter is

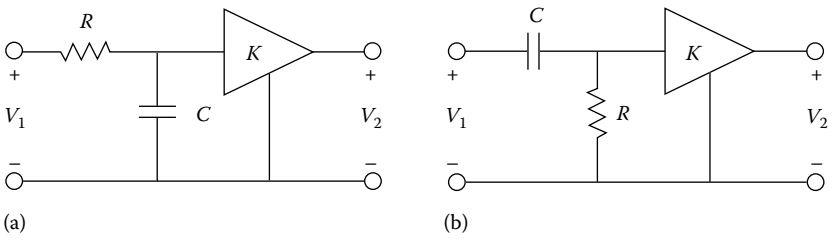


FIGURE 11.5 (a) Low-pass and (b) high-pass first-order filters.

$$T(s) = \frac{V_2(s)}{V_1(s)} = \frac{K/RC}{s + 1/RC} = \frac{K\omega_o}{s + \omega_o} = \frac{T_{LP}(0)\omega_o}{s + \omega_o} \tag{11.20}$$

The low-frequency ($\omega \ll \omega_o$) gain magnitude and polarity are set by K.

First-order high-pass filters can also be realized in a single-amplifier low-gain circuit (see Figure 11.5b). The transfer function for this filter is given in Equation 11.4, and in this case $T_{HP}(j\infty) = K$. Both low-pass and high-pass magnitude responses are shown in Figure 11.1. Note that the first-order filters exhibit no peaking and the frequency-dependent part of the magnitude response approaches ± 20 dB/decade.

11.4 Second-Order Positive-Gain Filters

Practical realizations for second-order filters using positive gain amplifiers are presented here. These filters are easy to design and have been extensively used in many applications. The first realization is a low-pass Sallen and Key filter and is shown in Figure 11.6 [8].

The transfer function of Figure 11.6 is

$$\frac{V_2(s)}{V_1(s)} = \frac{\frac{K}{R_1R_3C_2C_4}}{s^2 + s\left(\frac{1}{R_3C_4} + \frac{1}{R_1C_2} + \frac{1}{R_3C_2} - \frac{K}{R_3C_4}\right) + \frac{1}{R_1R_3C_2C_4}} \tag{11.21}$$

Equating this transfer function with the standard form of a second-order low-pass filter given in Equation 11.6 gives

$$\omega_o = \frac{1}{\sqrt{R_1R_3C_2C_4}} \tag{11.22}$$

$$\frac{1}{Q} = \sqrt{\frac{R_3C_4}{R_1C_2}} + \sqrt{\frac{R_1C_4}{R_3C_2}} + (1 - K)\sqrt{\frac{R_1C_2}{R_3C_4}} \tag{11.23}$$

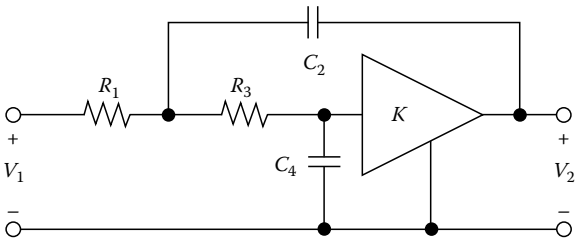


FIGURE 11.6 Low-pass Sallen and Key filter.

and

$$T_{LP}(j0) = K \tag{11.24}$$

These three equations have five unknowns, giving the designer some freedom in selecting component values. Two examples showing different techniques for defining these values of the circuit in Figure 11.6 are given below.

Example 11.1

An equal-resistance equal-capacitance low-pass filter. For this example, $R = R_1 = R_3$ and $C = C_2 = C_4$. A Butterworth low-pass filter characteristic is needed with $\omega_o = 6283$ rad/s (1 kHz) and $Q = 0.7071$. With these constraints,

$$\omega_o = \frac{1}{RC} \quad \frac{1}{Q} = 3 - K \tag{11.25}$$

and $RC = 159$ μs, $T_{LP}(j0) = K = 1.586$. Selecting $C = 0.1$ μF yields $R = 1.59$ kΩ.

Example 11.2

A unity-gain low-pass filter. For this example let $K = 1$. Therefore, Equation 11.23 becomes

$$\frac{1}{Q} = \sqrt{\frac{R_3 C_4}{R_1 C_2}} + \sqrt{\frac{R_1 C_4}{R_3 C_2}} = \sqrt{\frac{C_4}{C_2}}\left(\sqrt{\frac{R_3}{R_1}} + \sqrt{\frac{R_1}{R_3}}\right) \tag{11.26}$$

The desired transfer function, with the complex frequency normalized by a factor of 10^4 rad/s, is

$$\frac{V_2(s)}{V_1(s)} = \frac{0.988}{s^2 + 0.179s + 0.988} = \frac{T_{LP}(j0)\omega_o^2}{s^2 + \frac{\omega_o}{Q}s + \omega_o^2} \tag{11.27}$$

and $\omega_o = 0.994$ rad/s, $Q = 5.553$. To obtain a real-valued resistor ratio, pick

$$\frac{C_4}{C_2} \le \frac{1}{4Q^2} = 0.00811 \tag{11.28}$$

or $C_4/C_2 = 0.001$ in this case. Equation 11.26 yields two solutions for the ratio R_3/R_1, 30.3977 and 0.0329. From Equation 11.22 with $\omega_o = 9.94$ krad/s, $R_1C_2 = 577$ μs. If C_2 is selected as 0.1 μF, then $C_4 = 100$ pF, $R_1 = 5.77$ kΩ, and $R_3 = 175.4$ kΩ.

A Sallen and Key bandpass circuit is shown in Figure 11.7, and its voltage transfer function is

$$\frac{V_2(s)}{V_1(s)} = \frac{\frac{sK}{R_1C_5}}{s^2 + s\left(\frac{1}{R_1C_5} + \frac{1}{R_2C_5} + \frac{1}{R_4C_5} + \frac{1}{R_4C_3} - \frac{K}{R_2C_5}\right) + \frac{1}{R_4C_3C_5}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)} \tag{11.29}$$

Equating this transfer function to that of Equation 11.11 results in

$$\omega_o = \sqrt{\frac{1 + \frac{R_1}{R_2}}{R_1R_4C_3C_5}} \tag{11.30}$$

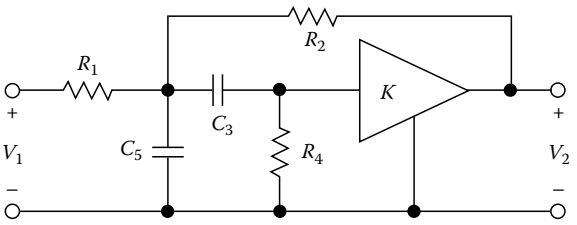


FIGURE 11.7 Bandpass Sallen and Key filter.

$$\frac{1}{Q} = \frac{\left[1 + \frac{R_1}{R_2}(1-K)\right]\sqrt{\frac{R_4 C_3}{R_1 C_5}} + \sqrt{\frac{R_1 C_3}{R_4 C_5}} + \sqrt{\frac{R_1 C_5}{R_4 C_3}}}{\sqrt{1 + \sqrt{\frac{R_1}{R_2}}}} \tag{11.31}$$

and

$$T_{BP}(j0) = \frac{\frac{K}{R_1 C_5}}{\frac{1}{R_1 C_5} + \frac{1}{R_2 C_5} + \frac{1}{R_4 C_5} + \frac{1}{R_4 C_3} + \frac{K}{R_2 C_5}} \tag{11.32}$$

These three equations contain six unknowns. A common constraint is to set $K = 2$, requiring the gain block to have two equal-valued resistors connected around an op-amp.

Example 11.3

An equal-capacitance gain-of-2 bandpass filter. Design a bandpass filter with $\omega_n = 1$ and $Q = 2$. Arbitrarily select $C_3 = C_5 = 1$ F and $K = 2$. If R_1 is selected to be 1 Ω, then Equations 11.11, 11.29, and 11.30 give

$$\omega_o^2 = \left(1 + \frac{R_1}{R_2}\right)\frac{1}{R_4} = 1 \tag{11.33}$$

and

$$\frac{\omega_o}{Q} = 1 - \frac{1}{R_2} + \frac{2}{R_4} = \frac{1}{2} \tag{11.34}$$

Solving these equations gives $R_2 = 0.7403$ Ω and $R_4 = 2.3508$ Ω. Practical values are achieved after frequency and impedance denormalizations are made.

The high-pass filter, the third type of filter to be discussed, is shown in Figure 11.8. Its transfer function is

$$\frac{V_2(s)}{V_1(s)} = \frac{s^2 K}{s^2 + s\left(\frac{1}{R_2 C_1} + \frac{1}{R_4 C_3} + \frac{1}{R_4 C_1} - \frac{K}{R_2 C_1}\right) + \frac{1}{R_2 R_4 C_1 C_3}} \tag{11.35}$$

Equating Equation 11.35 to Equation 11.10 results in

$$\omega_o = \frac{1}{\sqrt{R_2 R_4 C_1 C_3}} \tag{11.36}$$

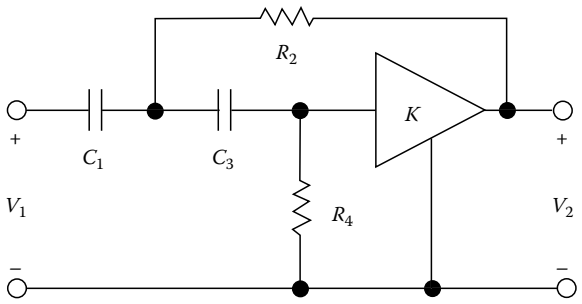


FIGURE 11.8 High-pass Sallen and Key filter.

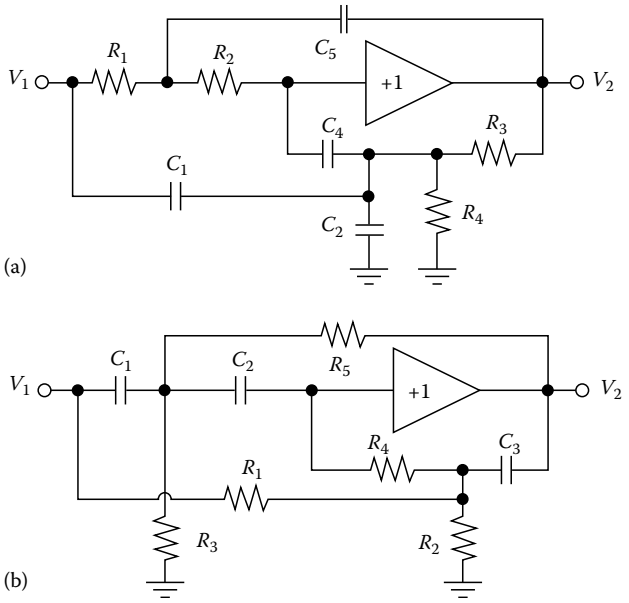


FIGURE 11.9 Notch filters derived from center-loaded twin-*T* networks: (a) $\omega_z \leq \omega_p$ and (b) $\omega_z < \omega_p$. (From Sedra, A. S. and Brackett, P. O., *Filter Theory and Design: Active and Passive*, Matrix, Portland, OR, 1978. With permission.)

$$\frac{1}{Q} = \sqrt{\frac{R_4 C_3}{R_2 C_1}} + \sqrt{\frac{R_2 C_1}{R_4 C_3}} + \sqrt{\frac{R_2 C_3}{R_4 C_1}} - K\sqrt{\frac{R_4 C_3}{R_2 C_1}} \tag{11.37}$$

and

$$T_{HP}(j\infty) = K \tag{11.38}$$

The design procedure using these equations is similar to that for the low-pass and bandpass filters.

The last type of second-order filter in this section is the notch filter, shown in Figure 11.9. The general transfer function for the notch filter is

$$\frac{V_2(s)}{V_1(s)} = T_N(j0)\frac{s^2 + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2} \tag{11.39}$$

where

ω_p and ω_z are the pole and zero frequencies (ω_o), respectively
Q_p is the pole Q

For the circuit of Figure 11.9a,

$$T_N(j0) = \frac{\omega_z^2}{\omega_p^2}, \quad R_2C_4 = \frac{\sqrt{1+a}}{\omega_p} \tag{11.40}$$

and

$$R_2 = (1+a)R_1 = 2R_3 = aR_4, \quad C_4 = \frac{\omega_z^2}{\omega_p^2}C_1 = C_2 + C_1 = \frac{C_5}{2} \tag{11.41}$$

For the circuit of Figure 11.9b.

$$T_N(j0) = 1, \quad R_2C_4 = \frac{\sqrt{1+a}}{\omega_p} \tag{11.42}$$

and

$$R_4 = (1+a)\frac{\omega_z^2}{\omega_p^2}R_1 = (1+a)(R_1+R_2) = aR_3 = 2R_5, \quad C_1 = C_2 = \frac{C_3}{2} \tag{11.43}$$

The Q for both notch circuits is

$$Q = \frac{\sqrt{1+a}}{a} \tag{11.44}$$

so a is a helpful design parameter for these circuits. For typical values of Q, say 0.5, 1, and 5, the corresponding values of a are 4.828, 1.618, and 0.2210, respectively.

11.5 Second-Order Biquadratic Filters

Next, second-order biquadratic functions will be described. These functions are general realizations of the second-order transfer function. Filters implementing biquadratic functions, often referred to as biquads, are found in many signal processing applications.

11.5.1 Biquadratic Transfer Function

The general form of the second-order biquadratic transfer function is

$$T(s) = H\frac{s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0} = H\frac{s^2 + \frac{\omega_z}{Q_z}s + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2} \tag{11.45}$$

with the pole locations given by Equation 11.8 and the zero locations by

$$z_1, z_2 = -\frac{\omega_z}{2Q_z} \pm j\omega_z\sqrt{1 - \frac{1}{4Q_z^2}} \tag{11.46}$$

where Q_p and Q_z are the pole and zero Q, respectively. Filters capable of implementing this voltage transfer function are called biquads since both the numerator and denominator of their transfer functions contain biquadratic expressions. The zeros described by the numerator of Equation 11.45 strongly influence the magnitude response of the biquadratic transfer function and determine the filter type (low-pass, high-pass, etc.).

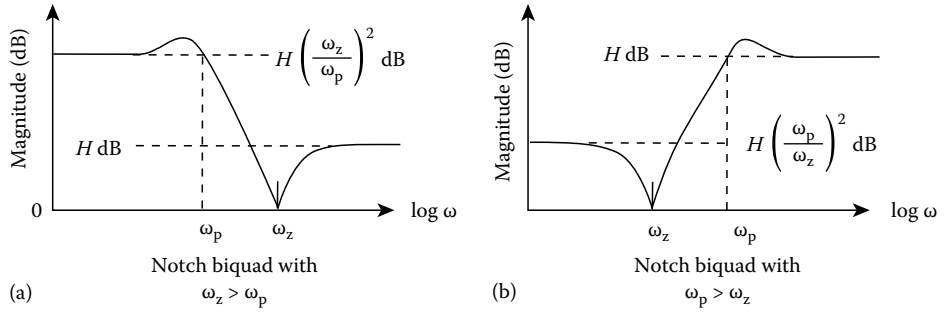


FIGURE 11.10 Frequency response of second-order notch filters. (a) $\omega_z > \omega_p$ and (b) $\omega_p > \omega_z$.

The notch filter form of the biquadratic transfer function is

$$T_{NF}(s) = H\frac{s^2 + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2} \tag{11.47}$$

with zeros located at $s = \pm j\omega_z$. Attenuation of high or low frequencies is determined by selection of ω_z relative to ω_p. The low-pass notch filter (LPN) requires $\omega_z > \omega_p$, shown in Figure 11.10a and the high-pass notch filter (HPN) requires $\omega_p > \omega_z$, shown in Figure 11.10b.

An all-pass filter implemented using the biquadratic transfer function has the general form

$$T_{AP}(s) = H\frac{s^2 - \frac{\omega_z}{Q_z}s + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2} \tag{11.48}$$

The all-pass magnitude response is independent of frequency; i.e., its magnitude is constant. The all-pass filter finds use in shaping the phase response of a system. To accomplish this, the all-pass has right-half plane zeros that are mirror images around the imaginary axis of left-half plane poles.

11.5.2 Biquad Implementations

A single-amplifier low-gain realization [4] of the biquadratic transfer function with transmission zeros on the $j\omega$ axis is shown in Figure 11.11. Zeros are generated by the circuit's input "twin-*T*" *RC* network. This filter is well suited for implementing elliptic functions. Its transfer function has the form

$$T(s) = H\frac{s^2 + \omega_z^2}{s^2 + \frac{\omega_p}{Q_p}s + \omega_p^2} = H\frac{s^2 + b_o^2}{s^2 + a_1 s + a_o^2} \tag{11.49}$$

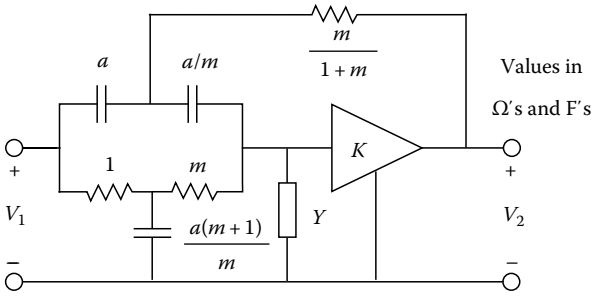


FIGURE 11.11 Low-gain realization for biquadratic network functions. (From Huelsman, L. P. and Allen, P. E., *Introduction to the Theory and Design of Active Filters*, McGraw-Hill, New York, 1980. With permission.)

The selection of element $Y(s)$ in the filter circuit can be a resistor or capacitor. If the condition $\omega_p > \omega_z$ is desired, $Y = 1/R$ is chosen. For this choice the resulting transfer function has the form

$$T(s) = \frac{V_2(s)}{V_1(s)} = \frac{K(s^2 + 1/a^2)}{s^2 + (m+1/a)[1/R + (2-K)/m]s + [1 + (m+1)/R]/a^2} \tag{11.50}$$

where, K, m, and a are defined in Figure 11.11. The design parameter m is used to control the spread of circuit element values. The design equations for this filter resulting from equating Equation 11.49 and 11.50 are

$$a = \frac{1}{\sqrt{b_o}} \tag{11.51}$$

$$R = \frac{m+1}{a_o/b_o - 1} \tag{11.52}$$

$$K = 2 + \frac{m}{m+1}\left(\frac{a_o}{b_o} - 1 - \frac{a_1}{\sqrt{b_o}}\right) \tag{11.53}$$

and

$$H = K \tag{11.54}$$

If the condition $\omega_z > \omega_p$ is desired, then $Y = s(aC)$ is chosen. This choice has the following transfer function:

$$\frac{V_2(s)}{V_1(s)} = \frac{\dfrac{K(s^2 + 1/a^2)}{(m+1)C + 1}}{s^2 + \left\{\dfrac{(m+1)[C + (2-K)/m]}{a[(m+1)C + 1]}\right\}s + \dfrac{1}{a^2[(m+1)C + 1]}} \tag{11.55}$$

The design equations follow from equating Equations 11.49 and 11.55 to get

$$a = \frac{1}{\sqrt{b_o}} \tag{11.56}$$

$$C = \frac{b_o/a_o - 1}{m+1} \tag{11.57}$$

$$K = 2 + \frac{m}{m+1}\left(\frac{b_o}{a_o} - 1 - \frac{a_1\sqrt{b_o}}{a_o}\right) \tag{11.58}$$

and

$$H = \frac{a_o}{b_o}K \tag{11.59}$$

As before, the factor m is chosen arbitrarily to control the spread of element values.

Example 11.4

A low-pass elliptic filter. It is desired to realize the elliptic voltage transfer function given as

$$\frac{V_2(s)}{V_1(s)} = \frac{H(s^2 + 2.235990)}{s^2 + 0.641131s + 1.235820}$$

This transfer function will have a 1 dB ripple in the passband, and at least 6 dB of attenuation for all frequencies greater than 1.2 rad/s [12]. Obviously, the second choice described above applies. From Equations 11.56 through 11.59, with $m = 0.2$, we find that $a = 0.66875$, $C = 0.67443$, $K = 2.0056$, and $H = 1.1085$.

Biquadratic transfer functions can also be implemented using the two-amplifier low-gain configuration in Figure 11.12 [5]. When zeros located off the $j\omega$ axis are desired, the design equations for the two-amplifier low-gain configuration are simpler than those required by the single-amplifier low-gain configuration. Also note that the required gain blocks of -1 and $+2$ are readily implemented using operational amplifiers. The transfer function for this configuration is

$$\frac{V_2(s)}{V_1(s)} = \frac{2(Y_1 - Y_2)}{Y_3 - Y_4} \tag{11.60}$$

The values of the admittances are determined by separately dividing the numerator and denominator by $s + c$, where c is a convenient value greater than zero. Partial fraction expansion results in expressions that can be implemented using *RC* networks. From the partial fraction expansion of the numerator divided by $s + c$, the pole residue at $s = -c$ is

$$k_b = \frac{H(c^2 - cb_1 + b_o)}{-c} \tag{11.61}$$

Depending on c, b_1, and b_o, the quality k_b can be positive or negative. If positive, Y_1 is

$$Y_1 = \frac{Hs}{2} + \frac{Hb_o}{2c} + \frac{1}{2/k_b + 2c/k_b s} \tag{11.62}$$

with $Y_2 = 0$, removing the inverting gain amplifier from the circuit. The *RC* network used to realize Y_1 when k_b is positive is shown in Figure 11.13a. If k_b is negative, Y_1 and Y_2 become

$$Y_1 = \frac{Hs}{2} + \frac{Hb_o}{2c} \quad \text{and} \quad Y_2 = \frac{1}{2/|k_b| + 2c/|k_b|s} \tag{11.63}$$

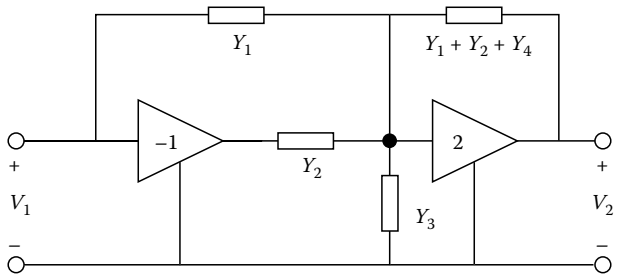


FIGURE 11.12 Two-amplifier low-gain biquad configuration.

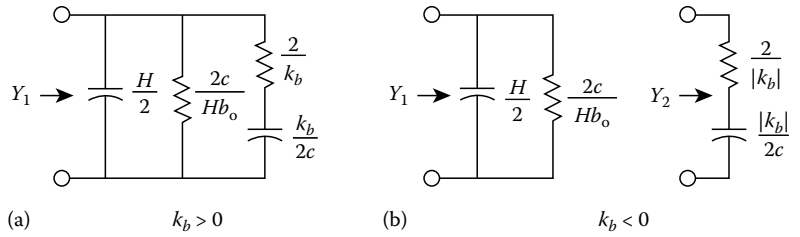


FIGURE 11.13 Realizations for (a) $Y_1(s)$ and (b) $Y_2(s)$ in Figure 11.12.

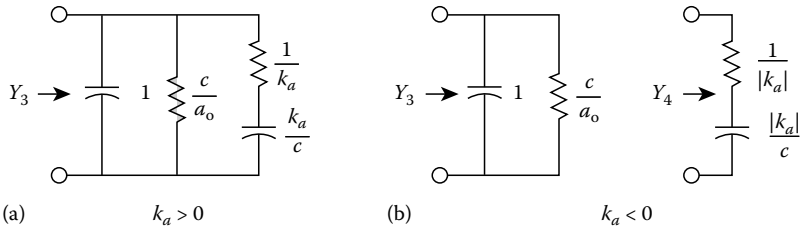


FIGURE 11.14 Realizations for (a) $Y_3(s)$ and (b) $Y_4(s)$ in Figure 11.12.

Realizations for Y_1 and Y_2 for this case are shown in Figure 11.13b. In determining Y_3 and Y_4, the partial fraction expansion of the denominator divided by $s+c$ yields a pole residue at $s=-c$,

$$k_a = \frac{c_2 - ca_1 + a_o}{-c} \tag{11.64}$$

If k_a is positive, then $Y_4 = 0$ and Y_3 is

$$Y_3 = s + \frac{a_o}{c} + \frac{1}{1/k_a + c/k_a s} \tag{11.65}$$

The realization of Y_3 is shown in Figure 11.14a. For a negative k_a, Y_3 and Y_4 become

$$Y_3 = s + \frac{a_o}{c} \tag{11.66}$$

and

$$Y_4 = \frac{1}{1/|k_a| + c/|k_a|s} \tag{11.67}$$

RC network realizations for Y_3 and Y_4 are shown in Figure 11.14b.

Example 11.5

An all-pass function [5]. It is desired to use the configuration of Figure 11.12 to realize the following all-pass function:

$$\frac{V_2(s)}{V_1(s)} = \frac{s^2 - 4s + 4}{s^2 + 4s + 4}$$

The choice of $c = +2$ will simplify the network. For this choice the numerator partial-fraction expansion is

$$\frac{s^2 = 4s + 4}{s + 2} = s + 2 - \frac{8s}{s + 2}$$

Hence $Y_1(s) = (s + 2)/2$ and $Y_2(s) = 4s/(s + 2)$. The denominator partial-fraction expansion is

$$\frac{s^2 + 4s + 4}{s + 2} = s + 2$$

Thus, $Y_3(s) = (s + 2) + [4s/(s + 2)]$ and $Y_4(s) = 0$.

11.6 Higher Order Filters

Many applications require filters of order greater than two. One way to realize these filters is simply to cascade second-order filters to implement the higher order filter. If the order is odd, then one first-order (or third-order) section will be required. The advantage of the cascade approach is that it builds on the precious techniques described in this section. The desired high-order transfer function of the filter $T(s)$ will be broken into second-order functions $T_k(s)$ so that

$$T(s) = T_1(s)T_2(s)T_3(s)\cdots T_{n/2}(s) \tag{11.68}$$

Since the output impedance of the second-order sections is low, the sections can be cascaded without significant interaction.

If $T(s)$ is an odd-order function then a first-order passive network (like those shown in Figure 11.15) can be added to the cascade of second-order functions. Both of the sections shown have nonzero output impedances, so it is advisable to use them only in the last stage of a cascade filter. Alternatively, Figure 11.5 could be used.

Several considerations should be taken into account when cascading second-order sections together. Dynamic range is one of these considerations. To maximize the dynamic range of the filter, the peak gain of each of the individual transfer functions should be equal to the maximum gain of the overall transfer function [10].

To maximize the signal-to-noise of the cascade filter, the magnitude curve in the passband of each of the individual transfer functions should be flat when possible. Otherwise, signals with frequencies in a minimum gain region of an individual transfer function will have a lower signal-to-noise ratio than other signals [10]. Another consideration is minimum noise which is achieved by placing high gain stages first.

Designing a higher order filter as a cascade of second-order sections allows the designer many options. For instance, a fourth-order bandpass filter could result from the cascade of two second-order bandpass sections or one second-order low-pass and a second-order high-pass. Since the bandpass section is the easiest of the three types to tune [10] it might be selected instead of the low-pass–high-pass combination.

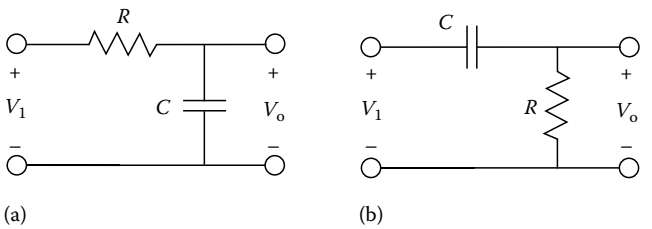


FIGURE 11.15 First-order filter sections: (a) low-pass and (b) high-pass.

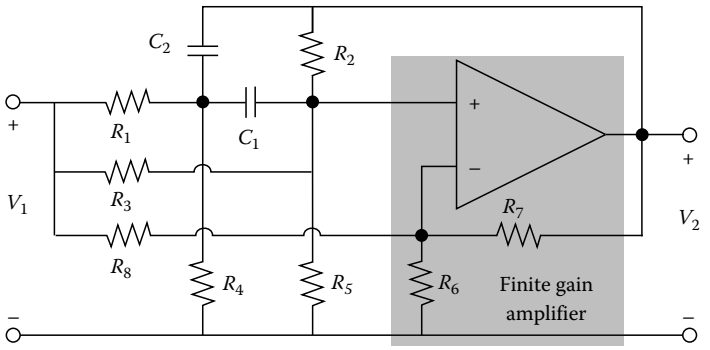


FIGURE 11.16 Biquad multiloop feedback stage.

Other guidelines for the cascade of second-order sections include putting the low-pass or bandpass sections at the input of the cascade in order to decrease the high-frequency content of the signal, which could avoid slewing in following sections. Using a bandpass or high-pass section at the output of the cascade can reduce dc offsets or low-frequency ripple.

One option that is now available is to use CAD tools in the design of filters. When high-order filters are needed, CAD tools can save the designer time. The examples that are worked later in this section require the use of filter tables interspersed with calculations. A CAD tool such as MicroSim Filter Designer [6] can hasten this process since it eliminates the need to refer to tables. The designer can also compare filters of different orders and different approximation methods (Butterworth, Chebyshev, etc.) to determine which one is the most practical for a specific application. The Filter Designer is able to design using the cascade method as discussed in this section, and after proposing a circuit it can generate a netlist so that a program such as SPICE can simulate the filter along with additional circuitry, if necessary.

The Filter Designer allows the user to specify a filter with either a passband/stopband description, center frequency/bandwidth description, or a combination of filter order and type (low-pass, high-pass, etc.). Then an approximation type is chosen, with Butterworth, Chebyshev, inverse Chebyshev, Bessel, and elliptic types available. A Bode plot, pole-zero plot, step or impulse response plot of the transfer function can be inspected. The designer can then select a specific circuit that will be used to realize the filter. The Sallen and Key low-pass and high-pass stages and the biquad multiloop feedback (Figure 11.16) stages are included. The *s*-coefficients of the filter can also be displayed.

The Filter Designer allows modifications to the specifications, *s*-coefficients, or actual components of the circuit in order to examine their effect on the transfer function. There is also a rounding function that will round components to 1%, 5%, or 10% values for resistors and 5% for capacitors and then recompute the transfer function based on these new component values. Finally, the package offers a resize function that will rescale components without changing the transfer function.

Example 11.6

A fifth-order low-pass filter. Design a low-pass filter using the Chebyshev approximation with a 1 dB equiripple passband, a 500 Hz passband, 45 dB attenuation at 1 kHz, and a gain of 10. Using a nomograph for Chebyshev magnitude functions, the order of this function is found to be five [5]. Using this information and a chart of quadratic factors of 1.0 dB equal-ripple magnitude low-pass functions, the transfer function is found to be

$$T(s_n) = \frac{0.9883K_1}{s_n^2 + 0.1789s_n + 0.9883} \cdot \frac{0.4293K_2}{s_n^2 + 0.4684s_n + 0.4293} \cdot \frac{0.2895K_3}{s_n + 0.2895}$$

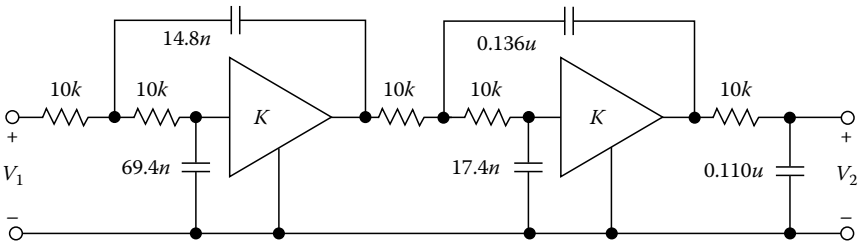


FIGURE 11.17 Fifth-order low-pass filter.

Denormalizing the frequency with $s_n = s/500(2\pi)$ gives

$$T_1(s) = \frac{9,751,000K_1}{(s^2 + 562s + 9,751,000)}$$
$$T_2(s) = \frac{4,240,000K_2}{(s^2 + 1470s + 4,240,000)}$$

and

$$T_3(s) = \frac{909K_3}{s + 909}$$

Stages 1 and 2 will be Sallen and Key low-pass filters with $R = 10$ kΩ. Following the guidelines given above, stage 1 will have the gain of 10 and successive stages will have unity gain. Using Equation 11.23 for stage 1 ($R_2 = R_4 = R$),

$$\frac{1}{Q} = 2\sqrt{\frac{C_4}{C_2}} + (1 - K)\sqrt{\frac{C_2}{C_4}} = 0.1800$$

So $C_4 = 4.696C_2$. Substituting this into Equation 11.22,

$$\omega_n = \frac{1}{2.167RC_2} = 3123\,\text{rad/s}$$

Thus, $C_2 = 14.8$ nF and $C_4 = 69.4$ μF. Analysis for stage 2 with $K_2 = 1$ yields $C_2 = 0.135$ μF and $C_4 = 17.4$ nF. Stage 3 is simply an *RC* low-pass section with $R = 10$ kΩ and $C = 0.110$ μF. The schematic for this filter is shown in Figure 11.17.

Example 11.7

A fourth-order bandpass filter. Design a bandpass filter centered at 4 kHz with a 1 kHz bandwidth. The time delay of the filter should be 0.5 ms (with less than 5% error at the center frequency).

The group of filters with constant time delays are called Thomson filters. For this design, begin with a low-pass filter with the normalized bandwidth of 1/4 kHz, or 0.25. Consulting a table for delay error in Thomson filter [5] shows that a fourth-order filter is needed. From tables for Thomson filters, the quadratic factors for the fourth-order low-pass filter give the overall transfer function

$$T_{LP}(s_n) = \frac{1}{(s_n^2 + 4.208s_n + 11.488)(s_n^2 + 5.792s_n + 9.140)}$$

To convert $T_{LP}(s_n)$ to a bandpass function $T_{BP}(s_n)$, the transformation

$$s_n = \frac{\omega_R}{BW}\frac{s_{bn}^2 + 1}{s_{bn}} = 4\frac{s_{bn}^2 + 1}{s_{bn}}$$

is used, giving

$$T_{BPn}(s) = \frac{s^4/256}{(s^4 + 1.05s^3 + 2.72s^2 + 1.05s + 1)(s^4 + 1.45s^3 + 2.57s^2 + 1.45s + 1)}$$

ω_R and BW are the geometric center frequency and bandwidth of the bandpass filter. The polynomials in the denominator of the expression are difficult to factor, so an alternate method can be used to find the poles of the normalized bandpass function. Each pole p_{In} of the normalized low-pass transfer function generates two poles in the normalized bandpass transfer function [5–10] given by

$$p_{bn} = \frac{BW\ p_{1n}}{2\omega_R} \pm \sqrt{\left(\frac{BW\ p_{1n}}{2\omega_R}\right)^2 - 1} \tag{11.69}$$

Using Equation 11.69 with each of the poles of $T_{LP}(s_n)$ gives the following pairs of poles (which are also given in tables for Thomson filters):

$$p_{b1n} = -0.1777 \pm j0.6919 \quad p_{b2n} = -0.3483 \pm j1.356$$
$$p_{b3n} = -0.3202 \pm j0.8310 \quad p_{b4n} = -0.4038 \pm j1.0478$$

Note that each low-pass pole does not generate a conjugate pair, but when taken together, the bandpass poles generated by the low-pass poles will be conjugates. The unnormalized transfer functions generated by the normalized bandpass poles are

$$T_{BP1}(s) = \frac{6283s}{(s^2 + 8932s + 322.3 \times 10^6)}$$
$$T_{BP2}(s) = \frac{6283s}{(s^2 + 1751s + 1238 \times 10^6)}$$
$$T_{BP3}(s) = \frac{6283s}{(s^2 + 16100s + 501.0 \times 10^6)}$$

and

$$T_{BP4}(s) = \frac{6283s}{(s^2 + 20300s + 796.5 \times 10^6)}$$

The overall transfer function realized by these four intermediate transfer functions could also be grouped into two low-pass stages followed by two high-pass stages, however, this example will use four bandpass filters. Using the equal-capacitance gain of two-design strategy for the Sallen and Key bandpass stage,

$$\omega_n^2 = \left(1 + \frac{R_1}{R_2}\right)\frac{1}{R_1 R_4 C^2} \tag{11.70}$$

$$\frac{1}{Q} = \left[\left(1 - \frac{R_1}{R_2}\right)\sqrt{\frac{R_4}{R_1}} + 2\sqrt{\frac{R_1}{R_4}}\right]\sqrt{\frac{R_2}{R_1 + R_2}} \tag{11.71}$$

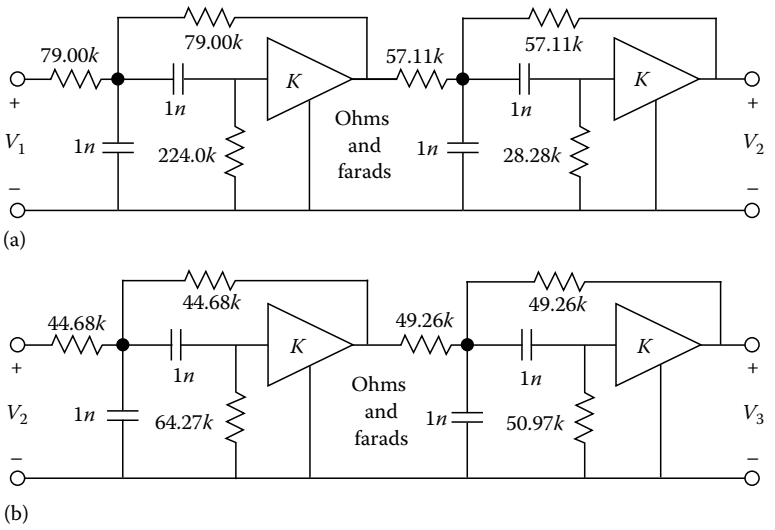


FIGURE 11.18 Fourth-order bandpass filter. (a) Stages 1 and 2 and (b) stages 3 and 4.

and

$$H_o = \left(\frac{\dfrac{K}{R_1}}{\dfrac{1}{R_1} + \dfrac{1}{R_2} + \dfrac{2}{R_4} - \dfrac{K}{R_2}} \right) \tag{11.72}$$

Making the further assignment $R_2 = R_1$ simplifies the equations to

$$\omega_n^2 = \frac{2}{R_1 R_4 C^2}, \quad \frac{1}{Q} = 2\sqrt{\frac{R_1}{R_4}}, \quad \text{and} \quad H_o = 2Q^2$$

Selecting $C = 1$ nF gives the component values shown in Figure 11.18, the schematic diagram for the fourth-order bandpass filter. While the gain of the filter is greater than unity, the output can be attenuated if unity gain is desired.

Example 11.8

A sixth-order high-pass filter. Design a Butterworth high-pass filter with a −3 dB frequency of 100 kHz and a stopband attenuation of 30 dB at 55 kHz. A nomograph for Butterworth (maximally flat) functions indicates that this filter must have an order of at least six. A table for quadratic factors of Butterworth functions [5] gives

$$T(s_n) = \frac{K_1}{\left(s_n^2 + 0.518 s_n + 1\right)} \quad \frac{K_2}{\left(s_n^2 + 1.414 s_n + 1\right)} \quad \frac{K_3}{\left(s_n^2 + 1.932 s_n + 1\right)}$$

Using $s_{nH} = 1/s_n$ for low-pass to high-pass transformation and $s_n = s/2\pi \times 10^5$ to denormalize the frequency,

$$T_1(s) = \frac{s^2 K_1}{s^2 + 325,500 s + 394.8 \times 10^9}$$

$$T_2(s) = \frac{s^2 K_2}{s^2 + 888,600 s + 394.8 \times s10^9}$$

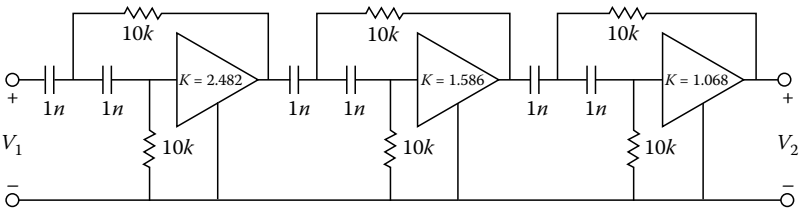


FIGURE 11.19 A sixth-order high-pass filter (all component values are Ω or F).

and

$$T_3(s) = \frac{s^2 K_3}{s^2 + 1,214,000s + 394.8 \times 10^9}$$

The Sallen and Key high-pass filter can be used to implement all three of these stages. Picking a constant R, constant C design style with $\omega_n = 10^5$ rad/s gives $R = 10$ kΩ and $C = 1$ nF for each stage. Equation 11.23 then simplifies to

$$\frac{1}{Q} = 3 - K$$

for each stage. Thus $K_1 = 2.482$, $K_2 = 1.586$, and $K_3 = 1.068$, and the resulting high-pass filter is shown in Figure 11.19.

Example 11.9

A fifth-order low-pass filter. Design a low-pass filter with an elliptic characteristic such that it has a 1 dB ripple in the passband (which is 500 Hz) and 65 dB attenuation at 1 kHz. From a nomograph for elliptic magnitude functions, the required order of this filter is five. A chart for a fifth-order elliptic filter with 1 dB passband ripple gives

$$T(s_n) = \frac{H\left(s_n^2 + 4.365\right)\left(s_n^2 + 10.568\right)}{\left(s_n + 0.3126\right)\left(s_n^2 + 0.4647s_n + 0.4719\right)\left(s_n^2 + 0.1552s_n + 0.9919\right)}$$

This transfer function can be realized with two notch filters like Figure 11.9a and a first-order low-pass section. Denormalizing the frequency with $s_n = s/1000\pi$ gives

$$T_1(s) = \frac{H_1(s^2 + 43,080,000)}{s^2 + 1,460s + 4,657,000}$$

$$T_2(s) = \frac{H_2(s^2 + 104,300,000)}{s^2 + 487.6s + 9,790,000}$$

and

$$T_3(s) = \frac{982.1}{s + 982.1}$$

For the first stage, use Equations 11.40 and 11.44, $H_1 = 0.1081$, and $a = 0.9429$. Setting $C_1 = 1$ nF and applying Equations 11.40 and 11.41,

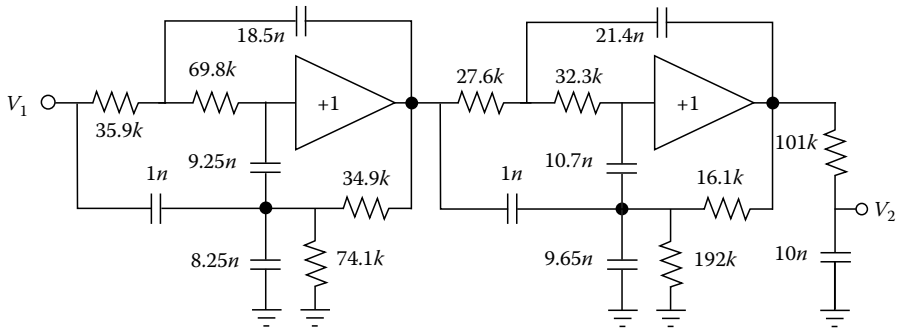


FIGURE 11.20 Fifth-order low-pass filter (all component values are Ω or F).

$$C_1 = 1\,\text{nF}, \quad C_4 = 9.25\,\text{nF}, \quad C_2 = 8.25\,\text{nF}, \quad C_5 = 18.5\,\text{nF}$$
$$R_2 = 69.8\,\text{k}\Omega, \quad R_1 = 35.9\,\text{k}\Omega, \quad R_3 = 34.9\,\text{k}\Omega, \quad R_1 = 74.1\,\text{k}\Omega$$

Similarly, in the second stage $H_2 = 0.0939$, $a = 0.1684$, and

$$C_1 = 1\,\text{nF}, \quad C_4 = 10.7\,\text{nF}, \quad C_2 = 9.65\,\text{nF}, \quad C_5 = 21.4\,\text{nF}$$
$$R_2 = 32.3\,\text{k}\Omega, \quad R_1 = 27.6\,\text{k}\Omega, \quad R_3 = 16.1\,\text{k}\Omega, \quad R_4 = 192\,\text{k}\Omega$$

The third stage can be a simple *RC* low-pass with $C = 10$ nF and $R = 101$ kΩ. The resulting cascade filter is shown in Figure 11.20.

These examples illustrate both the use of the Filter Designer and the method of cascading first- and second-order stages to achieve higher order filters.

11.7 Influence of Nonidealities

Effective filter design requires an understanding of how circuit nonidealities influence performance. Important nonidealities to consider include passive component tolerance (i.e., resistor and capacitor accuracy), amplifier gain accuracy, finite amplifier gain-bandwidth, amplifier slew rate, and noise.

11.7.1 Sensitivity Analysis

Classical sensitivity functions [5] are valuable tools for analyzing the influence of nonidealities on filter performance. The sensitivity function, s_x^Y [2], by definition, describes the change in a performance characteristic of interest, say y, due to the change in nominal value of some element x,

$$S_x^y = \left(\frac{\partial y}{\partial x}\right)\frac{x}{y} = \frac{\partial y/y}{\partial y/x} = \frac{\partial(l_n\, y)}{\partial(l_n\, x)} \tag{11.73}$$

The mathematical properties of the sensitivity function are given in Table 11.1 [5] for convenient reference.

A valuable result of the sensitivity function is that it enables us to estimate the percentage change in a performance characteristic due to variation in passive circuit elements from their nominal values. Consider, for example, a low-pass filter's cutoff frequency ω_o, which is a function of R_1, R_2, C_1, and C_2. Using sensitivity functions, the percentage change in ω_o is estimated as [3]

$$\frac{\Delta\omega_o}{\omega_o} \approx S_{R_1}^{\omega_o}\left(\frac{\Delta R_1}{R_1}\right) + S_{R_2}^{\omega_o}\left(\frac{\Delta R_2}{R_2}\right) + S_{C_1}^{\omega_o}\left(\frac{\Delta C_1}{C_1}\right) + S_{C_2}^{\omega_o}\left(\frac{\Delta C_2}{C_2}\right) \tag{11.74}$$

TABLE 11.1 Properties of Sensitivity Function

$S_x^{ky} = S_{kx}^y = S_x^y$	$S_x^{y_1/y_2} = S_x^{y_1} - S_x^{y_2}$
$S_x^x = S_x^{kx} = S_{kx}^{kx} = 1$	$S_{x_1}^y = S_{x_2}^y S_{x_1}^{y_2}$
$S_{1/x}^y = S_x^{1/y} = -S_x^y$	$S_x^y = S_x^{\lvert y\rvert} + j \arg y S_x^{\arg y}$
$S_x^{y_1 y_2} = S_x^{y_1} + S_x^{y_2}$	$S_x^{\arg y} = \frac{1}{\arg y} \operatorname{Im} S_x^y(\star)$
$S_x^{\prod_{i=1}^n y_i} = \sum_{i=1}^n S_x^{y_i}$	$S_x^{\lvert y\rvert} = \operatorname{Re} S_x^y(\star)$
$S_x^{y^n} = n S_x^y$	$S_x^{y+x} = \frac{1}{y+z}\left(y S_x^y + z S_x^z\right)$
$S_x^{x^n} = S_x^{kx^n} = n$	$S_x^{\sum_{i=1}^n y_i} = \frac{\sum_{i=1}^n y_i S_x^{y_i}}{\sum_{i=1}^n y_i}$
$S_{x^n}^y = \frac{1}{n} S_x^y$	$S_x^{\ln y} = \frac{1}{\ln y} S_x^y$
$S_{x^n}^x = S_{kx^n}^x = \frac{1}{n}$	

Source: Huelsman, L.P. and Allen, P.E., *Introduction to the Theory and Design of Active Filters*, McGraw-Hill, New York, 1980.

Note: Relations denoted by "(⋆)" use y to indicate a complex quantity and x to indicate a real quantity.

Not that the quantity $\Delta\omega_o/\omega_o$ represents a small differential change in the cutoff frequency ω_o. When considering the root locus of a filter design, the change in pole location due to a change in gain K, for example, could be described by [3]

$$S_K^{p_1} = \left(\frac{\partial \sigma_o}{\partial K}\right) + j\frac{\partial \omega_o}{\partial K}\left(\frac{K}{\omega_o}\right) \tag{11.75}$$

where the pole $p_1 = \sigma_o + j\omega_o$. Furthermore, the filter transfer function's magnitude and phase sensitivity is defined by [3]

$$S_x^{T(j\omega)} = S_x^{|T(j\omega)|} + S_x^{\theta(j\omega)} = \operatorname{Re} S_x^{T(j\omega)} + \frac{1}{\theta(\omega)} \operatorname{Im} S_x^{T(j\omega)} \tag{11.76}$$

and

$$S_x^{T(j\omega)} = \frac{x}{|T(j\omega)|}\frac{\partial}{\partial x}|T(j\omega)| + jx\frac{\partial \theta(\omega)}{\partial x} \tag{11.77}$$

where $T(j\omega) = |T(j\omega)|\exp[j\theta(\omega)]$. The interested reader is directed to Ref. [5, Chapter 3] for a detailed discussion of sensitivity.

11.7.2 Gain-Bandwidth

Now let us consider some of the amplifier nonidealities that influence the performance of our low-gain amplifier filter realizations. There are two amplifier implementations of interest each using an operational amplifier. They are the noninverting configuration and the inverting configuration shown in Figure 11.21.

The noninverting amplifier is described by

$$K = \frac{V_2}{V_1} = \frac{A_d(s)}{1 + A_d(s)/K_o} \tag{11.78}$$

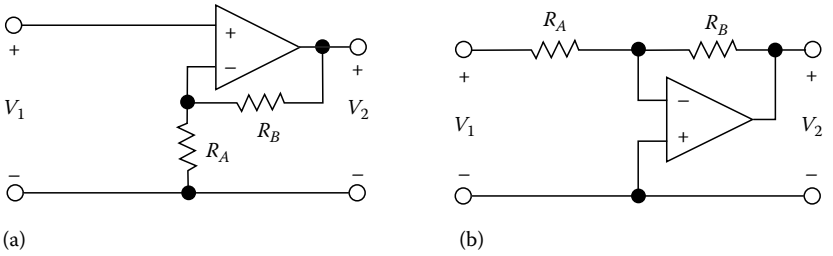


FIGURE 11.21 (a) Noninverting amplifier and (b) inverting amplifier.

where

$K_o = 1 + R_B/R_A$ is the ideal gain

$A_d(s)$ is the operational amplifier's differential gain

Using the dominant-pole model for the op-amp,

$$A_d(s) = \frac{GB}{s + \omega_a} = \frac{A_o \omega_a}{s + \omega_a} \tag{11.79}$$

where

A_o is the dc gain

ω_a is the dominant pole

GB is the gain-bandwidth product of the op-amp

Inserting Equation 11.79 into Equation 11.78 gives

$$K = \frac{GB}{s + \omega_a(1 + A_o/K_o)} \approx \frac{GB}{s + GB/K_o} \tag{11.80}$$

The approximation of the K expression is valid provided $A_o \gg K_o$. The magnitude and phase of Equation 11.80 are

$$|K(j\omega)| = \frac{GB}{\sqrt{\omega^2 + (GB/K_o)^2}} \tag{11.81}$$

and

$$\arg[K(j\omega)] = -\tan^{-1}\left(\frac{\omega}{GB} K_o\right) \tag{11.82}$$

Note that for $\omega > GB/K_o$, Equations 11.81 and 11.82 may be approximated by

$$|K(j\omega)| \approx K_o \tag{11.83}$$

and

$$\arg[K(j\omega)] \approx -\frac{\omega}{GB} K_o \tag{11.84}$$

The phase expression (Equation 11.84) describes the phase lag introduced to an active filter realization by the op-amp's finite gain-bandwidth product. To illustrate the importance of taking this into account, consider a positive-gain filter with $K_o = 3$ and operation frequency $\omega = \text{GB}/30$ [5]. According to Equation 11.84, a phase lag of 5.73° is introduced to the filter by the op-amp. Such phase lag might significantly impair the performance of some filter realizations.

The inverting amplifier of Figure 11.21b is described by

$$K = \frac{V_2}{V_1} = \frac{-A_d(s)[R_B/(R_A + R_B)]}{1 + A_d(s)[R_A/(R_A + R_B)]} \tag{11.85}$$

Inserting Equation 11.79 into Equation 11.85 gives

$$K = \frac{-[R_B/(R_A + R_B)]\text{GB}}{s + \omega_a[1 + A_o R_A/(R_A + R_B)]} \approx \frac{-[R_B/(R_A + R_B)]\text{GB}}{s + \text{GB}[R_A/(R_A + R_B)]} \tag{11.86}$$

where as before the approximation is valid if $A_o \gg (R_A + R_B)/R_A$. The magnitude and phase expressions for Equation 11.86 are

$$|K(j\omega)| = \frac{\text{GB}R_B/(R_A + R_B)}{\sqrt{\omega^2 + [\text{GB}R_A/(R_A + R_B)]^2}} \tag{11.87}$$

and

$$\arg[K(j\omega)] = \pi - \tan^{-1}\left[\frac{\omega}{\text{GB}}\left(1 + \frac{R_B}{R_A}\right)\right] \tag{11.88}$$

If $\omega < \text{GB}\ R_A/(R_A + R_B)$, then Equations 11.87 and 11.88 are approximated by

$$|K(j\omega)| \approx \frac{R_B}{R_A} \tag{11.89}$$

and

$$\arg[K(j\omega)] \approx \pi - \left[\frac{\omega}{\text{GB}}\left(1 + \frac{R_B}{R_A}\right)\right] \tag{11.90}$$

An important limitation of the inverting configuration is its bandwidth [5], compared to that of the noninverting configuration, when small values of gain K are desired. Take, for example, the case where a unity-gain amplifier is needed. Using the amplifier of Figure 11.21a, the resistor values would be $R_B = 0$ and $R_A = \infty$. From Equation 11.80 we see that this places a pole at $s = -\text{GB}$. Using the amplifier of Figure 11.21b, the resistors would be selected such that $R_B = R_A$. From Equation 11.86 we see that a pole located at $s = -\text{GB}/2$ results. Hence, for unity-gain realizations, the inverting configuration has half the bandwidth of the noninverting case. However, for higher gains, as the ratio R_B/R_A becomes large, both configurations yield a pole location that approaches $s = -(\text{GB})R_A/R_B$.

When using the inverting configuration described above to implement a negative-gain realization, the input impedance to the amplifier is approximately R_A. R_A must then be carefully selected to avoid loading the *RC* network. Another problem is related to maintaining low phase lag when a high Q is desired [5]. A Q of 10 requires that the ratio R_B/R_A be approximately 900. However, to avoid a phase lag of 6°, Equation 11.90 indicates that for $Q = 10$ the maximum filter design frequency cannot exceed

approximately GB/8500. If GB = 1 MHz, say if the 741 op-amp is being used, then the filter cannot function properly above a frequency of about 100 Hz.

We have seen that a practical amplifier implementation contributes a pole to the gain K of a filter due to the op-amp's finite gain-bandwidth product GB. This means that an active filter's transfer function will have an additional pole that is attributed to the amplifier. The poles then of an active filter will be influenced by this additional pole. Consider the positive-gain amplifier used in the Sallen–Key low-pass filter (Figure 11.6). The filter's ideal transfer function is expressed in Equation 11.21. For the equal R, equal C case ($R_1 = R_3$, $C_2 = C_4$) we have $R_B = R_A[2 - (1/Q)]$. By inserting Equation 11.80 into Equation 11.21, the transfer function becomes [3]

$$T(s_n) = \frac{V_2}{V_1} = \frac{\mathrm{GB_n}}{s_\mathrm{n}^3 + s_\mathrm{n}^2\left(3 + \frac{\mathrm{GB_n}}{1-\frac{1}{Q}}\right) + s_\mathrm{n}\left(1 + \frac{\mathrm{GB_n}}{3Q-1}\right) + \frac{\mathrm{GB_n}}{3-\frac{1}{Q}}} \tag{11.91}$$

where the normalized terms are defined by $s_\mathrm{n} = s/\omega_\mathrm{o}$ and $\mathrm{GB_n}/\omega_\mathrm{o}$. The manner in which the poles of Equation 11.91 are influenced by the amplifier's finite gain-bandwidth product is illustrated in Figure 11.22a. This plot is the upper left-half s_n plane. The solid lines correspond to constant values of Q and the circles correspond to discrete values of $\mathrm{GB_n}$, which were the same for each Q. For $Q = 0.5$ the $\mathrm{GB_n}$ values are labeled. Note that the $\mathrm{GB_n} = \infty$ case shown is the ideal case where amplifier GB is infinite. The poles shift toward the origin as ω_o approaches a value near GB. Furthermore, for large design values of Q, i.e., greater than 3, the actual Q value will decrease as ω_o approaches GB.

As another example, consider the Sallen–Key low-pass again but for the case where positive unity-gain and equal resistance ($R_1 = R_3$) are desired. When the amplifier's gain-bandwidth product is taken into account, the filter's transfer function becomes [3]

$$T(s) = \frac{V_2}{V_1} = \frac{\mathrm{GB_n}}{s_\mathrm{n}^3 + s_\mathrm{n}^2\left(2Q + \frac{1}{Q} + \mathrm{GB_n}\right) + s_\mathrm{n}\left(1 + \frac{\mathrm{GB_n}}{Q}\right) + \mathrm{GB_n}} \tag{11.92}$$

where again $s_\mathrm{n} = s/\omega_\mathrm{o}$ and $\mathrm{GB_n} = \mathrm{GB}/\omega_\mathrm{o}$. The poles of Equation 11.92 are shown in Figure 11.22b. The plots are similar but the Figure 11.22b) plot constant $\mathrm{GB_n}$ values for the range of Q shown are closer

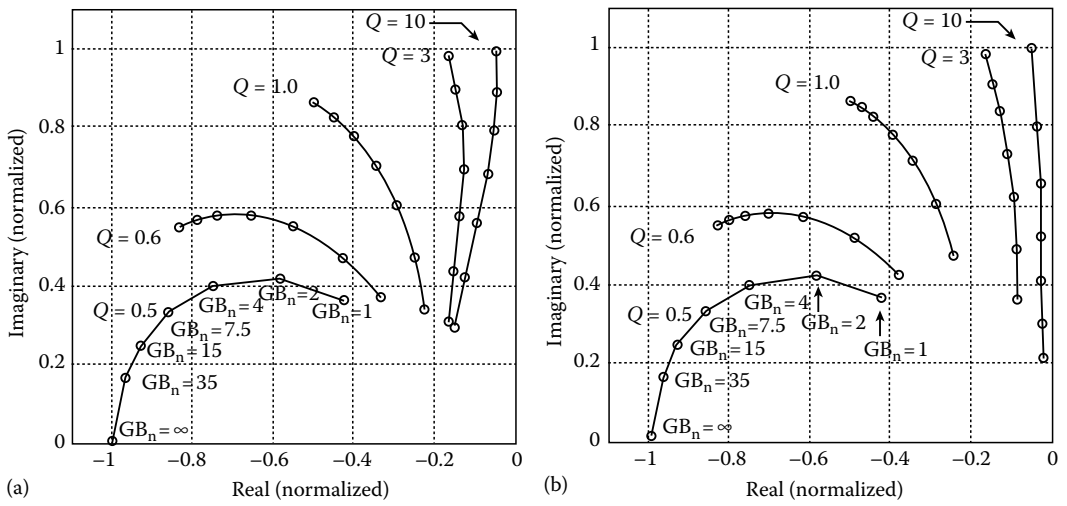


FIGURE 11.22 Effect of GB on the poles of the Sallen–Key second-order low-pass case. (a) Equal R, equal C case and (b) unity gain, equal R case.

TABLE 11.2 Summary of Finite GB on Filter Parameters

Sallen–Key Low-Pass Realizations	
Equal R, equal C	$\omega_0(\text{actual}) \approx \omega_o(\text{design})\left[1-\frac{1}{2}\left(3-\frac{1}{Q}\right)^2\frac{\omega_o}{GB}\right]$
	$Q(\text{actual}) \approx Q(\text{design})\left[1-\frac{1}{2}\left(3-\frac{1}{Q}\right)^2\frac{\omega_o}{GB}\right]$
	$GB \geq 45\left(1-\frac{1}{3Q}\right)^2\omega_0^*$
Unity-gain, equal R	$\omega_0(\text{actual}) \approx \omega_o(\text{design})\left(1-\frac{\omega_o Q}{GB}\right)$
	$Q(\text{actual}) \approx Q(\text{design})\left(1+\frac{\omega_o Q}{GB}\right)$
	$GB \geq 10\omega_0^*$

"*" denotes the condition for $\Delta\omega_o/\omega_o = \Delta Q/Q \leq 10\%$, i.e., for ω_o and Q to change less than 10% [3].

together than the Figure 11.22a plot. This means that the poles of the unity-gain, equal R design are less sensitive to amplifier gain-bandwidth product than the equal R, equal C case. The influence GB has on Sallen–Key low-pass filter parameters is summarized in Table 11.2.

11.7.3 Slew Rate

Slew rate is another operational amplifier parameter that warrants consideration in active filter design. When a filter signal level is large and/or the frequency is high, op-amp slew rate (SR) limitations can affect the small-signal behavior of amplifiers used in low-gain filter realizations. The amplifier's amplitude and phase response can be distorted by large-signal amplitudes. An analysis of slew rate induced distortion is presented in Ref. [1]. For the inverting amplifier (Figure 11.21b), slewing increases the small-signal phase lag. This distortion is described by Ref. [5]

$$\arg[K(j\omega)] = \pi - \tan^{-1}\left(\frac{(\omega/\text{GB}(1+R_B/R_A))}{N(A)}\right) \tag{11.93}$$

where

$N(A)$ is a describing function

A is the peak amplitude of the sinusoidal voltage input to the amplifier

$N(A) = 1$ while the op-amp is not slewing, yielding the same phase lag described by Equation 11.88. However, for conditions where slewing is present, $N(A)$ becomes less than 1. Under such conditions the filter may become unstable. Refer to Ref. [1] and Ref. [5, Chapter 4, Section 6] for a detailed discussion of slew rate distortion.

Selecting an operational amplifier with higher slew rate can reduce large-signal distortion. Given the choice of FET input or BJT input op-amps, FET input op-amps tend to have higher slew rates and higher input signal thresholds before slewing occurs. When selecting an amplifier for a filter it is important to remember that slewing occurs when the magnitude of the output signal slope exceeds SR and when the input level exceeds a given voltage threshold. Often the slew rate of a filter's amplifier sets the final limit of performance that can be attained in a given active filter realization at high frequencies.

11.7.4 Noise

The noise performance of a given filter design will ultimately determine the circuit's signal-to-noise ratio and dynamic range [7]. For these reasons, a strong understanding of noise analysis is required for effective filter design. In this section noise analysis of the noninverting and inverting op-amp

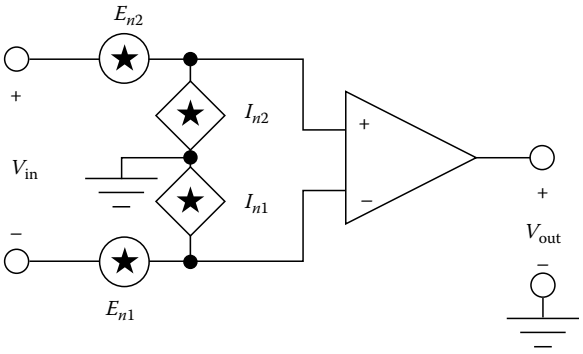


FIGURE 11.23 Operational amplifier noise model with four noise sources.

configurations are discussed for the implementation of the gain block *K*. An example noise analysis is illustrated for a simple filter design.

Proceeding with noise analysis, consider the four noise sources associated with operational amplifier. Shown in Figure 11.23, E_{n1} and E_{n2} represent the op-amp's voltage noise referred to the negative and positive inputs, respectively, and I_{n1} and I_{n2} represent the op-amp's current noise referred to the negative and positive inputs, respectively. E_{n1} and E_{n2} are related to E_n, from an op-amp manufacture's data sheet, through the relationship

$$E_n = \sqrt{E_{n1}^2 + E_{n2}^2} \left[\text{V}/\sqrt{\text{Hz}}\right] \tag{11.94}$$

I_n is also usually reported in an op-amp manufacturer's data sheet. The value given for I_n is used for I_{n1} and I_{n2}.

$$I_n = I_{n1} = I_{n2} \left[\text{A}/\sqrt{\text{Hz}}\right] \tag{11.95}$$

The analysis of the filter noise performance begins with the identification of all noise sources, which include the op-amp noise sources and all resistor thermal noise ($E_t = \sqrt{4\,\text{kt}\,R\Delta f}$ where 4kT is approximately 1.65×10^{-20} V^2/Ω; Hz at room temperature and Δf is the noise bandwidth). To begin the analysis, uncorrelated independent signal generators can be substituted for each noise source, voltage sources for noise voltage sources, and current sources for current noise sources. By applying superposition and using the rms value of the result, the noise contribution of each element to the total output noise E_{no} can be determined. Only squared rms values can be summed to determine E_{no}^2. The resultant expression then for E_{no}^2 should never involve subtraction of noise sources, only addition. Sources of noise in a circuit always add to the total noise at the circuit's output. The total noise referred to the input E_{ni} is simply E_{no} divided by the circuit's transfer function gain. Capacitors ideally only influence noise performance through their effect on the circuit's transfer function. That is, the transfer gain seen by a noise source to the circuit's output can be influenced by capacitors depending on the circuit. As a result, of course, a circuit's noise performance can be a strong function of frequency.

Consider again the noninverting op-amp configuration, which can be used to implement positive *K*. The circuit is shown in Figure 11.24a with appropriate noise sources included. Resistor R_s has been included to represent the driving source's (V_1) output resistance. The simplified circuit lumps together the amplifier's noise sources into E_{na}, shown in Figure 11.24b. E_{na} is described by Equation 11.96. The circuit in Figure 11.24b can be used in place of the positive gain block *K* to simplify the noise analysis of a larger circuit. R_s obviously will be determined by the circuit seen looking back from the amplifier's input:

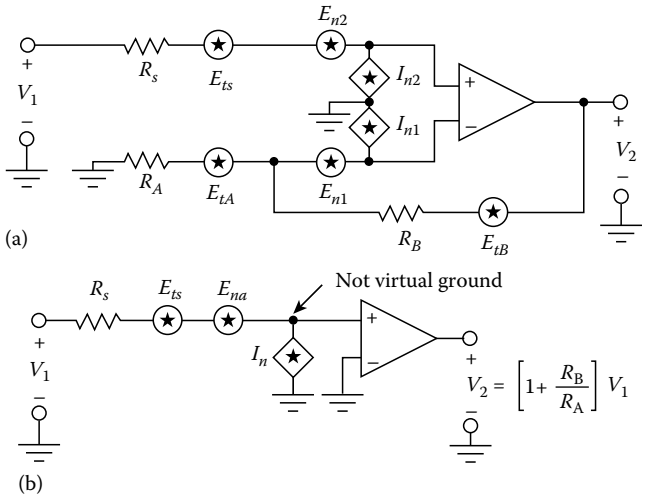


FIGURE 11.24 (a) Noninverting op-amp configuration with noise sources and (b) simplified noninverting amplifier circuit with noise sources.

$$E_{na}^2 = E_n^2 + \left(\frac{R_A}{R_A + R_B}\right)^2 \left(E_{tB}^2\right) + \left(\frac{R_B}{R_A + R_B}\right)^2 \left(E_{tA}^2\right) + I_{n1}^2 (R_A \| R_B)^2 \tag{11.96}$$

The inverting op-amp configuration for implementing a negative gain is shown in Figure 11.25a with noise sources included. Here again, the op-amp's noise sources can be lumped together and referred to the input as noise source E_{nb} shown in Figure 11.25b. For simplicity, R_s has been lumped together with R_A.

E_{nb} is described by Equation 11.97. The simplified circuit in Figure 11.25b is convenient for simplifying the noise analysis of larger circuit that uses a negative gain block.

$$E_{nb}^2 = \left(1 + \frac{R_A}{R_B}\right)^2 \left(E_n^2\right) + \left(\frac{R_A}{R_B}\right)^2 \left(E_{tB}^2\right) \tag{11.97}$$

Example 11.10

First-order low-pass filter noise analysis. Noise analysis of a low-pass filter with noninverting midband gain of 6 dB and −20 dB/decade gain roll-off after a 20 kHz cutoff frequency is desired. Recall from Section 11.3 that $\omega_o = RC$. Let us designate R_1 for R and C_2 for C. To obtain $\omega_o = 2\pi$(20 kHz), commercially available component values $R_1 = 16.9$ kΩ and $C_2 = 470$ pF (such as a 1% metal-film resistor for R_1 and a ceramic capacitor for C_2) can be used. Implement K with the noninverting op-amp configuration and use Figure 11.24b as a starting point of the noise analysis. The midband gain of 6 dB can be achieved by selecting $R_A = R_B$. For this example let $R_A = R_B = 10.0$ kΩ, also available in 1% metal film. Including the thermal noise source for R_1, the circuit shown in Figure 11.26 results.

From this circuit we can easily derive the output noise of the low-pass filter.

$$E_{\text{no}}^2 = K^2 \left[E_{na}^2 + \left| \frac{Z_2}{R_1 + Z_2} \right|^2 \left(E_{t1}^2\right) + |R_1 \| Z_2|^2 \left(I_n^2\right) \right] \tag{11.98}$$

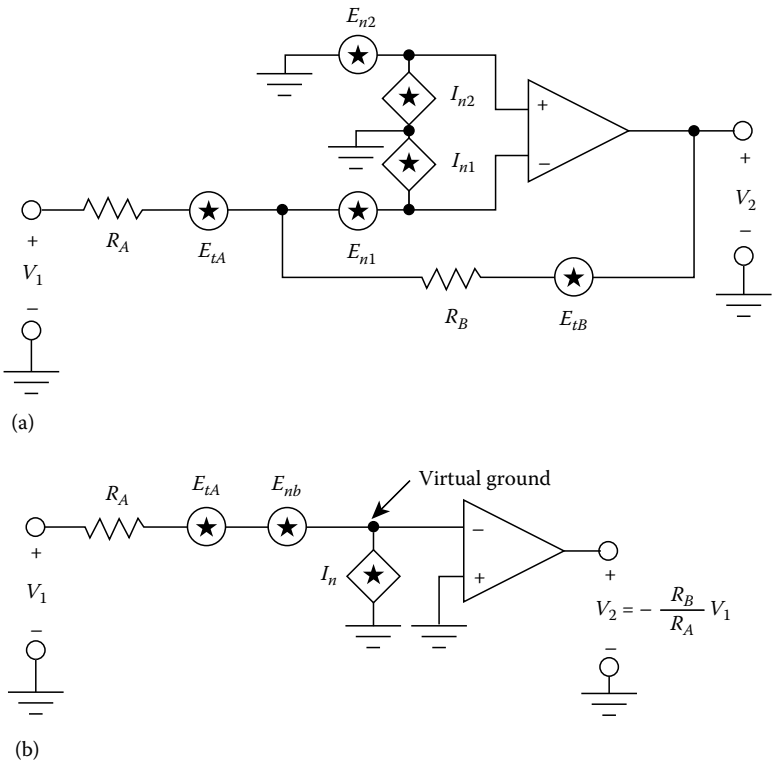


FIGURE 11.25 (a) Inverting op-amp configuration with noise sources and (b) simplified inverting amplifier circuit with noise sources.

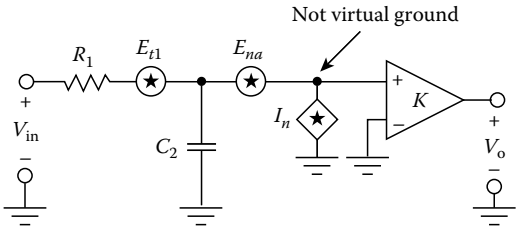


FIGURE 11.26 Positive gain first-order low-pass filter with noise sources.

where

$Z_2 = 1/sC_2$, the impedance of the capacitor

E_{na} is defined by Equation 11.96

The commercial op-amp OP-27 can be used to implement K and thus determines the values of E_n and I_n. For OP-27, $E_n = 3\,\text{nV}/\sqrt{\text{Hz}}$ and $I_n = 0.4\,\text{pA}/\sqrt{\text{Hz}}$. A comparison of the hand analysis to a PSpice simulation of this low-pass filter is shown in Figure 11.27. The frequency dependence of K was not included in the hand analysis since the filter required a low midband gain of two and low cutoff frequency relative to the OP-27's 8 MHz gain-bandwidth product. The simulated frequency response is shown in Figure 11.28. The computer simulation did include a complete model for the OP-27.

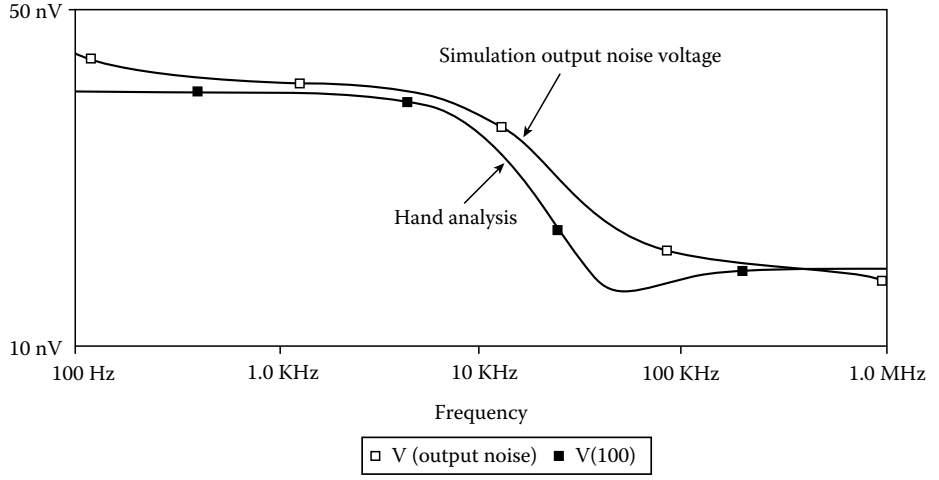


FIGURE 11.27 Comparison of calculated output noise with simulation.

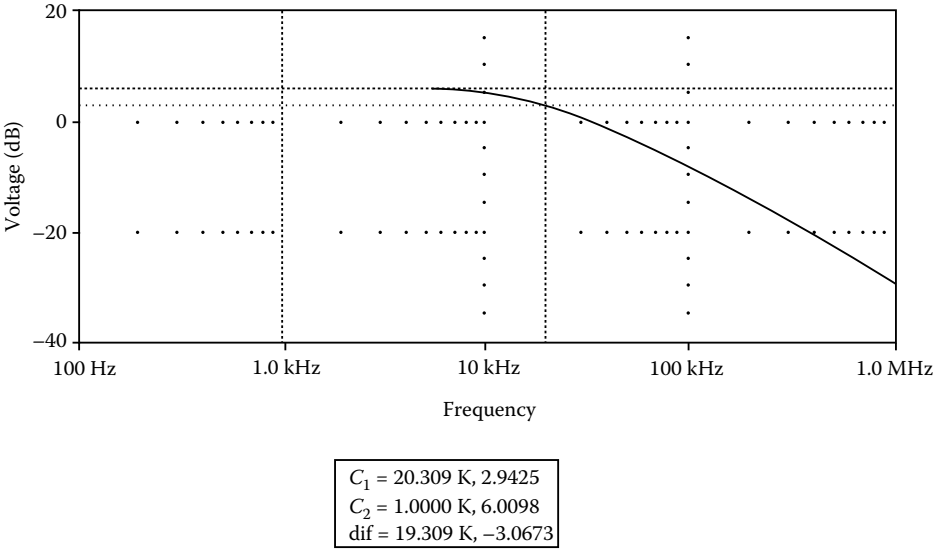


FIGURE 11.28 Simulated low-pass filter frequency response. The cursors indicate a −3 dB frequency of 20 kHz.

As indicated by Figure 11.27, the output noise predicted by Equation 11.98 agrees well with the simulation. Analyzing the noise contribution of each circuit element to the filter's total output noise can provide insight for improving the design. To this end, Table 11.3 was generated for this filter.

Note that at 1 kHz R_1 is the dominant source of noise. A smaller value of R_1 could be chosen at the expense of increasing C_2. Care must be taken in doing so because changing C_2 also changes the gain multiplier for I_n and R_1's noise source, as described in Equation 11.98. The noise bandwidth, using a single-pole response approximation, is 20 kHz $(\pi/2) = 31.4$ kHz. Hence, $E_{ni} = \left(20.2\,\text{nV}/\sqrt{\text{Hz}}\right)$. $(31.4\ \text{kHz})^{1/2} = 3.58\ \mu\text{V}$. To achieve then a signal-to-noise ratio of 10, the input signal level must be 35.8 μV. This would be more difficult to attain if 741 op-amp has been used, with its $E_n = 20.2\,\text{nV}/\sqrt{\text{Hz}}$, rather than the OP-27.

TABLE 11.3 Noise Contributions of Each Circuit Element at 1 kHz

Noise Source	Noise Value	Gain Multiplier	Output Noise Contribution	Input Noise Contribution
E_n	$3\,nV/\sqrt{Hz}$	2	$6\,nV/\sqrt{Hz}$	$3\,nV/\sqrt{Hz}$
I_n	$0.4\,pA/\sqrt{Hz}$	35.2 k	$14.1\,nV/\sqrt{Hz}$	$7.04\,nV/\sqrt{Hz}$
R_A	$12.65\,nV/\sqrt{Hz}$	1	$12.65\,nV/\sqrt{Hz}$	$6.325\,nV/\sqrt{Hz}$
R_B	$12.65\,nV/\sqrt{Hz}$	1	$12.65\,nV/\sqrt{Hz}$	$6.325\,nV/\sqrt{Hz}$
R_1	$16.4\,nV/\sqrt{Hz}$	1.998	$32.8\,nV/\sqrt{Hz}$	$16.4\,nV/\sqrt{Hz}$
Total noise contributions			$40.4\,nV/\sqrt{Hz}$	$20.2\,nV/\sqrt{Hz}$

Often in signal processing circuits several stages are cascaded to shape the response of the system. In doing so, the noise contributed by each stage is an important consideration. Placement of each stage within the cascade ultimately determines the E_{ni} of the entire cascade. The minimum E_{ni} is achieved by placing the highest gain first. By doing so the noise of all the following stages is divided by the single largest gain stage of the entire cascade. This comes, however, at the sacrifice of dynamic range. With the highest gain first in the cascade, each following stage has a larger input signal than in the case where the highest gain stage is last or somewhere between first and last in the cascade. Dynamic range is lost since there is a finite limit in the input signal level to any circuit before distortion results.

11.8 Summary

This chapter has examined active filters using low-gain amplifiers. These filters are capable of realizing any second-order transfer function. Higher order transfer functions are realized using cascaded first- and second-order stages. The nonideal behavior of finite gain-bandwidth, slew rate, and noise was examined. More information on this category of filters can be found in the references.

References

1. P. E. Allen, Slew induced distortion in operational amplifiers, *IEEE J. Solid-State Circuits,* SC1,2, 39–44, Feb. 1977.
2. H. W. Bode, *Network Analysis and Feedback Amplifier Design*, Princeton, NJ: Van Nostrand, 1945.
3. E. J. Kennedy, *Operational Amplifier Circuits—Theory and Applications,* New York: Holt, Rinehart & Winston, 1988.
4. W. J. Kerwin and L. P. Huelsman, The design of high-performance active *RC* bandpass filters, In *Proc. IEEE Int. Conv. Rec.,* pt. 10, Mar. 1966, pp. 74–80.
5. L. P. Huelsman and P. E. Allen, *Introduction to the Theory and Design of Active Filters*, New York: McGraw-Hill, 1980.
6. MicroSim Corporation, *PSpice Circuit Synthesis*, *version 4.05,* Irvine, CA: MicroSim, 1991.
7. C. D. Motchenbacher and J. A. Connelly, *Low Noise Electronic System Design*, New York: Wiley, 1993.
8. R. P. Sallen and E. L. Key, A practical method of designing *RC* active filters, *IRE Trans. Circuit Theory,* CT-2, 74–85, 1955.
9. R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters: Passive, Active RC, and Switched Capacitor*, Englewood Cliffs, NJ: Prentice-Hall, 1990.
10. A. S. Sedra and P. O. Brackett, *Filter Theory and Design: Active and Passive*, Portland, OR: Matrix, 1978.
11. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, New York: Holt, Rinehart & Winston, 1987.
12. A. Zverev, *Handbook of Filter Synthesis*, New York: Wiley, 1967.

12

Single-Amplifier Multiple-Feedback Filters

12.1 Introduction **12**-1
12.2 General Structure for Single-Amplifier Filters **12**-1
12.3 All-Pole Realizations of the MFB Structure **12**-3
Low-Pass Structure • High-Pass Structure • Bandpass Structure
12.4 MFB All-Pole Designs **12**-3
12.5 Practical Considerations in the Design of MFB Filters **12**-8
Sensitivity • Effect of Finite Amplifier Gain • Tuning
12.6 Modified Multiple-Loop Feedback Structure **12**-10
12.7 Biquadratic MMFB Structure **12**-12
12.8 Conclusions **12**-13
References **12**-13

F. William Stephenson
Virginia Polytechnic Institute and State University

12.1 Introduction

In this section we will consider the design of second-order sections that incorporate a single operational amplifier. Such designs are based upon one of the earliest approaches to *RC* active filter synthesis, which has proven to be a fundamentally sound technique for over 30 years. Furthermore, this basic topology has formed the basis for designs as technology has evolved from discrete component assemblies to monolithic realizations. Hence, the circuits presented here truly represent reliable well-tested building blocks for sections of modest selectivity.

12.2 General Structure for Single-Amplifier Filters

The general structure of Figure 12.1 forms the basis for the development of infinite-gain single-amplifier configurations. Simple circuit analysis may be invoked to obtain the open-circuit voltage transfer function. For the passive network

$$I_1 = y_{11}V_1 + y_{12}V_2 + y_{13}V_3 \tag{12.1}$$

$$I_2 = y_{21}V_1 + y_{22}V_2 + y_{23}V_3 \tag{12.2}$$

$$I_3 = y_{31}V_1 + y_{32}V_2 + y_{33}V_3 \tag{12.3}$$

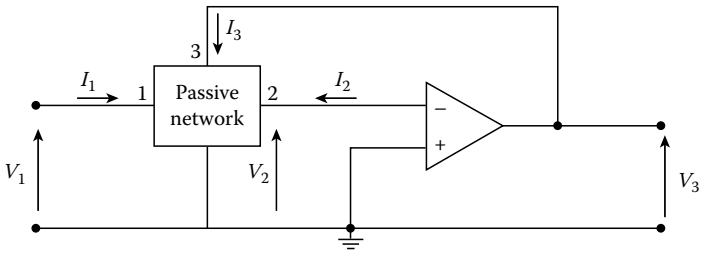


FIGURE 12.1 General infinite-gain single-amplifier structure.

For the amplifier, ideal except for finite gain A

$$V_3 = -AV_2 \tag{12.4}$$

Noting that $I_2 = 0$, the above equations reduce to the following expression for the voltage transfer function:

$$\frac{V_3}{V_1} = \frac{-y_{31}}{y_{32} + \frac{y_{33}}{A}} \tag{12.5}$$

As $A \to \infty$, which we can expect at low frequencies, the above expression reduces to the more familiar

$$\frac{V_3}{V_1} = -\frac{y_{31}}{y_{32}} \tag{12.6}$$

Theoretically, a wide range of transfer characteristics can be realized by appropriate synthesis of the passive network [1]. However, it is not advisable to extend synthesis beyond second-order functions for structures containing only one operational amplifier due to the ensuing problems of sensitivity and tuning. Furthermore, notch functions require double-element replacements [2] or parallel ladder arrangements [3], which are nontrivial to design, and whose performance is inferior to that resulting from other topologies such as those discussed in Chapters 13 and 14.

While formal synthesis techniques could be used to meet particular requirements, the most common approach is to use a double-ladder realization of the passive network, as shown in Figure 12.2. This arrangement, commonly referred to as the multiple-loop feedback (MFB) structure, is described by the following voltage transfer ratio:

$$\frac{V_3}{V_1} = \frac{-Y_1 Y_3}{Y_5(Y_1 + Y_2 + Y_3 + Y_4) + Y_3 Y_4} \tag{12.7}$$

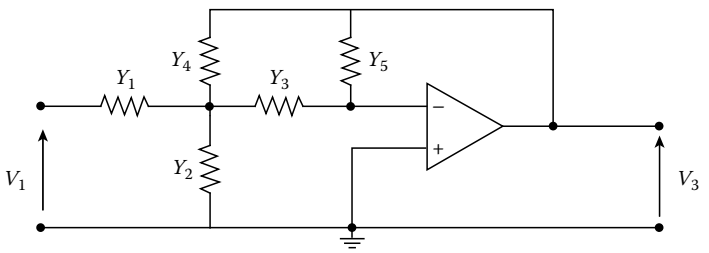


FIGURE 12.2 General double ladder multiple-feedback network.

This negative feedback arrangement yields highly stable realizations. The basic all-pole (low-pass, bandpass, high-pass) functions can be realized by single-element replacements for the admittances $Y_1, \ldots, Y_5$, as described in the following section.

12.3 All-Pole Realizations of the MFB Structure

12.3.1 Low-Pass Structure

The general form of the second-order all-pole low-pass structure is described by the following transfer ratio:

$$\frac{V_3}{V_1} = \frac{H}{s^2 + \frac{\omega_p s}{Q_p} + \omega_p^2} \tag{12.8}$$

By comparing the above requirement with Equation 12.7 it is clear that both Y_1 and Y_3 must represent conductances. Furthermore, by reviewing the requirements for the denominator, Y_5 and Y_2 must be capacitors, while Y_4 is a conductance.

12.3.2 High-Pass Structure

The general form of the second-order all-pole high-pass transfer function is

$$\frac{V_3}{V_1} = \frac{Hs^2}{s^2 + \frac{\omega_p s}{Q_p} + \omega_p^2} \tag{12.9}$$

With reference to Equation 12.7, it is seen that both Y_1 and Y_3 must represent capacitors. There is a need for a third capacitor ($Y_4 = sC_4$) to yield the s^2 term in the denominator function. The remaining two elements, Y_2 and Y_5, represent conductances.

12.3.3 Bandpass Structure

The general form of the second-order all-pole bandpass transfer function is

$$\frac{V_3}{V_1} = \frac{Hs}{s^2 + \frac{\omega_p s}{Q_p} + \omega_p^2} \tag{12.10}$$

Two solutions exist since Y_1 and Y_3 can be either capacitive or conductive. Choosing $Y_1 = G_1$ and $Y_3 = sC_3$ yields $Y_4 = sC_4$ and Y_2, Y_5 are both conductances.

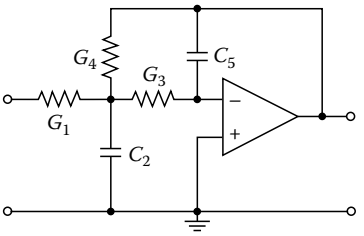
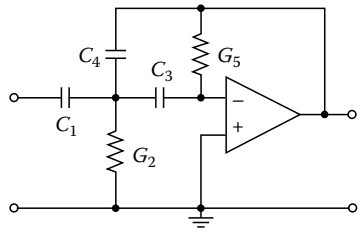
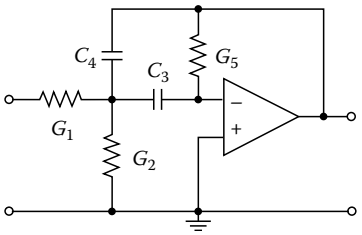
The general forms of the above realizations are summarized in Table 12.1 [4].

12.4 MFB All-Pole Designs

MFB designs are typically reserved for sections having a pole-Q of 10 or less. One of the reasons for this constraint is the reliance upon component ratios for achieving Q. This can be illustrated by consideration of the low-pass structure for which

$$\frac{V_3}{V_1} = \frac{-G_1 G_3}{s^2 C_2 C_5 + sC_5(G_1 + G_3 + G_4) + G_3 G_4} \tag{12.11}$$

TABLE 12.1 MFB All-Pole Realizations

Filter Type	Network	Voltage Transfer Function
Low-pass		$\frac{-G_1G_3}{s^2C_2C_5 + sC_5(G_1 + G_3 + G_4) + G_3G_4}$
High-pass		$\frac{-s^2C_1C_3}{s^2C_3C_4 + sG_5(C_1 + C_3 + C_4) + G_2G_5}$
Bandpass		$\frac{-sG_1C_3}{s^2C_3C_4 + sG_5(C_3 + C_4) + G_5(G_1 + G_2)}$

By comparison with Equation 12.8,

$$Q_p = \frac{\sqrt{C_2C_5G_3G_4}}{C_5(G_1 + G_3 + G_4)}$$

or, in terms of component ratios:

$$Q_p = \frac{\sqrt{C_2}}{\sqrt{C_5}}\left(\frac{1}{\frac{G_1}{\sqrt{G_3G_4}} + \sqrt{\frac{G_3}{G_4}} + \sqrt{\frac{G_4}{G_3}}}\right) \tag{12.12}$$

Hence, high Q_p can only be achieved by means of high component spreads. In general terms, a Q_p of value n requires a component spread proportional to n^2.

Filter design is effected by means of coefficient matching. Thus, for the low-pass case, comparison of like coefficients in Equation 12.8 and the transfer ratio in Table 12.1 yields

$$G_1G_3 = H \tag{12.13}$$

TABLE 12.2 Element Values for the MFB All-Pole Realizations

Element (Table 12.1)	Low-pass	Bandpass	High-Pass
Y_1	$G_1 = \frac{H}{\omega_p}$	$G_1 = H$	$C_1 = H$
Y_2	$C_2 = \dfrac{Q_p\left(2\omega_p^2 + H\right)}{\omega_p^2}$	$G_2 = 2\omega_p Q_p - H$	$G_2 = \omega_p(2+H)Q_p$
Y_3	$G_3 = \omega_p$	$C_3 = 1$	$C_3 = 1$
Y_4	$G_4 = G_3$	$C_4 = C_3$	$C_4 = C_3$
Y_5	$C_5 = \dfrac{\omega_p^2}{Q_p\left(2\omega_p^2 + H\right)}$	$G_5 = \dfrac{\omega_p}{2Q_p}$	$G_5 = \dfrac{\omega_p}{Q_p(2+H)}$

$$C_2 C_5 = 1 \tag{12.14}$$

$$C_5(G_1 + G_3 + G_4) = \frac{\omega_p}{Q_p} \tag{12.15}$$

$$G_3 G_4 = \omega_p^2 \tag{12.16}$$

These equations do not yield an equal-capacitor solution but can be solved for equal-resistor pairs. Hence, if $G_1 = G_3$,

$$G_1 = G_3 = \sqrt{H} \quad \text{(From Equation 12.13)}$$

$$G_4 = \frac{\omega_p^2}{\sqrt{H}} \quad \text{(From Equation 12.16)}$$

Then,

$$C_5 = \frac{\omega_p \sqrt{H}}{Q_p\left(2H + \omega_p^2\right)} = \frac{1}{C_2}$$

An alternative solution for which $G_3 = G_4$ is shown in Table 12.2, together with equal-capacitor designs for the bandpass and high-pass cases.

The conditions [4] for maximum Q_p in the bandpass realization require $C_3 = C_4$ and $G_1 = G_2 = nG_5$, where n is a real number. This yields a maximum Q_p of $\sqrt{n/2}$, and requires that $H = \omega_p Q_p$.

Example 12.1

Using the cascade approach, design a four-pole Butterworth bandpass filter having a Q of 5, a center frequency of 1.5 kHz, and midband gain of 20 dB. Assume that only 6800 pF capacitors are available.

Solution

The low-pass prototype is the second-order Butterworth characteristic having a dc gain of 10 (i.e., 20 dB). Thus,

$$T(s) = \frac{10}{s^2 + \sqrt{2}s + 1} \tag{i}$$

The low-pass-to-bandpass frequency transformation for a Q of 5 entails replacing s in (i) by $5(s + 1/s)$. This yields the following bandpass function for realization:

$$\frac{V_o}{V_i} = \frac{0.4s^2}{s^4 + 0.28284s^3 + 2.04s^2 + 0.28284s + 1}$$
$$= \underbrace{\frac{-sH_1}{(s^2 + 0.15142s + 1.15218)}}_{\text{(section 1)}} \cdot \underbrace{\frac{-sH_2}{(s^2 + 0.13142s + 0.86792)}}_{\text{(section 2)}} \qquad \text{(ii)}$$

$$Q_1 = Q_2 = 7.089$$
$$\omega_{p1} = 1.0734; \quad \omega_{p2} = 0.9316$$

As expected, the Q-factors of the cascaded sections are equal in the transformed bandpass characteristic. However, the order of cascade is still important. So as to reduce the noise output of the filter, it is necessary to apportion most of the gain to section 1 of the cascade. Section 2 then filters out the noise without introducing excessive passband gain. In the calculation that follows, it is important to note that the peak gain of a bandpass section is given by HQ/ω_p.

Since the overall peak gain of the cascade is to be 10, let this also be the peak gain of section 1. Hence,

$$\frac{H_1 Q_1}{\omega_{p1}} = 6.6041 H_1 = 10$$

giving $H_1 = 1.514$.

Furthermore, from (ii)

$$H_1 H_2 = 0.4$$

so that $H_2 = 0.264$.

The design of each bandpass section proceeds by coefficient matching, conveniently simplified by Table 12.2. Setting $C_3 = C_4 = 1$ F, the normalized resistor values for section 1 may be determined as

$$R_1 = 0.661\ \Omega; \quad R_2 = 0.073\ \Omega; \quad R_5 = 13.208\,\Omega$$

The impedance denormalization factor is determined as

$$z_n = \frac{10^{12}}{2\pi \times 1500 \times 6800} = 15{,}603$$

Thus, the final component values for section 1 are

$$\left.\begin{aligned} C_1 &= C_2 = 6800\ \text{pF} \\ R_1 &= 10.2\ \text{k}\Omega \\ R_2 &= 1.13\ \text{k}\Omega \\ R_5 &= 205\ \text{k}\Omega \end{aligned}\right\} \text{Standard 1\% values}$$

Note the large spread in resistance values ($R_5/R_2 \simeq 4Q^2$) and the fact that this circuit is only suitable for low-Q realizations. It should also be noted that the amplifier open-loop gain at ω_p must be much greater than $4Q^2$ if it is not to cause significant differences between the design and measured values of Q.

The component values for section 2 are determined in an identical fashion.

Example 12.2

Design the MFB bandpass filter characterized in Example 12.1 as a high-pass/low-pass cascade of second-order sections. Use the design equations of Table 12.2 and, where possible, set capacitors equal to 5600 pF. It is suggested that you use the same impedance denormalization factor in each stage. Select the nearest preferred 1% resistor values.

Solution

Since the peak gain of the overall cascade is to be 10, let this also be the gain of stage 1 (this solution yields the best noise performance). The peak gain of the low-pass section is given by

$$\frac{H_1 Q}{\sqrt{1 - 1/2Q^2}} = 7.16H_1 = 10 \quad \therefore H_1 = 1.397$$

The overall transfer function (from Example 12.1) is

$$\frac{V_o}{V_i} = \frac{0.4s^2}{s^4 + 0.2824s^3 + 2.04s^2 + 0.2824s + 1}$$
$$\therefore H_1 H_2 = 0.4 \quad \text{so that } H_2 = 0.286$$

Thus, assuming a low-pass/high-pass cascade, we have

$$\frac{V_o}{V_i} = \underbrace{\frac{1.397}{s^2 + 0.15145 + 1.1522}}_{\text{section 1}} \cdot \underbrace{\frac{0.286s^2}{s^2 + 0.1314s + 0.8679}}_{\text{section 2}}$$

Design the low-pass section (section 1) using Table 12.2 to yield

$$G_1 = \frac{H_1}{\sqrt{1.15218}} = 1.301, \quad \text{so that } R_1 = 0.7684\,\Omega$$
$$C_2 = \frac{\left(2\omega_p^2 + H_1\right)}{\omega_p^2} Q_p = 22.77\,\text{F}$$
$$G_3 = \omega_p = 1.0733, \quad \text{so that } R_3 = 0.9316\,\Omega = R_4$$
$$C_5 = 1/C_2 = 0.0439\,\text{F}$$

Now, design the high-pass section (section 2) to yield

$$C_1' = 0.286\,\text{F}$$
$$G_2' = \omega_p(2 + H_2)Q = 15.099, \quad \text{so that } R_2' = 0.0662\,\Omega$$
$$C_3' = C_4' = 1\ \text{F}$$
$$G_5' = \frac{\omega_p}{Q_p(2 + H)} = 0.0574, \quad \text{so that } R_5' = 17.397\,\Omega$$

To obtain as many 5600 pF capacitors as possible, the two sections should be denormalized separately. However, in this example, a single impedance denormalization will be used. Setting $C_3' = C_4' = 5600$ pF yields $z_n = 18{,}947$.

This leads to the following component values:

Low-Pass Stage	High-Pass Stage
$R_1 = 2.204\ \text{k}\Omega\ (2.21\ \text{k}\Omega)$	$C_1' = 1602\ \text{pF}$
$C_2 = 0.128\ \mu\text{F}$	$R_2' = 1.254\ \text{k}\Omega\ (1.24\ \text{k}\Omega)$
$R_3 = 17.651\ \text{k}\Omega\ (17.8\ \text{k}\Omega) = R_4$	$C_3' = C_4' = 5600\ \text{pF}$
$C_5 = 246\ \text{pF}$	$R_5' = 329.62\ \text{k}\Omega\ (332\ \text{k}\Omega)$

Note: 1% values in parentheses.

12.5 Practical Considerations in the Design of MFB Filters

Sensitivity, the effects of finite amplifier gain, and tuning are all of importance in practical designs. The following discussion is based upon the bandpass case.

12.5.1 Sensitivity

Taking account of finite amplifier gain A, but assuming $R_{\text{in}} = \infty$ and $R_{\text{o}} = 0$ for the amplifier, the bandpass transfer function becomes

$$\frac{V_3}{V_1} = \frac{-sG_1/C_4\left(1+\frac{1}{A}\right)}{s^2 + s\left\{\frac{G_5(C_3+C_4)}{C_3C_4} + \frac{G_1+G_2}{C_4(1+A)}\right\} + \left\{\frac{G_5(G_1+G_2)}{C_3C_4}\right\}} \tag{12.17}$$

which is identical to the expression in Table 12.1 if $A = \infty$.

Assuming a maximum Q design,

$$Q = \frac{Q_{\text{p}}}{1 + \frac{2Q_{\text{p}}^2}{(1+A)}} \tag{12.18}$$

where

Q_{p} is the desired selectively

Q is the actual Q-factor in the presence of finite amplifier gain

If $A \gg 2Q - 1$, the classical Q-sensitivity may be derived as

$$S_A^Q = \frac{2Q^2}{A} \tag{12.19}$$

which is uncomfortably high. By contrast, the passive sensitivities are relatively low:

$$S_{C_3,C_4}^Q = 0; \quad S_{G_5}^Q = -0.5; \quad S_{G_1,G_2}^Q = 0.25$$

while the ω_{p} sensitivities are all ± 0.5.

12.5.2 Effect of Finite Amplifier Gain

The effect of finite amplifier gain can be further illustrated by plotting Equation 12.18 for various Q-factors, and for two commercial operational amplifiers. Assuming a single-pole roll-off model, the frequency dependence of open-loop gain for μ *A*741 and *LF*351 amplifiers is as follows:

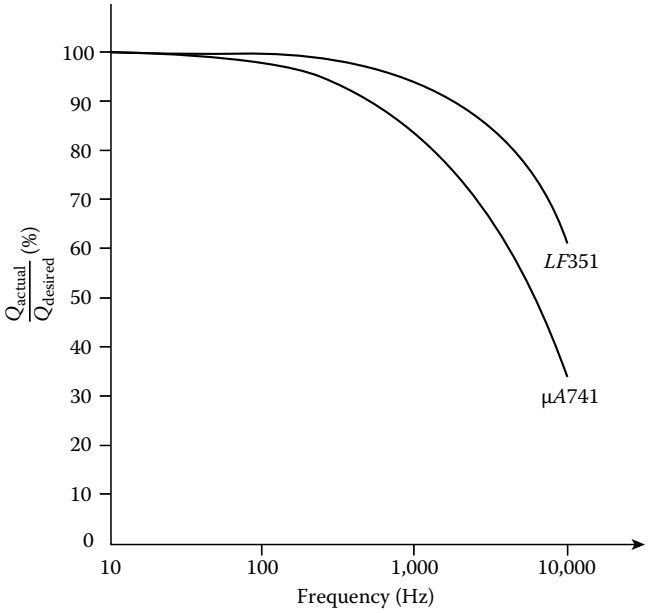


FIGURE 12.3 Effect of finite amplifier gain on the design Q for MFB bandpass realizations using two commercial operational amplifiers.

	Gain	
Frequency (Hz)	μ*A*741	*LF*351
10	1.3×10^5	3.16×10^5
100	10^4	3.16×10^4
1,000	10^3	3.16×10^3
10,000	10^2	3.16×10^2

Figure 12.3 shows the rather dramatic fall-off in actual Q as frequency increases (and hence gain decreases). Thus, for designs of modest Q (note that $A \gg 2Q - 1$), a very high quality amplifier is needed if the center frequency is more than a few kilohertz. For example, the *LF*351 with a unity gain frequency of 4 MHz will yield 6% error in Q at a frequency of only 1 kHz.

12.5.3 Tuning

Limited functional tuning of the bandpass section is possible. For example, the midband (peak) gain

$$K_o = \frac{G_1 C_3}{G_5(C_3 + G_4)} \tag{12.20}$$

may be adjusted by means of either G_1 or G_5.

Subsequent adjustment of either Q_p or ω_p is possible via G_2. In view of the discussion above, it is most likely that adjustment of Q_p will be desired. However, since the expressions for Q_p and ω_p are so similar, any adjustment of Q_p is likely to require an iterative procedure to ensure that ω_p does not change undesirably.

A more desirable functional tuning result is obtained in circumstances where it is necessary to preserve a constant bandwidth, i.e., in a spectrum analyzer. Since

$$\omega_p = \sqrt{G_5(G_1 + G_2)/C_3C_4} \tag{12.21}$$

and bandwidth, B, is

$$B = \frac{\omega_p}{Q_p} = G_5(C_3 + C_4) \tag{12.22}$$

adjustment of G_2 will allow for a frequency sweep without affecting K_o or B.

An alternative to functional tuning may be found by adopting deterministic [5] or automatic [6] tuning procedures. These are particularly applicable to hybrid microelectronic or monolithic realizations.

12.6 Modified Multiple-Loop Feedback Structure

In negative feedback topologies such as the MFB, "high" values of Q_p are obtained at the expense of large spreads in element values. By contrast, in positive feedback topologies such as those attributed to Sallen and Key, Q_p is enhanced by subtracting a term from the s^1 (damping) coefficient in the denominator. The two techniques are combined in the MMFB (Deliyannis) arrangement [7] of Figure 12.4.

Analysis of the circuit yields the bandpass transfer function as

$$\frac{V_o}{V_i} = \frac{-sC_3G_1(1+k)}{s^2C_3C_4 + s\{G_5(C_3 + C_4) - kC_3G_1\} + G_1G_5} \tag{12.23}$$

where $k = G_b/G_a$, and the Q-enhancement term "$-kC_3G_1$" signifies the presence of positive feedback. This latter term is also evident in the expression for Q_p:

$$Q_p = \frac{\sqrt{G_1/G_5}}{\left\{\sqrt{\frac{C_4}{C_3}} + \sqrt{\frac{C_3}{C_4}} - k\frac{G_1}{G_5}\sqrt{\frac{C_1}{C_2}}\right\}} \tag{12.24}$$

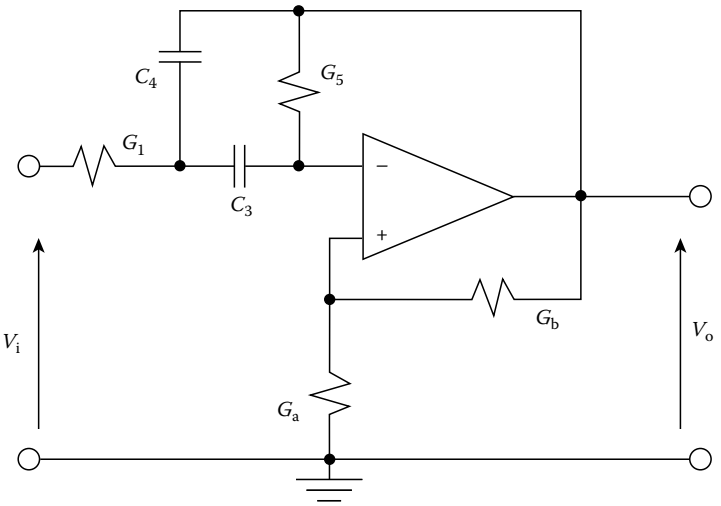


FIGURE 12.4 MMFB structure.

The design process consists of matching coefficients in Equation 12.23 with those of the standard bandpass expression of Equation 12.10. The design steps have been conveniently summarized by Huelsman [8] for the equal-capacitor solution and the following procedure is essentially the same as that described by him.

Example 12.3

Design a second-order bandpass filter with a center frequency of 1 kHz, a pole-Q of 8, and a maximum resistance spread of 50. Assume that the only available capacitors are of value 6800 pF.

1. The above constraint suggests an equal-valued capacitor solution. Thus, set $C_3 = C_4 = C$.
2. Determine the resistance ratio parameter n_o that would be required if there were no positive feedback. From Section 12.4, $n_o = 4Q_p^2 = 256$.
3. Select the desired ratio n (where n is greater than 1 but less than 256) and use it to determine the amount of positive feedback k. From Equation 12.24,

$$Q_p = \frac{\sqrt{n}}{2 - kn}$$

so that

$$k = \frac{1}{\sqrt{n}}\left\{\frac{2}{\sqrt{n}} - \frac{1}{Q_p}\right\}$$

Since $n = 50$ and $Q_p = 8$, $k = 0.0316$.

4. A convenient value may now be selected for R_B. If $R_B = 110\,\text{k}\boldsymbol{\Omega}$, then $R_A = R_B\,(0.0316) = 3.48\,\text{k}\boldsymbol{\Omega}$.
5. Since, from Equation 12.23,

$$\boldsymbol{\omega}_p = \sqrt{\frac{G_1 G_5}{C_3 C_4}}$$

and $G_1/G_5 = n$, we may determine G_5 as

$$G_5 = \frac{\boldsymbol{\omega}_p C}{\sqrt{n}}$$

Since $C = 6800$ pF, $n = 50$, and $G_5 = 1/R_5$:

$$R_5 = \frac{\sqrt{50}}{2\boldsymbol{\pi}10^3 \times 6.8 \times 10^{-9}} = 165.5\,\text{k}\boldsymbol{\Omega}$$

Hence $R_1 = R_5/n = 3.31\,\text{k}\boldsymbol{\Omega}$.

6. Using 1% preferred resistor values we have

$$R_B = 110\,\text{k}\boldsymbol{\Omega}; \quad R_A = 3.16\,\text{k}\boldsymbol{\Omega}$$
$$R_5 = 165\,\text{k}\boldsymbol{\Omega}; \quad R_1 = 3.48\,\text{k}\boldsymbol{\Omega}$$

Judicious use of positive feedback in the Deliyannis circuit can yield bandpass filters with Q values as high as 15–20 at modest center frequencies. A more detailed discussion of the optimization of this structure may be found elsewhere [9].

12.7 Biquadratic MMFB Structure

A generalization of the MMFB arrangement, yielding a fully biquadratic transfer ratio is shown in Figure 12.5. If the gain functions K_1, K_2, K_3 are realized by resistive potential dividers, the circuit reduces to the more familiar Friend biquad of Figure 12.6, for which

$$\frac{V_o}{V_i} = \frac{cs^2 + ds + e}{s^2 + as + b} \tag{12.25}$$

where

$$\begin{aligned} K_1 &= \frac{R_5}{R_4 + R_5}; \quad K_2 = \frac{R_D}{R_c + R_D}; \quad K_3 = \frac{R_7}{R_6 + R_7} \\ R_1 &= \frac{R_4 R_5}{R_4 + R_5}; \quad R_A = \frac{R_c R_D}{R_c + R_D}; \quad R_3 = \frac{R_6 R_7}{R_6 + R_7} \end{aligned} \tag{12.26}$$

This structure is capable of yielding a full range of biquads of modest pole Q, including notch functions derived as elliptic characteristics of low modular angle. It was used extensively in the Bell System, where the benefits of large-scale manufacture were possible. Using the standard tantalum thin film process, and deterministic tuning by means of laser trimming, quite exacting realizations were possible [10].

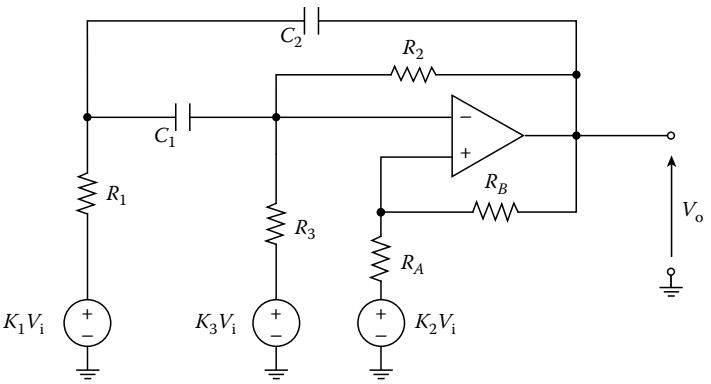


FIGURE 12.5 Generalization of the MMFB circuit.

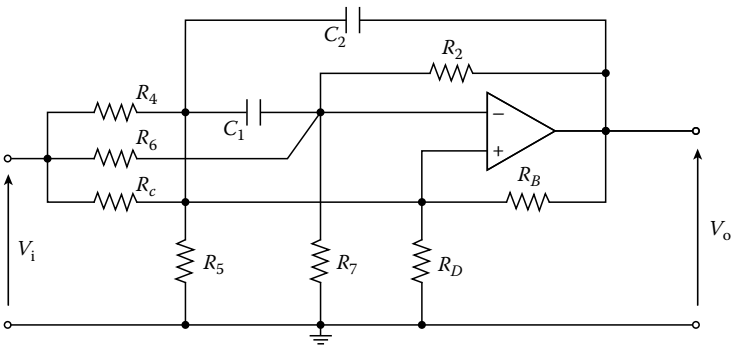


FIGURE 12.6 Friend biquad.

The structure is less suited to discrete component realizations. However, design is possible by coefficient matching. The reader is referred to an excellent step-by-step procedure developed by Huelsman [11].

12.8 Conclusions

The multiple-feedback structure is one of the most basic active filter building blocks. It is extremely reliable when used to realize low-Q (<10), low frequency (up to 15 kHz) second-order sections of the low-pass, bandpass, and high-pass forms. Stability is ensured by the negative feedback topology, though component spreads are proportional to Q^2.

The disadvantage of larger component spread may be reduced by the judicious use of positive feedback. This approach may be extended to yield the widely used Friend biquad, which allows the realization of notch and other approximations requiring a pair of imaginary zeros.

All networks described in this section readily lend themselves to the cascade method for realizing higher-order filters.

References

1. G. K. Aggarwal, On *n*th order simulation by one operational amplifier, *Proc. IEEE*, 52, 969, 1969.
2. P. L. Taylor, Flexible design method for active *RC* two-ports, *Proc. IEE*, 110, 1607–1616, 1963.
3. A. G. J. Holt and J. I. Sewell, Active *RC* filters employing a single operational amplifier to obtain biquadratic responses, *Proc. IEE*, 112, 2227–2234, 1965.
4. P. Bowron and F. W. Stephenson, *Active Filters for Communications and Instrumentation*, Berkshire, England: McGraw-Hill, 1979, p. 170.
5. R. A. Friedenson, R. W. Daniels, R. J. Dow, and P. H. McDonald, *RC* active filters for the *D*3 channel bank filter, *Bell Syst. Tech. J.*, 54(3), 507–529, 1975.
6. A. B. Grebene and H. R. Camenzind, Frequency-selective integrated circuits using phase-locked techniques, *IEEE J. Solid-State Circuits*, SC-4, 216–225, Aug. 1969.
7. T. Deliyannis, High-Q factor circuit with reduced sensitivity, *Electron. Lett.*, 4(26), 577–579, 1968.
8. L. P. Huelsman, *Active and Passive Filter Design*, New York: McGraw-Hill, 1993, ch. 5, pp. 277–278.
9. M. S. Ghausi and K. R. Laker, *Modern Filter Design*, Englewood Cliffs, NJ: Prentice Hall, 1981, ch. 4, pp. 197–201.
10. J. J. Friend, C. A. Harris, and D. Hilberman, STAR: An active biquadratic filter section, *IEEE Trans. Circuits Syst.*, CAS-22, 115–121, Feb. 1975.
11. L. P. Huelsman, Multiple-loop feedback filters, in *RC Active Filter Design Handbook* (F. W. Stephenson, Ed.), New York: Wiley, 1985, ch. 7, pp. 201–203.

13

Multiple-Amplifier Biquads

13.1 Introduction **13**-1
13.2 Biquads with Decoupled Time Constants **13**-2
13.3 Dual-Amplifier Twin-*T* Biquads **13**-5
13.4 GIC-Derived Dual-Amplifier Biquads **13**-9
13.5 GIC-Derived Three-Amplifier Biquads **13**-18
13.6 State-Variable-Based Biquads **13**-20
13.7 All-Pass-Based Biquads **13**-25
13.8 Summary **13**-26
References **13**-27

Norbert J. Fliege
University of Mannheim

13.1 Introduction

The step from single-amplifier to multiple-amplifier biquadratic filter sections provides several benefits. The most important benefits are as follows:

- Reduced passive element spread, i.e., the ratio between the largest and the smallest values of resistors and/or capacitors can be reduced compared to the single-amplifier case.
- The required amplifier gains in some circuits grow linearly or less with the *Q*-factor of the complex pole pairs.
- Multiple-amplifier biquads often provide lower sensitivities to both passive and active components.
- Most of the multiple-amplifier biquads are more universal filter structures realizing the general biquadratic transfer function.
- Most of the filter parameters such as pole and zero *Q*-factors, pole and zero frequencies, and the gain factor of the transfer function can be tuned independently.
- Designing multiple-amplifier filters, often the values of the capacitors can be chosen freely. The filter parameters are then determined by resistors, which is less costly than by capacitors.

On the other hand, these benefits must be paid for by increased space requirements and increased power dissipation. However, today there are low-cost and low-power integrated-circuit op-amps available with up to four op-amps on one chip. Therefore, size and power dissipation are often no longer the main problem.

In the following, we will first consider biquadratic filter sections and dual-amplifier twin-*T* biquads. Both circuit families are directly derived from single-amplifier circuits. Next we will derive filter circuits having a quite different origin: filters that are derived from the generalized impedance converter (GIC) and filters derived from state-variable representation of linear systems on the analog computer. Finally, we will briefly consider filter circuits based on first-order all-pass sections.

13.2 Biquads with Decoupled Time Constants

One of the simplest methods for improving the performance of a biquad is demonstrated in Figure 13.1.

Figure 13.1a shows a well-known Sallen and Key bandpass circuit [1], which offers moderate pole sensitivities with respect to the passive elements. It can easily be shown that this circuit has the following transfer voltage ratio:

$$H(s) = \frac{V_o}{V_i} = \frac{-K \cdot sT_2}{s^2 T_1 T_2 (1 + K) + s(T_{12} + T_1 + T_2) + 1} \tag{13.1}$$

with the amplifier gain K and the time constants $T_1 = R_1C_1$, $T_2 = R_2C_2$, and $T_{12} = R_1C_2$. The gain requirement is more than $4Q_p^2 - 1$ with Q_p being the Q-factor of the pole pair. This relationship limits the circuit to low or medium-Q applications.

If we insert another amplifier between the two RC networks R_1C_1 and R_2C_2, we obtain the biquad in Figure 13.1b [2,3], which possesses a transfer voltage ratio

$$H(s) = \frac{V_o}{V_i} = \frac{K_1K_2 \cdot sT_2}{s^2 T_1 T_2 (1 - K_1K_2) + s(T_1 + T_2) + 1} \tag{13.2}$$

Here the product K_1K_2 of the gain factors plays the role of the gain $-K$ in Figure 13.1a. One of the gain factors must be positive, the other negative. A comparison of Equations 13.1 and 13.2 shows that after isolating both RC networks the "cross time constant" T_{12} disappears. The two time constants T_1 and T_2 are decoupled. From this change we can derive two benefits. Both factors K_1 and K_2 require only a gain of approximately $2Q_p$ and, as will be shown later, this circuit can be designed with zero Q-sensitivity.

Next, we generalize the structure in Figure 13.1b by replacing the passive elements by general admittances; see Figure 13.2. In this circuit, the two subnetworks Y_{1a}, Y_{1b} and Y_{2a}, Y_{2b} are decoupled.

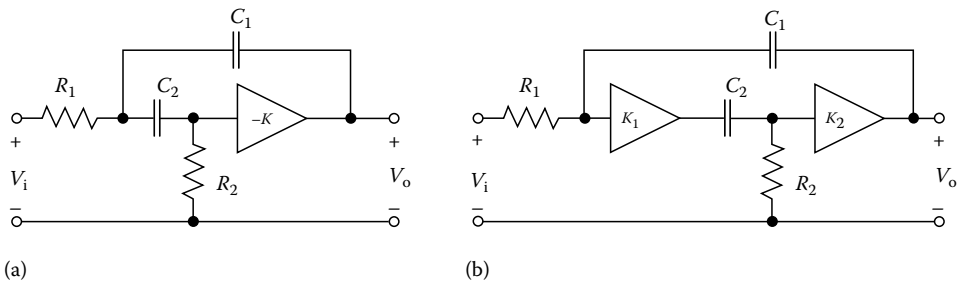


FIGURE 13.1 (a) Sallen and Key bandpass filter and (b) dual-amplifier bandpass filter with decoupled time constants.

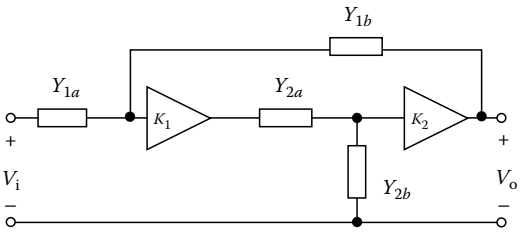


FIGURE 13.2 General biquad with decoupled networks.

By a simple analysis we obtain the transfer voltage ratio of the general circuit in Figure 13.2:

$$H(s) = \frac{V_o}{V_i} = \frac{K_1 K_2 Y_{1a} Y_{2a}}{(Y_{2a} + Y_{2b})(Y_{1a} + Y_{1b}) - K_1 K_2 Y_{1b} Y_{2a}} \tag{13.3}$$

By prespecifying the types of passive elements in Figure 13.2 and Equation 13.3, respectively, we next will derive a low-pass, a bandpass, and high-pass filter section.

With the prespecified elements $Y_{1a} = G_1$, $Y_{1b} = sC_1$, $Y_{2a} = G_2$, and $Y_{2b} = sC_2$ we obtain from Equation 13.3 the low-pass transfer function:

$$\begin{aligned} H_{LP}(s) = \frac{V_o}{V_i} &= \frac{K_1 K_2 G_1 G_2}{s^2 C_1 C_2 + s[G_1 C_2 + G_2 C_1 (1 - K_1 K_2)] + G_1 G_2} \\ &= \frac{K_1 K_2}{s^2 T_1 T_2 + s[T_2 + T_1 (1 - K_1 K_2)] + 1} \end{aligned} \tag{13.4}$$

If we predefine $K_1 = K_2 = 1$ we simply get voltage followers in the filter circuit and additionally a simple design procedure resulting in a low-sensitivity filter section. Figure 13.3 shows the low-pass filter section with two op-amps and four passive elements.

For $K_1 = K_2 = 1$ we will have dc gain $H_0 = 1$, a pole frequency

$$\omega_p = 1/\sqrt{T_1 T_2} \tag{13.5}$$

and a Q-factor

$$Q_p = \sqrt{T_2/T_1} \tag{13.6}$$

Thus, designing the filter section, from the predefined parameters ω_p and Q_p we determine the time constants

$$T_2 = \frac{Q_p}{\omega_p}, \quad T_1 = \frac{1}{Q_p \omega_p} \tag{13.7}$$

Finally, we choose the values of the capacitors and calculate the resistors from the time constants.

From Equations 13.5 and 13.6, we can immediately read the pole sensitivities:

$$S_{R_1}^{\omega_p} = S_{C_1}^{\omega_p} = S_{R_2}^{\omega_p} = S_{C_2}^{\omega_p} = -\frac{1}{2} \tag{13.8}$$

$$S_{R_1}^{Q_p} = S_{C_1}^{Q_p} = -S_{R_2}^{Q_p} = -S_{C_2}^{Q_p} = -\frac{1}{2} \tag{13.9}$$

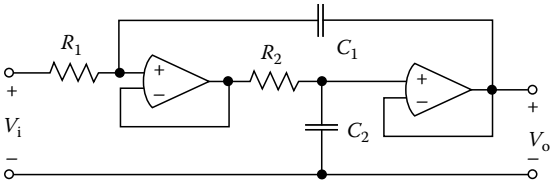


FIGURE 13.3 Dual op-amp low-pass filter section.

This result is comparable with the pole sensitivities of passive second-order *RLC* networks.

Next we will develop a high-pass filter section by prespecifying $Y_{1a} = sC_1$, $Y_{1b} = G_1$, $Y_{2a} = sC_2$, and $Y_{2b} = G_2$ resulting in a transfer voltage ratio:

$$H_{\mathrm{HP}}(s) = \frac{V_\mathrm{o}}{V_\mathrm{i}} = \frac{s^2 K_1 K_2 C_1 C_2}{s^2 C_1 C_2 + s[G_2 C_1 + G_1 C_2(1 - K_1 K_2)] + G_1 G_2}$$
$$= \frac{s^2 K_1 K_2 T_1 T_2}{s^2 T_1 T_2 + s[T_1 + T_2(1 - K_1 K_2)] + 1} \tag{13.10}$$

The only difference of this transfer function compared with the low-pass transfer function in Equation 13.4 is the high-pass term (with s^2) in the numerator and the fact that the time constants T_1 and T_2 are interchanged in the denominator polynomial. Thus we can transfer the results in Equation 13.5 through 13.9 to the high-pass filter by only replacing the index 1 by 2 and vice versa. Figure 13.4 shows the high-pass filter section.

The bandpass filter circuit mentioned at the beginning of this section is defined by the following types of elements: $Y_{1a} = G_1$, $Y_{1b} = sC_1$, $Y_{2a} = sC_2$, and $Y_{2b} = G_2$. Its transfer function is written in Equation 13.2. The bandpass design is somewhat different from that of the low-pass and the high-pass filters described above. If we set the design values

$$G_1 = G_2 = 1, \quad C_1 = C_2 = \frac{1}{2Q_\mathrm{p}} \tag{13.11}$$

and

$$-K_1 K_2 = 4Q_\mathrm{p}^2 - 1 \tag{13.12}$$

we obtain a special design where all passive Q-sensitivities are zero and all other sensitivities are very low [2,3]:

$$S_{R_1}^{Q_\mathrm{p}} = S_{C_1}^{Q_\mathrm{p}} = S_{R_2}^{Q_\mathrm{p}} = S_{C_2}^{Q_\mathrm{p}} = 0 \tag{13.13}$$

$$S_{R_1}^{\omega_\mathrm{p}} = S_{C_1}^{\omega_\mathrm{p}} = S_{R_2}^{\omega_\mathrm{p}} = S_{C_2}^{\omega_\mathrm{p}} = -\frac{1}{2} \tag{13.14}$$

$$S_{K_1}^{Q_\mathrm{p}} = S_{K_2}^{Q_\mathrm{p}} = \frac{-K_1 K_2}{2(1 - K_1 K_2)} < \frac{1}{2} \tag{13.15}$$

$$S_{K_1}^{\omega_\mathrm{p}} = S_{K_2}^{\omega_\mathrm{p}} = -\frac{1}{2}\left(1 - \frac{1}{4Q_\mathrm{p}^2}\right) \tag{13.16}$$

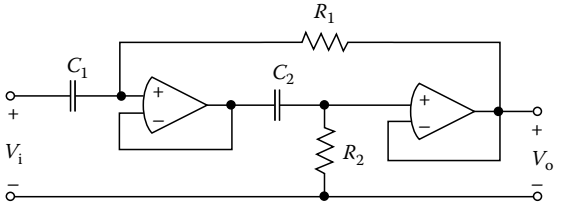


FIGURE 13.4 Dual op-amp high-pass filter section.

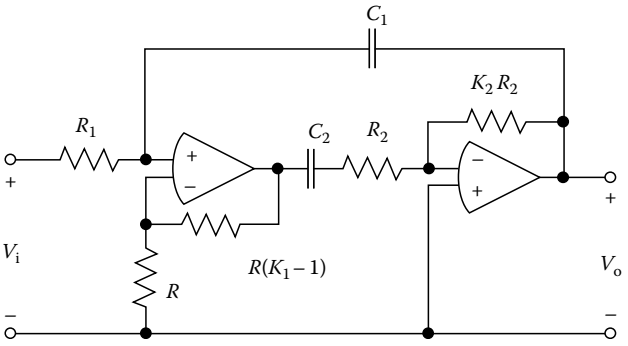


FIGURE 13.5 Dual op-amp bandpass filter section.

Figure 13.5 shows the bandpass filter section realized with two op-amps. In this circuit K_1 is a noninverting and K_2 an inverting amplifier. The resistor R_2 of the passive network R_2, C_2 is realized by the input resistor of the inverting amplifier.

13.3 Dual-Amplifier Twin-*T* Biquads

Twin-*T* feedback networks are easily tunable and provide relatively favorable sensitivity properties. In active filters, a twin-*T* network is connected between the input and the output of an inverting amplifier. It has been shown [4] that the sensitivity to the active element can be substantially reduced when a symmetrical feedback network is used, i.e., when the dc gain of the feedback network is equal to the high-frequency gain.

In a single-amplifier twin-*T* biquad, the output port of the feedback network is loaded by the input resistance of the inverting amplifier. As a consequence, the feedback network is no longer symmetrical. Here again we can significantly improve the behavior of the twin-*T* biquad by introducing a second amplifier: inserting a voltage follower between the twin-*T* network and the inverting amplifier maintains the symmetry of the feedback network. Figure 13.6 shows the corresponding dual-amp twin-*T* resonator* with the usually chosen passive element relations.

From the resonator in Figure 31.6 several filter circuits can be derived by inserting the input voltage source in one of the grounded branches and by taking one of the two amplifier outputs as output terminal

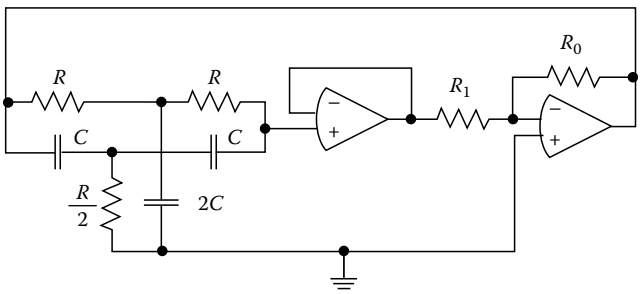


FIGURE 13.6 Dual-amp twin-*T* resonator.

* A resonator is a circuit without input and output terminals that only serves to show the feedback mechanism.

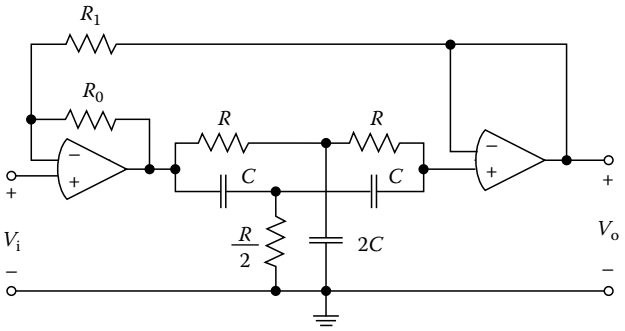


FIGURE 13.7 Dual-amp twin-T band-rejection filter.

of the filter section. The voltage transfer ratios of all these circuits will have different numerator polynomials but the same denominator polynomial. As an example, Figure 31.7 shows a band-rejection filter derived from the dual-amp twin-T resonator in Figure 13.6.

The voltage transfer ratio of the band-rejection filter in Figure 13.7 can be calculated to be

$$H_{BR}(s) = \frac{V_o}{V_i} = \frac{s^2T^2 + 1}{s^2T^2 + s4T/K + 1} \tag{13.17}$$

with the time constant $T = RC$ and the gain $K = (R_0 + R_1)/R_1$ of the series connection of the two op-amp circuits. Designing this filter, we determine the time constant T from the notch frequency ω_z or pole frequency ω_p, respectively,

$$T = \frac{1}{\omega_z} = \frac{1}{\omega_p} \tag{13.18}$$

and the amplifier gain from the Q-factor:

$$K = 4 \cdot Q_p \tag{13.19}$$

In Ref. [4], it is shown that the filter has favorably low gain-sensitivity products.

In order to obtain further filter variants, in Ref. [4] a complementary circuit is derived from the resonator in Figure 13.6; see Figure 13.8.

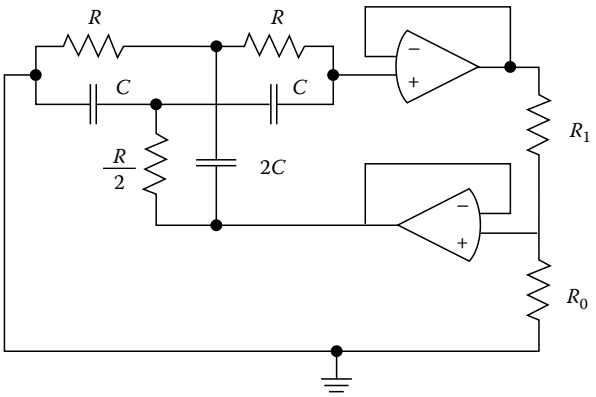


FIGURE 13.8 Complementary circuit of the resonator in Figure 13.6.

This resonator is especially useful for deriving filter sections with frequently applied transfer functions. In Figures 13.9 through 13.13 some of these filter sections are depicted. Their voltage transfer ratios are given by

$$H(s) = \frac{V_o}{V_i} = \frac{N(s)}{s^2T^2 + s4T/K + 1} \tag{13.20}$$

The band-rejection filter in Figure 13.9 has a numerator polynomial $N(S) = s^2T^2 + 1$. The filter section in Figure 13.10 is low-pass with a numerator $N(s) = 1$, the filter section in Figure 13.11 is high-pass

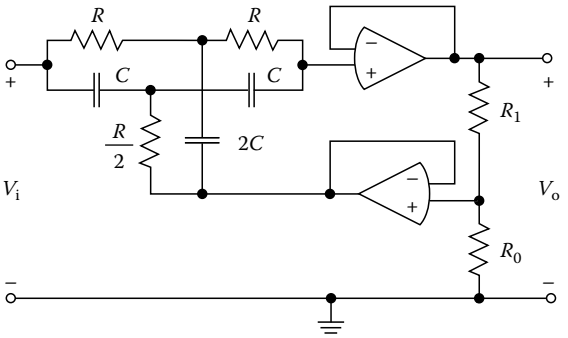


FIGURE 13.9 Twin-*T* band-rejection filter.

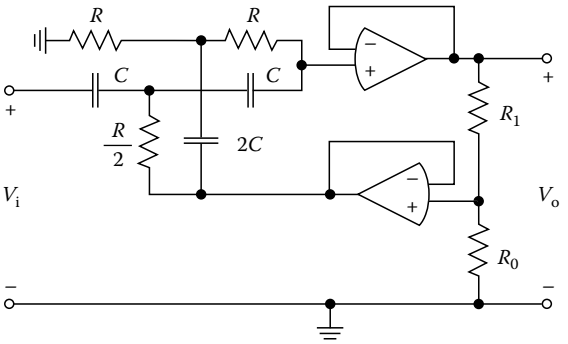


FIGURE 13.10 Twin-*T* low-pass filter.

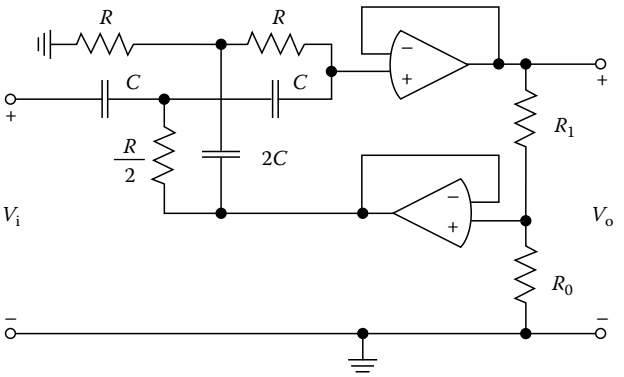


FIGURE 13.11 Twin-*T* high-pass filter.

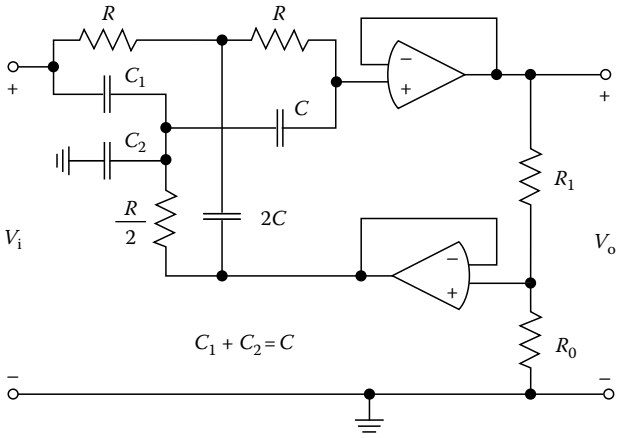


FIGURE 13.12 Twin-*T* Cauer low-pass filter section.

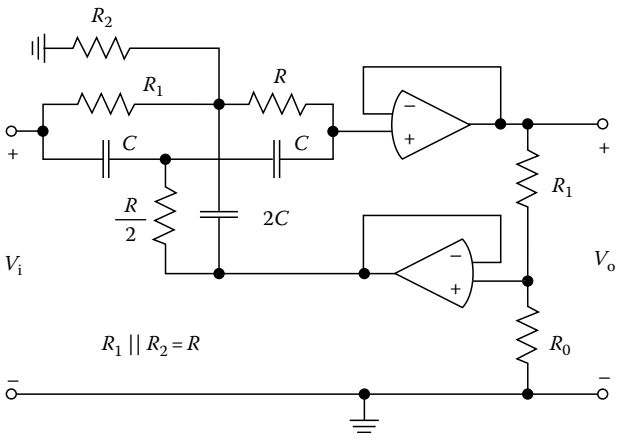


FIGURE 13.13 Twin-*T* Cauer high-pass filter section.

with $N(s) = s^2T^2$. The design of these sections is exactly the same as that of the filter in Figure 13.7. It also provides the same pole sensitivities.

The two structures in Figure 13.12 and 13.13 are particularly suitable for realizing elliptical filter sections (also called Cauer filters). The Cauer low-pass filter section in Figure 13.12 has a numerator polynomial:

$$N(s) = s^2R^2CC_1 + 1, \quad C_1 + C_2 = C \tag{13.21}$$

In this circuit, the first capacitor C is split into a parallel connection of two capacitors C_1 and C_2, where C_2 is grounded. The zeros are on the $j\omega$ axis at positions $\pm j\omega_z$ with $\omega_z = 1/\sqrt{R^2CC_1}$. The notch frequency ω_z is greater than the pole frequency ω_p. Designing the filter section, we determine the parameters T and K as described above. The splitting ratio of the input capacitor is determined by the poles and zeros:

$$\frac{C_2}{C_1} = \frac{\omega_z^2}{\omega_p^2} - 1 \tag{13.22}$$

Finally, in Figure 13.13 we have a Cauer high-pass filter section with a numerator polynomial:

$$N(s) = s^2 RR_1C^2 + 1, \quad R_1 \| R_2 = R \tag{13.23}$$

Here the zero frequency ω_z is less than the pole frequency ω_p. As in the low-pass case, the splitting ratio of the input resistors is determined by these two frequencies:

$$\frac{R_1}{R_2} = \frac{\omega_p^2}{\omega_z^2} - 1 \tag{13.24}$$

The circuit in Figure 13.6 is actually of third order. By matching the passive components, as shown in Figure 13.6, we obtain a second-order transfer function. The question may arise as to what happens if we have a small mismatch due to the tolerances of practical components. In this case, a third pole and a third zero appear on the negative real axis of the s-plane. They do not cancel each other exactly, but approximately. In general, the existence of the third pole and the third zero does not affect the frequency response of the filter significantly. However, there is a second effect due to a small mismatch of passive components that is more severe: the position of the pole pair desired by design is changed. This change can be estimated by the pole sensitivities with respect to the passive elements [4].

13.4 GIC-Derived Dual-Amplifier Biquads

In this section, we consider a class of biquadratic building blocks with two op-amps that are derived from the generalized impedance converter (GIC). A catalog of such building blocks realizing a wide variety of network functions, including elliptic and all-pass ones, was published by Fliege [5].

Figure 13.14b shows the general filter structure, which is based on the resonator with two nullors in Figure 13.14a. Each nullor consists of one nullator and one norator and constitutes a model for the ideal op-amp. Combining the norator between the node between the admittances Y_2 and Y_3 and ground and the nullator across Y_4 and Y_6 yields the op-amp μ_1 in Figure 13.14b. It can readily be verified that the voltage transfer ratio of the circuit in Figure 13.14b with $\mu_i = \infty$, $i = 1, 2$ is given by

$$H(s) = \frac{V_o}{V_i} = \frac{Y_{6b}(Y_2Y_4 + Y_{1a}Y_4) + Y_{1b}(Y_3Y_5 - Y_{6a}Y_4)}{Y_1Y_3Y_5 + Y_2Y_4Y_6} \tag{13.25}$$

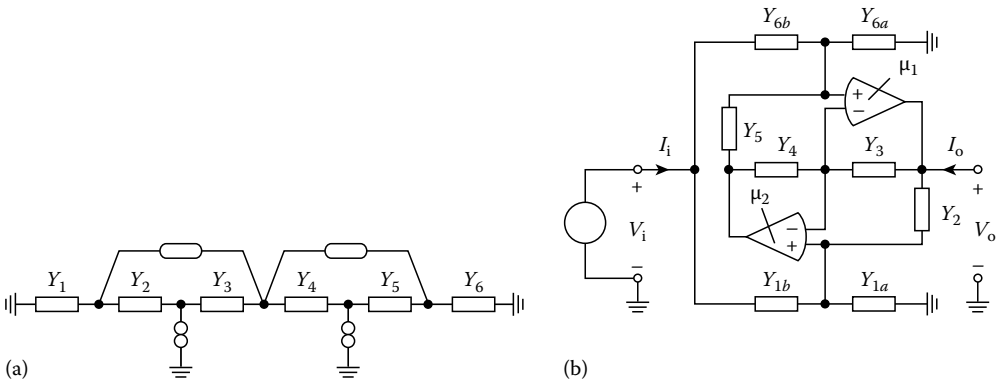


FIGURE 13.14 Resonator with (a) two nullors and (b) the corresponding general dual-amplifier filter structure.

with

$$Y_1 = Y_{1a} + Y_{1b}, \quad Y_6 = Y_{6a} + Y_{6b} \tag{13.26}$$

If we choose the node between Y_4 and Y_5 as the output of the building block we will obtain a similar transfer function. We only have to interchange the admittances Y_1 by Y_6, Y_2 by Y_5, and Y_3 by Y_4.

First, we will derive a second-order low-pass building block from the general structure and we will take this low-pass filter as a prototype for the whole circuit family to explain their advantageous characteristics. If we choose the passive elements of the building block as $Y_{1a} = G_1$, $Y_{1b} = 0$, $Y_2 = G_2$, $Y_3 = sC_3$, $Y_4 = G_4$, $Y_5 = G_5 + sC_5$, $Y_{6a} = 0$, and $Y_{6b} = G_6$ we obtain the low-pass building block in Figure 13.15. Its voltage transfer ratio reads

$$\begin{aligned} H_{\mathrm{LP}}(s) = \frac{V_\mathrm{o}}{V_\mathrm{i}} &= \frac{G_6 G_4 (G_2 + G_1)}{s^2 C_3 C_5 G_1 + s C_3 G_5 G_1 + G_2 G_4 G_6} \\ &= H_0 \cdot \frac{\omega_\mathrm{p}^2}{s^2 + s\omega_\mathrm{p}/Q_\mathrm{p} + \omega_\mathrm{p}^2} \end{aligned} \tag{13.27}$$

with dc gain

$$H_0 = \frac{1 + \alpha_{21}}{\alpha_{21}} \tag{13.28}$$

pole frequency

$$\omega_\mathrm{p} = \sqrt{\frac{\alpha_{21}}{T_{34} T_{56}}} \tag{13.29}$$

and Q-factor

$$Q_\mathrm{p} = \omega_\mathrm{p} \cdot T_{55} \tag{13.30}$$

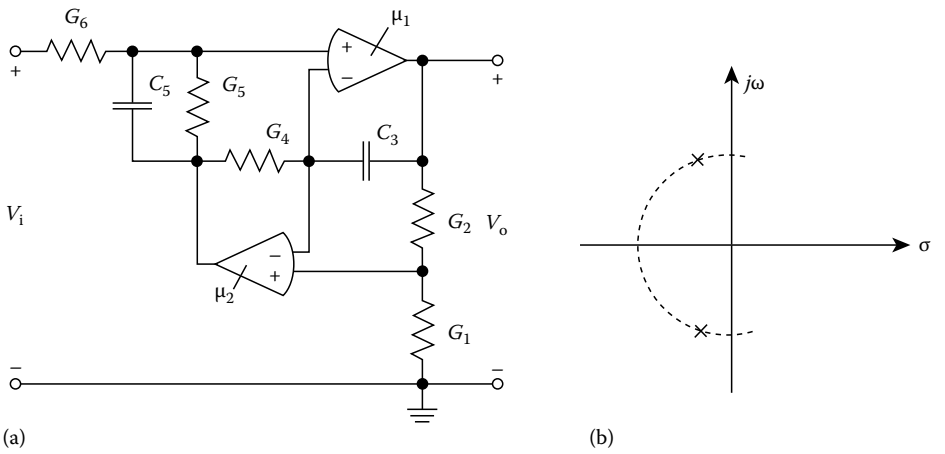


FIGURE 13.15 Second-order low-pass building block. (a) The circuitry. (b) The pole locations.

Here, the time constants are defined as

$$T_{\mu\nu} = \frac{C_\mu}{G_\nu} \tag{13.31}$$

and the conductance ratios as

$$\alpha_{\mu\nu} = \frac{G_\mu}{G_\nu} \tag{13.32}$$

Figure 13.15b shows the location of the finite poles and zeros in the *s*-plane.

In the following, we predefine $G_1 = G_2$, i.e., $\alpha_{21} = 1$, and $T_{34} = T_{56} = T$. This provides optimum pole sensitivities with respect to the passive elements and with respect to the gain-bandwidth products of the operational amplifiers. Hence, the gain factor H_0 always has the value 2.

Based on these parameter restrictions, we can establish an easy design procedure. First, we can choose the capacitors $C_3 = C_5 = C$. Then, given the pole parameters ω_p and Q_p, we determine the two resistors

$$R_4 = R_6 = R = \frac{1}{\omega_p \cdot C} \tag{13.33}$$

and from Equation 13.30 the resistor

$$R_5 = Q_p \cdot R \tag{13.34}$$

There are three key features that make these dual-amplifier biquads particularly favorable for practical applications.

- Biquad building blocks have a low spread of elements. The resistors $R_1 = R_2$ can be chosen freely. The same holds for the two capacitors $C_3 = C_5$. The frequency-determining resistors $R_4 = R_6$ have equal values, too. There is only the *Q*-determining resistor R_5, which differs by a factor of Q_p.
- Relative sensitivities of the gain and the pole parameters with respect to the passive elements are of the same order of magnitude as in case of a second-order passive *RLC* network; see Equation 13.35.
- Impact of the op-amp gain and gain-bandwidth product on the pole parameter is extremely low.

A sensitivity analysis of the biquad circuit in Figure 13.15 gives the following results:

$$\begin{aligned}
&S_{R_1}^{H_0} = -S_{R_2}^{H_0} = -\frac{1}{2}s \\
&S_{R_1}^{\omega_p} = -S_{R_2}^{\omega_p} = -S_{C_3}^{\omega_p} = -S_{R_4}^{\omega_p} = -S_{C_5}^{\omega_p} = -S_{R_6}^{\omega_p} = \frac{1}{2} \\
&S_{R_1}^{Q_p} = -S_{R_2}^{Q_p} = -S_{C_3}^{Q_p} = -S_{R_4}^{Q_p} = +S_{C_5}^{Q_p} = -S_{R_6}^{Q_p} = \frac{1}{2} \\
&S_{R_5}^{Q_p} = 1.
\end{aligned} \tag{13.35}$$

In most practical applications, we can describe the op-amp dynamics by a one-pole model:

$$\mu(s) = \frac{\mu_0}{1 + (s/\omega_c)} = \frac{1}{(1/\mu_0) + (s/\omega_T)} \tag{13.36}$$

with the dc gain μ_0, the 3-dB cutoff frequency ω_c, and the gain-bandwidth product $\omega_T = \mu_0\omega_c = 1/T_T$.

If the circuit elements are chosen properly, e.g., choosing network parameters $\alpha_{21} = 1$ and $T_{34} = T_{56} = T = 1/\omega_p$, an analysis of the impact of the parameters μ_{0i} and $T_{Ti} = 1/\omega_{Ti}$ of the two op-amps μ_i, $i = 1, 2$, on the pole parameters of the transfer function yields

$$\frac{\Delta\omega_p}{\omega_p} \approx \frac{1}{\mu_{01}} - \frac{1}{\mu_{02}} - \frac{T_{T1} + T_{T2}}{T} \tag{13.37}$$

$$\frac{\Delta Q_p}{Q_p} \approx -2Q_p\left(\frac{1}{\mu_{01}} + \frac{1}{\mu_{02}} + \frac{T_{T1} - T_{T2}}{T}\right) \tag{13.38}$$

In case of high pole frequencies the impact of the dc gains μ_{01} and μ_{02} can be neglected against that of the gain-bandwidth products. From Equation 13.37, after some intermediate steps, we obtain

$$\omega_p + \Delta\omega_p \approx \frac{1}{T + T_{T1} + T_{T2}} \tag{13.39}$$

In case of an ideal op-amp we have $\omega_P = 1/T$. The nonideal op-amp causes a pole frequency change $\Delta\omega_p$ due to the time constants T_{Ti}. The frequency determining time constant T of the passive network is increased by the two time constants T_{Ti} of the op-amps, which are reciprocals of the gain-bandwidth products ω_{Ti}.

The most interesting result is the Q-factor change in Equation 13.38. If the frequency responses of the two op-amps are matched, as is usually the case with dual packages, the impact of the two gain-bandwidth products cancels out. Thus, we have nearly no Q enhancement at higher pole frequencies.

To make the GIC-derived biquad almost independent of op-amp parameters, the above-mentioned conditions ($\alpha_{21} = 1$, $T_{34} = T_{56}$) must be met by the passive elements. In a more general view, this result holds for the whole family of GIC-derived building blocks. Each of these building blocks has two frequency-determining time constants and one resistive voltage divider (G_1 and G_2 or G_3 and G_4). In any case, to obtain independence of op-amp parameters we have to choose equal time constants and a voltage divider with equal resistors. If the time constants or the resistors, respectively, do not match exactly, the independence of op-amp parameters remains nearly unchanged.

Next we will derive a bandpass building block from the general biquad in Figure 13.14b. If we choose the passive elements as $Y_{1a} = G_1$, $Y_{1b} = 0$, $Y_2 = G_2$, $Y_3 = G_3$, $Y_4 = sC_4$, $Y_5 = G_5$, $Y_{6a} = sC_6$, and $Y_{6b} = G_6$, we obtain the bandpass building block in Figure 13.16, which has a voltage transfer ratio:

$$\begin{aligned} H_{BP}(s) = \frac{V_o}{V_i} &= \frac{G_6(sC_4G_2 + sC_4G_1)}{s^2C_4C_6G_2 + sC_4G_2G_6 + G_1G_3G_5} \\ &= H_0 \cdot \frac{s}{s^2 + s\omega_p/Q_p + \omega_p^2} \end{aligned} \tag{13.40}$$

with pole frequency

$$\omega_p = \sqrt{\alpha_{12}/(T_{43}T_{65})} \tag{13.41}$$

Q-factor

$$Q_p = \omega_p \cdot T_{66} \tag{13.42}$$

and midband gain

$$H_0Q_p = (1 + \alpha_{12}) \tag{13.43}$$

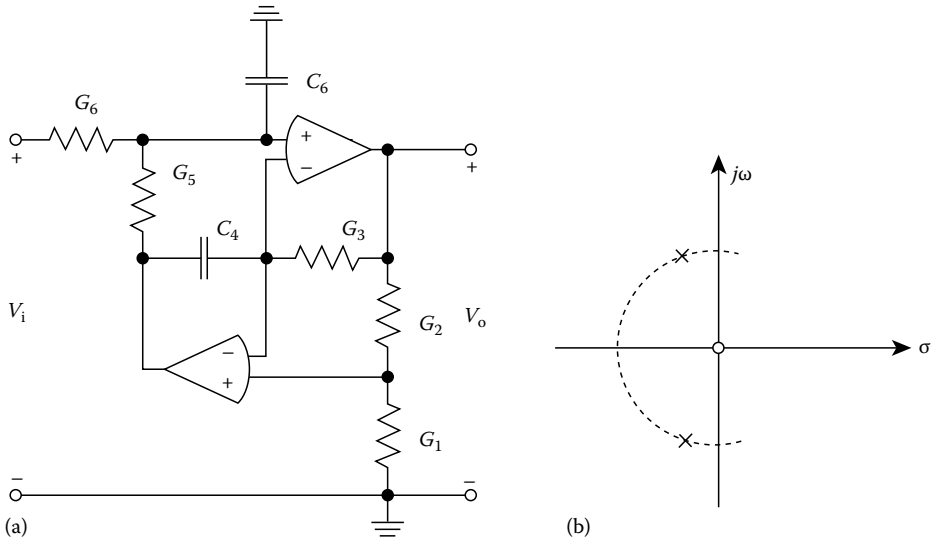


FIGURE 13.16 Second-order bandpass building block. (a) The circuitry. (b) The pole and zero locations.

The filter design is similar to that of the low-pass building block. First we choose the two resistors $R_1 = R_2$ and the two capacitors $C_4 = C_6 = C$. Then, from the pole frequency ω_p we determine the two resistors:

$$R_3 = R_5 = R = \frac{1}{\omega_p \cdot C} \tag{13.44}$$

and from the Q-factor Q_p the resistor

$$R_6 = Q_p \cdot R \tag{13.45}$$

The pole sensitivities with respect to the passive elements and the relationship between the gain and the gain-bandwidth product of the op-amps are the same as in case of the low-pass filters. It should be mentioned that the maximum of the magnitude response $|H(j\omega)|$ occurs at $\omega = \omega_p$ and has exactly the value 2, independent of the Q-factor and the time constants of the circuit*; see Equation 13.43.

We can derive a high-pass building block by choosing the following elements: $Y_{1a} = G_1$, $Y_{1b} = 0$, $Y_2 = G_2$, $Y_3 = G_3$, $Y_4 = sC_4$, $Y_5 = G_5$, $Y_{6a} = G_6$, and $Y_{6b} = sC_6$; see Figure 13.17.

This filter circuit has a voltage transfer ratio

$$\begin{aligned} H_{HP}(s) = \frac{V_o}{V_i} &= \frac{sC_6(sC_4G_2 + sC_4G_1)}{s^2C_4C_6G_2 + sC_4G_2G_6 + G_1G_3G_5} \\ &= (1 + \alpha_{12}) \cdot \frac{s^2}{s^2 + s\omega_p/Q_p + \omega_p^2} \end{aligned} \tag{13.46}$$

with ω_p as in Equation 13.41 and Q_p as in Equation 13.42.

The design of the high-pass building block in Figure 13.17 is identical to that of the bandpass described above. Both building blocks also have the same pole sensitivities and the same impact of the op-amps on the pole parameters.

* Only assuming $\alpha_{12} = 1$.

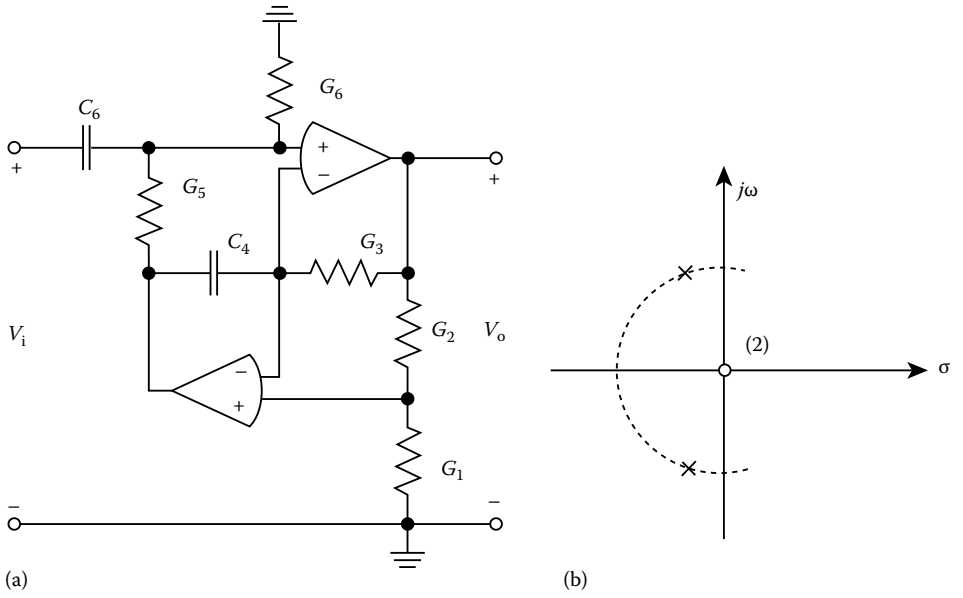


FIGURE 13.17 Second-order high-pass building block. (a) The circuitry. (b) The pole and zero locations.

If we feed the input signal not only through the capacitor C_6, but additionally through the element G_1, and if we set again $\alpha_{12} = 1$ we obtain the all-pass building block in Figure 13.18.

Its voltage transfer ratio reads

$$H_{AP}(s) = \frac{V_o}{V_i} = \frac{s^2 C_6 C_4 G_2 - sC_4 G_1 G_6 + G_1 G_3 G_5}{s^2 C_4 C_6 G_2 + sC_4 G_2 G_6 + G_1 G_3 G_5} = \frac{s^2 - \dfrac{s}{T_{66}} + \dfrac{1}{T_{43} T_{65}}}{s^2 + \dfrac{s}{T_{66}} + \dfrac{1}{T_{43} T_{65}}} \tag{13.47}$$

The all-pass building block is designed exactly as the high-pass or bandpass circuit.

From the all-pass building block, we can derive a second-order notch filter by adding a conductor G_{6b} in parallel with the capacitor G_6; see Figure 13.19.

The circuit in Figure 13.19 has the general voltage transfer ratio

$$H(s) = \frac{V_o}{V_i} = \frac{sC_6 C_4 G_2 + sC_4 (G_2 G_{6b} - G_1 G_{6a}) + G_1 G_3 G_5}{s^2 C_4 C_6 G_2 + sC_4 G_2 G_6 + G_1 G_3 G_5} \tag{13.48}$$

with $G_6 = G_{6a} + G_{6b}$. By setting $G_2 G_{6b} = G_1 G_{6a}$ (normally $G_1 = G_2$ and $G_{6a} = G_{6b}$) we obtain a second-order notch filter with a voltage transfer ratio:

$$H(s) = \frac{s^2 + \dfrac{\alpha_{12}}{T_{43} T_{65}}}{s^2 + \dfrac{s}{T_{66}} + \dfrac{\alpha_{12}}{T_{43} T_{65}}} \tag{13.49}$$

where $\alpha_{12} = 1$ if $G_1 = G_2$.

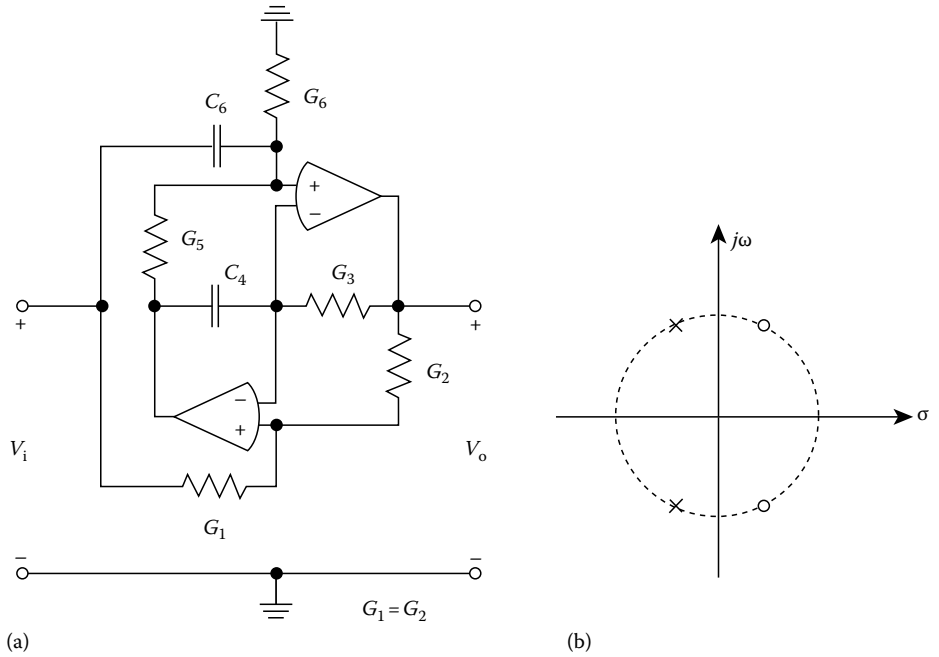


FIGURE 13.18 Second-order all-pass building block. (a) The circuitry. (b) The pole and zero locations.

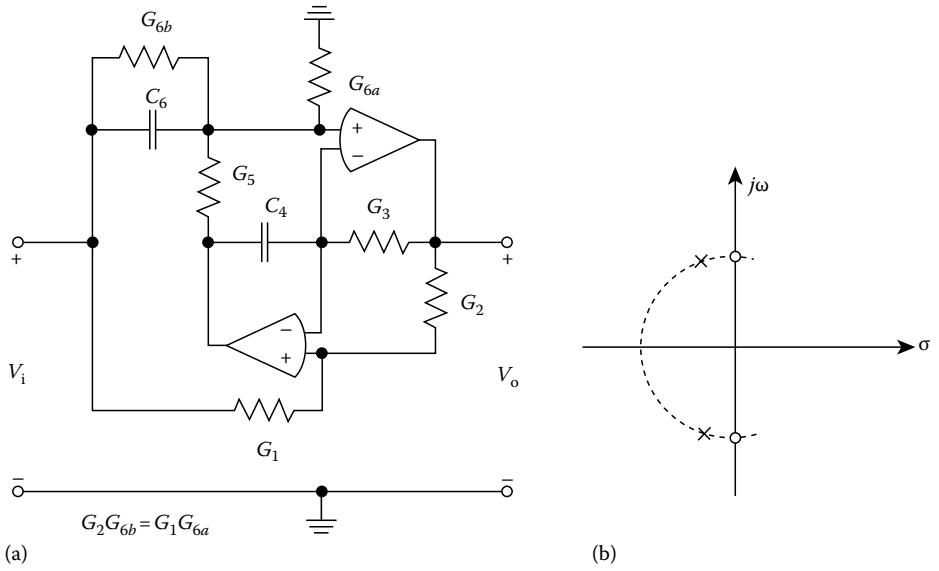


FIGURE 13.19 Notch filter building block. (a) The circuitry. (b) The pole and zero locations.

Finally we will derive two second-order elliptic or Cauer-type building blocks from the general dual-amplifier structure in Figure 13.14b. To get a Cauer low-pass building block we take $Y_{1a} = G_1$, $Y_{1b} = C_1$, $Y_2 = G_2$, $Y_3 = G_3$, $Y_4 = G_4$, $Y_5 = sC_5$, $Y_{6a} = 0$, and $Y_{6b} = G_6$; see Figure 13.20.

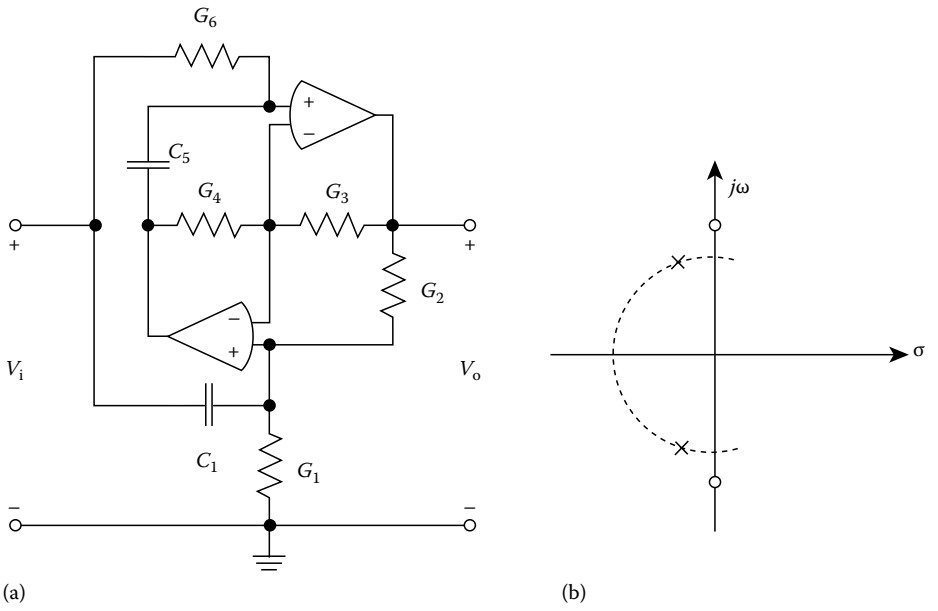


FIGURE 13.20 Second-order Cauer low-pass building block. (a) The circuitry. (b) The pole and zero locations.

This filter circuit has a voltage transfer ratio

$$\begin{aligned} H(s) = \frac{V_o}{V_i} &= \frac{s^2 C_1 G_3 C_5 + G_6(G_2 G_4 + G_1 G_4)}{s^2 C_1 G_3 C_5 + s G_1 G_3 C_5 + G_2 G_4 G_6} \\ &= \frac{s^2 + \dfrac{\alpha_{43}(1+\alpha_{12})}{T_{12} T_{56}}}{s^2 + \dfrac{s}{T_{11}} + \dfrac{\alpha_{43}}{T_{12} T_{56}}} \\ &= \frac{s^2 + \omega_z^2}{s^2 + s\omega_p / Q_p + \omega_p^2} \end{aligned} \tag{13.50}$$

The magnitude ω_z of the zeros is always greater than the magnitude ω_p of the poles. Additionally, without any matching of elements, the real part of the zeros is always zero.

The design can proceed in the following steps. First we predefine $\alpha_{43} = 1$ and choose due to practical considerations the values of the resistors $R_3 = R_4$. We also select the two capacitors C_1 and C_5. Then, comparing the first coefficient in the denominator of $H(s)$ in Equation 13.50, namely, $T_{11} = Q_p/\omega_p$ with $T_{11} = C_1 R_1$, we determine the resistor

$$R_1 = \frac{Q_p}{\omega_p C_1} \tag{13.51}$$

Further, comparing the last coefficients in the numerator and denominator yields

$$\frac{\omega_z^2}{\omega_p^2} = 1 + \alpha_{12} \tag{13.52}$$

With $\alpha_{12} = R_2/R_1$ we can solve Equation 13.52 for

$$R_2 = R_1 \cdot \left(\frac{\omega_z^2}{\omega_p^2} - 1 \right) \tag{13.53}$$

Finally, from $\omega_p^2 = 1/(T_{12}T_{56})$ we can determine the resistor

$$R_6 = \frac{1}{\omega_p^2 C_1 C_5 R_2} \tag{13.54}$$

In order to get the three resistors R_1, R_2, and R_6 in the same order of magnitude it might be advisable to predefine different values for the capacitors C_1 and C_5.

A Cauer high-pass building block is depicted in Figure 13.21. Its voltage transfer ratio reads

$$\begin{aligned} H(s) = \frac{V_o}{V_i} &= \frac{s^2 C_2 G_4 C_6 + G_1(G_3 G_5 - G_6 G_4)}{s^2 C_2 G_4 C_6 + s G_6 G_4 C_2 + G_1 G_3 G_5} \\ &= \frac{s^2 + \dfrac{\alpha_{34} - \alpha_{65}}{T_{21} T_{65}}}{s^2 + s\dfrac{1}{T_{11}} + \dfrac{\alpha_{43}}{T_{12} T_{56}}} \\ &= \frac{s^2 + \omega_z^2}{s^2 + s\omega_p/Q_p + \omega_p^2} \end{aligned} \tag{13.55}$$

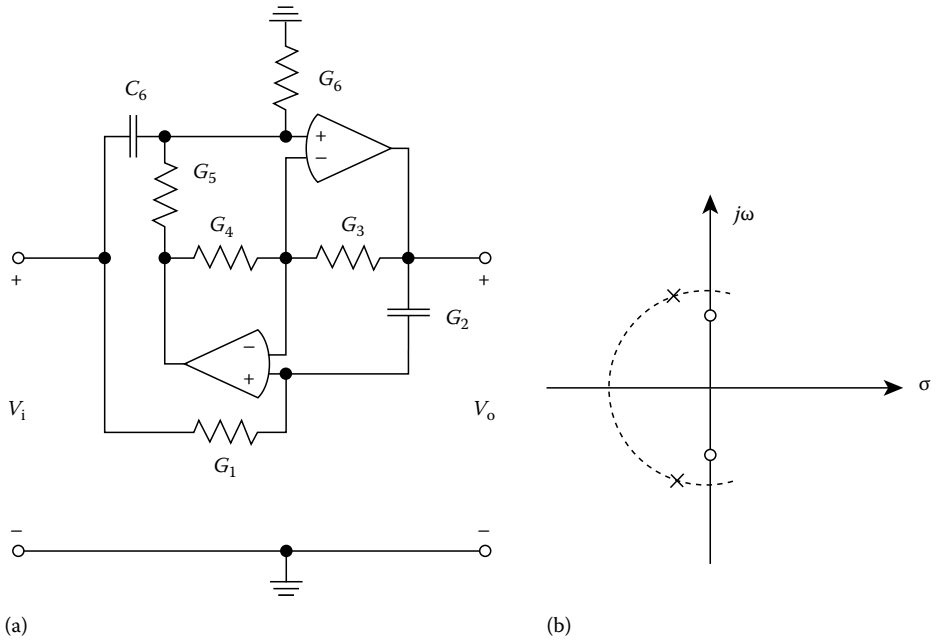


FIGURE 13.21 Second-order Cauer high-pass building block. (a) The circuitry. (b) The pole and zero locations.

The design of this building block is similar to that of the low-pass circuit. First we choose the resistors $R_3 = R_4$ and the capacitors C_2 and C_6 and then, with predefined parameters Q_p, ω_p, and ω_z, we determine the remaining three resistors:

$$R_6 = \frac{Q_p}{\omega_p C_6} \tag{13.56}$$

$$R_5 = R_6 \cdot \left(1 - \frac{\omega_z^2}{\omega_p^2}\right) \tag{13.57}$$

$$R_1 = \frac{1}{\omega_p^2 C_6 C_2 R_5} \tag{13.58}$$

13.5 GIC-Derived Three-Amplifier Biquads

In order to get more flexibility for realizing arbitrary second-order transfer functions and to obtain less resistor spread for realizing a given pole-Q, we can extend the resonator in Figure 13.14a to a resonator with three nullors; see Figure 13.22.

There are two different biquads known from the literature that are based on the resonator in Figure 13.22. The first one is proposed by Mikhael and Bhattacharyya [6] and is shown in Figure 13.23.

This filter circuit requires a small resistor spread to realize high pole-Q. The zeros of the transfer function are formed with a resistive feed-forward network providing a flexible design with arbitrary numerator coefficients. The voltage transfer ratio V_3/V_i is

$$H_3(s) = \frac{V_3}{V_i} = \frac{N_3(s)}{D(s)} \tag{13.59}$$

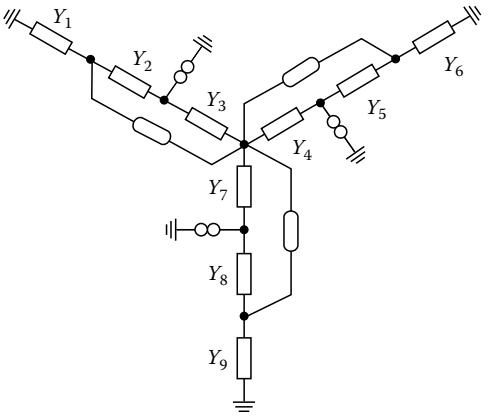


FIGURE 13.22 Resonator with three nullors.

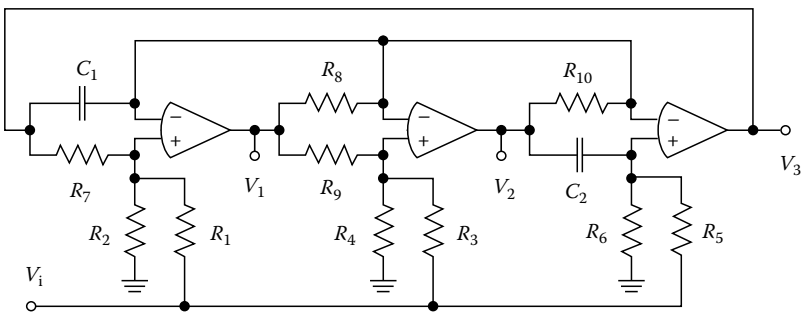


FIGURE 13.23 Mikhael–Bhattacharyya biquad.

with

$$N_3(s) = s^2C_1C_2G_1G_9 + sC_2(G_7G_8G_3 + G_8G_2G_3 - G_8G_1G_4) + (G_9G_7G_{10}G_5 + G_9G_{10}G_2G_5 - G_9G_{10}G_4G_6) \tag{13.60}$$

and

$$D(s) = s^2C_1C_2(G_1G_9 + G_2G_9) + sC_2(G_7G_8G_3 + G_7G_8G_4) + (G_9G_7G_{10}G_5 + G_9G_{10}G_7G_6) \tag{13.61}$$

The two other output nodes lead to similar transfer expressions.

The second biquad, which is based on the resonator in Figure 13.22, was proposed by Padukone et al. [7] and is depicted in Figure 13.24. Assuming ideal op-amps and choosing V_3 as output voltage, we obtain the following transfer function:

$$H_3(s) = \frac{V_3}{V_i} = \frac{N_3(s)}{D(s)} \tag{13.62}$$

with

$$N_3(s) = s^2[C_2C_3(G_2G_6 - G_1G_3) + C_1C_2G_1G_8] + s[C_1G_2G_5G_9 - C_3G_2G_5G_7 + C_2G_1G_4G_8] + (G_2G_4G_5G_9) \tag{13.63}$$

and

$$D(s) = s^2(C_1 + C_3)C_2G_2G_6 + sC_2G_1G_4(G_3 + G_8) + G_2G_4G_5(G_7 + G_9) \tag{13.64}$$

It has been shown [7] that the pole sensitivities to all passive components are not greater than unity. The filter section has the main advantage of being particularly insensitive to gain-bandwidth variations even when the op-amps are mismatched.

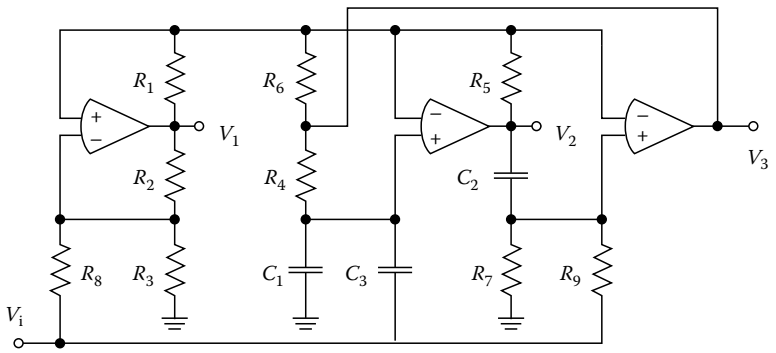


FIGURE 13.24 Padukone–Mulawka–Ghausi biquad.

13.6 State-Variable-Based Biquads

A frequently-used multiple-amplifier biquad is the circuit proposed by Kerwin et al. [8]. This filter circuit has extreme flexibility, good performance, and low sensitivities to the passive components. The filter is based on analog computer structures [9], which are derived from the state-variable representation of linear continuous systems. Therefore, these filters are also referred to as *state-variable filters*.

Figure 13.25a shows the basic analog computer structure consisting of one summing amplifier and two integrators. We assume both integrators to have the same transfer function $-1/(sT)$, where T is called the integrator time constant. Analyzing this structure yields

$$V_1 = -K_1 \cdot V_3 + K_2 \cdot V_i + K_3 \cdot V_2 \tag{13.65}$$

$$V_2 = -\frac{1}{sT} \cdot V_1 \tag{13.66}$$

$$V_3 = -\frac{1}{sT} \cdot V_2 \tag{13.67}$$

which results in

$$H_{HP}(s) = \frac{V_1}{V_i} = \frac{s^2T^2K_2}{s^2T^2 + sTK_3 + K_1} \tag{13.68}$$

Using Equation 13.66, we can immediately derive the voltage transfer ratio V_2/V_i from Equation 13.68:

$$H_{BP}(s) = \frac{V_2}{V_i} = \frac{-sTK_2}{s^2T^2 + sTK_3 + K_1} \tag{13.69}$$

Finally, with Equation 13.67 we obtain from Equation 13.69

$$H_{LP}(s) = \frac{V_3}{V_i} = \frac{K_2}{s^2T^2 + sTK_3 + K_1} \tag{13.70}$$

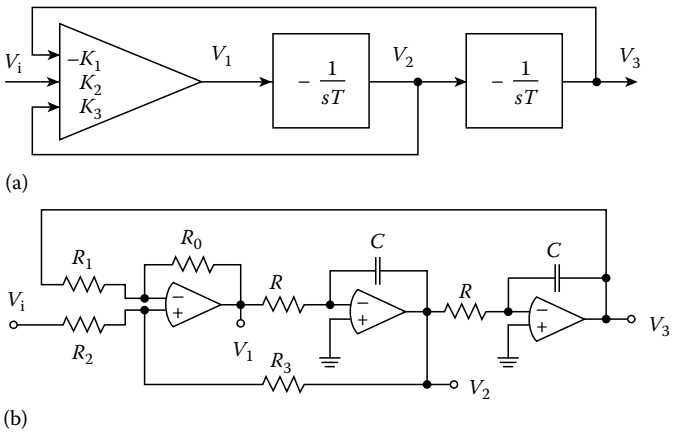


FIGURE 13.25 (a) Second-order analog computer structure and (b) state-variable filter section proposed by Kerwin, Huelsman, and Newcomb.

Thus, the structure in Figure 13.25a simultaneously realizes a high-pass filter, a bandpass filter, and a low-pass filter. The corresponding filter circuit proposed by Kerwin, Huelsman, and Newcomb is depicted in Figure 13.25b. The integrators consist of one op-amp, one resistor R, and one capacitor C. The time constant is given by $T = RC$. The three gain factors of the summing amplifier are determined by the four resistors $R_0 - R_3$:

$$K_1 = \frac{R_0}{R_1} \tag{13.71}$$

$$K_2 = \frac{R_3}{R_2 + R_3}\left(1 + \frac{R_0}{R_1}\right) \tag{13.72}$$

$$K_3 = \frac{R_2}{R_2 + R_3}\left(1 + \frac{R_0}{R_1}\right) \tag{13.73}$$

Obviously, the integrator time constant T plays the role of a reciprocal normalization frequency. Thus, if we refer the frequency variable s to $1/T$, we obtain from Equation 13.70 the normalized low-pass transfer function:

$$H_{\mathrm{LP}}(s) = \frac{V_3}{V_\mathrm{i}} = \frac{K_2}{s^2 + sK_3 + K_1} \tag{13.74}$$

Given the normalized pole frequency ω_p and the pole Q-factor Q_p, we can design the filter section by equating

$$K_1 = \omega_\mathrm{p}^2 \tag{13.75}$$

and

$$K_3 = \frac{\omega_\mathrm{p}}{Q_\mathrm{p}} \tag{13.76}$$

Then K_2 is fixed by the dc gain of the transfer function:

$$K_2 = H_0 \tag{13.77}$$

The state-variable filter circuit can be extended to a general biquad by adding an output amplifier that sums the three voltages V_1, V_2, and V_3. Figure 13.26a shows a state-variable filter with an output amplifier summing the voltages V_1, V_2, and V_3 of the circuit in Figure 13.25b. This biquad has been proposed also by Kerwin et al. [8]. As alternative circuits, the amplifiers in Figure 13.26b and c can be used. Figure 13.26b shows an output amplifier that realizes the following sum:

$$V_\mathrm{o} = \alpha_1 V_1 + \alpha_2 V_2 + \alpha_3 V_3 \tag{13.78}$$

with

$$\alpha_1 = -\frac{R_{10}}{R_{11}}, \quad \alpha_2 = \frac{R_{14}}{R_{12} + R_{14}}\left(1 + \frac{R_{10}}{R_{11} \| R_{13}}\right), \quad \alpha_3 = -\frac{R_{10}}{R_{13}} \tag{13.79}$$

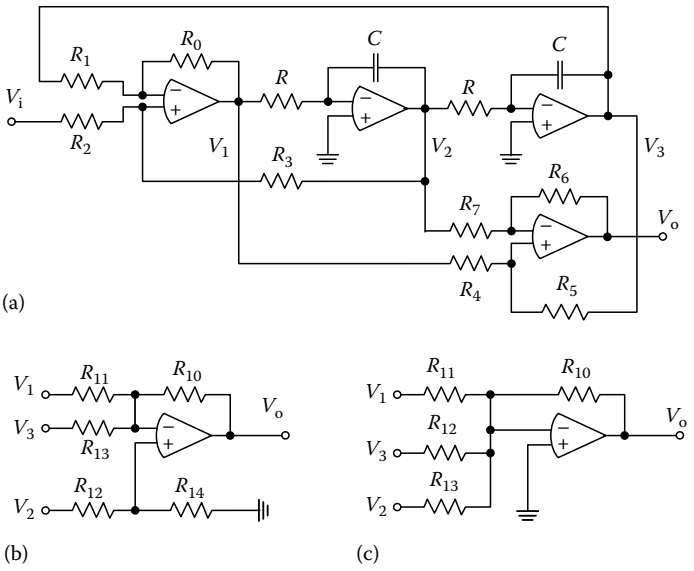


FIGURE 13.26 (a) State-variable filter with output amplifier, (b) output amplifier with inverting and noninverting inputs, and (c) output amplifier with three inverting inputs.

If we solve Equations 13.68 through 13.70 for the three output voltages and substitute them in Equation 13.79 with normalized variables s we obtain

$$H(s) = \frac{V_o}{V_i} = -K_2 \frac{|\alpha_1|s^2 + |\alpha_2|s + |\alpha_3|}{D(s)} \tag{13.80}$$

All numerator coefficients have the same sign. Therefore, the zeros of the transfer function are in the left-s-half plane. If we set $\alpha_2 = 0$, i.e., if we delete the voltage divider R_{12}, R_{14} and ground the noninverting input terminal of the op-amp, we obtain zeros on the $j\omega$ axis.

When designing the biquad, R_{14} and R_{10} may be used to scale the impedance level of the two resistive subnetworks. Then from the three numerator coefficients or from the overall gain constant of the transfer function, the zero frequency ω_z, and the zero Q-factor Q_z we can easily determine the remaining resistors R_{11}, R_{12}, and R_{13}.

The output amplifier in Figure 13.26c has three inverting inputs. Summing the voltages V_1, V_2, and V_3 leads to a numerator polynomial where the sign of the middle coefficient is different from the sign of the other two. Thus, the zeros are in the right-s-half plane. Again, we can delete the resistor R_{12} to realize zero on the $j\omega$ axis.

A second state-variable biquad circuit proposed by Tow and Thomas [10–12] yields similar performance to that of the Kerwin–Huelsman–Newcomb circuit. It uses a feedback loop with one damped integrator, one integrator, and one inverting amplifier; see Figure 13.27a. Figure 13.27b shows the Tow–Thomas circuit with three op-amps.

The damped integrator has a transfer function

$$\frac{V_1}{V_3} = \left.\frac{-1}{sT + \alpha}\right|_{V_i=0} \tag{13.81}$$

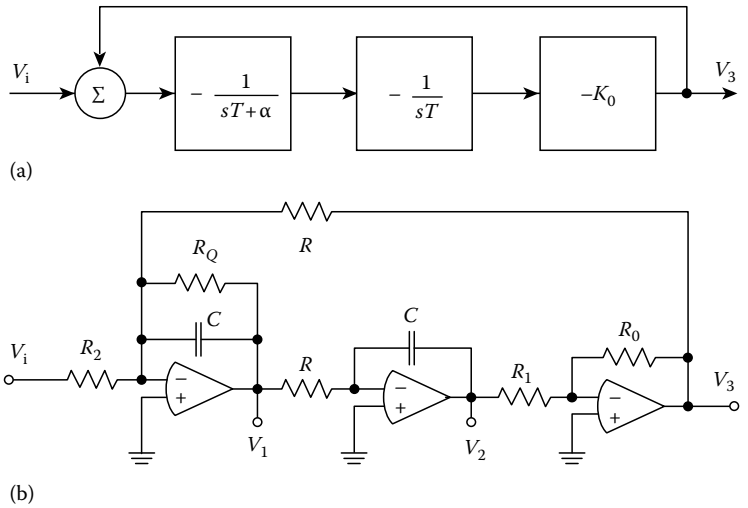


FIGURE 13.27 (a) Principle and (b) three op-amp realization of Tow–Thomas filter.

with $T = RC$ and $\alpha = R/R_Q$. The inverting amplifier has a gain

$$\frac{V_3}{V_2} = -\frac{R_0}{R_1} = -K_0 \tag{13.82}$$

An analysis of the circuit in Figure 13.27b with V_i being the input voltage and V_3 the output voltage yields a transfer function

$$H_{LP}(s) = \frac{V_3}{V_i} = \frac{-K_0\alpha_2}{s^2T^2 + sTK_Q + K_0} \tag{13.83}$$

with

$$K_0 = \frac{R_0}{R_1}, \quad \alpha_2 = \frac{R}{R_2}, \quad K_Q = \frac{R}{R_Q} \tag{13.84}$$

For the design of the filter, the integrator time constant T serves as a reciprocal normalization frequency. Then, from the predefined normalized pole frequency we can determine the resistor ratio K_0, from the pole Q-factor the ratio K_Q, and from the dc gain of the filter section the ratio α_2. Choosing convenient values for C and R_0, we finally determine the resistors R, R_1, R_Q, and R_2 from the parameters T, K_0, K_Q, and α_2, respectively.

The filter circuit in Figure 13.26 requires an additional op-amp to realize a transfer function with a general second-degree numerator polynomial. An alternative method is to feed fractions of the input signal forward into the input of each op-amp. This is realized in the multiple-input Tow–Thomas biquad [10]; see Figure 13.28. The transfer function of this circuit can be calculated to be

$$H(s) = \frac{V_o}{V_i} = -\frac{s^2T^2\alpha_4 + sT(K_Q\alpha_4 - K_0\alpha_3) + [K_0(\alpha_2 - K_Q\alpha_3)]}{s^2T^2 + sTK_Q + K_0} \tag{13.85}$$

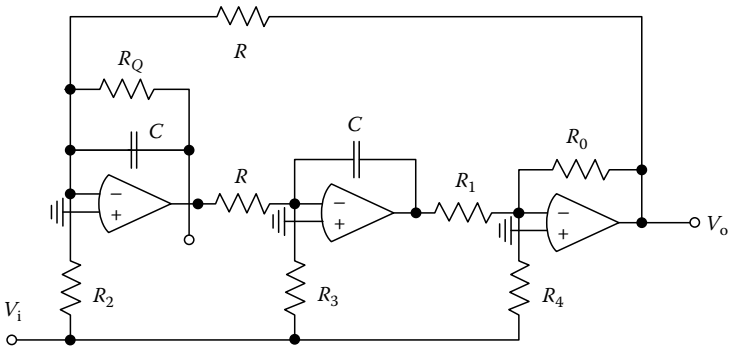


FIGURE 13.28 Generalized Tow–Thomas biquad.

with K_0, α_2, and K_Q as defined in Equation 13.84 and

$$\alpha_3 = \frac{R}{R_3}, \quad \alpha_4 = \frac{R_0}{R_4} \tag{13.86}$$

Thus, arbitrary numerator coefficients can be predescribed. In particular, if we choose $\alpha_3 = \alpha_4 = 0$ we obtain the low-pass filter circuit in Figure 13.27b and the transfer function in Equation 13.83.

When the state-variable filters described above are used to realize high-Q filter functions, the Q practically obtained is usually higher than that desired in the design. This effect is called Q enhancement and is caused by the phase lag introduced by the nonideal op-amps. One way to solve this problem is to use integrators with phase compensation.

Figure 13.29 shows a noninverting integrator with an additional op-amp for phase lag compensation. A detailed description of this circuit can be found in Ref. [13]. Putting this noninverting integrator together with an inverting integrator in a feedback loop results in a resonator with a Q-factor that is almost independent of the gain-bandwidth product of the op-amps. Thus, nearly no Q enhancement occurs.

Exactly this feedback loop is used in the Åkerberg–Mossberg biquad [14]; see Figure 13.30. In this circuit, a noninverting integrator with phase lag compensation together with an inverting damped integrator is connected as a feedback loop. More details about this filter section can be found in Refs. [4,13,14].

Finally, let us consider the general biquad proposed by Berka and Herpy [15]; see Figure 13.31. This biquad is also based on a state-variable representation and requires a second-order differentiator and a damped integrator. One of the main advantages of this circuit is extremely low sensitivities. A detailed description of the filter circuit and its design can be found in Ref. [4].

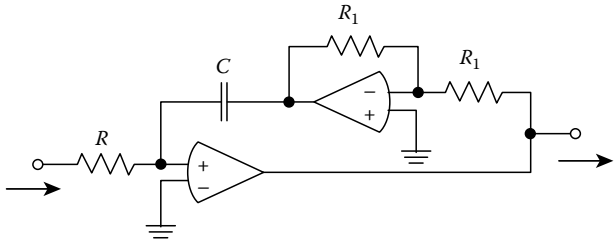


FIGURE 13.29 Noninverting integrator using an additional op-amp to compensate for phase lag.

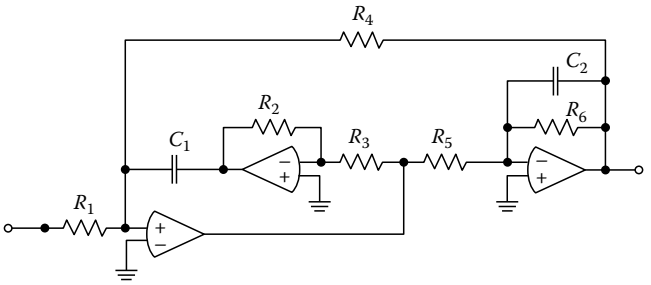


FIGURE 13.30 Åkerberg–Mossberg biquad.

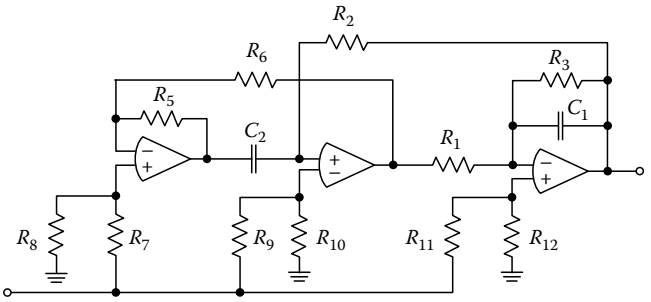


FIGURE 13.31 Berka–Herpy biquad.

13.7 All-Pass-Based Biquads

Finally, we will briefly consider two circuits that are based on first-order all-pass sections. Figure 13.32 shows the classical filter circuit introduced by Tarmy and Ghausi [16].

In this circuit, amplifiers with differential inputs and differential outputs are used. In the feedback loop, two all-pass sections with voltage transfer ratios

$$H_i(s) = \frac{sT_i - 1}{sT_i + 1} \tag{13.87}$$

$T_i = R_iC_i$, $i = 1, 2$, and one inverting amplifier are cascaded. The overall transfer function is

$$H(s) = \frac{V_o}{V_i} = \alpha_1 \frac{s^2T_1T_2 - s(T_1 + T_2) + 1}{s^2T_1T_2 + s\alpha_2(T_1 + T_2) + 1} \tag{13.88}$$

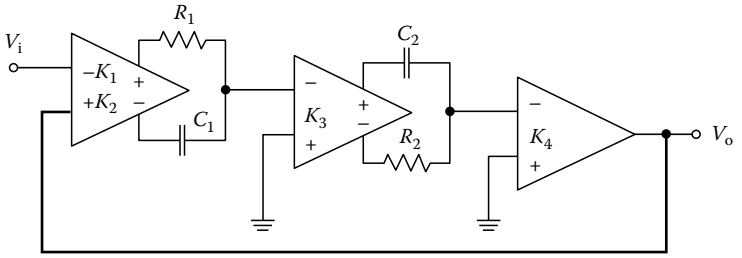


FIGURE 13.32 Tarmy–Ghausi circuit.

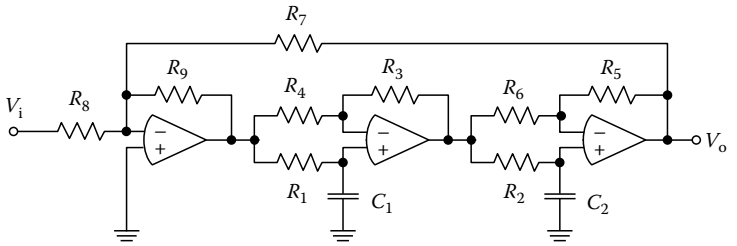


FIGURE 13.33 Tarmy–Ghausi circuit modified by Moschytz.

with

$$\alpha_1 = \frac{K_1K_3K_4}{1 + K_2K_3K_4}, \quad \alpha_2 = \frac{1 - K_2K_3K_4}{1 + K_2K_3K_4} \tag{13.89}$$

The main advantages of the Tarmy–Ghausi circuit are its low sensitivities to the gain-bandwidth products of the op-amps and thus its favorable performance at high pole frequencies and high pole Q-factors.

Most of the popular low-cost op-amps are not configured for differential output operations. This problem is bypassed by a circuit proposed by Moschytz [17]; see Figure 13.33.

The circuit in Figure 13.33 is also based on first-order all-pass sections. But the all-pass circuits are realized by means of only single-ended op-amps. Therefore, this circuit is more convenient for practical realizations.

13.8 Summary

In this chapter, we have shown that a lot of different multiple-amplifier biquads are known from the literature. Hence, a design engineer who aims to realize a high-performance active filter is in the favorable situation to find a rich variety of alternative circuits and design methods. On the other hand, this variety is also confusing and the design engineer may need assistance in deciding among all these circuits and methods. Unfortunately, the multiple-amplifier biquads mentioned in this chapter cannot be classified in a simple way. Therefore, only a rough guidance can be given. In a practical case, it is advisable to compare two or three solutions next to the predefined demands by a thorough analysis and then to find the final solution.

The biquads with decoupled time constants offer some of the benefits mentioned at the beginning of this section at low costs, namely, small passive element spread and low sensitivities. The dual-amplifier twin-T biquads and the GIC-derived dual-amplifier biquads offer a trade-off between costs and performance. The twin-T biquads are particularly suitable for the design of elliptic filters. The special merits of the GIC-derived biquads are the small number of passive components and the independence of the op-amp parameters. Thus, this biquad represents a robust filter solution for many different applications. Additionally, these filters can be easily designed.

The GIC-derived three-amplifier biquads and the state-variable-based filters offer additional flexibility with respect to an independent choice and an independent tuning of the filter parameters. Typically, these circuits are used in applications with switched parameters, e.g., filters with switched cutoff frequencies. This flexibility is paid for by a higher number of op-amps and passive components. It should be mentioned that the original state-variable circuits, i.e., two integrators in a loop, are rather sensitive to the phase lag introduced by the nonideal op-amps. This is always a problem when simultaneously high-frequency and high-Q performance shall be achieved. In such applications, biquads with compensated phase lag should be applied, e.g., the biquads proposed by Åkerberg and Mossberg, Berka and Herpy, or Tarmy and Ghausi.

References

1. R. P. Sallen and E. L. Key, A practical method of designing *RC* active filters, *IRE Trans. Circuits Theory*, CT-2, 75–85, 1955.
2. P. R. Geffe, A Q-invariant active resonator, *Proc. IEEE* (Lett.), 57, 1442, 1969.
3. M. A. Soderstrand and S. K. Mitra, Extremely low sensitivity active *RC* filter, *Proc. IEEE (Lett.)*, 57, 2175, 1969.
4. M. Herpy and J.-C. Berka, *Active RC Filter Design*, Amsterdam: Elsevier, 1986.
5. N. J. Fliege, A new class of second-order *RC*-active filters with two operational amplifiers, *Nachrichtentech. Z.*, 26, 279–282, 1973.
6. W. B. Mikhael and B. B. Bhattacharyya, A practical design for insensitive *RC*-active filters, *IEEE Trans. Circuits Syst.*, CAS-22, 407–415, 1975.
7. P. Padukone, J. Mulawka, and M. S. Ghausi, An active biquadratic section with reduced sensitivity to operational amplifier imperfections, *J. Franklin Inst.*, 30(1), 27–40, 1980.
8. W. J. Kerwin, L. P. Huelsman, and R. W. Newcomb, State-variable synthesis for insensitive integrated circuit transfer functions, *IEEE J. Solid-State Circuits*, SC-2, 87–92, 1967.
9. W. Schüßler, Schaltung und Messung von Übertragungsfunktionen an einem Analogrechner, *Archiv der Elektrischen Übertragung*, 13, 405–419, 1959.
10. P. E. Fleischer and J. Tow, Design formulas for biquad active filters using three operational amplifiers, *Proc. IEEE*, 61, 662–663, 1973.
11. L. C. Thomas, The biquad: Part I—Some practical design considerations, *IEEE Trans. Circuits Theory*, CT-18, 350–357; —, The biquad: Part II—A multipurpose active filtering system, *IEEE Trans. Circuits Theory*, CT-18, 358–361, 1971.
12. J. Tow, Design formulas for active *RC* filters using operational-amplifier biquad, *Electron. Lett.*, 5, 339–341, 1969.
13. L. P. Huelsman and P. E. Allen, *Introduction to the Theory and Design of Active Filters*, New York: McGraw-Hill, 1980.
14. D. Åkerberg and K. Mossberg, A versatile active *RC* building block with inherent compensation for the finite bandwidth of the amplifier, *IEEE Trans. Circuit Syst.*, CAS-21, 75–78, 1974.
15. J. C. Berka and M. Herpy, Novel active *RC* building block with optimal sensitivity, *Electron. Lett.*, 17, 887–888, 1981.
16. R. Tarmy and M. S. Ghausi, Very high-*Q* insensitive *RC* networks, *IEEE Trans. Circuits Theory*, CT-17, 358–366, 1970.
17. G. S. Moschytz, High-Q factor insensitive active *RC* networks, similar to the Tarmy–Ghausi circuit but using single-ended operational amplifiers, *Electron. Lett.*, 8, 458–459, 1972.

14

The Current Generalized Immittance Converter Biquads*

14.1 Introduction **14**-1
14.2 Biquadratic Structure Using the Antoniou CGIC (2-OA CGIC Biquad) **14**-2
14.3 Realization of the Most Commonly Used Biquad Functions Using the 2-OA CGIC **14**-4
14.4 Stability during Activation and Sensitivity Using the 2-OA CGIC Biquad **14**-4
Stability Properties • Sensitivity Analysis
14.5 Design and Tuning Procedure of the 2-OA CGIC Biquad **14**-6
14.6 Design Examples Using the 2-OA CGIC Biquad **14**-7
14.7 Practical High-Order Design Examples Using the 2-OA CGIC Biquad **14**-9
Low-Pass Filter • Bandpass Filter
14.8 Universal 2-OA CGIC Biquad **14**-11
14.9 3-OA CGIC Biquadratic Structure **14**-11
14.10 Design and Tuning Procedure of the 3-OA CGIC Biquad **14**-15
14.11 Practical Sixth-Order Elliptic BP Filter Design Using the 3-OA CGIC Biquad **14**-16
14.12 Composite GIC Biquad: 2 C 2-OA CGIC **14**-18
14.13 Summary **14**-19
References **14**-19

Wasfy B. Mikhael
University of Central Florida

14.1 Introduction

Current generalized immittance convertors (CGICs) have been used to realize high-performance active biquads with 2 OAs, 3 OAs, or n OAs per sections [1,5].

In this chapter two- and three-amplifier biquads are presented that are based on CGICs. Although several biquads have been reported, the ones presented here have proved to be clearly superior

* Wasfy B. Mikhael, "Chapter 9: Biquad II," in *RC Active Filter Design Handbook*, Stephenson, New York: Wiley, 1985.

to single-amplifier biquads. This is because the design carried out [1] was constrained by stringent performance criteria satisfying important properties and features such as stability, versatility, insensitivity to component tolerances and drift, low dependence on the op-amp (OA) frequency limitations, finite gain, tunability, small spread in component values, and minimum total capacitance.

In addition, the performance of the CGIC-based biquads presented here are comparable to multiple OA biquads. On the other hand, the 3-OA CGIC biquad is shown to yield additional performance improvements over the 2-OA CGIC biquad. Also, the CGIC biquads use the OAs in the differential mode.

In the following discussion the generalized structure of the 2-OA and 3-OA CGIC biquads are presented. Illustrative examples of the element identification to realize the most commonly used biquads are tabulated. Stability and sensitivity properties are discussed. A design procedure for each biquad is described that minimizes the active sensitivities while maintaining the filter's stability. Several second-order design examples are given. A sixth-order Chebyshev LPF and a sixth-order elliptic BPF are designed using the design values and tuning procedure suggested. The excellent performance of the resulting realizations is experimentally verified. A universal 2-OA GIC hybrid implementation using thick film is also described. 2-OA CGIC biquadratic active filter realizations for extended high-frequency applications employing the composite operational amplifiers technique [2,3] are given.

14.2 Biquadratic Structure Using the Antoniou CGIC (2-OA CGIC Biquad)

Consider the network of Figure 14.1, which is simply Antoniou's CGIC [4], with two new ports created across 3-G and 4-G. This is represented symbolically in Figure 14.2.

A new configuration is now obtained, as shown in Figure 14.3. A synthesis procedure is now described that uses this configuration. The transfer functions between the input and output terminals 2, 3, and 4, assuming ideal OAs are readily obtained as

$$\frac{V_3}{V_1} = T_1 = \{Y_5 + h(s)[Y_7(1 + Y_6/Y_2) - Y_5Y_8/Y_2]\}/D(s) \tag{14.1a}$$

$$\frac{V_4}{V_i} = T_2 = [Y_5(1 + Y_8/Y_4) - Y_6Y_7/Y_4 + h(s)Y_7]/D(s) \tag{14.1b}$$

$$\frac{V_2}{V_i} = T_3 = [Y_5 + h(s)Y_7]/D(s) \tag{14.1c}$$

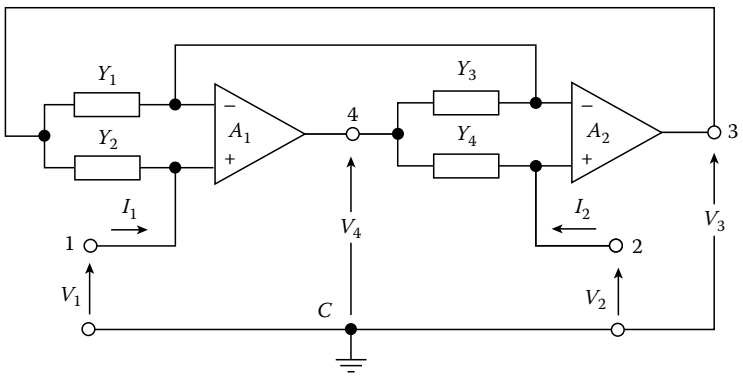


FIGURE 14.1 Antoniou's CGIC with additional ports 3*G* and 4*G*.

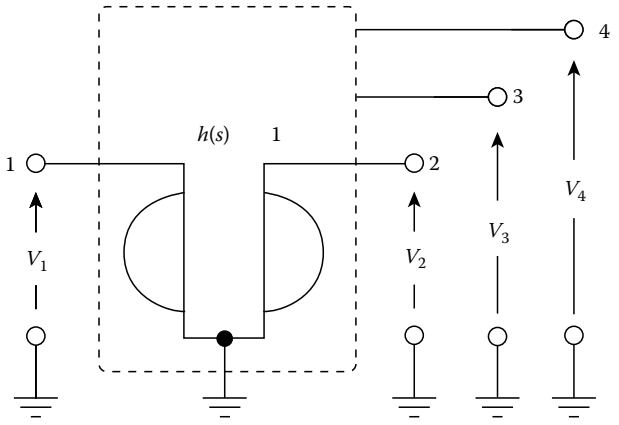


FIGURE 14.2 Symbolic representation of the CGIC in Figure 14.1.

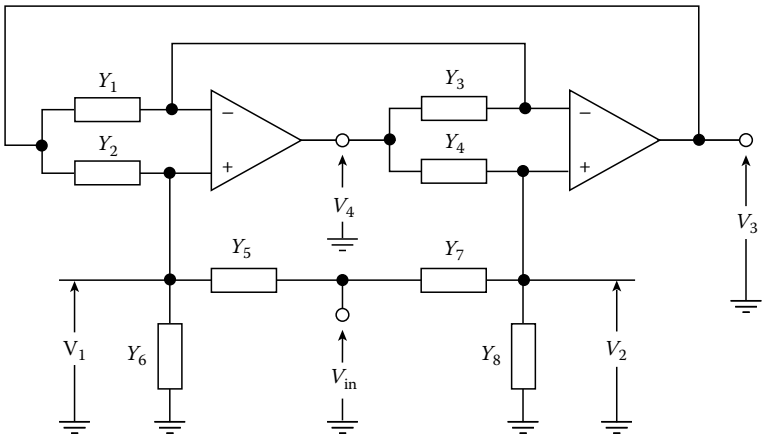


FIGURE 14.3 Basic configuration.

where

$$\begin{aligned} h(s) &= Y_2Y_3/Y_1Y_4 \\ D(s) &= (Y_5 + Y_6) + h(s)(Y_7 + Y_8) \end{aligned} \tag{14.1d}$$

The conversion function $h(s)$ and the admittances Y_5–Y_8 can be selected in many different ways, and it is clear that any stable second-order transfer function with any desired zero and pole locations can be realized.

Letting

$$Y_i = sC_i + G_i \tag{14.2}$$

where $i = 1$–4, we have from Equation 14.1

$$h(s) = \frac{(sC_2 + G_2)(sC_3 + G_3)}{(sC_1 + G_1)(sC_4 + G_4)} \tag{14.3}$$

Clearly, by omitting one or more conductances and/or capacitances, a number of specific conversion functions can be generated.

14.3 Realization of the Most Commonly Used Biquad Functions Using the 2-OA CGIC

The most frequently used second-order transfer functions have already been described in Chapter 11. Realizations of these functions are given in Table 14.1. The LP, HP, BP, and BS sections are produced by choosing $h(s)$ in a simple manner such as $K_1 s$, $K_2 s^2$, $K_3 s + K_4 s^2$ or their reciprocals. Circuits 3, 4, and 7 can be regarded as realizations of simple RLC networks [4,5]. Second-order AP sections can be obtained from circuits 11 and 12 of Table 14.1.

Figures 14.1 and 14.2 and Table 14.1 show that, with the exception of circuit 10, the response is obtained from the output of an OA. Owing to the low output resistance of the amplifier, any number of sections can be cascaded without using isolation amplifiers.

14.4 Stability during Activation and Sensitivity Using the 2-OA CGIC Biquad

14.4.1 Stability Properties

It has been shown [6] that some networks using CGICs can be conditionally stable, that is, a circuit can lock in an unstable mode during activation (just after switching on the power supply). For amplifiers with a finite open loop gain A, the circuit of Figure 14.3 gives

$$\frac{V_k}{V_i} = \frac{N_k(s)}{D(s)}$$

where $k = 2, 3, 4$ and

$$\begin{aligned} D(s) &= M_1 Y_1 + M_2 Y_3 \\ &\quad + (1 + M_1)(1 + M_2)[Y_1/A_1 + Y_3/A_2 + Y_1/A_1A_2 + Y_3/A_1A_2] \end{aligned} \tag{14.4}$$

$$M_1 = \frac{Y_5 + Y_6}{Y_2}$$

$$M_2 = \frac{Y_7 + Y_8}{Y_4}$$

It can be easily shown that for the circuit in Figure 14.3 low-frequency unstable modes cannot arise during activation. This is due to the absence of differences and changes of sign in the denominator coefficients. Thus any combination of OA gains that occur during transients and power supply switching does not result in saturation or low-frequency instability before reaching the steady state [7].

14.4.2 Sensitivity Analysis

The pole Q-factor Q_p, the undamped frequency of oscillation ω_p, the notch frequency ω_n, and the multiplier constants H_{BS}, H_{LP}, H_{HP}, and H_{BP}, as well as Q_z, ω_z, and H_{AP}, have been previously defined in

TABLE 14.1 Element Identification for Realizing the Most Commonly Used Transfer Functions

Circuit Number	$H(s)$	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Transfer Function	Remarks
1	$\frac{sC_2(sC_3+G_3)}{G_1G_4}$	G_1	SC_2	sC_3+G_3	G_4	G_5	0	0	G_8	$T_2=\frac{G_1G_5(G_4+G_8)}{G_1G_5G_4+sC_2G_3G_8+s^2C_2C_3C_8}$	LP
2	$\frac{G_2G_3}{sC_1(sC_4+G_4)}$	sC_1	G_2	G_3	$sC_4+G_4^1$	0	G_6	G_7	$sC_8^1+G_8^a$	$T_1=\frac{G_2G_3G_7\left(1+\frac{G_8}{G_2}\right)}{G_2G_3(G_7+G_8)+s(C_2G_4G_6+C_8G_2G_3)+s^2C_1C_4C_6}$	LP
3	$\frac{sC_3G_2}{G_1G_4}$	G_1	G_2	sC_3	G_4	0	G_6	sC_7	$sC_8^a+G_8$	$T_1=\frac{s^2C_3C_7(G_2+G_6)}{G_6G_1G_4+sC_3G_2G_8+s^2(C_7+C_8)G_2C_3}$	HP
4	$\frac{sC_3G_2}{G_1G_4}$	G_1	G_2	sC_3	G_4	0	G_6	$\frac{sC_7}{1+sC_7R_7}$	0	$T_1=\frac{s^2C_3C_7(G_2+G_6)}{G_6G_1G_4+sC_7G_1G_4G_6R_7+s^2C_3C_7G_2}$	HP
5	$\frac{s^2C_2C_3}{G_1G_4}$	G_1	G_2	sC_3	G_4	0	G_6	0	G_8	$T_2=\frac{sC_5\left(1+\frac{G_8}{G_4}\right)G_1G_4}{G_1G_4G_6+sC_5G_1G_4+s^2C_2C_3G_8}$	BP
6	$\frac{G_2G_3}{sC_1(sC_4+G_4)}$	sC_1	G_2	G_3	$sC_4+G_4^a$	0	G_6	sC_7	G_8	$T_1=\frac{sC_7G_2G_3\left(1+\frac{G_6}{G_2}\right)}{G_2G_3G_8+s(C_7G_2G_3+C_1G_4G_8)+s^2C_1C_4G_6}$	BP
7	$\frac{sC_3G_2}{G_1G_4}$	G_1	G_2	sC_3	G_4	0	G_6	sC_7	$sC_8+G_8^a$	$T_1=\frac{sC_3G_7(G_2+G_6)}{G_1G_4G_6+sC_3G_2(G_7+G_8)+s^2C_3C_8G_2}$	BP
8	$\frac{G_2G_3}{s^2C_1C_4}$	sC_1	G_2	G_3	sC_4	G_6	0	G_7	sC_8	$T_2=\frac{s^2C_1G_5(C_4+C_8)+G_2G_3G_7}{s^2C_1G_5C_4+sC_8G_2G_3+G_2G_3G_7}$	N
9	$\frac{sG_2C_3}{G_1G_4}$	G_1	G_2	sC_3	G_4	G_5	0	sC_7	G_8	$T_2=\frac{s^2C_3C_7G_2+G_1G_5(G_4+G_8)}{s^2C_3C_7G_2+sC_3G_2G_8+G_1G_4G_5}$	N
10	$\frac{sG_2C_3}{G_1G_4}$	G_1	G_2	sC_3	G_4	G_5	G_6^a	sC_7	sC_8+G_8	$T_1=\frac{\left(s^2+\frac{G_1G_4G_5}{C_3C_7G_2}\right)\left(\frac{C_7}{C_7+C_8}\right)}{s^2+\frac{G_8}{C_7+C_8}+\frac{G_1G_4(G_5+G_6)}{C_3G_2(C_7+C_8)}}$	N
11	$\frac{G_2G_3}{s^2C_1C_4}$	sC_1	G_2	G_3	sC_4	G_5	G_6^a	G_7	sC_8	$T_1=\frac{s^2C_1G_4G_5-sG_5C_8G_3+G_3G_7(G_2+G_6)}{s^2C_1C_4(G_5+G_6)+sC_8G_2G_3+G_2G_3G_7}$	Nonminimum phase For all pass: $G_6=0$, $G_5=G_2$
12[b]	$\frac{sG_2G_3}{G_1G_4}$	G_1	G_2	sC_3	G_4	G_5	G_6	sC_7	G_8	$T_1=\frac{s^2C_3C_7(G_2+G_6)-sC_3G_5G_8+G_1G_4G_5}{s^2C_3C_7G_2+sC_3G_2G_8+(G_5+G_6)G_1G_4}$	Nonminimum phase For all pass: $G_6=0$, $G_2=G_5$

[a] These elements can be set equal to zero.

[b] A special case of Circuit 12—namely, the all-pass case—has also been independently proposed by J. T. Lim, Bell Northern Research, Canada, as communicated privately by him to Dr. A. Antoniou.

Chapter 7. The sensitivity of a quantity x with respect to variations in an element e is denoted by S_e^x. For ideal amplifiers, the use of Table 14.1 leads to

$$0 \leq S_e^x \leq 1 \tag{14.5}$$

where

x represents any one of these H, ω, or Q qualities
e represents any capacitance or conductance

14.5 Design and Tuning Procedure of the 2-OA CGIC Biquad

A design procedure is now described. Table 14.1 shows that there are several degrees of freedom in the choice of element values. These may be used to minimize $S_A^{Q_p}$ or the spread of element values. By using minimum sensitivity (to the OA parameters) constraints in circuits 1, 3, 7, 10, and 12, possible sets of element values for LP, HP, BP, BS, and AP sections have been obtained, as shown in Table 14.2. In addition, this choice of elements guarantees stable operation with real OA's. It is seen that the notch frequency ω_n, the undamped frequency of oscillation ω_p, and Q-factor Q_p can be easily adjusted by sequentially trimming three distinct resistors. A tuning sequence is also given in Table 14.2.

It is to be noted from Table 14.2 for the design of the BS sections, that

$$\omega_n^2 = \frac{\omega_p G_s}{C_s} \tag{14.6}$$

where

$$G_5 + G_6 = G_1, \quad C_7 + C_8 + C, \quad \text{and} \quad \omega_p = \frac{G}{C}$$

TABLE 14.2 Design Values and Tuning Procedure

Circuit Number (from Table 14.1)	Design Values	Transfer Function Realized	Tuning Sequence		
			ω_o	ω_p	Q_p
1	$G_1 = G_4 = G_5 = G_8 = G$, $G_3 = G/Q_p$, $C_2 = C_3 = C$, where $C = \frac{G}{\omega_p}$	$T_2 = \frac{2\omega_p^2}{D(s)}$	—	G_8	G_3
3	$G_1 = G_2 = G_4 = G_6 = G$, $G_8 = G/Q_p$, $C_8 = 0$, $C_3 = C_7 = C$, where $C = \frac{G}{\omega_p}$	$T_1 = \frac{2s^2}{D(s)}$	—	G_4	G_8
7	$G_1 = G_2 = G_4 = G_6 = G$, $G_7 = G/Q_p$, $G_8 = 0$, $C_3 = C_8 = C$, where $C = \frac{G}{\omega_p}$	$T_1 = \frac{\left(\frac{2\omega_p}{Q_p}\right)s}{D(s)}$	—	G_2	G_7
10	$G_1 = G_2 = G_4 = G_5 + G_6 = G$, $G_8 = G/Q_p$, $C_3 = C_7 + C_8 = C$, where $\omega_p = \frac{G}{C}$ and $\omega_n^2 = \omega_p \frac{G_5}{C_7}$	$T_3 = \frac{C_7}{C} \frac{\left(s^2 + \omega_n^2\right)}{D(s)}$	G_2	G_6	G_8
12	$G_1 = G_2 = G_4 = G_5 = G$, $G_6 = 0$, $C_3 = C_7 = C$, $C_8 = G/Q_p$, where $\omega_p = \frac{G}{C}$	$T_1 = \frac{D(-s)}{D(s)}$	G_4		G_8

Note: $D(s) = s^2 + (\omega_p/Q_p)s + \omega_p^2$.

Hence

$$\frac{w_n^2}{w_p^2} = \frac{G_5}{G} \cdot \frac{C}{C_7} \tag{14.7}$$

where

$$\frac{G_5}{G} \text{ is always} \leq 1 \quad \text{and} \quad \frac{C}{C_7} \text{ is always} \geq 1$$

It is seen from Equation 14.7 that for $\omega_n \leq \omega_p$, C_8 can be set to zero ($C_7 = C$), and only two capacitors per section are used. For $\omega_n > \omega_p$, three capacitors are required. In all cases, regardless of the choice of C_7 and C_8, the total capacitance per section is equal to $2C$.

In most applications where notch sections are used, ω_n/ω_p is close to unity and care should be taken when choosing G_5 and C_7 in Equation 14.7. A suitable choice of G_5, G_6, G_7, and C_8 values may be obtained by letting

$$\frac{G_5}{G} = \frac{1}{K} \quad \text{and} \quad \frac{C}{C_7} = 2$$

Thus K is approximately equal to 2 ($K < 2$ for $\omega_n > \omega_p$ and $K > 2$ for $\omega_n \leq \omega_p$). The suggested choice yields a capacitor spread of 2 and both R_5 and R_6 are approximately equal to $2R$.

14.6 Design Examples Using the 2-OA CGIC Biquad

Example 14.1

Design a second-order Butterworth ($Q_p = 0.707$) LP filter having a cutoff frequency $f_p = 20{,}000/2\pi$ Hz.

Procedure

1. Circuit 1 in Table 14.2 realizes an LPF. The design equations are also given in Table 14.2.
2. First we choose an appropriate value for C, say 10 nF. Thus $C = C_2 = C_3 = 10$ nF.
3. Now, $R = \frac{1}{C\omega_p} = \frac{1}{20{,}000 \times 10^{-8}}$
   Therefore, $R = 5$ kΩ
4. Consequently, $R = R_1 = R_4 = R_5 = R_8 = 5$ kΩ and $R_3 = RQ_p = 3.535$ kΩ.
5. The circuit is shown in Figure 14.4a. It is noted that the low-frequency gain of the LP filter H_{LP} is 2.

A simple procedure can be followed to scale H_{LP} by a factor x less than unity, that is, effectively multiplying the transfer function realized by x. This is done by replacing the resistance R_5 by two resistors R_A, and R_B (in series with the input V_{in}) in the manner shown in Figure 14.4b, where

$$\begin{aligned} R_5 &= R = 5 \text{ k}\Omega \\ &= R_A \parallel R_B \end{aligned}$$

The desired gain and scale factor $x = R_B/(R_A + R_B)$. Thus for $x = \frac{1}{2}$, resulting in a dc gain of the LP filter of unity, the choice of resistors R_A and R_B is $R_A = R_B = 10$ kΩ.

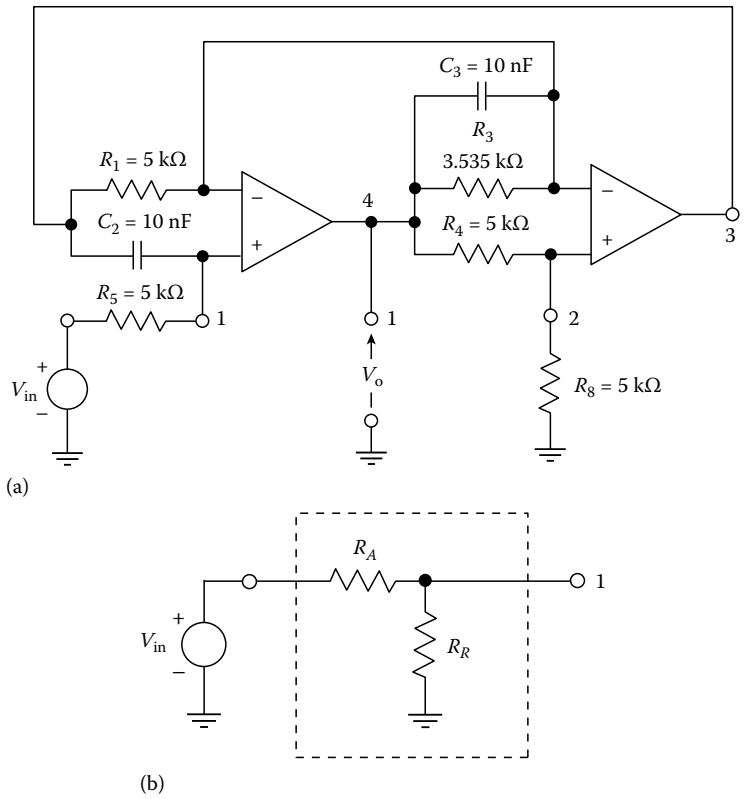


FIGURE 14.4 (a) Second-order Butterworth LPF design example and (b) controlling the gain factor of the LPF.

If functional tuning of the filter is desired, the tuning sequence of circuit 1 in Table 14.2 can be followed. First, ω_p is adjusted by applying a sinusoidal input at the desired ω_p frequency. Then R_B is tuned until ω_p realized equals the desired value. This can be detected by monitoring the phase angle of the output relative to the input. When the proper ω_p is reached, the output lags the input by 90°. Next, to adjust the Q_p, the filter gain H_{dc} of the LPF at a frequency much lower than ω_p is determined. Then an input at ω_p is applied. R_3 is adjusted until the gain of the LPF at ω_P is Q_P desired H_{dc}.

Example 14.2

Design a second-order BP filter with $Q_p = 10$ and $f_p = 10{,}000/2\pi$ Hz.

Procedure

1. Circuit 7 in Table 14.2 realizes a BP filter. The design equations are also given in Table 14.2.
2. First we choose a suitable value for C, say 10 nF.
3. Thus $C_3 = C_8 = C = 10$ nF.
4. Hence $R = (1/C\omega_p) = 10\,\text{k}\Omega$. Consequently, $R = R_1 = R_2 = R_4$, $R_6 = 10\,\text{k}\Omega$, and $R_7 = RQ_p = 100\,\text{k}\Omega$.
5. The circuit is shown in Figure 14.5. The gain at resonance, that is, at $\omega = \omega_p$, is equal to 2. To scale the gain by a factor x less than 2, the resistor R_7 is split into two resistors in a manner similar to that in Figure 14.4b and explained in Example 14.1.

Again, if functional tuning is desired, the sequence in Table 14.2 can be followed.

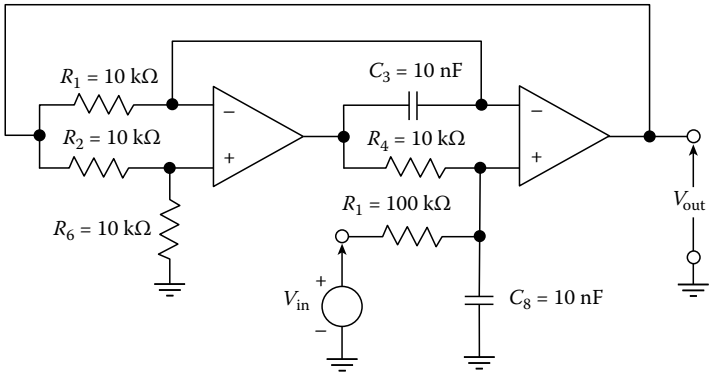


FIGURE 14.5 Design of a second-order BPF.

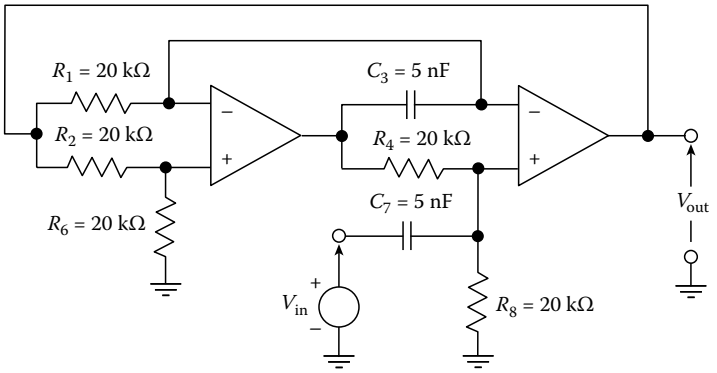


FIGURE 14.6 Design of a second-order HPF.

Example 14.3

Design a second-order HP filter with $Q_p = 1$ and $f_p = 10{,}000/2\pi$ Hz.

Procedure

1. Circuit 3 in Table 14.2 realizes an HP filter.
2. Let us choose $C_3 = C_7 = C = 5$ nF. Hence R can be computed as $R = \frac{1}{\omega_p C} = \frac{1}{10{,}000 \times 50 \times 10^{-9}}$ Therefore, $R = 20$ kΩ.
3. Consequently, $R_1 = R_2 = R_4 = R_6 = R = 20$ kΩ and $R_8 = RQ = 20$ kΩ.
4. The realization is shown in Figure 14.6. The gain at high-frequency H_{HP} is equal to 2.

14.7 Practical High-Order Design Examples Using the 2-OA CGIC Biquad

Using Table 14.2, 1% metal-film resistors, 2% polystyrene capacitors and μ A741 OA's, a sixth-order Chebyshev LP filter, and a sixth-order elliptic BP filter were designed and constructed. The LP filter has a

maximum passband attenuation of 1.0 dB; bandwidth = 3979 Hz. The BP filter has the following specifications:

Center frequency = 1500 Hz
Passband = 60 Hz
Maximum passband attenuation = 0.3 dB
Minimum stopband attenuation outside the frequency range 1408 → 1595 Hz = 38 dB

14.7.1 Low-Pass Filter

The realization uses cascaded section of type 1, in Table 14.2, as shown in Figure 14.7a. The measured frequency response (input level = 50 mV), shown in Figure 14.7b and c, agrees with the theoretical response. The effect of dc-supply variations is illustrated in Figure 14.7d. The deviation in the passband ripple is about 0.1 dB for supply voltage in the range 5–15 V. The effect of temperature variations is illustrated in Figure 14.7e, which shows the frequency response at −10°C (right-hand curve), 20°C, and 70°C (left-hand curve). The last peak has been displaced horizontally by 42 Hz, which corresponds to a

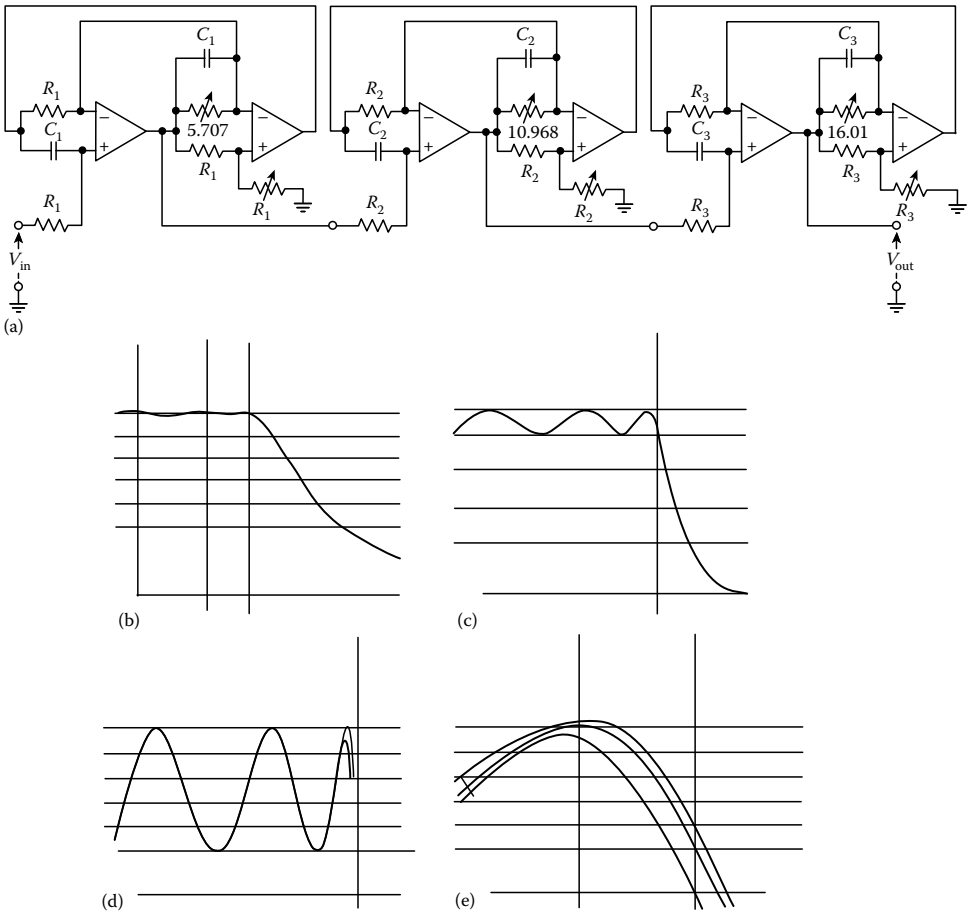


FIGURE 14.7 (a) Realization of the sixth-order Chebyshev low-pass filter. Frequency responses: (b) logarithmic gain scale and linear frequency scale; (c) linear gain and frequency scales; (d) for supply voltages ±5 V (lower curve) and ±15 V, input level = 0.05 V; (e) at temperatures −10°C (right-hand curve), 20°C, and 70°C (left-hand curve).

change of 133 ppm/°C. The frequency displacement is due to passive element variations and is within the predicted value.

14.7.2 Bandpass Filter

The realization uses cascaded sections of the types 7 and 10, in Table 14.2, and is shown in Figure 14.8a. The measured frequency response is shown in Figure 14.8b and c, and it is in agreement with the theoretical response. Figure 14.8d shows the frequency response for supply voltages of 7.5 V (lower curve) and 15 V; the input is 0.3 V. The passband ripple remains less than 0.39 dB and the deviation in the stopband is negligible. Figure 14.8e and f illustrate the effect of temperature variations. The passband ripple remains less than 0.35 dB in the temperature range −10°C to 70°C. A center frequency displacement of 15 Hz has been measured that corresponds to a change of 125 ppm/°C.

14.8 Universal 2-OA CGIC Biquad

Study of Table 14.2 suggests that several circuits may be combined to form a universal biquad. This can be achieved on a single substrate using thick-film technology.

Upon examining the element identification and design values in Table 14.2 it is easy to see that one common thick-film substrate can be made to realize circuits 1, 3, 7, and 10 in Table 14.2 (other circuits from Table 14.1 can be included if desired) with no duplication in OAs and chip capacitors and minimal duplication in resistors. The superposition of circuits 1, 3, 7, and 10 is shown in Figure 14.9. The following items should be noted.

1. Each resistor having the same subscript represents one resistor only and needs to appear once in a given biquad realization and thus once on the substrate. As an example, for $R_J = RQ_p$, only one R_J is needed with connection to several nodes. The unwanted connections may be opened during the trimming process according to the type of circuit required.
2. Three capacitor pads are needed; they are marked 1, 2, and 3 in Figure 14.9. To obtain capacitor 4, either capacitor 2 or 3 connections are made common with capacitor 4 terminals. The capacitor pad terminals are available on the external terminals of the substrate.* The chip capacitors are reflow-soldered in the appropriate locations based on the circuit realized.

A dual CGIC universal biquad implemented using thick-film resistors, chip NPO capacitors, and one quad OA is shown in Figure 14.10. Note that this hybrid array is capable of realizing gyrators and simulating inductors and super capacitors. Sample results are given in Figure 14.11 for realizing different biquadratic functions using this implementation.

14.9 3-OA CGIC Biquadratic Structure

Consider the circuit shown in Figure 14.12 where the output can be taken to be V_1, V_2, or V_3. Assuming ideal OAs, the transfer functions between the input and output terminals 1, 2, and 3 can be readily obtained as

$$\begin{aligned} V_1/V_i &= T_1 \\ &= \{Y_{11}Y_{12}[Y_1(1 + Y_4/Y_9) - Y_2Y_3/Y_9] \\ &\quad + (Y_{12}Y_3Y_7Y_8)/Y_9 + [(1 + Y_4/Y_9)Y_5 - Y_3Y_6/Y_9]Y_7Y_{10}\}/D(s) \end{aligned} \tag{14.8a}$$

* This can readily be understood by examining the different realizations to be obtained from this layout.

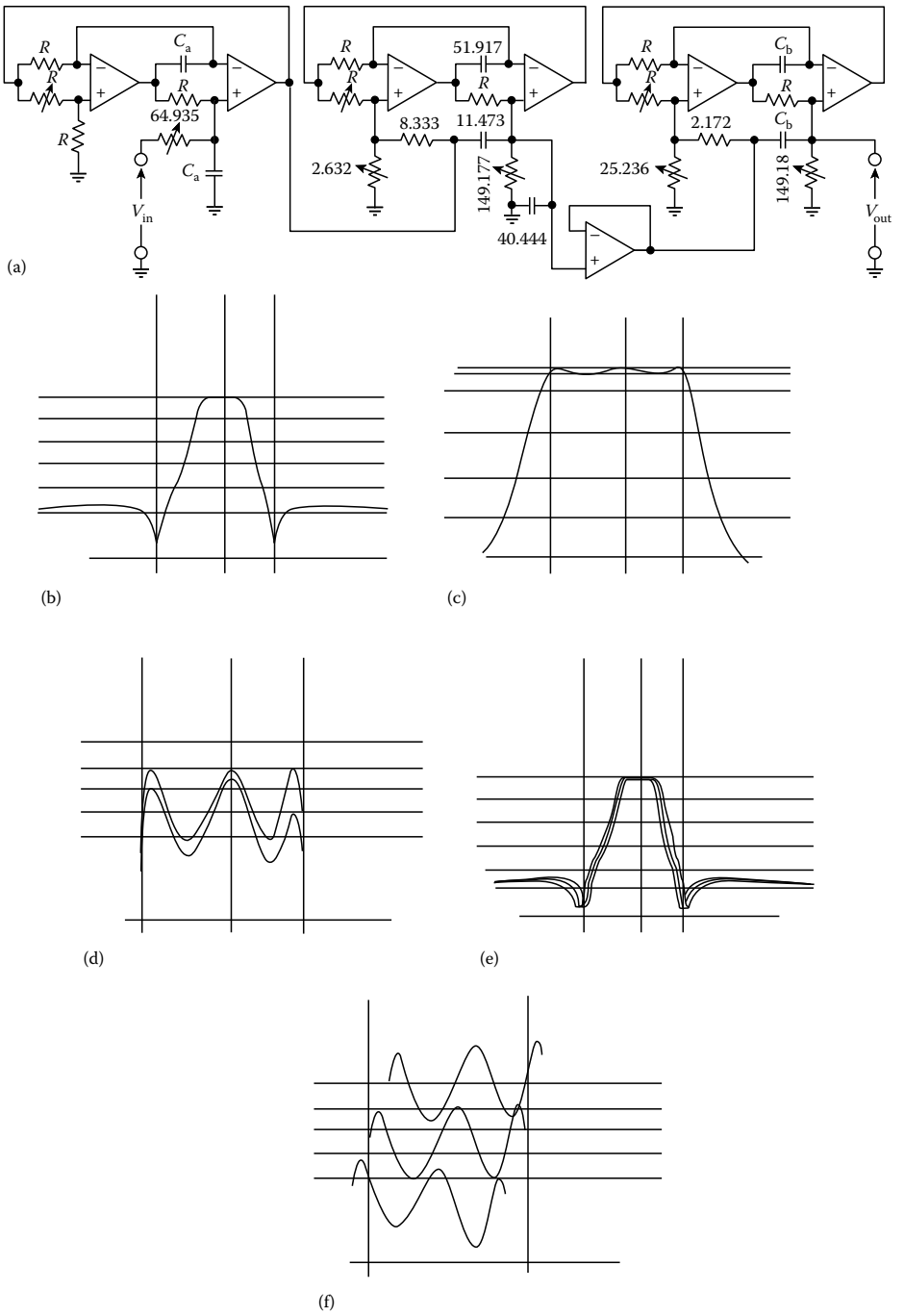


FIGURE 14.8 (a) Realization of the sixth-order elliptic bandpass filter. Frequency responses of bandpass filter: (b) logarithmic gain and linear frequency scales; (c) linear gain and frequency scales; (d) for supply voltages of ±7.5 V (lower curve) and ±15 V, input level of 0.3 V; (e) at temperatures of 10°C (right-hand curve), 20°C, and 70°C (left-handed curve); and (f) expanded passband of Figure 14.8e.

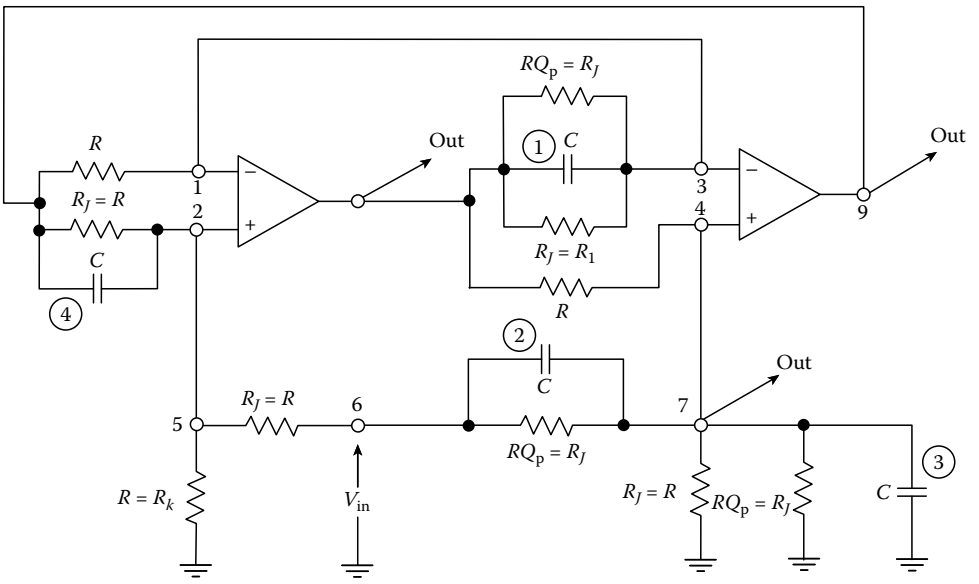


FIGURE 14.9 Superposition of circuits 1, 3, 7, and 10 from Table 14.2. Note: Out = output terminals.

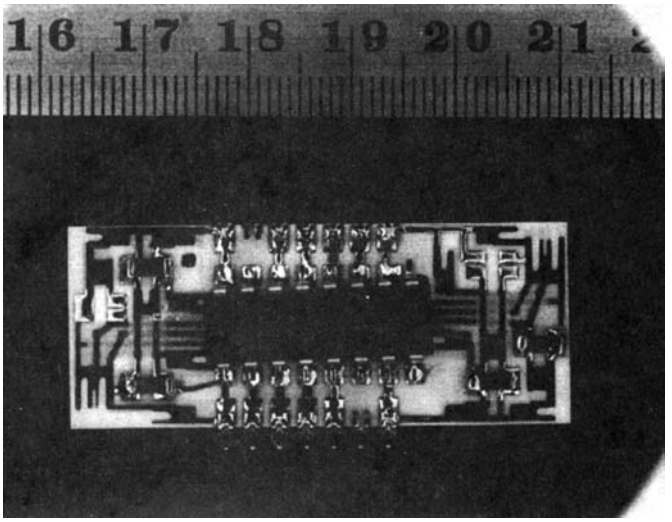


FIGURE 14.10 Dual CGIC universal biquad implemented using thick-film resistors; chip NPO capacitors and one quad OA.

$$V_2/V_i = T_2$$
$$= \{Y_{11}Y_{12}Y_1 + Y_{12}[Y_1Y_6 + (Y_3Y_7Y_8)/Y_9 - Y_5Y_6] + [Y_3Y_6 - Y_4Y_5](Y_7Y_8Y_{12})/Y_9Y_{11} + Y_5Y_7Y_{10}\}/D(s) \quad (14.8b)$$

$$V_3/V_i = T_3$$
$$= \{Y_1Y_{11}Y_{12} + Y_{12}[(1 + Y_2/Y_7)Y_3 - Y_1Y_4/Y_7]Y_8/Y_9 + [(1 + Y_2/Y_7)Y_5 - Y_1Y_6/Y_7]Y_7Y_{10}\}D(s) \quad (14.8c)$$

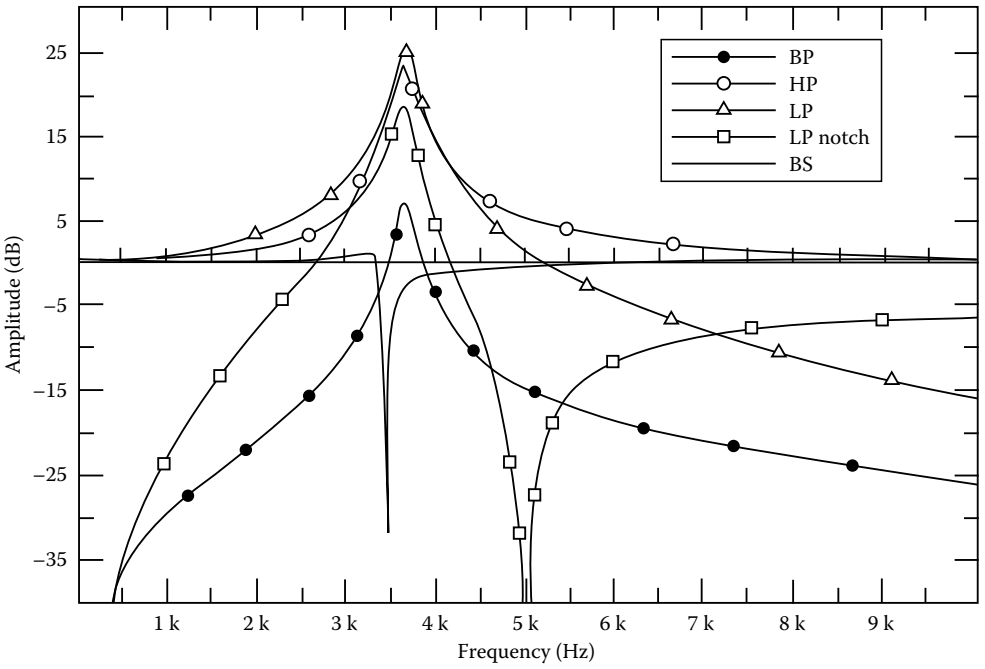


FIGURE 14.11 Different second-order realizations obtained from the universal CGIC biquad.

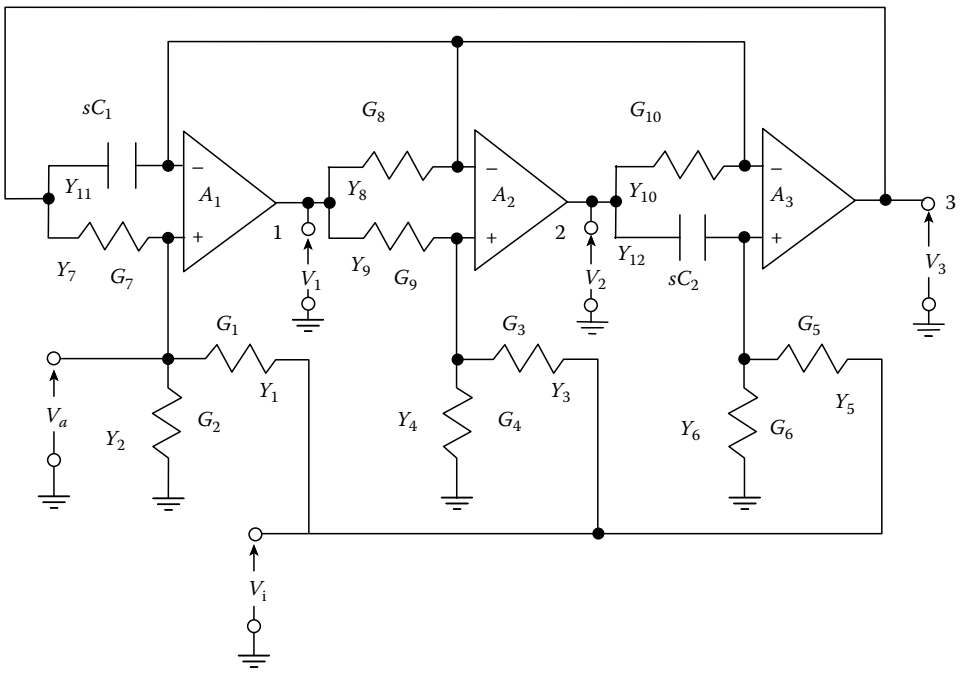


FIGURE 14.12 3-OA CGIC biquad.

where

$$D(s) = Y_{11}Y_{12}(Y_1 + Y_2) + Y_{12}(Y_3 + Y_4)Y_7Y_8/Y_9 + (Y_5 + Y_6)Y_7Y_{10} \tag{14.9}$$

The admittances Y_1–Y_{12} can be selected in many ways and any stable second-order transfer function can be realized. For the purposes of this chapter, Y_1–Y_{10} are taken to be purely conductive, while Y_{11} and Y_{12} are purely capacitive, that is,

$$\begin{aligned} Y_1 - Y_{10} &= G_1 - G_{10} \\ Y_{11} &= sC_1, \quad Y_{12} = sC_2 \end{aligned} \tag{14.10}$$

Any rational and stable transfer function can be expressed as a product of second-order transfer functions of the form

$$T(s) = \frac{a_2s^2 + a_1s + a_0}{b_2s^2 + b_1s + b_0} \tag{14.11}$$

where $a_1 = a_2 = 0$, $a_0 = a_1 = 0$, $a_0 = a_2 = 0$, or $a_1 = 0$, for an LP, HP, BP, or N section, respectively. These section can be realized by choosing the G_i's ($i = 1$–10) properly in Equation 14.8. By comparing Equations 14.8 through 14.11, circuits 1–4 in Table 14.3 can be obtained.

All-pass transfer functions can be realized by setting $a_2 = b_2$, $a_1 = -b_1$, $a_0 = b_0$; these can be obtained from circuit 5 of Table 14.3.

It can be easily shown that this biquad possesses similar excellent low sensitivity properties and stability during activation as those of the 2-OA CGIC biquad, given in Section 14.4.

14.10 Design and Tuning Procedure of the 3-OA CGIC Biquad

Several degrees of freedom exist in choosing element values, as shown in Table 14.3. These are used to satisfy the constraints of the given design, namely, those of low sensitivity, reduced dependence on the

TABLE 14.3 Element Identification for Realizing the Most Commonly Used Transfer Functions

Circuits Number	G_1	G_2	G_3	G_4	G_5	G_6	Transfer Function	Remarks
1	0		0			[a]	$T_3 = \dfrac{\left(1 + \frac{G_2}{G_7}\right)\frac{G_5G_7G_{10}}{C_1C_2}}{s^2G_2 + s\frac{G_4G_7G_8}{C_1G_9} + \frac{(G_5 + G_6)G_7G_{10}}{C_1C_2}}$	LP
2		[a]	0		0		$T_1 = \dfrac{s^2G_1\left(1 + \frac{G_4}{G_9}\right)}{s^2(G_1 + G_2) + s\frac{G_4G_7G_8}{C_1G_9} + \frac{G_6G_7G_{10}}{C_1C_2}}$	HP
3	0			[a]	0		$T_3 = \dfrac{s\left(1 + \frac{G_2}{G_7}\right)\frac{G_3G_7G_8}{C_1G_9}}{s^2G_2 + s\frac{(G_3 + G_4)G_7G_8}{C_1G_9} + \frac{G_6G_7G_{10}}{C_1C_2}}$	BP
4		[a]	0			[a]	$T_1 = G_1\left(1 + \dfrac{G_4}{G_9}\right)\dfrac{s^2 + \frac{G_5G_7G_{10}}{C_1C_2G_1}}{s^2(G_1 + G_2) + s\frac{G_4G_7G_8}{C_1G_9} + \frac{(G_5 + G_6)G_7G_{10}}{C_1C_2}}$	N
5		[a]	0			0	$T_3 = \dfrac{s^2G_1 - s\frac{G_1G_4G_8}{C_1G_9} + \left(1 + \frac{G_2}{G_7}\right)\frac{G_5G_7G_{10}}{C_1C_2}}{s^2(G_1 + G_2) + s\frac{G_4G_7G_8}{C_1G_9} + \frac{G_5G_7G_{10}}{C_1C_2}}$	Nonminimum phase[b]

Notes: $Y_7 - Y_{10} = G_7 - G_{10}$ always.
$Y_{11} - sC_1$, $Y_{12} = sC_2$.

[a] These elements can be set to zero.

[b] For all-pass $G_2 = 0$, $G_7 = G_1$.

TABLE 14.4 Design Values and Tuning Procedures

Circuit Number	Design Values	Transfer Function Realized	Tuning Sequence			
			ω_n	ω_p	Q_p	Q_z
1	$R_2 = R$, $R_4 = RQ_p^{1/2}$, $R_5 = 2R/(\alpha H_{LP})$, $R_6 = R/[\alpha(1 - H_{LP}/2)]$, $H_{LP} < 2$	$T_3 = H_{LP}\dfrac{\omega_p^2}{D(s)}$	—	R_5	R_4	—
2	$R_1 = R\left(1 + \bar{Q}_p^{1/2}\right)/H_{HP}$, $R_2 = R\left(1 + \bar{Q}_p^{1/2}\right)\left[1 + \bar{Q}_p^{1/2} - H_{HP}\right]$, $R_4 = RQ_p^{1/2}$, $R_6 = R/\alpha$, $H_{HP} < 1 + \bar{Q}_p^{1/2}$	$T_1 = H_{HP}\dfrac{s^2}{D(s)}$	—	R_6	R_4	—
3	$R_2 = R$, $R_3 = 2RQ_p^{1/2}H_{BP}$, $R_4 = RQ_p^{1/2}/1 - H_{BP}/2$, $R_6 = R/\alpha$, $-H_{BP} < 2$	$T_3 = H_{BP}\dfrac{\frac{s\omega_p}{Q_p}}{D(s)}$	—	R_6	R_3	—
4	$R_1 = R\left(1 + \bar{Q}_p^{1/2}\right)/H_N$, $R_2 = 1/(G - G_1)$, $R_4 = R\bar{Q}_p^{1/2}$, $R_6 = 1/(\alpha G - G_5)$, $R_5 = R\omega_p^2\left(1 + \bar{Q}_p^{1/2}\right)/\left(\alpha H_N \omega_n^2\right)$, $H_N < \left(1 + \bar{Q}_p^{1/2}\right)$, $H_N =< \omega_p^2\left(1 + \bar{Q}_p^{1/2}\right)/\omega_n^2$ for $\omega_n > \omega_p$	$T_1 = H_N\dfrac{\left(s^2 + \omega_n^2\right)}{D(s)}$	R_5	R_6	R_4	—
5	For all-pass: $R_1 = R$, $R_4 = RQ_p^{1/2}$ $R_5 = R/\alpha$, $R_1 = R_7$, $R_2 = \infty$, $H_{AP} = 1$	$T_1 = H_{AP}\dfrac{s^2 - \frac{\omega_z}{Q_z}s + \omega_z^2}{D(s)}$	R_5	R_4		

Notes: $D(s) = s^2 + (\omega_p/Q_p)s + \omega_p^2$, $\alpha = 2Q_p^{1/2}/\left(1 + Q_p^{1/2}\right)$.
$C_1 = C_2 = C = 1/(\omega_p R)$, $R_{10} = \alpha R$, $R_8 = RQ_p^{1/2}$, $R_7 = R_8 = R$.

OA finite gain and bandwidth, and low element-spread design, in circuits 1–5 in Table 14.3. Using these constraints, a possible design for LP, HP, BP, AP, and N sections is obtained, as indicated in Table 14.4. It is seen that Q_p, ω_p, ω_n, Q_z, and ω_z, can be independently adjusted by trimming at most three resistors. A trimming sequence is also given in Table 14.4.

14.11 Practical Sixth-Order Elliptic BP Filter Design Using the 3-OA CGIC Biquad

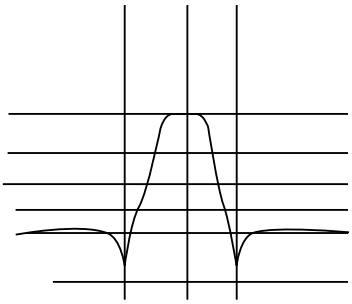
The sixth-order elliptic bandpass filter specified in Section 14.7 was designed using the 3-OA CGIC biquad and similar components to those in Section 14.7.

The realization shown in Figure 14.13a uses cascaded sections of the types 3 and 4 in Table 14.4. The element design values are also given in Figure 14.13a. The measured frequency response is shown in Figure 14.13b and c; it is in agreement with the theoretical response. Figure 14.13d shows the frequency response for supply voltages of ±7.5 V (lower curve), and ±15 V (upper curve); the input voltage is 0.3 V. The passband ripple remains less than 0.34 dB and the deviation in the stopband is negligible. Figure 14.13e and f illustrate the effect of temperature variations. The passband ripple remains less than 0.5 dB in the temperature range from −10°C (right-hand curve) to 70°C (left-hand curve). A center frequency displacement of 9 Hz has been measured, which corresponds to a change of 75 ppm/°C.

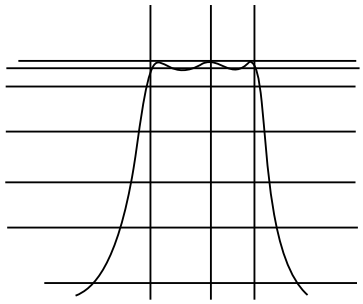
These results illustrate the additional performance improvements compared with the results in Section 14.7 using the 2-OA CGIC biquad.

Section No	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	C_1	C_2
1	∞	2	11.396	∞	∞	1.175	2	11.396	2	3.403	53.052	53.052
2	8.939	2.576	∞	17.273	4.592	1.426	2	17.273	2	3.585	51.917	51.917
3	4.309	3.732	∞	17.273	2.611	1.948	2	17.273	2	3.585	54.211	54.211

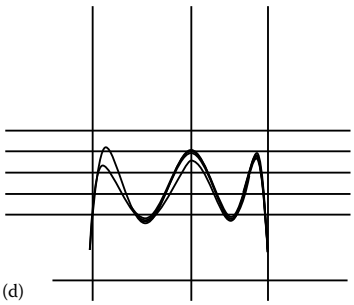
(a) Resistors in kilo ohms and capacitors in nanofarads



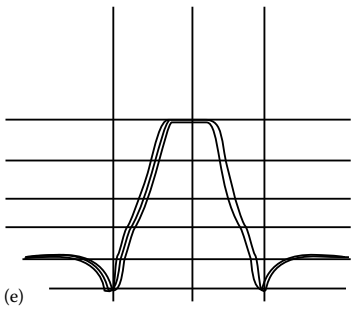
(b)



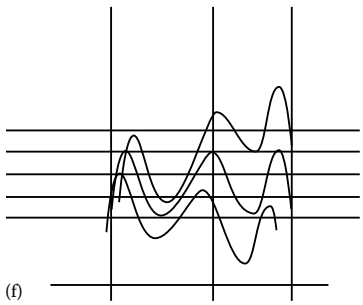
(c)



(d)



(e)



(f)

FIGURE 14.13 (a) Sixth-order elliptic bandpass filter. Resistors in kiloohms and capacitors in nanofarads. (b)–(f) Frequency response using 3-OA CGIC. (b) Logarithmic gain and linear frequency scales; (c) linear gain and frequency scales; (d) frequency response for supply voltages of ±7.5 V (lower curve) and ±15 V, input level of 0.3 V; (e) frequency response; and (f) at temperatures of −10°C (right-hand curve), and 20°C and 70°C (left-hand curve).

14.12 Composite GIC Biquad: 2 C 2-OA CGIC

To obtain the composite CGIC biquad [3], each single OA in the original 2-OA CGIC biquad is replaced by a composite amplifier each constructed using two regular OAs [2] and denoted C2OA. All possible combinations of the four C2OA structures in Ref. [3] were used to replace the two OAs in the CIC network. Although several useful combinations were obtained, it was found that the best combination is shown in Figure 14.14, where A_1 is replaced by C2OA-4 and A_2 is replaced by C2OA-3 in the CGIC of Figure 14.1.

Computer simulation plots and experimental results of Figure 14.15 show clearly the considerable improvements of the new CGIC filter responses over those of a 2-OA CGIC implemented using regular OAs.

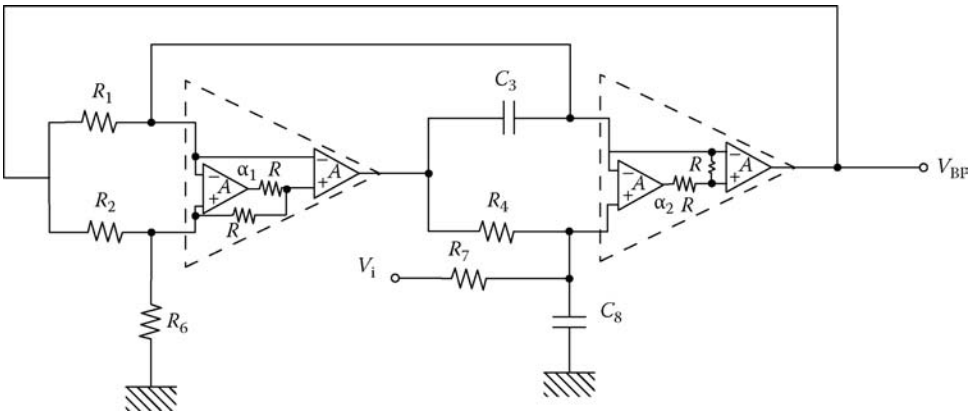
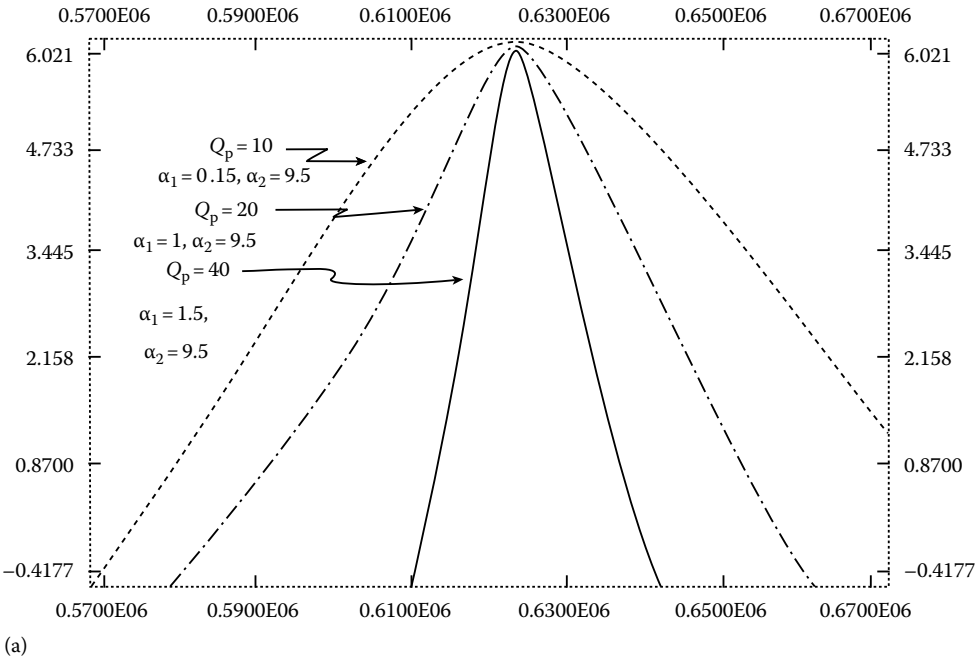


FIGURE 14.14 Practical BP filter realization of the composite GIC using C2OA-4 and C2OA-3.



(a)

FIGURE 14.15 (a) Computer plots of the composite GIC BP filter frequency responses of Figure 14.14 for $f_0 = 100$ kHz and $Q_p = 10, 20, 40$.

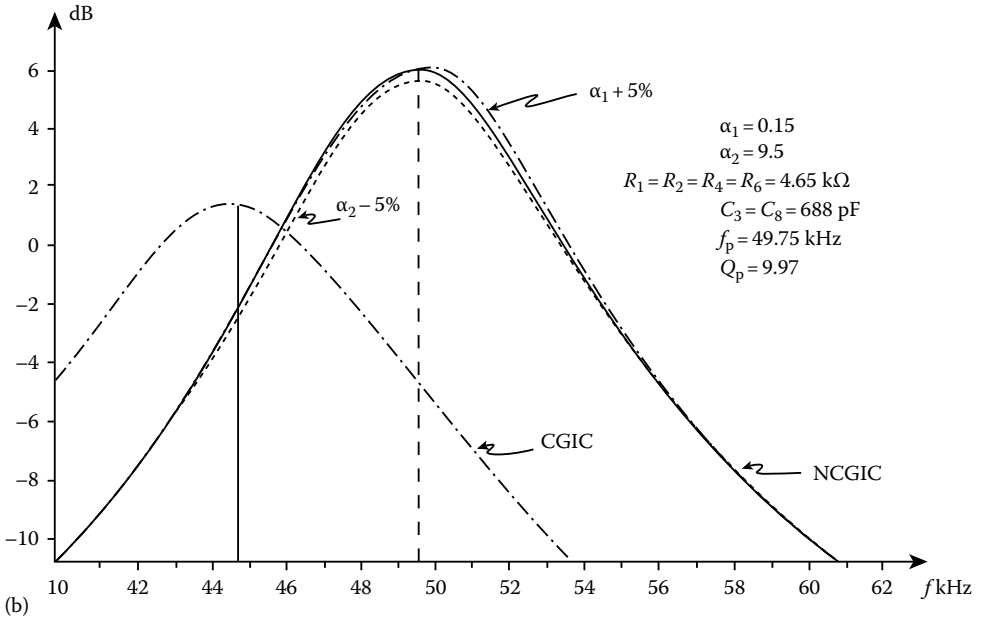


FIGURE 14.15 (continued) (b) Experimental results of single and composite GIC BP filters of Figure 6.3, and the response sensitivity for a_1 and a_2 variations ($f_0 = 50$ kHz, $Q_p = 10$).

Similar improvements in inductance simulation applications, employing the 2C2-OA CGIC, have been reported.

14.13 Summary

Two configurations have been presented for the synthesis of *RC*-active networks. These have been used to design a number of universal second-order sections such as low-pass, high-pass, bandpass, and bandstop sections. By using these sections, most of the practical filter specifications can be realized. Each section employs a CGIC, which can be implemented by using 2 OAs in the 2-OA CGIC biquad and 3 OAs in the 3-OA CGIC design. The sensitivities of Q_p, ω_p, Q_z, ω_z, and also the multiplier constant of the realization, have been found to be low with respect to the passive- and active-element variations. A simple functional tuning sequence in which only resistors are adjusted has been described. With the exception of one of the notch sections using the 2-OA CGIC, of all the circuits considered, the output can be located at an OA output. Consequently, these sections can be cascaded without the need for isolating amplifiers. Although exact matching of the gain bandwidth products of the OAs in the CGIC biquad is not essential, optimum results are obtained when they are matched within practical ranges. This is achieved easily by using dual or quad OAs. Also, the choice of the element values for optimum performance is also given. The additional performance improvements using the 3-OA CGICiquads are illustrated experimentally. Useful operating frequencies extension employing the composite operational amplifier technique in the 2-OA CGIC biquad is also given.

References

1. W. B. Mikhael and B. B. Bhattacharyya, A practical design for insensitive *RC*-active filters, *IEEE Trans. Circuits Syst.*, CAS-21, 75–78, Jan. 1974.
2. W. B. Mikhael and S. Michael, Composite operational amplifiers: Generations and finite gain applications, *IEEE Trans. Circuit Syst.*, CAS-34, 449–460, May 1987.

3. S. Micheal and W. B. Mikhael, High frequency filtering and inductance simulation using new composite generalized immittance converters, in *Proc. IEEE ISCAS*, Kyoto, Japan, June 1985, pp. 299–300.
4. A. Antoniou, Realization of gyrators using operational amplifiers and their use in *RC* active network synthesis, *Proc. IEEE*, 116, 1838–1850, Nov. 1969.
5. J. Valihora, Modern technology applied to network implementation, in *Proc. IEEE Int. Symp. Circuit Theory*, Los Angeles, CA, Apr. 1972, pp. 169–173.
6. A. Antoniou, Stability properties of some gyrator circuits, *Electron. Lett*., 4, 510–512, 1968.
7. B. B. Bhattacharyya, W. B. Mikhael, and A. Antoniou, Design of *RC*-active networks by using generalized immittance converters, in *Proc. IEEE Int. Symp. Circuit Theory*, Toronto, Canada, Apr. 1973, pp. 290–294.

15

High-Order Filters

15.1 Introduction **15**-1
15.2 Cascade Realizations **15**-2
15.3 Multiple-Loop Feedback Realizations **15**-6
Follow-the-Leader Feedback Topology • Leapfrog Topology
15.4 Ladder Simulations **15**-12
Signal-Flow Graph Methods • Maximization of the Dynamic Range • Element Substitution
15.5 Summary **15**-27
References **15**-28

Rolf Schaumann
Portland State University

15.1 Introduction

With the realization of second-order filters discussed in the previous chapters of this section, we will now treat methods for practical filter implementations of order higher than two. Specifically, we will investigate how to realize efficiently, with low sensitivities to component tolerances, the input-to-output voltage transfer function

$$H(s) = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{N(s)}{D(s)} = \frac{a_m s^m + a_{m-1}s^{m-1} + \cdots + a_1 s + a_0}{s^n + b_{n-1}s^{n-1} + \cdots + b_1 s + b_0} \tag{15.1}$$

where $n \geq m$ and $n > 2$. The sensitivity behavior of high-order filter realizations shows that, in general, it is not advisable to realize the transfer function $H(s)$ in the so-called direct form [5, Chapter 3] (see also Chapter 5). By direct form we mean an implementation of Equation 15.1 that uses only one or maybe two active devices, such as operational amplifiers (op-amps) or operational transconductance amplifiers (OTAs), embedded in a high-order passive *RC* network. Although it is possible in principle to realize Equation 15.1 in direct form, the resulting circuits are normally so sensitive to component tolerances as to be impractical. Since the direct form for the realization of high-order functions is ruled out, in this section we present those methods that result in designs of practical manufacturable active filters with acceptably low sensitivity, the *cascade* approach, the *multiple-loop feedback topology,* and *ladder simulations*. Both cascade and multiple-loop feedback techniques are modular, with active biquads used as the fundamental building blocks. The ladder simulation method seeks active realizations that inherit the low passband sensitivity properties of passive doubly terminated *LC* ladder filters (see Chapter 9).

In the cascade approach, a high-order function $H(s)$ is factored into low (first or second) order subnetworks, which are realized as discussed in the previous chapters of this section and connected in cascade such that their product implements the prescribed function $H(s)$. The method is widely

employed in industry; it is well understood, very easy to design, and efficient in its use of active devices. It uses a modular approach and results in filters that, for the most part, show satisfactory performance in practice. The main advantage of cascade filters is their generality, i.e., any arbitrary stable transfer function can be realized as a cascade circuit, and tuning is very easy because each biquad is responsible for the realization of only one pole pair (and zero pair): the realizations of the individual critical frequencies of the filter are decoupled from each other. The disadvantage of this decoupling is that for filters of high order, say $n > 8$, with stringent requirements and tight tolerances, the passband sensitivity of cascade designs to component variations is often found to remain still too sensitive. In these cases, the following approaches lead to more reliable circuits.

The multiple-loop feedback or coupled-biquad methods also split the high-order transfer function into second-order subnetworks. These are interconnected in some type of feedback configuration that introduces coupling chosen to reduce the transfer function sensitivities. The multiple-loop feedback approach retains the modularity of cascade designs but at the same time yields high-order filter realizations with noticeably better passband sensitivities. Of the numerous topologies that have been proposed in the literature, see, e.g., Ref. [5, Chapter 6], we discuss only the follow-the-leader feedback (FLF) and the LF (leapfrog) methods. Both are particularly well suited for all-pole characteristics but can be extended to realizations of general high-order transfer functions. Although based on coupling of biquads in a feedback configuration, the LF procedure is actually derived from an *LC* ladder simulation and will, therefore, be treated as part of that method.

As the name implies, the ladder simulation approach uses an active circuit to simulate the behavior of doubly terminated *LC* ladders in an attempt to inherit their excellent low passband-sensitivity properties. The methods fall into two groups. One is based on element replacement or substitution, where the inductors are simulated via electronic circuits whose input impedance is inductive over the necessary frequency range; the resulting active ''components'' are then inserted into the *LC* filter topology. The second group may be labeled operational simulation of the *LC* ladder, where the active circuit is configured to realize the internal operation, i.e., the equations, of the *LC* prototype. Active filters simulating the structure or the behavior of *LC* ladders have been found to have the lowest passband sensitivities among active filters and, consequently, to be the most appropriate if filter requirements are stringent. They can draw on the wealth of information available for lossless filters, e.g., Refs. [5, Chapter 2], [9, Chapter 13], [7] that can be used directly in the design of active ladder simulations. A disadvantage of this design method is that a passive *LC* prototype must, of course, exist* before an active simulation can be attempted.

15.2 Cascade Realizations

Without loss of generality we may assume in our discussion of active filters that the polynomials $N(s)$ and $D(s)$ of Equation 15.1 are even, i.e., both n and m are even. An odd function can always be factored into the product of an even function and a first-order function, where the latter can easily be realized by a passive *RC* network and can be appended to the high-order active filter as an additional section. Thus, we can factor Equation 15.1 into the product of second-order pole-zero pairs, so that the high-order transfer function $H(s)$ is factored into the product of the second-order functions

$$T_i(s) = k_i \frac{\alpha_{2i}s^2 + \alpha_{1i}s + \alpha_{0i}}{s^2 + s\omega_{0i}/Q_i + \omega_{0i}^2} = k_i t_i(s) \tag{15.2}$$

* The realizability conditions for passive *LC* filters are more restrictive than those for active *RC* filters; specifically, the numerator $N(s)$ of Equation 16.1 must be strictly even or odd so that only $j\omega$-axis transmission zeros can be realized.

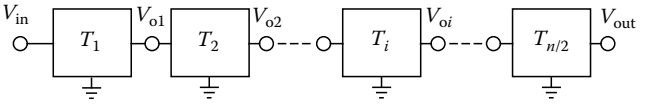


FIGURE 15.1 Cascade realization of an *n*th-order transfer function; *n* is assumed even.

such that

$$H(s) = \prod_{i=1}^{n/2} T_i(s) = \prod_{i=1}^{n/2} k_i \frac{\alpha_{2i}s^2 + \alpha_{1i}s + \alpha_{0i}}{s^2 + s\omega_{0i}/Q_i + \omega_{0i}^2} = \prod_{i=1}^{n/2} k_i t_i(s) \tag{15.3}$$

In Equation 15.2, ω_{0i} is the pole frequency and Q_i the pole quality factor; the coefficients α_{2i}, α_{1i}, and α_{0i} determine the type of second-order function $T_i(s)$ that can be realized by an appropriate choice of biquad from the literature, e.g., Refs. [5, Chapter 5], [9, Chapters 4 and 5] (see also Chapters 12 through 14). The transfer functions $T_i(s)$ of the individual biquads are scaled by a suitably defined gain constant k_i, e.g., such that the leading coefficient in the numerator of the gain-scaled transfer function $t_i(s)$ is unity or such that $|t_i(j\omega)| = 1$ at some desired frequency. If we assume now that the output impedances of the biquads are small compared to their input impedances, all second-order blocks can be connected in cascade as in Figure 15.1 without causing mutual interactions due to loading. The product of the biquadratic functions is then realized as required by Equation 15.3.

The process is straightforward and leads to many possibilities for cascading the identified circuit blocks, but several questions must still be answered:

1. Which pair of zeros of Equation 15.1 should be assigned to which pole-pair when the biquadratic functions $T_i(s)$ are formed? Since we have $n/2$ pole pairs and $n/2$ zero pairs (counting zeros at 0 and at ∞) we can select from $(n/2)!$ possible pole-zero pairings.
2. In which order should the biquads be cascaded? For $n/2$ biquads, we have $(n/2)!$ possible ways of section ordering.
3. How should the gain constants k_i in Equation 15.2 be chosen to determine the signal level for each biquad? In other words, what is the optimum gain assignment?

Because the total prescribed transfer function $H(s)$ is simply the product of $T_i(s)$, the choices in 1, 2, and 3 are quite arbitrary as far as $H(s)$ is concerned. However, they determine significantly the dynamic range,* i.e., the distance between the maximum possible undistorted signal (limited by the active devices) and the noise floor, because the maximum and minimum signal levels throughout the cascade filter depend on the pole-zero pairings, cascading sequence, and gain assignment.

There exist well-developed techniques and algorithms for answering the questions of pole-zero pairing, section ordering, and gain assignment exactly [3, Chapter 1], [5, Chapter 6]; they rely heavily on computer routines and are too lengthy to be treated in this chapter. We will only give a few rules-of-thumb, which are based on the intuitive observation that in the passband the magnitude of each biquad should vary as little as possible. This keeps signal levels as equal as possible versus frequency and avoids in-band attenuation and the need for subsequent amplification that will at the same time raise the noise floor and thereby further limit the dynamic range. Note also that in-band signal amplification may overdrive the amplifier stages and cause distortion. The simple rules below provide generally adequate designs.

* Pole-zero pairing also affects to some extent the sensitivity performance, but the effect usually is not very strong and will not be discussed in this chapter. For a detailed treatment see Ref. [3].

1. Assign each zero or zero-pair to the closest pole-pair.
2. Sequence the sections in the order of increasing values of Q_i, i.e., $Q_1 < Q_2 < \cdots < Q_{n/2}$, so that the section with the flattest transfer function magnitude comes first, the next flattest one follows, and so on. If the position of any section is predetermined,* use the sequencing rule for the remaining sections.

After performing steps (1) and (2), assign the gain-scaling constants k_i such that the maximum output signals of all sections in the cascade are the same, i.e.,

$$\max|V_{oi}(j\omega)| = \max\left|V_{o,n/2}(j\omega)\right| = \max|V_{out}(j\omega)| \quad i = 1, \ldots, n/2 - 1 \tag{15.4}$$

This can be done readily with the help of a network analysis routine or simulator, such as SPICE, by computing the output signals at each biquad: using the notation, from Equation 15.3,

$$H(s) = \prod_{i=1}^{n/2} T_i(s) = \prod_{i=1}^{n/2} k_i t_i(s) = \prod_{i=1}^{n/2} k_i \prod_{i=1}^{n/2} t_i(s) \tag{15.5}$$

and

$$H_i(s) = \prod_{j=1}^{i} T_j(s) = \prod_{j=1}^{i} k_j t_j(s) = \prod_{j=1}^{i} k_j \prod_{j=1}^{i} t_j(s) \tag{15.6}$$

we label the total prescribed gain constant

$$K = \prod_{i=1}^{n/2} k_i \tag{15.7}$$

such that

$$\max\left|\prod_{i=1}^{n/2} t_i(j\omega)\right| = M_{n/2} \tag{15.8a}$$

is some given value. Further, let us denote the maxima of the intermediate gain-scaled transfer functions by M_i, i.e.,

$$\max\left|\prod_{k=1}^{i} t_k(j\omega)\right| = M_i \quad i = 1, \ldots, n/2 - 1 \tag{15.8b}$$

then we obtain [5, Chapter 6]

$$k_1 = K\frac{M_{n/2}}{M_1} \quad \text{and} \quad k_j = \frac{M_{j-1}}{M_j} \quad j = 2, \ldots, n/2 \tag{15.9}$$

* Such as, e.g., placing a lowpass at the input may be preferred to avoid unnecessary high-frequency signals from entering the filter.

Choosing the gain constants as in Equation 15.9 guarantees that the same maximum voltage appears at all biquad outputs to assure that the largest possible signal can be processed without distortion.

To illustrate the process, let us realize the sixth-order transfer function

$$H(s) = \frac{0.7560s^3}{(s^2 + 0.5704s + 1)(s^2 + 0.4216s + 2.9224)(s^2 + 0.1443s + 0.3422)} \tag{15.10}$$

where the frequency parameter s is normalized with respect to $\omega_n = 130{,}590\ \text{s}^{-1}$. It defines a sixth-order bandpass filter with a 1 dB equiripple passband in 12 kHz $\leq f \leq$ 36 kHz and at least 25 dB attenuation in $f \leq 4.8$ kHz and $f \geq 72$ kHz [5, Chapter 1]. $H(s)$ can be factored into the product of

$$T_1(s) = k_1 t_1(s) = \frac{k_1 s}{s^2 + 0.5704s + 1} \quad \text{with}$$

$$Q_1 = \frac{\sqrt{1}}{0.5704} \approx 1.75 \tag{15.11a}$$

$$T_2(s) = k_2 t_2(s) = \frac{k_2 s}{s^2 + 0.4216s + 2.9224} \quad \text{with}$$

$$Q_2 = \frac{\sqrt{2.9224}}{0.4216} \approx 4.06 \tag{15.11b}$$

$$T_3(s) = k_3 t_3(s) = \frac{k_3 s}{s^2 + 0.1443s + 0.3422} \quad \text{with}$$

$$Q_3 = \frac{\sqrt{0.3422}}{0.1443} \approx 4.06 \tag{15.11c}$$

which are cascaded in the order of increasing values of Q, i.e., $T_1T_2T_3$. We can compute readily the maximum values at the output of the sections in the cascade as $M_1 = |t_1|_{\max} \approx 1.75$, $M_2 = |t_1t_2|_{\max} \approx 1.92$, and $M_3 = |t_1t_2t_3|_{\max} \approx 1.32$ to yield by Equation 15.9

$$k_1 = 0.7560\frac{1.32}{1.75} \approx 0.57 \quad k_2 = \frac{1.75}{1.92} \approx 0.91 \quad k_3 = \frac{1.92}{1.32} \approx 1.45 \tag{15.12}$$

Let us build the sections with the Åckerberg–Mossberg circuit [9, Chapter 4.4] shown in Figure 15.2. It realizes

$$\frac{V_o}{V_i} = -\frac{k\omega_0^2}{s^2 + s\omega_0/Q + \omega_0^2} \quad \text{with } \omega_0 = \frac{1}{RC} \tag{15.13}$$

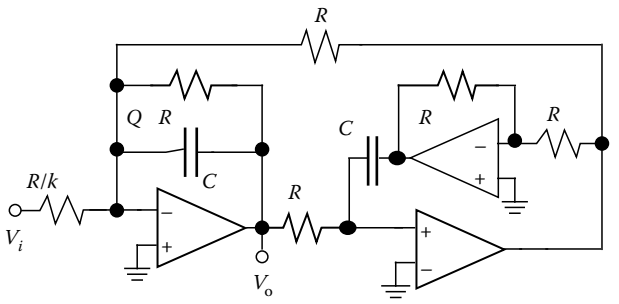


FIGURE 15.2 Åckerberg–Mossberg bandpass circuit.

We obtain readily from Equations 15.11 through 15.13 the following (rounded) component values:

$$\text{for } T_1\text{: } R = 1.5\ \text{k}\Omega \quad QR = 2.7\ \text{k}\Omega \quad R/k = 2.7\ \text{k}\Omega$$

$$\text{for } T_2\text{: } R = 0.9\ \text{k}\Omega \quad QR = 3.6\ \text{k}\Omega \quad R/k = 1.7\ \text{k}\Omega$$

$$\text{for } T_3\text{: } R = 2.6\ \text{k}\Omega \quad QR = 10\ \text{k}\Omega \quad R/k = 1\ \text{k}\Omega$$

The three sections are then interconnected in cascade in the order $T_1T_2T_3$.

15.3 Multiple-Loop Feedback Realizations

These topologies are also based on biquad building blocks, which are then embedded, as the name implies, into multiple-loop resistive feedback configurations. The resulting coupling between sections is selected such that transfer function sensitivities are reduced below those of cascade circuits. It has been shown that the sensitivity behavior of the different available configurations is comparable. We shall, therefore, concentrate our discussion only on the FLF and, as part of the ladder simulation techniques, on the leapfrog (LF) topologies, which have the advantage of being relatively easy to derive without any sacrifice in performance. Our derivation will reflect the fact that both configurations* are particularly convenient for geometrically symmetrical bandpass functions and that the LF topology is obtained from a direct simulation of an *LC* lowpass ladder.

15.3.1 Follow-the-Leader Feedback Topology

The FLF topology consists of a cascade of biquads whose outputs are fed back into a summer at the filter's input. At the same time, the biquad outputs may be fed forward into a second summer at the filter's output to permit an easy realization of arbitrary transmission zeros. The actual implementation of the summers and the feedback factors is shown in Figure 15.3; if there are n noninteracting biquads, the order of the realized transfer function $H(s)$ is $2n$. Assuming that the two summer op-amps are ideal, routine analysis yields

$$-V_0 = \frac{R_{F0}}{R_{\text{in}}} V_{\text{in}} + \sum_{i=1}^{n} \frac{R_{F0}}{R_{Fi}} V_i = \alpha V_{\text{in}} + \sum_{i=1}^{n} F_i V_i \tag{15.14}$$

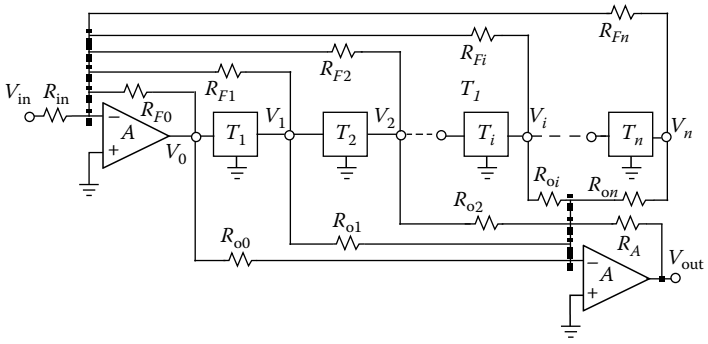


FIGURE 15.3 FLF circuit built from second-order sections $T_i(s)$ and a feedback network consisting of an op-amp summer with resistors R_{Fi}. Also shown is an output summer with resistors R_{oi} to facilitate the realization of arbitrary transmission zeros.

* As are all other multiple-loop feedback circuits.

where we defined $\alpha = R_{F0}/R_{\text{in}}$ and the feedback factors $F_i = R_{F0}/R_{Fi}$. Similarly, we find for the output summer

$$V_{\text{out}} = -\sum_{i=0}^{n} K_i V_i = -\sum_{i=0}^{n} \frac{R_A}{R_{\text{o}i}} V_i \tag{15.15}$$

from which the definition of the resistor ratios K_i is apparent. Any of the parameters F_i and K_i may, of course, be reduced to zero by replacing the corresponding resistor, R_{Fi} or $R_{\text{o}i}$, respectively, by an open circuit. Finally, the internal voltages V_i can be computed from

$$V_i = V_0 \prod_{j=1}^{i} T_j(s) \quad i = 1, \ldots, n \tag{15.16}$$

so that with Equation 15.14

$$H_0(s) = \frac{V_0}{V_{\text{in}}} = -\frac{\alpha}{1 + \sum_{k=1}^{n} \left[F_k \prod_{j=1}^{k} T_j(s) \right]} \tag{15.17}$$

which with Equation 15.16 yields

$$H_i(s) = \frac{V_i}{V_{\text{in}}} = -\frac{N_i(s)}{D(s)} = -\frac{\alpha \prod_{j=1}^{i} T_j(s)}{1 + \sum_{k=1}^{n} \left[F_k \prod_{j=1}^{k} T_j(s) \right]} \quad i = 1, \ldots, n \tag{15.18}$$

Note that

$$H_n(s) = \frac{V_n}{V_{\text{in}}} = -\frac{N_n(s)}{D(s)} = -\frac{\alpha \prod_{j=1}^{n} T_j(s)}{1 + \sum_{k=1}^{n} \left[F_k \prod_{j=1}^{k} T_j(s) \right]} \tag{15.19}$$

is the transfer function of the FLF network without the output summer, i.e., with $R_{\text{o}i} = \infty$ for all i.

By Equation 15.19, the transmission zeros of $H_n(s)$ are set by the zeros of $T_j(s)$, i.e., by the feedforward path, whereas the poles of $H_n(s)$ are determined by the feedback network and involve both the poles and zeros of the biquads $T_j(s)$ and the feedback factors F_k. Typically, an FLF network is designed with second-order bandpass biquads

$$T_i(s) = A_i \frac{s/Q_i}{s^2 + s/Q_i + 1} = A_i t_i(s) \tag{15.20}$$

so that $H_n(s)$ has all transmission zeros at the origin, i.e., it is an all-pole bandpass function. Note that in Equation 15.20 the frequency parameter s is normalized to the pole frequency ω_0 ($\omega_{0i} = \omega_0$ for all i is assumed), and Q_i and A_i are the section's pole quality factor and midband gain, respectively. Designing an FLF network with arbitrary zeros requires second-order sections with finite transmission zeros, which leads to quite difficult design procedures. It is much simpler to use Equation 15.15 with Equation 15.18 to yield

$$H(s) = \frac{V_{\text{out}}}{V_{\text{in}}} = \frac{N_{2n}(s)}{D_{2n}(s)} = \alpha \frac{K_0 + \sum_{k=1}^{n} \left[K_k \prod_{j=1}^{k} T_j(s) \right]}{1 + \sum_{k=1}^{n} \left[F_k \prod_{j=1}^{k} T_j(s) \right]} \tag{15.21}$$

which realizes the transfer function of the complete circuit in Figure 15.3 with an arbitrary numerator polynomial $N_{2n}(s)$, even for second-order bandpass functions $T_i(s)$ as in Equation 15.20. It is a ratio of two polynomials whose roots can be set by an appropriate choice of the functions $T_i(s)$, the parameters K_i for the transmission zeros, and the feedback factors F_i for the poles.

We illustrate the design procedure by considering a specific case. For $n = 3$, Equation 15.21 becomes

$$H(s) = \alpha \frac{K_0 + K_1 T_1 + K_2 T_1 T_2 + K_3 T_1 T_2 T_3}{1 + F_1 T_1 + F_2 T_1 T_2 + F_3 T_1 T_2 T_3} \tag{15.22}$$

Next we transform the bandpass functions $T_i(s)$ in Equation 15.20 into lowpass functions by the lowpass-to-bandpass transformation (see Chapter 3)

$$p = Q \frac{s^2 + 1}{s} \tag{15.23}$$

where

$Q = \omega_0 / B$ is the "quality factor" of the high-order bandpass with bandcenter ω_0 and bandwidth B
p is the normalized lowpass frequency

This step transforms the bandpass functions (Equation 15.20) with all identical pole frequencies ω_0 into the first-order lowpass functions

$$T_{i\text{LP}}(p) = \frac{A_i Q / Q_i}{p + Q/Q_i} = \frac{q_i}{p + q_i} \tag{15.24}$$

where

$q_i = Q/Q_i$
A_i is the dc gain of the lowpass section

Applying Equation 15.23 to the prescribed function $H(s)$ of order $2n$ in Equation 15.22 converts it into a prototype lowpass function $H_{\text{LP}}(p)$ of order n. Substituting Equation 15.24 into the numerator and denominator expressions of that function, of order $n = 3$ in our case, shows that the zeros and poles, respectively, are determined by

$$N_3(p) = \alpha K_0 \prod_{j=1}^{3} (p + q_j) + \sum_{j=1}^{2} \left[k_j \prod_{i=j+1}^{3} (p + q_i) \right] + k_3 \tag{15.25a}$$

and

$$D_3(p) = \prod_{j=1}^{3} (p + q_j) + \sum_{k=1}^{2} \left[f_k \prod_{i=k+1}^{3} (p + q_i) \right] + f_3 \tag{15.25b}$$

where we introduced the abbreviations

$$f_i = F_i \prod_{j=1}^{i} A_j q_j \quad \text{and} \quad k_i = \alpha K_i \prod_{j=1}^{i} A_j q_j \tag{15.26}$$

To realize the prescribed third-order function

$$H_{\mathrm{LP}}(p) = \frac{V_o}{V_i} = \frac{a_3 p^3 + a_2 p^2 + a_1 p + a_0}{p^3 + b_2 p^2 + b_1 p + b_0} \tag{15.27}$$

we compare coefficients between Equations 15.25 and 15.27. For the denominator terms we obtain

$$\begin{aligned} b_2 &= q_1 + q_2 + q_3 + f_1 \\ b_1 &= q_1 q_2 + q_1 q_3 + q_2 q_3 + f_1(q_2 + q_3) + f_2 \\ b_0 &= q_1 q_2 q_3 + f_1 q_2 q_3 + f_2 q_3 + f_3 \end{aligned}$$

These are three equations in six unknowns, f_i and q_i, $i = 1, \ldots, 3$, which can be written more conveniently in matrix form:

$$\begin{pmatrix} 1 & 0 & 0 \\ q_2 + q_3 & 1 & 0 \\ q_2 q_3 & q_3 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} b_2 - (q_1 + q_2 + q_3) \\ b_1 - (q_1 q_2 + q_1 q_3 + q_2 q_3) \\ b_0 - q_1 q_2 q_3 \end{pmatrix} \tag{15.28}$$

The transmission zeros are found via an identical process: the unknown parameters k_i are computed from an equation of the form (Equation 15.28) with f_i replaced by $k_i/(\alpha K_0)$ and b_i replaced by a_i/a_3, $i = 1, \ldots, 3$. Also, $K_0 = a_3/\alpha$.

The unknown parameters f_i can be solved from the matrix expression (Equation 15.28). It is a set of linear equations whose coefficients are functions of the prescribed coefficients b_i and of the numbers q_i which for given Q are determined by the quality factors Q_i of the second-order sections $T_i(s)$. Thus, the Q_i are free parameters that may be selected to satisfy any criteria that may lead to a better-working circuit. The free design parameters may be chosen, for example, to reduce a circuit's sensitivity to element variations. This leads to a multiparameter (i.e., the n Q_i-values) optimization problem whose solution requires the availability of the appropriate computer algorithms. If such software is not available, specific values of Q_i can be chosen. The design becomes particularly simple if all the Q_i-factors are equal, a choice that has the additional practical advantage of resulting in all identical second-order building blocks, $T_i(s) = T(s)$. For this reason, this approach has been referred to as the "Primary Resonator Block" (PRB) technique. The passband sensitivity performance of PRB circuits is almost as good as that of fully optimized FLF structures. The relevant equations are derived in the following:

With $q_i = q$ for all i we find from Equation 15.28

$$\begin{pmatrix} 1 & 0 & 0 \\ 2q & 1 & 0 \\ q^2 & q & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} b_2 - 3q \\ b_1 - 3q^2 \\ b_0 - q^3 \end{pmatrix} \tag{15.29}$$

It shows that

$$\begin{aligned} F_1 A_1 q &= f_1 = b_2 - 3q \\ F_2 A_1 A_2 q^2 &= f_2 = b_1 - 3q^2 - 2q f_1 \\ F_3 A_1 A_2 A_3 q^3 &= f_3 = b_0 - q^3 - q^2 f_1 - q f_2 \end{aligned} \tag{15.30}$$

The system (Equation 15.30) represents three equations for the four unknowns, q, f_i, $i = 1, \ldots, 3$, (in general, one obtains n equations for $n+1$ unknowns) so that one parameter, q, can still be used for optimization purposes. This single degree of freedom is often eliminated by choosing

$$q = \frac{b_{n-1}}{nb_n} \tag{15.31a}$$

i.e., $q = b_2/3$ in our example, which means $f_1 = 0$. The remaining feedback factors can then be computed recursively from Equation 15.30. The systematic nature of the equations makes it apparent how to proceed for $n > 3$. As a matter of fact, it is not difficult to show that, with $f_1 = 0$, in general

$$f_2 = b_{n-2} - \frac{n(n-1)}{2!} q^2 b_n \tag{15.31b}$$

$$f_i = b_{n-i} - \frac{q^i}{(n-i)!}\left[\frac{n!}{i!} b_n + \sum_{j=1}^{i-1} \frac{f_j}{q^j} \frac{(n-j)!}{(i-j)!}\right] \quad i = 3, \ldots, n \tag{15.31c}$$

Note that b_n usually equals unity. As mentioned earlier, equations of identical form, with f_i replaced by $k_i/(\alpha K_0)$ and b_i by a_i/a_n with $K_0 = a_n/\alpha$, are used to determine the summing coefficients K_i of the output summer, which establishes the transmission zeros. Thus, given a geometrically symmetrical bandpass function with center frequency ω_0, quality factor Q, and bandwidth B, Equation 15.31 can be used to calculate the parameter q and all feedback and summing coefficients required for a PRB design. All second-order bandpass sections, Equation 15.20, are tuned to the same pole-frequency, $\omega = \omega_0$, and have the same pole quality factor, $Q_p = Q/q$, where $Q = \omega_0/B$.

As discussed, the design procedure computes only the *products* f_i and k_i; the actual values of F_i, K_i, and A_i are not uniquely determined. As a matter of fact, the gain constants A_i are *free* parameters that are selected to maximize the circuit's dynamic range in much the same way as for cascade designs.* With a few simple approximations, it can be shown [5, Chapter 6] that the appropriate choice of gain constants in the FLF circuit is

$$A_i \approx \sqrt{1 + (Q_i/Q)^2} = \sqrt{1 + q_i^{-2}} \quad i = 1, \ldots, n \tag{15.32a}$$

The same equation holds for the PRB case where $Q_i = Q_p$ for all i so that

$$A_i = A \approx \sqrt{1 + (Q_p/Q)^2} = \sqrt{1 + q^{-2}} \tag{15.32b}$$

The following example demonstrates the complete FLF (PRB) design process.

Let us illustrate the multiple-loop feedback procedure by realizing again the bandpass function (Equation 15.10), but now as an FLF (PRB) circuit. The previous data specify that the bandpass function should be converted into a prototype lowpass function with bandcenter $\omega_0/(2\pi) = \sqrt{12 \cdot 36}$ kHz and bandwidth $B/(2\pi) = (36 - 12)$ kHz by the transformation Equation 15.23,

$$p = Q\frac{s^2 + 1}{s} = \frac{\omega_0}{B}\frac{s^2 + 1}{s} = \frac{\sqrt{12.36}}{36 - 12}\frac{s^2 + 1}{s} = 0.866\frac{s^2 + 1}{s} \tag{15.33}$$

* For practical FLF (PRB) designs, this step of scaling the signal levels is very important because an inadvertently poor choice of gain constants can result in very large internal signals and, consequently, very poor dynamic range.

where s is normalized by ω_0 as before. Substituting Equation 15.33 into Equation 15.10 results in

$$H_{\mathrm{LP}}(s) = \frac{0.491}{p^3 + 0.984p^2 + 1.236p + 0.491} \tag{15.34}$$

which corresponds to Equation 15.27 with $a_i = 0$, $i = 1, 2, 3$. To realize this function, we need three first-order lowpass sections of the form (Equation 15.24), i.e.,

$$T_{\mathrm{LP}}(p) = A\frac{q}{p+q} \tag{15.35}$$

where $A = \sqrt{1 + 1/q^2}$ from Equation 15.32b. We proceed with Equation 15.31a to choose $q = 0.984/3 = 0.328$, so that $A = 3.209$. With these numbers, we can apply Equation 15.33 to Equation 15.35 to obtain the second-order bandpass that must be realized:

$$T_{\mathrm{BP}}(p) = 3.209\frac{0.328}{0.866\frac{s^2+1}{s} + 0.328} = \frac{1.2153s}{s^2 + 0.379s + 1} \tag{15.36}$$

The feedback resistors we obtain by Equation 15.30:

$$\begin{aligned} 1.052F_1 &= f_1 = 0 \rightarrow F_1 = 0 \\ 1.108F_2 &= f_2 = b_1 - 3q^2 = 0.913 \rightarrow F_2 = 0.777 \\ 1.166F_3 &= f_3 = b_0 - q^3 - qf_2 = 0.371 \rightarrow F_3 = 0.318 \end{aligned} \tag{15.37}$$

Choosing, e.g., $R_{F0} = 10\ \mathrm{k}\Omega$ results in $R_{Fi} = R_{F0}/F_i$, that is, $R_{F1} = \infty$, $R_{F2} = 12.9\ \mathrm{k}\Omega$, and $R_{F3} = 31.5\ \mathrm{k}\Omega$. Also, $R_{F0}/R_{\mathrm{in}} = \alpha = 0.491$ $(Aq)^{-3} = 0.421 \rightarrow R_{\mathrm{in}} = 23.7\ \mathrm{k}\Omega$.

There remains the choice of second-order bandpass sections. Notice from Equation 15.19 that the sections must have positive, i.e., noninverting, gain to keep the feedback loops stable. We choose GIC (general impedance converter) sections [5, Chapter 4], [9, Chapter 4.5], one of which is shown explicitly in Figure 15.4. They realize

$$T(s) = \frac{V_{\mathrm{o}}}{V_i} = \frac{\left(1 + \frac{G_3}{G_2}\right)\frac{G_1}{C}s}{s^2 + \frac{G_1}{C}s + \left(\frac{G}{C}\right)^2\frac{G_3}{G_2}} = \frac{K\frac{\omega_0}{Q}s}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \tag{15.38}$$

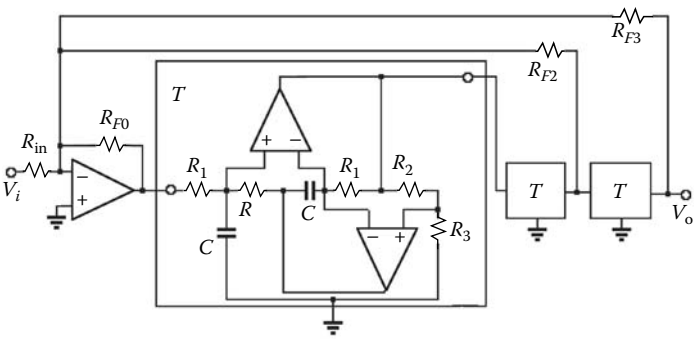


FIGURE 15.4 PRB topology with three identical GIC bandpass sections. The circuit for one section is shown explicitly, as is the input summer with the feedback resistors.

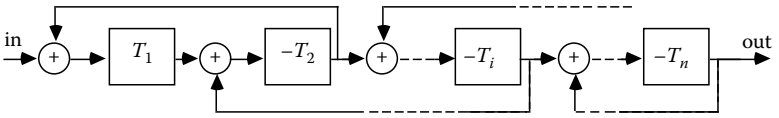


FIGURE 15.5 Leapfrog topology.

Comparing Equation 15.38 with Equation 15.36, choosing $R_3 = 5$ kΩ, $C = 2$ nF, and remembering that s is normalized with respect to $\omega_0 = 130{,}590$ s^{-1} results in the rounded values

$$R = 5.5 \text{ k}\Omega \quad R_1 = 10 \text{ k}\Omega \quad R_2 = 10 \text{ k}\Omega$$

15.3.2 Leapfrog Topology

The LF configuration is pictured in Figure 15.5. Each of the boxes labeled T_i realizes a second-order transfer function. The feedback loops always comprise two sections; thus, inverting and noninverting sections must alternate to keep the loop gains negative and the loops stable. If the circuit is derived from a resistively terminated lossless ladder filter as is normally the case, T_1 and T_n are lossy and all the internal sections are lossless. A lossless block implies a function T_i with infinite Q, which may not be stable by itself, but the overall feedback connection guarantees stability.

An LF circuit can be derived from the configuration in Figure 15.5 by direct analysis with, e.g., bandpass transfer functions as in Equation 15.20 assumed for the blocks T_i. Comparing the resulting equation with that of a prescribed filter yields expressions that permit determining all circuit parameters in a similar way as for FLF filters [1]. Because the topology is identical to that derived from a signal-flow graph representation of an *LC* ladder filter, we do not consider the details of the LF approach here, but instead proceed directly to the ladder simulation techniques.

15.4 Ladder Simulations

Although transfer functions of *LC* ladders are more restrictive than those realizable by cascade circuits (see footnote in Section 15.2), lossless ladder filters designed for maximum power transfer have received considerable attention in the active filter literature because of their significant advantage of having the lowest possible sensitivities to component tolerances in the passband. Indeed, the majority of high-order active filters with demanding specifications are being designed as simulated *LC* ladders. Many active circuit structures have been developed, which simulate the performance of passive *LC* ladders and inherit their good sensitivity performance. The ladder simulations can be classified into two main groups: *operational simulation* and *element substitution*. Starting from an existing *LC* prototype ladder, the operational simulation models the internal operation of the ladder by simulating the circuit equations, i.e., Kirchhoff's voltage and current laws and the *I–V* relationships of the ladder arms. Fundamentally, this procedure simulates the *signal-flow graph* (SFG) of the ladder where all voltages and all currents are considered signals, which are integrated on the inductors and capacitors, respectively. The SFG method is developed in Section 15.4.1. The element substitution procedure replaces all inductors or inductive branches by active networks whose input impedance is inductive over the frequency range of interest. A practical approach to this method is presented in Section 15.4.3. For more detailed discussions of these important and practical procedures, we refer the reader to a modern text on active filters, such as Refs. [5,6,9] and, for approaches using operational transconductance amplifiers, to Section 16.3.

15.4.1 Signal-Flow Graph Methods

To derive the SFG method, consider the ladder structure in Figure 15.6, whose branches may contain arbitrary combinations of capacitors and inductors. In general, resistors are permitted to allow for lossy components. Labeling the combination of elements in the series arms as admittances Y_i, $i = 2, 4$, and those in the shunt branches as impedances Z_j, $j = 1, 3, 5$, we can readily analyze the ladder by writing Kirchhoff's laws and the I–V relationships for the ladder arms as follows:

$$\begin{aligned} I_i &= G_i(V_i - V_1) \quad V_1 = Z_1 I_1 = Z_1(I_i - I_2) \\ I_2 &= Y_2 V_2 = Y_2(V_1 - V_3) \quad V_3 = Z_3 I_3 = Z_3(I_2 - I_4) \\ I_4 &= Y_4 V_4 = Y_4(V_3 - V_5) \quad V_5 = V_o = Z_5 I_5 = Z_5(I_4 - I_6) \quad I_6 = G_o V_o \end{aligned} \tag{15.39}$$

In the active simulation of this circuit, all currents and voltages are to be represented as *voltage* signals. To reflect this in the expressions, we use a resistive scaling factor R as shown in one of these equations as an example:

$$V_3 = \frac{Z_3}{R} I_3 R = \frac{Z_3}{R}(I_2 R - I_4 R) \tag{15.40}$$

and introduce the notation

$$I_k R = i_k \quad V_k = v_k \quad G_i R = g_i \quad Z_k/R = z_k \quad Y_k R = y_k \tag{15.41}$$

The lower-case symbols are used to represent the *scaled* quantities; notice that z_k and y_k are dimensionless voltage transfer functions, also called *transmittances*, and that both i_k and v_k are voltages. We shall retain the symbol i_k to remind ourselves of the origin of that signal as a current in the original ladder. Equation 15.39 then takes on the form

$$\begin{aligned} i_i &= g_i[v_i + (-v_1)] \quad -v_1 = -z_1\ i_1 = -z_1\ [i_i + (-i_2)] \\ -i_2 &= y_2(-v_2) = y_2[(-v_1) + v_3] \quad v_3 = -z_3(-i_3) = -z_3[(-i_2) + i_4] \\ i_4 &= y_4\ v_4 = y_4[v_3 + (-v_5)] \quad -v_5 = -v_o = -z_5 i_5 = -z_5[i_4 + (-i_6)] \quad -i_6 = g_o(-v_o) \end{aligned} \tag{15.42}$$

where we have made all the transmittances z_i inverting and assigned signs to the signals in a consistent fashion, such that only *signal addition* is required in the circuit to be derived from these equations. We made this choice because addition can be performed at op-amp summing nodes with no additional circuitry (see below), whereas subtraction would require either differential amplifiers or inverters. The price to be paid for this convenience is that in some cases the overall transfer function suffers a sign inversion (a 180° phase shift), which is of no consequence in most cases. The signal-flow block diagram

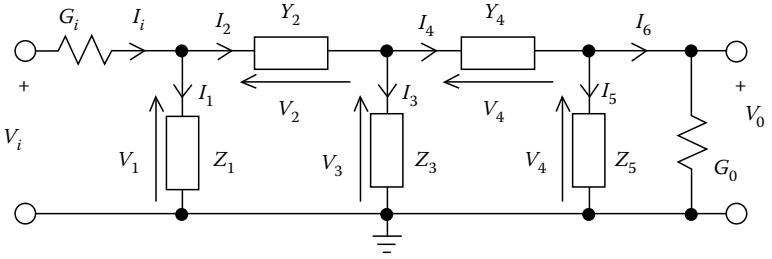


FIGURE 15.6 Ladder network.

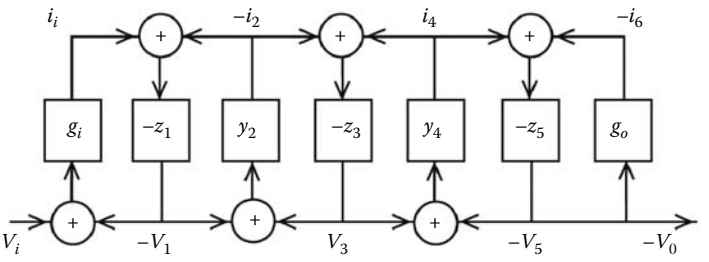


FIGURE 15.7 Signal-flow graph block diagram realizing Equation 15.42.

implementing Equation 15.42 is shown in Figure 15.7. As is customary, all voltage signals are drawn at the bottom of the diagram, and those derived from currents at the top. We observe that the circuit consists of a number of interconnected loops of two transmittances each and that all loop-gains are negative as required for stability. Notice that redrawing this figure in the form of Figure 15.5 results in an identical configuration, i.e., as mentioned earlier, the leapfrog method is derived from a ladder simulation technique.

To determine how the transmittances are to be implemented, we need to know which elements are in the ladder arms. Consider first the simple case of an all-pole lowpass ladder where $Z_i = 1/(sC_i)$ and $Y_j = 1/(sL_j)$, i.e., $z_i = 1/(sC_iR)$ and $y_i = 1/(sL_i/R)$ (see Figure 15.11). Evidently then, for this case all transmittances are integrators. Suitable circuits are shown in Figure 15.8 where for each integrator we have used two inputs in anticipation of the final realization, which has to sum two signals as indicated in Figure 15.7. The circuits realize

$$V_o = \pm \frac{G_1 V_1 + G_2 V_2}{sC + G_3} \tag{15.43}$$

where the plus sign is valid for the series transmittances $y_i(s)$ in Figure 15.8b and the minus sign for the shunt transmittances $z_i(s)$* in Figure 15.8a. G_3 is zero if the integration is lossless as required in an internal branch of the ladder; in the two end branches, G_3 is used to implement the source and load resistors of the ladder.

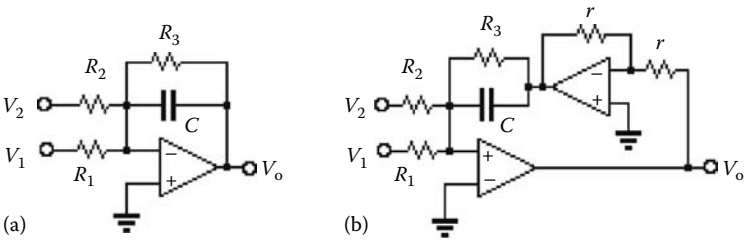


FIGURE 15.8 (a) Inverting lossy Miller integrator; (b) noninverting lossy phase-lead integrator.

* These two circuits are good candidates for building two-integrator loops as required in Figure 15.7 because the op-amp in the Miller integrator causes a phase lag, whereas the op-amps in the noninverting integrator cause a phase lead of the same magnitude. In the loop, these two phase shifts just cancel and cause no errors in circuit performance. Notice that the Åckerberg–Mossberg biquad in Figure 15.2 is also constructed as a loop of these two integrators.

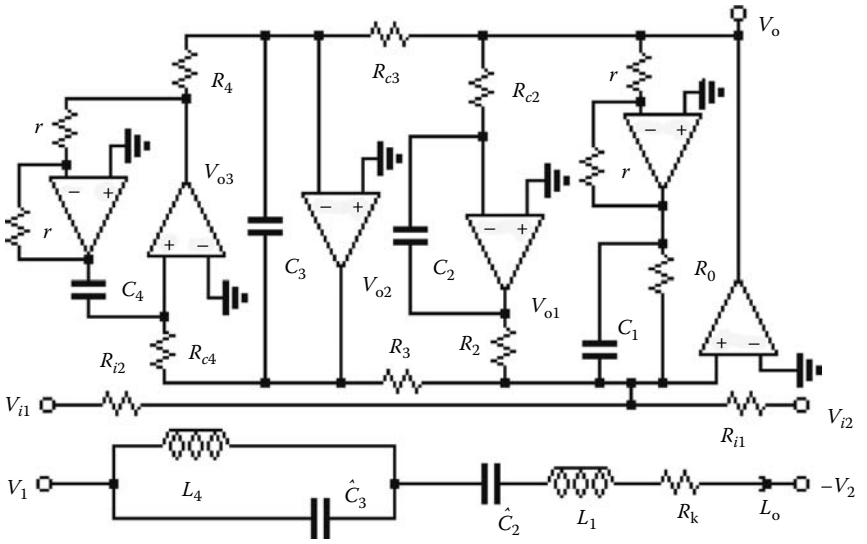


FIGURE 15.9 Active realization of a series ladder branch (plus sign in Equation 15.46).

The next question to be answered is what circuitry will realize more general ladder arms which may contain both series and parallel *LC* elements, and resistors to handle the source or load terminations (or losses if required). Such general series and shunt branches are shown at the bottom of Figures 15.9 and 15.10, respectively, where we have labeled the capacitors in the *passive* network as "$\hat{C}$" to be able to distinguish them from the capacitors in the active circuit (labeled *C* without circumflex). We have chosen the signs of the relevant voltages and currents in Figures 15.9 and 15.10 appropriately to obtain

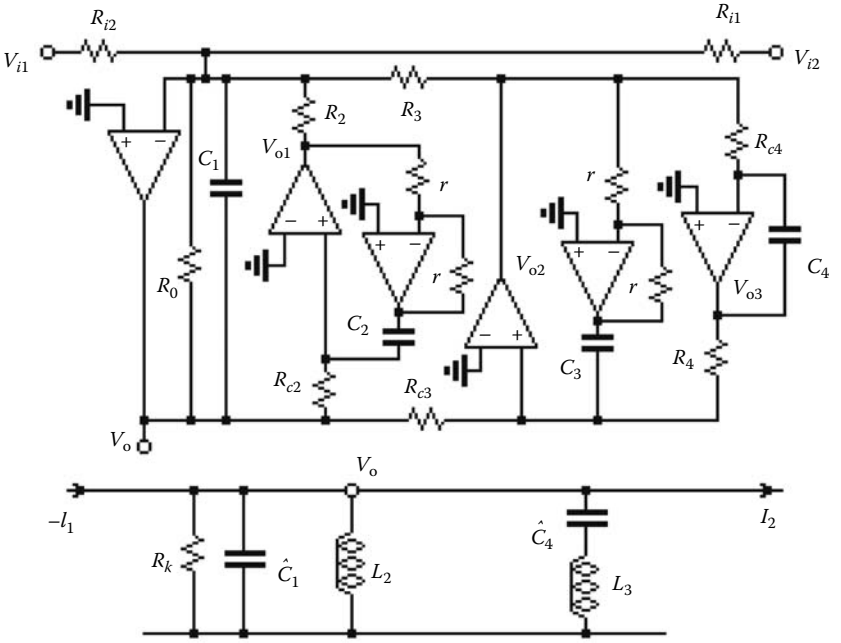


FIGURE 15.10 Active realization of a shunt ladder branch minus sign in Equation 15.46.

noninverting transmittances $y(s)$ for the series arms and *inverting transmittances* $z(s)$ for the shunt arms as requested in Figure 15.7. Recall that this choice permits the flowgraph to be realized with only *summing* functions. For easy reference, the active *RC* circuits realizing these branches are shown directly above the passive *LC* arms [5, Chapter 6], [9, Chapter 15].

The passive series branch in Figure 15.9 realizes the current I_o

$$\begin{aligned} I_o R_\mathrm{p} &= +Y(s)R_\mathrm{p}(aV_1 + bV_2) \\ &= +\frac{1}{R_k/R_\mathrm{p} + sL_1/R_\mathrm{p} + \frac{1}{s\hat{C}_2 R_\mathrm{p}} + \frac{1}{s\hat{C}_3 R_\mathrm{p} + \frac{1}{sL_4/R_\mathrm{p}}}}(aV_1 + bV_2) \end{aligned} \tag{15.44a}$$

which was converted into a voltage through multiplication with a scaling resistor R_p (p stands for *passive*; R_p is the resistor used to scale the *passive* circuit). Also, we have multiplied the input signals by two constants, a and b, in anticipation of future scaling possibilities in the *active* circuit. Using, as before, *lower-case* symbols for the *normalized* variables, we obtain for the series branch

$$i_o = +y(s)(av_1 + bv_2) = +\frac{1}{r_k + sl_1 + \frac{1}{sc_2} + \frac{1}{sc_3 + \frac{1}{sl_4}}}(av_1 + bv_2) \tag{15.44b}$$

In an analogous fashion we find for the voltage V_o in the passive shunt branch in Figure 15.10, after impedance-level scaling with R_p and signal-level scaling with a and b, the expression

$$\begin{aligned} V_\mathrm{o} &= -\frac{Z(s)}{R_\mathrm{p}}(aI_1 R_\mathrm{p} + bI_2 R_\mathrm{p}) \\ &= -\frac{1}{G_k R_\mathrm{p} + s\hat{C}_1 R_\mathrm{p} + \frac{1}{sL_2/R_\mathrm{p}} + \frac{1}{sL_3/R_\mathrm{p} + \frac{1}{sC_4 R_\mathrm{p}}}}(aI_1 R_\mathrm{p} + bI_2 R_\mathrm{p}) \end{aligned} \tag{15.45a}$$

which with lower-case notation gives

$$v_o = -z(s)(ai_1 + bi_2) = -\frac{1}{g_k + sc_1 + \frac{1}{sl_2} + \frac{1}{sl_3 + \frac{1}{sc_4}}}(ai_1 + bi_2) \tag{15.45b}$$

Turning now to the active *RC* branches in Figures 15.9 and 15.10, elementary analysis of the two circuits, assuming ideal op-amps,* results in

$$V_\mathrm{o} = \pm\frac{R_\mathrm{a}G_{i1}V_{i1} + R_\mathrm{a}G_{i2}V_{i2}}{R_\mathrm{a}G_0 + sC_1R_\mathrm{a} + \frac{R_\mathrm{a}G_2}{sC_2R_{c2}} + \frac{R_\mathrm{a}G_3}{sC_3R_{c3} + \frac{G_4R_{c3}}{sC_4R_{c4}}}} \tag{15.46}$$

where we used a normalizing resistor R_a (a stands for *active*; R_a is the resistor used to scale the *active* circuit). In Equation 15.46 the plus sign is valid for the series arm, Figure 15.9, and the minus sign for the shunt arm, Figure 15.10.

---

* Using a more realistic op-amp model, $A(s) \approx \omega_t/s$, one can show [5, Chapter 6] that Equation 16.46 to a first-order approximation is multiplied by $(1 + j\omega/\omega_t) \approx \exp(j\omega/\omega_t)$ for the series arm (plus sign in Equation 16.46) and by $(1 - j\omega/\omega_t) \approx \exp(-j\omega/\omega_t)$ for the shunt arm [minus sign in Equation 16.46]. Thus, in the loop gains determined by the product of an inverting and a noninverting branch, op-amp effects cancel to a first order, justifying the assumption of ideal op-amps. See footnote in Section 15.4.1.

Notice that Equation 15.46 is of the same form as Equations 15.44 and 15.45 so that we can compare the expressions term-by-term to obtain the component values required for the active circuit (s) to realize the prescribed passive ladder arms. Thus, we find for the *series* branch, Figure 15.9, by comparing coefficients between Equations 15.46 and 15.44a, and assuming all equal capacitors C in the active circuit:

$$\begin{gathered} R_{i1} = \frac{R_\mathrm{a}}{a} \quad R_{i2} = \frac{R_\mathrm{a}}{b} \quad R_0 = \frac{R_\mathrm{a}R_\mathrm{p}}{R_k} \quad C = \frac{L_1}{R_\mathrm{a}R_\mathrm{p}} \\ R_{c2}R_2 = \frac{\hat{C}_2}{C}R_\mathrm{a}R_\mathrm{p} \quad R_{c3}R_3 = \frac{\hat{C}_3}{C}R_\mathrm{a}R_\mathrm{p} \quad R_{c4}R_4 = \frac{L_4\hat{C}_3}{C^2} \end{gathered} \tag{15.47a}$$

In an identical fashion we obtain from Equations 15.46 and 15.45a for the components of the *shunt* arm

$$\begin{gathered} R_{i1} = \frac{R_\mathrm{a}}{a} \quad R_{i2} = \frac{R_\mathrm{a}}{b} \quad R_0 = R_k\frac{R_\mathrm{a}}{R_\mathrm{p}} \quad C = \hat{C}_1\frac{R_\mathrm{p}}{R_\mathrm{a}} \\ R_{c2}R_2 = \frac{L_2}{C}\frac{R_\mathrm{a}}{R_\mathrm{p}} \quad R_{c3}R_3 = \frac{L_3}{C}\frac{R_\mathrm{a}}{R_\mathrm{p}} \quad R_{c4}R_4 = \frac{L_3\hat{C}_4}{C^2} \end{gathered} \tag{15.47b}$$

The scaling resistors R_a and R_p are chosen to obtain convenient element values. Note that each of the last three equations for both circuits determines only the *product* of two resistors. This leaves three degrees of freedom that are normally chosen to maximize dynamic range by equalizing the maximum signal levels at all op-amp outputs. We provide some discussion of these matters in Section 15.4.2.

We have displayed the active and passive branches in Figures 15.9 and 15.10 together to illustrate the one-to-one correspondence of the circuits. For example, if we wish to design an all-pole lowpass filter, such as the one in Figure 15.11, where the internal series arms consist of a single inductor and each internal shunt arm of a single capacitor, the corresponding active realizations reduce to those of Figure 15.8b and a, respectively, with $R_3 = \infty$. For the end branches, R_s in series with L_1 and R_l in parallel with $\hat{C}_4$, we obtain the circuits in Figure 15.8b and a, respectively, with R_3 *finite* to account for the source and load resistors. The remaining branches in the active circuits are absent. Assembling the resulting circuits as prescribed in Figure 15.7 leads to the active SFG filter in Figure 15.12, where we have for convenience chosen all identical capacitors, C, and have multiplied the input by an arbitrary constant K because the active circuit may realize a gain scaling factor. The component values are computed from a set of equations similar to Equation 15.47. To show in some detail how they are arrived at, we derive the equations for each integrator in Figure 15.12 and compare them with the corresponding arm in the passive ladder. Recalling that signals with lower-case symbols in the active circuit are voltages, we obtain

$$\frac{G_1R_\mathrm{a}V_\mathrm{in} + G_2R_\mathrm{a}(-v_2)}{sCR_\mathrm{a} + G_3R_\mathrm{a}} \rightarrow \frac{KV_\mathrm{in} - V_2}{sL_1/R_\mathrm{p} + R_\mathrm{s}/R_\mathrm{p}} \tag{15.48a}$$

$$\frac{G_4R_\mathrm{a}i_1 + G_5R_\mathrm{a}(-i_3)}{sCR_\mathrm{a}} \rightarrow \frac{I_1 - I_3}{s\hat{C}_2R_\mathrm{p}} \tag{15.48b}$$

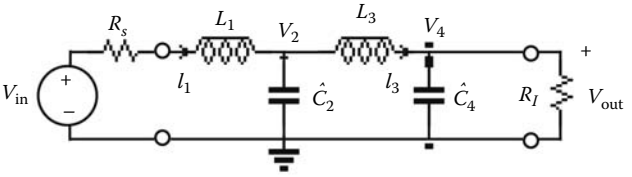


FIGURE 15.11 Fourth-order all-pole lowpass ladder.

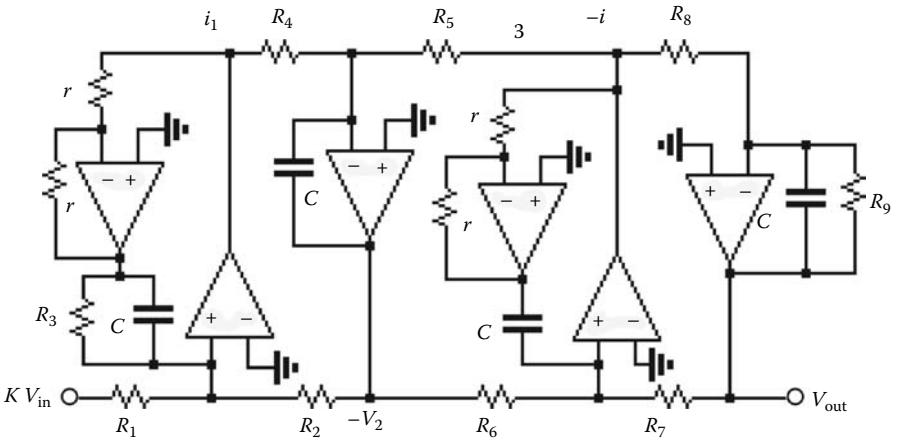


FIGURE 15.12 Active realization of the ladder in Figure 15.11.

$$\frac{G_6R_a(-v_2) + G_7R_aV_{out}}{sCR_a} \rightarrow \frac{-V_2 + V_4}{sL_3/R_p} \tag{15.48c}$$

$$\frac{G_8R_a(-i_3)}{sCR_a + G_9R_a} \rightarrow \frac{-i_3}{s\hat{C}_4R_p + R_p/R_l} \tag{15.48d}$$

where we used scaling resistors for the active (R_a) and the passive (R_p) circuits as before. Choosing a convenient value for C in the active circuit and equating the time constants and the dc gains in Equation 15.48a results in the following expressions:

$$R_3 = \frac{L_1}{C}\frac{1}{R_s} \quad R_1 = R_3\frac{R_s}{KR_p} \quad R_2 = R_3\frac{R_s}{R_p} \tag{15.49a}$$

Similarly,

$$R_4 = R_5 = \frac{\hat{C}_2}{C} \quad R_6 = R_7 = \frac{L_3}{C}\frac{1}{R_p} \quad R_8 = \frac{\hat{C}_4}{C} \quad R_9 = \frac{\hat{C}_4}{C}\ R_l = \frac{R_l}{R_p}R_8 \tag{15.49b}$$

15.4.2 Maximization of the Dynamic Range

The remaining task in a signal-flow graph simulation of an *LC* ladder is that of voltage-level scaling for dynamic range maximization. As in cascade design, we need to achieve that all op-amps in $0 \leq \omega \leq \infty$ see the same maximum signal level so that no op-amp becomes overdriven sooner than any other one. It may be accomplished by noting that a scale factor can be inserted into any signal line in a signal-flow graph *as long as the loop gains are not changed*. Such signal-level scaling will not affect the transfer function except for an overall gain factor. The procedure can be illustrated in the flow diagram in Figure 15.7. If we simplify the circuit by combining the self-loops at input and output and employ the scale factors α, β, γ, δ and their inverses to the loops in Figure 15.7, we obtain the modified flow diagram in Figure 15.13. Simple analysis shows that the transfer function has not changed except for a multiplication by the factor $\alpha\beta\gamma\delta$, which is canceled by the multiplier $(\alpha\beta\gamma\delta)^{-1}$ at the input. To understand how the scale factors are

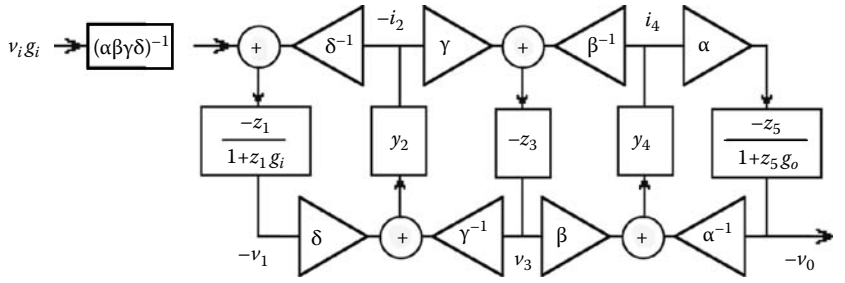


FIGURE 15.13 Using scale factor for signal-level equalization.

computed, assume that in Figure 15.7, i.e., before scaling, the maximum of $|i_4(j\omega)|$ is α times as large as the maximum of $|v_o(j\omega)|$, where α may be less than or larger than unity. Since

$$-v_o = \frac{-z_5}{1 + z_5 g_o} i_4$$

the maxima can be equalized, i.e., the level of v_o can be increased by α, if we apply a gain scale factor α as indicated in Figure 15.13. To keep the loop gain unchanged, a second scale factor α^{-1} is inserted as shown. Continuing, if in Figure 15.7 max $|v_3(j\omega)|$ is β times as large as max $|i_4(j\omega)|$, we raise the level of $|i_4|$ by a gain factor β and correct the loop gain by a second factor β^{-1} as shown in Figure 15.13. In a similar fashion we obtain the maxima of the remaining voltages and apply the appropriate scale factors γ, δ, γ^{-1}, and δ^{-1} in Figure 15.13.

It is easy to determine the relevant maxima needed for calculating the gain factors. Recall that the node voltages v_i, v_1, i_2, v_3, i_4, and $v_5 = v_o$ in the signal-flow graph of Figure 15.13 correspond directly to the actual currents and voltages in the original ladder, Figure 15.6, and that their maxima in $0 \le \omega \le \infty$ can be evaluated readily with any network analysis program. For the circuit in Figure 15.13, the scale factors that ensure that the currents (normalized by R_p) in all series ladder arms and the voltages in all shunt ladder arms have the same maxima are then obtained as

$$\alpha = \frac{\max|i_4|}{\max|v_o|} \quad \beta = \frac{\max|v_3|}{\max|i_4|} \quad \gamma = \frac{\max|i_2|}{\max|v_3|} \quad \delta = \frac{\max|v_1|}{\max|i_2|} \tag{15.50}$$

The procedure thus far takes care of ladders with only one element in each branch, such as all-pole lowpass filters (Figure 15.11). In the more general case, we must also equalize the maxima of the magnitudes of V_{o1}, V_{o2}, and V_{o3} at the *internal* op-amp outputs in Figures 15.9 and 15.10. This is achieved easily if we remember that the "external" voltage maxima of $|V_{i1}|$, $|V_{i2}|$, and $|V_o|$ are already equalized by the previous steps leading to Equation 15.50, and that in the passive *series* branch V_{o1} represents the voltage on $\hat{C}_2$, V_{o2} stands for the voltage on $\hat{C}_3$, and V_{o3} corresponds to the current (times R_p) through the inductor L_4. After finding the maxima of these signals with the help of a network analysis program and computing the scale factors

$$m_1 = \frac{\max|V_{o1}|}{\max|V_o|} \quad m_2 = \frac{\max|V_{o2}|}{\max|V_o|} \quad m_3 = \frac{\max|V_{o3}|}{\max|V_o|} \tag{15.51}$$

we can equalize all internal branch voltages of the *series* arm (Figure 15.9) by modifying the design Equations 15.47a as follows [5, Chapter 6]:

$$R_{i1} = \frac{R_a}{a} \quad R_{i2} = \frac{R_a}{b} \quad R_0 = \frac{R_a R_p}{R_k} \quad C = \frac{L_1}{R_a R_p}$$
$$m_1 R_{c2} \frac{R_2}{m_1} = \frac{\hat{C}_2}{C} R_a R_p \quad m_2 R_{c3} \frac{R_3}{m_2} = \frac{\hat{C}_3}{C} R_a R_p \quad \frac{m_3}{m_2} R_{c4} \frac{R_4}{m_3/m_2} = \frac{L_4 \hat{C}_3}{C^2} \tag{15.52}$$

A possible choice for the element values given by products is

$$m_1 R_{c2} = \frac{R_2}{m_1} = \sqrt{\frac{\hat{C}_2}{C} R_a R_p} \quad m_2 R_{c3} = \frac{R_3}{m_2} = \sqrt{\frac{\hat{C}_3}{C} R_a R_p}$$
$$\frac{m_3}{m_2} R_{c4} = \frac{m_2}{m_3} R_4 = \frac{\sqrt{L_4 \hat{C}_3}}{C} \tag{15.53a}$$

In the passive *shunt* branch, V_{o1} and V_{o2} represent the currents (times R_p) through the inductors L_2 and L_3, respectively, and V_{o3} stands for the voltage across $\hat{C}_4$. In an identical fashion we obtain then with Equation 15.50 from Equation 15.47b for the components of the *shunt* arm

$$R_{i1} = \frac{R_a}{a} \quad R_{i2} = \frac{R_a}{b} \quad R_0 = R_k \frac{R_a}{R_p} \quad C = \hat{C}_1 \frac{R_p}{R_a}$$
$$m_1 R_{c2} = \frac{R_2}{m_1} = \sqrt{\frac{L_2}{C} \frac{R_a}{R_p}} \quad m_2 R_{c3} = \frac{R_3}{m_2} = \sqrt{\frac{L_3}{C} \frac{R_a}{R_p}}$$
$$\frac{m_3}{m_2} R_{c4} = \frac{m_2}{m_3} R_4 = \sqrt{\frac{L_3 \hat{C}_4}{C}} \tag{15.53b}$$

Next we present an example [2], [5, Chapter 6] in which the reader may follow the different steps discussed.

To simulate the fourth-order elliptic lowpass ladder filter in Figure 15.14a by the signal-flow graph technique, we first reduce the loop count by a source transformation. The resulting circuit, after

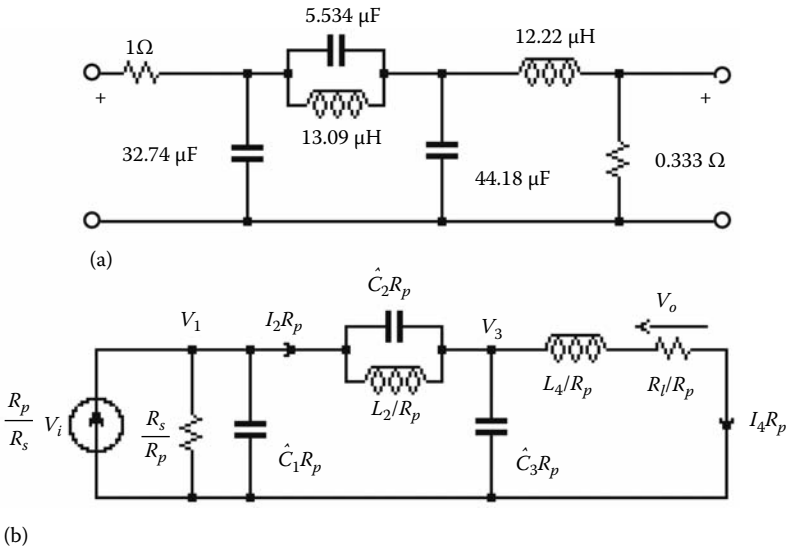


FIGURE 15.14 (a) Fourth-order elliptic *LC* lowpass ladder; (b) the circuit after source transformation and impedance scaling.

TABLE 15.1 Voltage and Current Maxima for Figure 15.14a

Voltage or Current	Maximum of Voltage or Current
Voltage across $\hat{C}_1$	0.699 V
Current through L_2	1.550 A
Voltage across $\hat{C}_2$	1.262 V
Current through $L_1 \| \hat{C}_2$	1.125 A
Voltage across $\hat{C}_2$	0.690 V
Current through L_4	0.866 A

impedance scaling, is shown in Figure 15.14b. To use signal-level scaling for optimizing the dynamic range, the relevant maxima of the *LC* ladder currents and voltages are needed. Table 15.1 lists the relevant data obtained with the help of network analysis software. From these numbers we find

$$\alpha = \frac{\max|v_3|}{\max|i_4 R_p|} = \frac{0.69}{0.866} = 0.797 \quad \beta = \frac{\max|i_2 R_p|}{\max|v_3|} = \frac{1.125}{0.690} = 1.630$$
$$\gamma = \frac{\max|v_1|}{\max|i_2 R_p|} = \frac{0.699}{1.125} = 0.621 \tag{15.54}$$

For ease of reference, the signal-flow graph with scaling factors is shown in Figure 15.15. Let $R_p = 1\ \Omega$. Note that the input signal voltage $i_{in} = R_p V_i / R_s$ has been multiplied by a factor K to permit realizing an arbitrary signal gain. If the desired gain is unity, and since the *dc* gain in the passive *LC* ladder equals

$$\frac{V_o}{V_i} = \frac{R_l}{R_s + R_l} = \frac{0.333}{1.333} = 0.25$$

we need to choose $K = 1/0.25 = 4$. Thus, the input signal i_i is multiplied by

$$\frac{K}{\alpha\beta\gamma} = \frac{4}{0.797 \cdot 1.630 \cdot 0.621} = 4.958 \tag{15.55}$$

The transmittances are defined as

$$z_1 = \frac{1}{s\hat{C}_1 R_p + G_s R_p} \quad y_2 = \frac{1}{\frac{1}{s\hat{C}_2 R_p + R_p/(sL_p)}}$$
$$z_3 = \frac{1}{s\hat{C}_3} \quad y_4 = \frac{1}{sL_4/R_p + R_l/R_p} \tag{15.56}$$

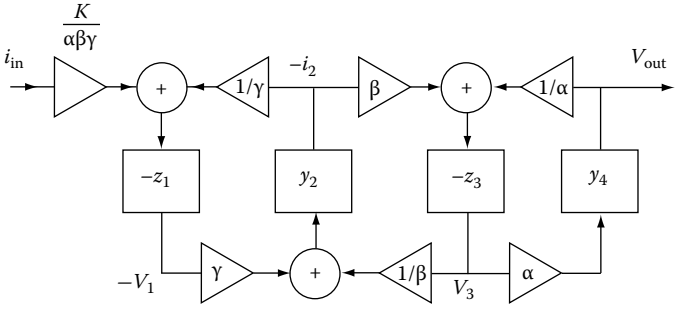


FIGURE 15.15 Signal-flow graph for the filter in Figure 15.14.

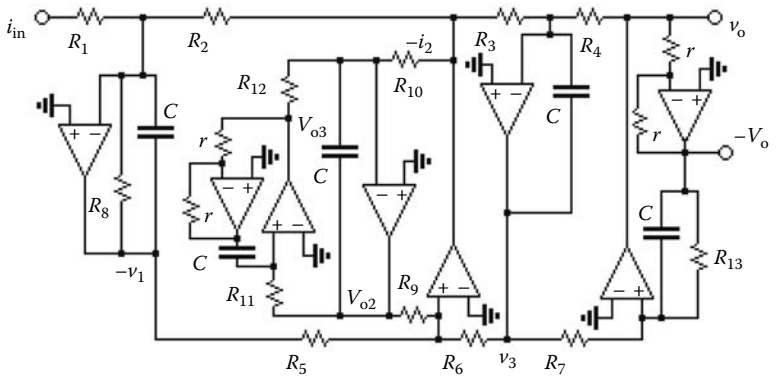


FIGURE 15.16 Active realization of the circuit in Figure 15.14.

With the numbers in Equations 15.54 and 15.55 we obtain from the signal-flow graph

$$\begin{aligned} -v_1 &= -z_1\left(4.958 i_{\text{in}} - \frac{1}{0.621} i_2\right) \quad -i_2 = y_2\left(-0.621 v_1 + \frac{1}{1.630} v_3\right) \\ v_3 &= -z_3\left(-1.630 i_2 + \frac{1}{0.797} v_{\text{out}}\right) \quad v_{\text{out}} = y_4 0.792 v_3 \end{aligned} \tag{15.57}$$

The reader is encouraged to verify from Equations 15.56 and 15.57 and Figures 15.9 and 15.10 that the active SFG realization of the ladder is as shown in Figure 15.16. For instance, the circuitry between the nodes v_1, v_3, and i_2 implements the parallel LC series branch to realize the finite transmission zero, obtained from Figure 15.9 by setting the components G_0, C_1, G_2, G_{c2}, and C_2 to zero. The element values in the active circuit are now determined readily by comparing the circuit equations for the active circuit,

$$\begin{aligned} -v_1 &= -\frac{G_1 R_{a1} i_{\text{in}} + G_2 R_{a1}(-i_2)}{sCR_{a1} + G_8 R_{a1}} \quad -i_2 = \frac{G_5 R_{a2}(-v_1) + G_6 R_{a2} v_3}{\frac{G_9 R_{a2}}{sCR_{10} + \frac{G_{12} R_{10}}{sCR_{11}}}} \\ v_3 &= -\frac{G_3 R_{a3}(-i_2) + G_4 R_{a3} v_o}{sCR_{a3}} \quad v_o = \frac{G_7 R_{a4} v_3}{sCR_{a4} + G_{13} R_{a4}} \end{aligned} \tag{15.58}$$

to Equation 15.57 and using Equation 15.56.* Comparing the coefficients results in

$$\begin{aligned} G_1 R_{a1} &= 4.958 \quad G_2 R_{a1} = \frac{1}{0.621} \quad G_5 R_{a2} = 0.621 \quad G_6 R_{a2} = \frac{1}{1.630} \\ G_3 R_{a3} &= 1.630 \quad G_4 R_{a3} = \frac{1}{0.797} \quad G_7 R_{a4} = 0.797 \end{aligned}$$

Further,

$$\hat{C}_1 R_p = CR_{a1} \quad \hat{C}_2 R_p = C\frac{R_9 R_{10}}{R_{a2}} \quad \hat{C}_3 R_p = CR_{a3} \quad \frac{L_4}{R_p} = CR_{a4}$$

Thus, with $R_p = 1\ \Omega$ and choosing $C = 5$ nF,

* Observe that for greater flexibility we permitted a different scaling resistor R_{ai}, $i = 1, \ldots, 4$, in each branch.

$$R_{a1} = \frac{\hat{C}_1}{C} R_p = 6.55\ \text{k}\Omega \quad R_{a3} = \frac{\hat{C}_3}{C} R_p = 8.84\ \text{k}\Omega \quad R_{a4} = \frac{L_4}{CR_p} = 2.44\ \text{k}\Omega$$

R_{a2} is undetermined; choosing $R_{a2} = 5\ \text{k}\Omega$ leads to the feed-in resistors for each branch

$$R_1 = \frac{R_{a1}}{4.958} = 0.800\ \text{k}\Omega \quad R_2 = 0.621 R_{a1} = 4.07\ \text{k}\Omega \quad R_3 = \frac{R_{a3}}{1.630} = 5.42\ \text{k}\Omega$$
$$R_4 = 0.797\ R_{a3} = 7.05\ \text{k}\Omega \quad R_5 = \frac{R_{a2}}{0.621} = 8.05\ \text{k}\Omega$$
$$R_6 = 1.630\ R_{a3} = 8.15\ \text{k}\Omega \quad R_7 = \frac{R_{a4}}{0.797} = 3.06\ \text{k}\Omega$$

The remaining components are determined from the equations

$$R_8 = R_s \frac{R_{a1}}{R_p} = 6.55\ \text{k}\Omega \quad R_{13} = \frac{R_{a4} R_p}{R_l} = 7.33\ \text{k}\Omega$$

and

$$R_9 R_{10} = R_p R_{a2} \frac{\hat{C}_2}{C} = (2.35\ \text{k}\Omega)^2 \quad R_{11} R_{12} = \frac{L_2}{C} \frac{R_9 R_{10}}{R_{a2} R_p} = (1.70\ \text{k}\Omega)^2$$

Since only the products of these resistors are given, we select their values uniquely for dynamic range maximization. According to Equation 15.51 we compute from Table 15.1

$$m_2 = \frac{\max|v_{c2}|}{\max|i_2|} = \frac{1.262}{1.125} = 1.12 \quad \text{and} \quad m_3 = \frac{\max|i_{L2}|}{\max|i_2|} = \frac{1.550}{1.125} = 1.38$$

to yield from Equation 15.53a:

$$R_9 = m_2\ 2.35\ \text{k}\Omega = 2.64\ \text{k}\Omega \quad R_{10} = \frac{2.35\ \text{k}\Omega}{m_2} = 2.10\ \text{k}\Omega$$
$$R_{11} = \frac{m_2}{m_3} 1.70\ \text{k}\Omega = 1.38\ \text{k}\Omega \quad R_{12} = \frac{m_3}{m_2} 1.70\ \text{k}\Omega = 2.09\ \text{k}\Omega$$

15.4.3 Element Substitution

LC ladders are known to yield low-sensitivity filters with excellent performance, but high-quality inductors cannot be implemented in microelectronic form. An appealing solution to the filter design problem is, therefore, to retain the ladder structure and to simulate the behavior of the inductors by circuits consisting of resistors, capacitors, and op-amps. A proven technique uses *impedance converters*, electronic circuits whose input impedance is proportional to frequency when loaded by the appropriate element at the output. The best-known impedance converter is the gyrator, a two-port circuit whose input impedance is inversely proportional to the load impedance, i.e.,

$$Z_{in}(s) = \frac{r^2}{Z_L(s)} \tag{15.59}$$

The parameter r is called the *gyration resistance*. Clearly, when the load is a capacitor, $Z_L(s) = 1/(sC)$, $Z_{in}(s) = sr^2C$ is the impedance of an inductor of value $L = r^2C$. Gyrators are very easy to realize with

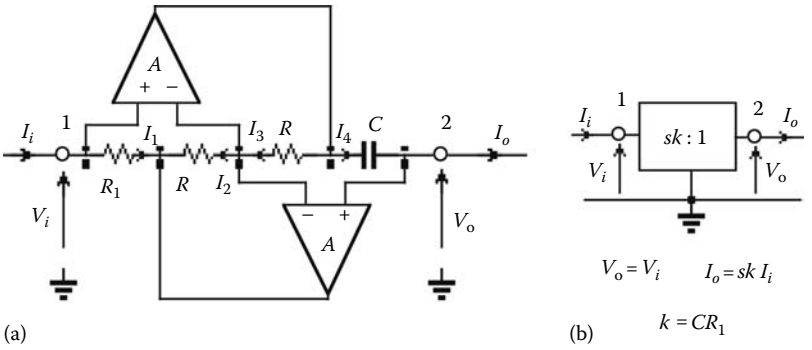


FIGURE 15.17 General impedance converter: (a) circuit; (b) symbolic representation.

transconductors (voltage-to-current converters) and are widely used in transconductance-*C* filters; see Section 16.3. However, no high-quality gyrators with good performance beyond the audio range have been designed to date with op-amps. If op-amps are to be used, a different kind of impedance converter is employed, one that converts a load resistor R_L into an inductive impedance, such that

$$Z_{in}(s) = (sk)R_L \tag{15.60}$$

A good circuit that performs this function is Antoniou's general impedance converter (GIC) shown in Figure 15.17a. The circuit, with elements slightly rearranged, was encountered in Figure 15.4, where we used the GIC to realize a second-order bandpass function. The circuit is readily analyzed if we recall that the voltage measured between the op-amp input terminals and the currents flowing into the op-amp input terminals are zero. Thus, we obtain from Figure 15.17a the set of equations

$$V_o = V_i \quad \frac{I_4}{sC} = I_3 R \quad I_2 R = I_1 R_1 \quad I_2 = I_3 \quad I_i = I_1 \quad I_4 = I_o \tag{15.61}$$

These equations indicate that the terminal behavior of the general impedance converter is described by

$$V_o = V_i \quad I_o = sCR_1 I_i = skI_i \tag{15.62}$$

that is

$$\frac{V_i}{I_i} = Z_{in}(s) = sk\frac{V_o}{I_o} = sk\, Z_L(s) \tag{15.63}$$

Notice that the input impedance is inductive as prescribed by Equation 15.60 if the load is a resistor.* Figure 15.17b also shows the circuit symbol we will use for the GIC in the following to keep the circuit diagrams simple. This impedance converter and its function of converting a resistive load into an inductive input impedance is the basis for Gorski-Popiel's embedding technique [5, Chapter 6], [9, Chapter 14.4], which permits replacing the inductors in an *LC* filter by resistors.

* To optimize the performance of the GIC, i.e., to make it optimally independent of the finite gain-bandwidth product of the op-amps, the GIC elements should be chosen as follows: For an arbitrary load $Z_L(s)$ one chooses $\omega_c C = 1/{}^*Z_L(j\omega_c)^*$. ω_c is some critical frequency, normally chosen at the upper passband corner. If the load is resistive, $Z_L = R_L$, select $C = 1/(\omega_c R_L)$ [6].

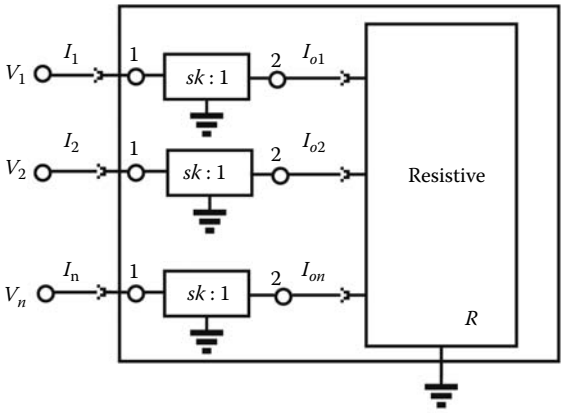


FIGURE 15.18 Simulation of an inductance network.

15.4.3.1 Inductor Replacement: Gorski-Popiel's Method

To understand the behavior of the circuit in Figure 15.18, recall from Equation 15.62 that the voltages at terminals 1 and 2 of the GICs are the same. Then, by superposition the typical input current I_{oi} into the resistive network $\boldsymbol{R}$ can be written as

$$I_{oi} = \frac{1}{R_{i1}} V_1 + \frac{1}{R_{i2}} V_2 + \cdots + \frac{1}{R_{in}} V_n = \sum_{j=1}^{n} \frac{1}{R_{ij}} V_j \quad i = 1, \ldots, n \tag{15.64}$$

where the parameters R_{ij} are given by the resistors and the configuration of $\boldsymbol{R}$. Using the current relationship $I_{oi} = skI_i$ of the impedance converters in Equation 15.64 results in

$$I_i = \sum_{j=1}^{n} \frac{1}{skR_{ij}} V_j \quad i = 1, \ldots, n \tag{15.65}$$

which makes the combined network consisting of the n GICs and the network $\boldsymbol{R}$ look purely inductive, i.e., each resistor R_r in the network $\boldsymbol{R}$ appears replaced by an inductor of value

$$L_r = skR_r = sCR_1R_r \tag{15.66}$$

Examples of this process are contained in Figure 15.19. Figure 15.19a and b illustrate the conventional realizations of a grounded and a floating inductor requiring one and two converters, respectively. This is

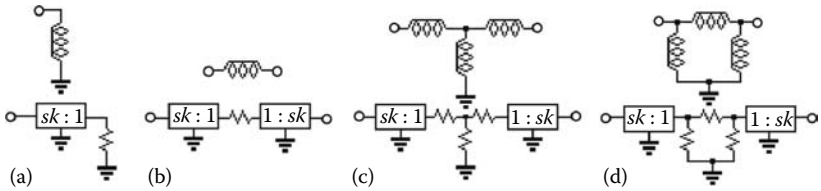


FIGURE 15.19 Elementary inductance networks and their GIC-*R* equivalents: (a) grounded inductor (b) floating inductor (c) inductive T (d) inductive Π.

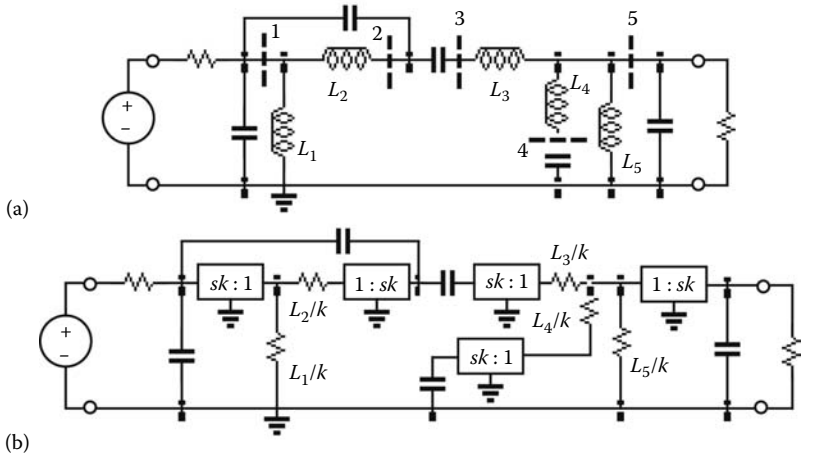


FIGURE 15.20 *LC* ladder realization with GICs: (a) a choice of cuts; (b) the cuts replaced by GICs.

completely analogous to the use of gyrators, where a grounded inductor requires one gyrator, but a floating inductor is realized with two gyrators connected back-to-back. Figure 15.19c and d show how the process is extended to complete inductive subnetworks: the GICs are used to isolate the subnetworks from the remainder of the circuit; there is no need to convert each inductor separately. The method is further illustrated with the filter in Figure 15.20. The *LC* ladder is designed to specifications by use of the appropriate design tables [4,7] or filter design software [8]. In the resulting ladder, the inductive subnetworks are separated as shown in Figure 15.20a by the five dashed cuts. The cuts are repaired by inserting GICs with conversion factor k and the correct orientation (output, terminal 2 in Figure 15.17, toward the inductors). Note that all GICs must have the same conversion factor k. Finally, the inductors L_i are replaced by resistors of value L_i/k as shown in Figure 15.20b. A numerical example will illustrate the design steps:

Assume we wish to implement a sixth-order bandpass filter with the following specifications:
Maximally flat passband with $\leq$3 dB attenuation in 900 Hz $\leq f \leq$ 1200 Hz
Transmission zero at $f_z = 1582.5$ Hz; source and load resistors $R = 3$ kΩ

Using the appropriate software or tables, the *LC* filter in Figure 15.21a is found with element values in kiloohm, megahertz, and nanofarad. Figure 15.21b shows the circuit redrawn to help identify the inductive subnetwork and the locations of the cuts. Note that only three impedance converters are used rather than six (two each for the two floating inductors and one each for the grounded ones), if a direct conversion of the individual inductors had been attempted.

For the final active realization, we assumed a conversion parameter $k = CR_1 = 30$ μs. Finally, to design the GICs we compute the impedances seen into the nodes where the GICs see the (now) resistive subnetworks. Using analysis software to compute $|V_i/I_i|$ at these nodes, we find

$$\text{for GIC}_a,\ |Z_a| \approx 18\ \text{k}\Omega,\ \text{and for GIC}_b \text{ and GIC}_c, |Z_b| \approx |Z_c| \approx 4.7\ \text{k}\Omega$$

so that with $\omega_c = 2\pi \cdot 1.2$ kHz and $k = 30$ μs the design elements are

$$C_a = \frac{1}{\omega_c |z_a|} \approx 7.4\ \text{nF} \quad R_{1a} = \frac{k}{C_a} \approx 4\ \text{k}\Omega \quad C_b = C_c \approx 28\ \text{nF} \quad R_b = R_c \approx 1\ \text{k}\Omega$$

To complete the design, we choose the resistors R in Figure 15.17a as $R = 1$ kΩ.

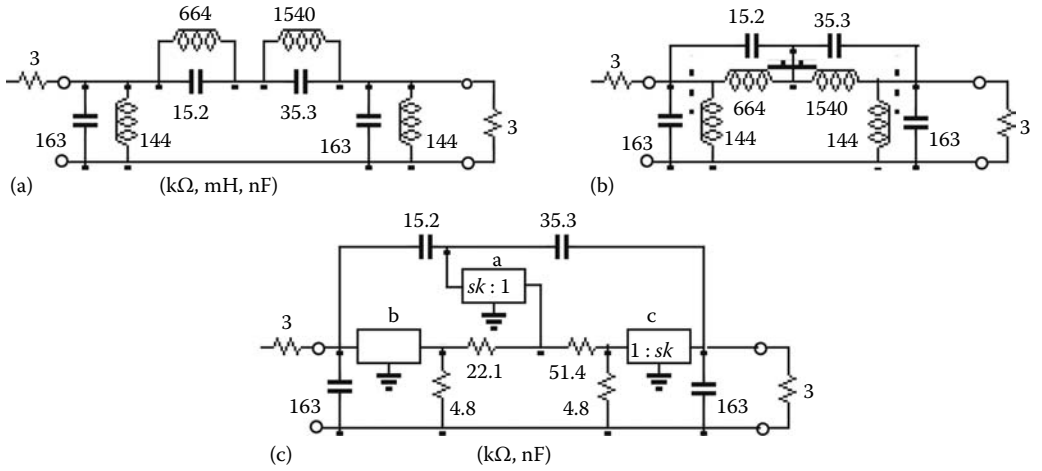


FIGURE 15.21 Sixth-order *LC* bandpass filter and active realization using impedance converters. (a) The LC filter with element values in kiloohm, megahertz, and nanoforad. (b) The circuit redrawn to help identify the inductive subnetwork and the locations of the cuts. (c) The final active realization using a conversion parameter $k = CR_i = 30\mu s$.

15.5 Summary

In this chapter, we discussed the more practical techniques for the design of active filters of order higher than two: *cascade* design, *multiple-loop feedback* approaches, and methods that *simulate the behavior of LC ladder* filters. We pointed out that direct realization methods are impractical because they result in high sensitivities to component values. In many applications, a cascade design leads to satisfactory results. The practical advantages of cascade circuits are modularity, ease of design, flexibility, very simple tuning procedures, and economical use of op-amps with as few as one op-amp per pole pair. Also, we pointed out again that an arbitrary transfer function can be realized with the cascade design method, i.e., no restrictions are placed on the permitted locations of poles and zeros.

For challenging filter requirements, the cascade topologies may still be too sensitive to parameter changes. In those cases the designer can use the FLF configuration or, for best performance, a ladder simulation, provided that a passive prototype ladder exists. FLF circuits retain the advantage of modularity, but if optimal performance is desired, computer-aided optimization routines must generally be used to adjust the available free design parameters. However, excellent performance with very simple design procedures, no optimization, and high modularity can be obtained by use of the *primary-resonator-block* (PRB) method, where all biquad building blocks are identical.

From the point of view of minimum passband sensitivity to component tolerances, the best active filters are obtained by simulating *LC ladders*. If the prescribed transfer characteristic can at all be realized as a passive *LC* ladder, the designer can make use of the wealth of available information about the design of such circuits, and simple procedures are available for ''translating'' the passive *LC* circuit into its active counterpart. We may either take the passive circuit and replace the inductors by active networks, or we imitate the mathematical behavior of the whole *LC* circuit by realizing the integrating action of inductors and capacitors via active *RC* integrators. Both approaches result in active circuits of high quality; a disadvantage is that more op-amps may be required than in cascade or FLF methods. This drawback is offset, however, by the fact that the sensitivities to component tolerances are very nearly as low as those of the originating ladder.

An alternative method for the active realization of a passive *LC* ladder is obtained by scaling each impedance of the passive circuit by $1/(ks)$ (Bruton's transformation) [5, Chapter 6], [9, Chapter 14.5].

This impedance transformation converts resistors into capacitors, inductors into resistors, and capacitors into "*frequency-dependent negative resistors* (FDNRs)," which can readily be realized with Antoniou's GIC circuit. The procedure is especially useful for passive prototype circuits that have only *grounded* capacitors because capacitors are converted to FDNRs and floating FDNRs are very difficult to implement in practice. The method results in biasing difficulties for active elements, and because the *entire* ladder must be transformed, the active circuit no longer contains source and load resistors. If these two components are prescribed and must be maintained as in the original passive circuit, additional buffers are required. The method was not discussed in this text because it shows no advantages over the Gorski-Popiel procedure.

An important practical aspect is the *limited dynamic range* of active filters, restricted by noise and by the finite linear signal swing of op-amps. Dynamic range maximization should be addressed whenever possible. Apart from designing low-noise circuits, the procedures always proceed to equalize the op-amp output voltages by exploiting available free gain constants or impedance scaling factors. It is a disadvantage of the element substitution method that no general dynamic range scaling method appears to be available.

The methods discussed in this chapter dealt with the design of filters in *discrete* form, that is, separate passive components and operational or transconductance amplifiers are assembled on, e.g., a printed circuit board to make up the desired filter. Chapter 16 addresses modifications in the design approach that lead to filters realizable in fully integrated form.

References

1. Laker, K. R. and M. S. Ghausi, A comparison of active multiple-loop feedback techniques for realizing high-order bandpass filters, *IEEE Trans. Circuits Syst.*, CAS-21, 774–783, November 1974.
2. Martin, K. and A. S. Sedra, Design of signal-flow graph (SFG) active filters, *IEEE Trans. Circuits Syst.*, CAS-25, 185–195, September 1978.
3. Moschytz, G. S., *Linear Integrated Networks—Design*, New York: Van Nostrand Reinhold, 1975.
4. Saal, R., *Handbook of Filter Design*, Berlin: AEG-Telefunken, 1979.
5. Schaumann, R., M. S. Ghausi, and K. R. Laker, *Design of Analog Filters: Passive, Active RC and Switched Capacitor*, Englewood Cliffs, NJ: Prentice-Hall, 1990.
6. Sedra, A. S. and P. O. Brackett, *Filter Theory and Design: Active and Passive*, Portland, OR: Matrix, 1978.
7. Zverev, A. I., *Handbook of Filter Synthesis*, New York: Wiley, 1967.
8. Silveira, B. M., C. Ouslis, and A. S. Sedra, Passive ladder synthesis in filtorX, *IEEE Int. Symp. Circuits Syst.*, 1431–1434, 1993.
9. Schaumann, R. and M. Van Valkenburg, *Design of Analog Filters*, New York: Oxford University Press, 2001.

16

Continuous-Time Integrated Filters

16.1 Introduction **16**-1
16.2 MOSFET-*C* Filters **16**-3
Basic Building Blocks • Cascade Realizations • Ladder Simulations
16.3 g_m-*C* Filters **16**-14
Basic Building Blocks • Cascade Realizations • Ladder Simulations
16.4 Tuning **16**-31
16.5 Discussion and Conclusions **16**-33
References **16**-34

Rolf Schaumann
Portland State University

16.1 Introduction

All modern signal-processing systems include various types of electrical filters that the designer has to realize in an appropriate technology. The literature contains many well-defined filter design techniques [1–3], and computer programs are available, which help the designer find the appropriate transfer function that describes the required filter characteristics mathematically. The reader may also refer to Section I in this book and the other chapters of this section (Section II).

Once the filter's transfer function is obtained, implementation methods must be found that are compatible with the technology selected for the design of the total system. In some situations, considerations of power consumption, frequency range, signal level, or production numbers may dictate *discrete* (*passive* or *active*) filter realizations. Often, however, as much as possible of the total system must be *fully integrated* in microelectronic form, so that the filters can be implemented in the same technology.

Often, digital (Section III) or sampled-data (Chapter 18) implementations are suitable for realizing the filter requirements. However, in modern communications applications, the required frequency range is so high that digital or sampled-data circuitry is inappropriate or too expensive so that *continuous-time* (*c-t*) filters are necessary. In addition, filters in many signal-processing situations must interface with the ''real world,'' where the input and output signals take on continuous values as functions of the continuous variable time, i.e., they are *c-t* signals. In these situations *c-t* antialiasing and reconstruction filters are often required. Because the performance of the total filter system is of relevance and not just the performance of the intrinsic filter, the designer may have to consider if it might not be preferable to implement the entire system in the *c-t* domain rather than as a digital or sampled-data system. At least at low frequencies the latter methods have the advantages of very high accuracy, better signal-to-noise ratio, and little or no parameter drifts, but they entail a number of problems connected with

analog-to-digital (A/D) and digital-to-analog (D/A) conversion (see Chapter 10 of *Analog and VLSI Circuits*), sample-and-hold, switching, antialiasing, and reconstruction circuitry.

Traditionally, *c-t* filters were implemented as discrete designs. Well-understood procedures exist for deriving passive *LC* filters (Section I) from a given transfer function with prescribed complex natural frequencies, e.g., [1, Chapter 2], [2], [3, Chapter 13]. To date no practical methods exist for building high-quality, i.e., low-loss, inductors on an integrated circuit (IC) chip.* The required complex natural frequencies must, therefore, be realized by using *gain*, i.e., as we saw earlier in this section, by embedding an operational amplifier (op-amp; see Chapter 16 of *Fundamentals of Circuits and Filters*) in an *RC* feedback network [1,3]. Since op-amps, resistors, and capacitors can be implemented on an integrated circuit, it appears that with active *RC* networks the problem of monolithic filter design is solved in principle: all active devices and any necessary capacitors and resistors can be integrated together on one silicon chip. Although this conclusion is correct, the designer needs to consider four other factors that are important in integrated *c-t* filter design and perhaps are not immediately obvious.

The first item concerns the most important design task for achieving commercially practical designs: integrated filters must be *electronically tunable*, preferably by an *automatic* tuning scheme. Because of its importance, we shall devote a separate section, Section 16.4, to this topic. The second item deals with the economics of practical implementations of active filters: in *discrete* designs, the cost of components and stocking them usually necessitate designing the filter with a minimum number of active devices. One, two, or possibly three op-amps per pole pair are used and the smallest number of different (if possible, all identical) capacitors. In *integrated* realizations, capacitors are determined by processing mask dimensions and the number of different capacitor values is unimportant, as long as the *element spread* is not excessive. Further, active devices frequently occupy less chip area than passive elements so that it is often preferable to use active elements instead of passive ones.† Also, the designer should remember that in IC technology it is not easy to generate accurate absolute component values, but that *ratios* of like components, such as capacitor ratios, can be realized very precisely. The third observation pertains to the fact that filters usually have to share an integrated circuit with other, possibly switched or digital, systems so that the ac ground lines (power supply and ground wires) are likely to contain switching transients and generally are noisy. Measuring the analog signals relative to ac ground, therefore, may result in designs with poor signal-to-noise ratio and low power-supply rejection. The situation is remedied in practice by building continuous-time filters in fully differential, balanced form, where the signals are referred to each other as $V = V^+ - V^-$ as shown in Figure 16.4b through d. An additional advantage of this arrangement is that the signal range is doubled (for an added 6 dB of signal-to-noise ratio) and that the even-order harmonics in the nonlinear operation of the active devices cancel. All filters in this chapter are understood, therefore, to be designed in fully differential form. Finally, we point out that communication circuitry is often required to operate at hundreds of megahertz or higher, where op-amp–based active *RC* filters will not function because of the op-amps' limited bandwidth.

Today, *c-t* filters integrated in bipolar, CMOS, or BiCMOS technology are no longer academic curiosities but a commercial reality (see Ref. [5] for some recent advances in the field). In the following we discuss the main methods that have proven to be reliable. First, we present *MOSFET-C filters*, whose design methods resemble most closely the standard active *RC* procedures discussed in Chapters 11–14; they can, therefore, be most readily understood by the reader without requiring further background. Next, we introduce the *transconductance-C* (also referred to as g_m-*C*) technique, which is currently the predominant method for

* Spiral inductors of a few nanohenry or microhenry in size, however, can be used at gigahertz frequencies. Typically, they are extremely lossy, with quality factors of the order of only 10. Since such low values of Q are unacceptable for the design of high-quality selective filters, the inductor losses must be reduced. This is accomplished by placing a simulated negative resistor (Figure 16.17c) in series or parallel with the inductor. Similarly, since the spiral inductor itself cannot be tuned, variable current-shunting circuitry is employed to change the effective inductor value; see Ref. [4].

† However, keeping the number of active devices small remains important because active devices consume power and generate noise.

c-t integrated filters. Designs based on transconductors lead to filters for the higher operating frequencies that are important for modern communication systems.

16.2 MOSFET-*C* Filters

As mentioned in the introduction, the MOSFET-*C* method follows standard op-amp–based active filter techniques [6], which, as we saw in previous chapters, rely heavily on the availability of integrators and summers. The only difference is that the method replaces the resistors used in the conventional active *RC* integrating and summing circuitry by MOSFET devices (Figure 16.1) based in the triode (ohmic) region ($V_C - V_T > V_D$, see Section 1.2 of *Analog and VLSI Circuits*). Defining the source (V_S), drain (V_D), gate (V_C), and substrate (V_B) voltages as shown in Figure 16.1, the resulting nonlinear drain current I becomes

$$\begin{aligned} I &= \frac{W}{L}\mu C_{ox}\left[(V_C - V_T)(V_D - V_S) + a_2\left(V_D^2 - V_S^2\right) + a_3\left(V_D^3 - V_S^3\right) + \cdots\right] \\ &= \left[\frac{W}{L}\mu C_{ox}(V_C - V_T)\right](V_D - V_S) + b_2\left(V_D^2 - V_S^2\right) + b_3\left(V_D^3 - V_S^3\right) + \cdots \end{aligned} \tag{16.1}$$

where

W and L are the channel width and length, respectively
μ is the effective mobility
C_{ox} is the gate capacitance per unit area
V_T is the threshold voltage of the device

The term in brackets is a tunable conductor,

$$G(V_C) = \frac{W}{L}\mu C_{ox}(V_C - V_t) \tag{16.2}$$

where the gate voltage V_C is used for control or *tuning*. Thus, the drain current of a MOSFET in the triode region is proportional to the drain-to-source voltage, but contains nonlinear second- and higher order terms. Third- and higher order terms can be shown to be small and will be neglected in the following. Now consider placing two of these MOS conductances in parallel as shown in Figure 16.2a. Note that the two devices are driven in balanced form by $+V_1$ and $-V_1$ at one pair of terminals and that the other terminals are at the same voltage V. Applying these conditions to Equation 16.1 results in

$$I_1 = G(V_C)(+V_1 - V) + b_2\left[(+V_1)^2 - V^2\right] \tag{16.3a}$$

$$I_2 = G(V_C)(-V_1 - V) + b_2\left[(-V_1)^2 - V^2\right] \tag{16.3b}$$

Consequently, apart from the neglected high-order odd terms, the difference current $I_1 - I_2$ is perfectly linear in the applied signal voltage V_1:

$$I_1 - I_2 = 2G(V_C)V_1 \tag{16.4}$$

Thus, in the MOSFET-*C* method, the even-order nonlinearities can be shown to be eliminated by carefully balanced circuit design, where all signals are measured

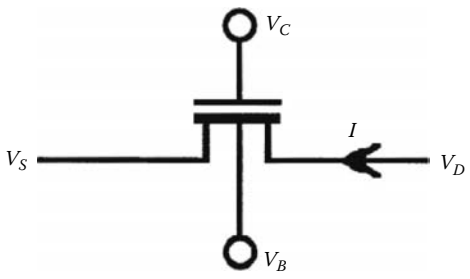


FIGURE 16.1 MOS transistor biased in the triode region.

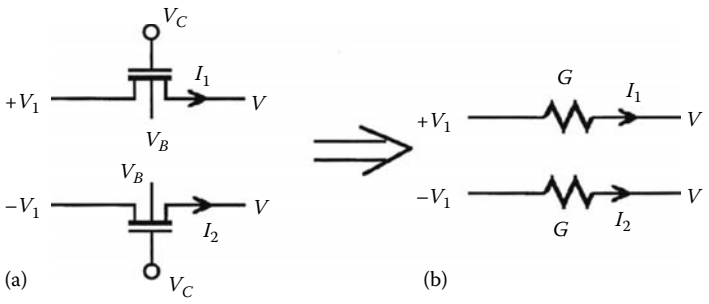


FIGURE 16.2 (a) Balanced linear MOSFET conductance; (b) equivalent resistor circuit.

strictly differentially [6]. Note that the expression for $I_1 - I_2$ in Equation 16.4 is the same as that for the linear resistive equivalent circuit in Figure 16.2b. This means that the MOSFET circuit in Figure 16.2a can be substituted for a pair of resistors in any *appropriate* active *RC* circuit; appropriate means that the resistor pair is driven by balanced signals at one end and that the voltages at the other two terminals are the same (in practice, usually virtual ground). In addition to these conditions, the substitution is valid if the MOSFETs are operated in the triode region and if $v_1(t)$ and $v(t)$ are safely within the range $|V_C - V_B^\star|$. Of course, in practice there remains some small distortion due to mismatches and the neglected odd-order terms. Odd-order nonlinearities are usually small enough to be negligible. Typically, the remaining nonlinearities, arising from odd harmonics, device mismatch, and body effects, are found to be of the order of 0.1% for 1 V signals.

In Section 16.2.1, we discuss the MOSFET-*C* integrator that is the fundamental building block used in *all* MOSFET-*C* active filter designs. The integrator will initially be used to construct first- and second-order MOSFET-*C* sections from which higher order filters can be assembled by the cascade method (Section 16.2.2). In Section 16.2.3, we show how simulated *LC* ladder filters are designed by use of integrators.

16.2.1 Basic Building Blocks

Substituting MOSFETs for resistors in an active *RC* prototype works correctly if the active *RC* circuit is of a form where all resistors come in balanced pairs, with one end of the resistor pair voltage-driven and the two other terminals seeing the same voltage (see Figure 16.2a). Active *RC* circuits do not normally satisfy these conditions, but many can readily be converted into that form if a balanced symmetrical op-amp as pictured in Figure 16.3 is available. It is important to note that this structure is not simply a differential op-amp, but that the input and output voltages are symmetrical with respect to a ground reference. Using a balanced op-amp, the conversion of many active *RC* prototypes into balanced form proceeds simply by taking the single-ended circuits and mirroring them at ground as shown below.

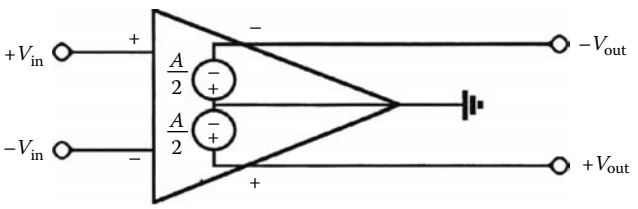


FIGURE 16.3 Balanced operational amplifier configuration realizing $V_{out} = -AV_{in}$.

16.2.1.1 Integrators

Mirroring the active *RC* integrator structure in Figure 16.4a at ground and using the balanced op-amp in Figure 16.3 leads to the balanced integrator in Figure 16.4b. Note that the two resistors are connected in the configuration prescribed in Figure 16.2 with $V = 0$ (virtual ground) so that they may be replaced by the MOSFET equivalent in Figure 16.4c. Analysis of the circuit in the time domain results in

$$-v_{\text{out}} = v^+ - \frac{1}{C}\int_{-\infty}^{t} i_2(t)\mathrm{d}t; \quad +v_{\text{out}} = v^- - \frac{1}{C}\int_{-\infty}^{t} i_1(t)\mathrm{d}t \tag{16.5a}$$

$$v_{\text{out}} - (-v_{\text{out}}) = v^- - v^+ - \frac{1}{C}\int_{-\infty}^{t} [i_1(t) - i_2(t)]\mathrm{d}t \tag{16.5b}$$

$$2v_{\text{out}} = 0 - \frac{1}{C}\int_{-\infty}^{t} 2G(V_C)v_1(t)\mathrm{d}t = -\frac{1}{C}\int_{-\infty}^{t} \frac{2v_1(t)}{R(V_C)}\mathrm{d}t \tag{16.5c}$$

or, after simplifying and using the Laplace transform, in the frequency domain,

$$v_{\text{out}} = -\frac{1}{CR(V_C)}\int_{-\infty}^{t} v_1(t)\mathrm{d}t \rightarrow V_{\text{out}}(s) = -\frac{1}{sCR(V_C)}V_1(s) \tag{16.5d}$$

We see that the MOSFET-*C* integrator realizes exactly the same transfer function as the active *RC* prototype and that the integration time constant $CR(V_C)$ is tunable by the control voltage V_C that is applied to all gates. Note that the circuit in Figure 16.4c is not a differential integrator. To build a

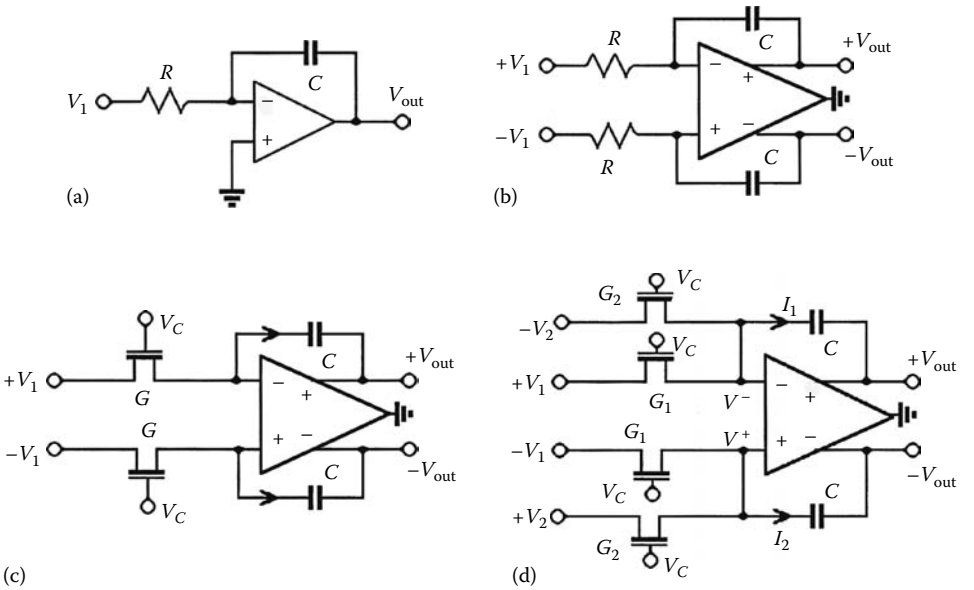


FIGURE 16.4 (a) Active *RC* integrator; (b) fully balanced equivalent; (c) MOSFET-*C* equivalent; and (d) differential MOSFET-*C* integrator.

differential integrator, one must connect a second pair of balanced resistors (MOSFETs) with inputs $-V_2$ and $+V_2$ as in Figure 16.4d to yield

$$v_{\text{out}} = -\frac{1}{C}\int_{-\infty}^{t}\left[\frac{v_1(t)}{R_1} - \frac{v_2(t)}{R_2}\right]\mathrm{d}t \rightarrow V_{\text{out}}(s) = -\frac{1}{sC}\left[\frac{V_1(s)}{R_1} - \frac{V_2(s)}{R_2}\right] \tag{16.6}$$

The same principle can also be used to build *programmable* integrators and, therefore, programmable filters: consider in Figure 16.4d the terminals for V_1 and $-V_2$ connected, i.e., $V_1 = -V_2$. The two resistors are then connected in parallel to give a resistive path $G_1 + G_2$, and the two resistor values can be controlled by different gate voltages V_{C1} and V_{C2}. Similarly, additional *balanced* MOSFET-resistor paths may be connected from V_1 or other signals to the integrator inputs (summing nodes). These paths can be turned on or off by an appropriate choice of gate voltages, so that transfer functions with different parameters (such as gains, quality factors, or pole-frequencies) or even transfer functions of different types (such as bandpass, low-pass, etc.) are obtainable.

If better linearity is required than is obtainable with the two-transistor MOSFET circuit in Figure 16.4c that replaces each resistor in the balanced active *RC* structure by one MOSFET, a four-MOSFET cross-coupled modification for each resistor pair can be used instead [7, paper 2-B.7]. The configuration is illustrated in Figure 16.5. The current difference can be shown to equal

$$\Delta I = I_1 - I_2 = K(V_{C1} - V_{C2})[V_1 - (-V_1)] \tag{16.7}$$

that is, ΔI is proportional to the product of the difference of the input signals and the difference of the applied gate voltages V_{Ci}, $i = 1, 2$. This indicates that one may interchange the input and gate-control voltages and reduce the drive requirements of the previous op-amp stage because no resistive current flows, only the small gate-capacitor current flows with the input applied at the control gates (now at $\pm V_1$). The price paid for this advantage is that the requirements on the control voltage source become more difficult and may necessitate an appropriate common-mode voltage on the signal lines.

Additional resistive or capacitive inputs may, of course, be used to design more general lossy integrators as illustrated in Figure 16.6. By writing the node equation at the summing node, this circuit may be analyzed to realize

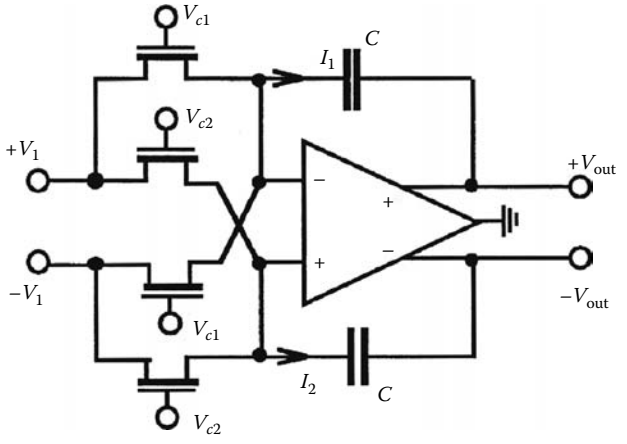


FIGURE 16.5 Integrator using a four-MOSFET configuration for greater linearity.

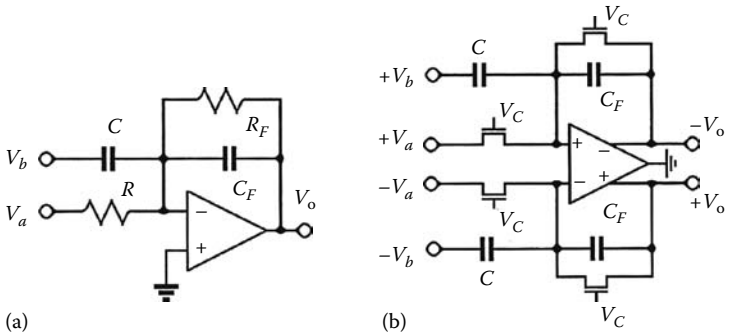


FIGURE 16.6 (a) Lossy integrator with capacitive feed-in branch; (b) MOSFET-*C* equivalent.

$$V_o = -\frac{1}{sC_F + G_F}(GV_a + sCV_b) \tag{16.8}$$

where the conductors are implemented in Figure 16.6b as

$$R(V_C) = \frac{1}{\mu C_{ox}(W/L)(V_C - V_T)}; \quad R_F(V_C) = \frac{1}{\mu C_{ox}(W_F/L_F)(V_C - V_T)} \tag{16.9}$$

as was derived in Equation 16.2. The controlling gate voltages V_C for the two resistors R and R_F may, in general, be different so that the resistor values may be tuned (or turned on or off) independently.

With lossy and lossless MOSFET-*C* integrators available, we can obtain not only first- and second-order filters as shown in Section 16.2.1.2, but also simulations of *LC* ladders in much the same way as was done in Chapter 15 for active *RC* circuits. We discuss this procedure in some detail in Section 16.2.3.

16.2.1.2 First- and Second-Order Sections

Based on the principle of balancing a single-ended structure, appropriate* standard classical active *RC* filters from the literature can be converted into balanced form and resistors can be replaced by MOSFETs. As an illustration, consider the single-ended prototype (Tow–Thomas) biquad in Figure 16.7a. As is typical for second-order active *RC* filter sections (see Chapter 13), the circuit is a two-integrator loop consisting of inverting lossy and lossless integrators (in addition to an inverter to keep the loop gain negative for stability reasons). The realized transfer function is not important to our discussion, but we point out that all resistors are voltage-driven (by the signal source or by an op-amp) and at their other ends all resistors are connected to an op-amp input, i.e., they are at virtual ground. Thus, this circuit satisfies our earlier condition for conversion to a MOSFET-*C* structure. Figure 16.7b shows the balanced active *RC* equivalent that is necessary to eliminate the nonlinear performance. Replacing the resistors by MOSFETs biased in the triode region leads to the final MOSFET-*C* version in Figure 16.7c. The W/L ratios are chosen appropriately to realize the prescribed resistor values

$$G_i = \frac{W_i}{L_i}\mu C_{ox}(V_C - V_T) \tag{16.10}$$

with excellent resistor matching given by aspect ratios

* All resistors must be voltage-driven from one side and see the same voltage, usually virtual ground, on the other.

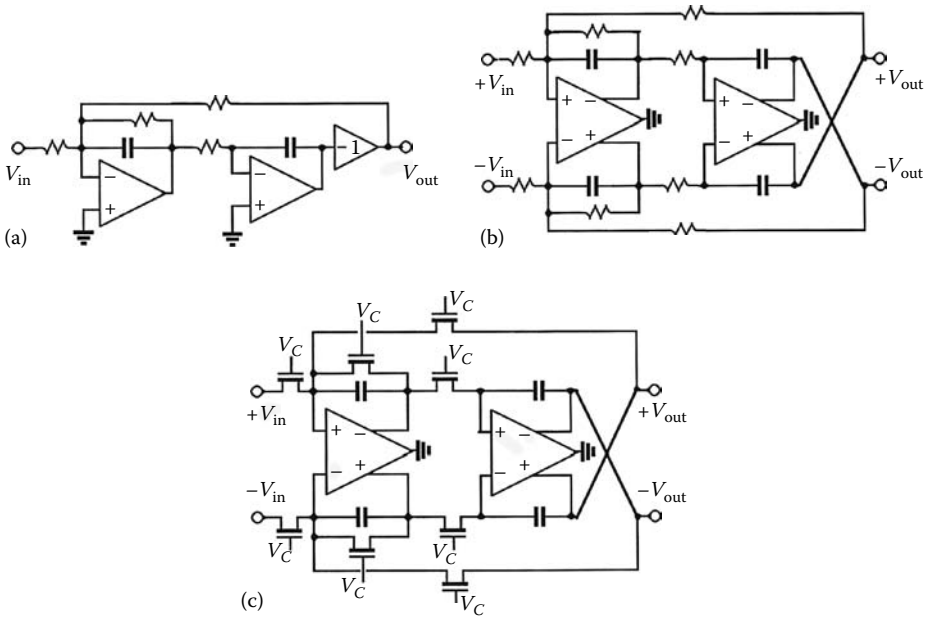


FIGURE 16.7 (a) Active *RC* prototype (Tow–Thomas biquad) for conversion into a MOSFET-*C* structure; (b) fully balanced version of the biquad with resistors; and (c) fully balanced version of the biquad with MOSFETs.

$$\frac{G_i}{G_k} = \frac{(W/L)_i}{(W/L)_k} \tag{16.11}$$

The voltage-variable resistors given by Equation 16.10 permit loss and time constants to be electronically tuned. Note that the inverter in the original circuit is not needed in Figure 16.7b and c because inversion in the differential topology is obtained by crossing wires.

The circuit may be made *programmable* by connecting additional MOSFET resistors with appropriate W/L ratios in parallel with the fundamental ones shown in Figure 16.7c and then switching them on or off as required by the desired values of the filter coefficients.

The above method indicates how first- or second-order sections can be obtained by choosing any suitable configuration from the active *RC* filter literature [1, Chapter 5], [3, Chapters 4 and 5] and converting it to balanced MOSFET-*C* form. Next we show a couple of entirely general first- and second-order sections that can be developed from the integrator in Figure 16.6b. The resulting circuits are shown in Figure 16.8 [7, paper 2-A.2]. If we combine the inputs $V_a = V_b = V_i$ of the integrator in Figure 16.6b and add two further cross-coupled feed-in capacitors, we obtain the first-order circuit in Figure 16.8a. Writing the node equation at the (inverting or noninverting)* op-amp input results in the transfer function

$$T_1(s) = \frac{V_o}{V_i} = \frac{s(C_1 - C_2) + G_1}{sC_F + G_2} \tag{16.12}$$

Note that this first-order circuit can realize zeros anywhere on the real axis. Similarly, if we combine two such integrators in a loop, with the individual signal paths and signs selected to assure negative feedback,

* Because the circuit is completely symmetrical, it is only necessary to derive the equations for one side, e.g., for V_o; the expressions for the other side, i.e., $-V_o$, are the same.

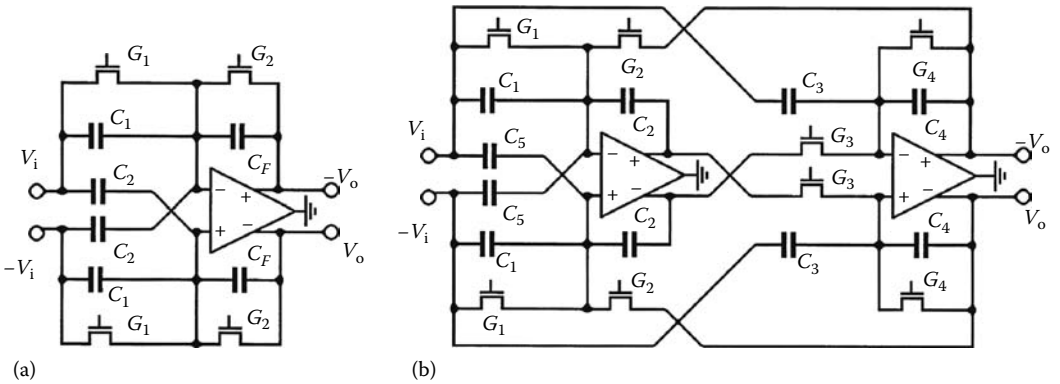


FIGURE 16.8 General (a) first- and (b) second-order MOSFET-*C* sections. The MOSFETs are labeled by the conductance values they are to implement.

TABLE 16.1 Functions Realizable with Figure 16.8b

Function	Choice of Elements
Bandpass	$G_1 = C_3 = 0$; $C_5 = 0$ if noninverting, $C_1 = 0$ if inverting
Low-pass	$C_1 = C_3 = C_5 = 0$
High-pass	$C_1 = C_5 = G_1 = 0$
Notch	$C_1 = C_5$
All-pass	$(C_1 - C_5)G_3 = -C_2G_4$

we obtain the general second-order section in Figure 16.8b. Writing again the node equations at the op-amp input nodes leads to the transfer function

$$T_2(s) = \frac{V_o}{V_i} = \frac{s^2C_2C_3 + s(C_1 - C_5)G_3 + G_1G_3}{s^2C_2C_4 + sC_2G_4 + G_2G_3} \tag{16.13}$$

Observe that this circuit can realize zeros anywhere in the *s*-plane, depending on the choice of element values. Consequently, the sections in Figure 16.8 can be used to implement arbitrary high-order transfer functions in a cascade topology. Specifically, for the indicated choice of elements the general biquad function in Equation 16.13 realizes the different transfer functions in Table 16.1.

16.2.2 Cascade Realizations

The realization of high-order transfer functions as a connection of low-order sections, including the *cascade*, *multiple-loop feedback*, and *coupled-biquad* approaches, is identical to that discussed for discrete active *RC* filters. The difference lies only in the implementation of the sections in fully integrated form. We do not repeat the process here but only discuss the most prevalent method, cascade design, in terms of an example and encourage the reader to refer to the earlier discussion in Chapter 15 for details. To repeat briefly, if a high-order function

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{N(s)}{D(s)} = \frac{a_ms^m + a_{m-1}s^{m-1} + \cdots + a_1s + a_0}{s^n + b_{n-1}s^{n-1} + \cdots + b_1s + b_0} \tag{16.14}$$

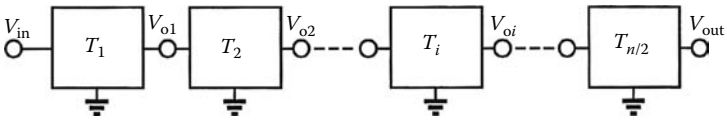


FIGURE 16.9 Cascade realization of an *n*th-order transfer function.

with $m \leq n$ and $n > 2$ is given, it is factored into second-order sections (and one first-order section if n is odd),

$$H(s) = \prod_{i=1}^{n/2} T_i(s) = \prod_{i=1}^{n/2} k_i \frac{a_{2i}s^2 + a_{1i}s + a_{0i}}{s^2 + s\omega_{0i}/Q_i + \omega_{0i}^2} = \prod_{i=1}^{n/2} k_i t_i(s) \tag{16.15}$$

where each of the second-order functions

$$T_i(s) = \frac{V_{oi}}{V_{oi-1}} = k_i \frac{a_{2i}s^2 + a_{1i}s + a_{0i}}{s^2 + s\omega_{0i}/Q_i + \omega_{0i}^2} = k_i t_i(s) \tag{16.16}$$

is implemented as a suitable biquad with specified pole quality factor Q_i and pole frequency ω_{0i}, such as the one in Figure 16.8b realizing Equation 16.13. As was explained in Chapter 15, k_i is a suitable gain constant, chosen to equalize the signal levels in order to optimize the dynamic range, and $t_i(s)$ is a gain-scaled transfer function. Note that in writing Equation 16.15 we assumed that n is even. Figure 16.9 shows the general (single-ended) structure of the filter.

To provide an example, assume we wish to realize a delay of $\tau_D = 0.187$ μs via a fifth-order Bessel approximation, but with a transmission zero at $f/f_n = 4.67$ to improve the attenuation in the stopband.* The transfer function is found to be

$$H_5(s) = \frac{43.3315(s^2 + 21.809)}{(s + 3.6467)(s^2 + 6.7040s + 14.2729)(s^2 + 4.6494s + 18.1563)} \tag{16.17}$$

The normalizing frequency is $f_n = 1/(2\pi\tau_D) = 850$ kHz [1, Chapter 1], [3, Chapter 10]. Let us choose a cascade implementation with the circuits in Figure 16.8. Factoring Equation 16.17 leads to the first-order and the two second-order functions on the left-hand side of Equation 16.18 to be realized by the functions on the right, which are obtained from Equations 16.12 and 16.13:

$$T_1(s) = \frac{3.6467}{s + 3.6467} \rightarrow T_1(s) = \frac{V_{o1}}{V_{in}} = \frac{G_1}{sC_F + G_2} \tag{16.18a}$$

$$T_2(s) = \frac{14.2729}{s^2 + 6.7040s + 14.2729} \rightarrow T_2(s) = \frac{V_{o2}}{V_{o1}} = \frac{G_1G_3}{s^2C_2C_4 + sC_2G_4 + G_2G_3} \tag{16.18b}$$

$$T_3(s) = \frac{0.8325(s^2 + 21.809)}{s^2 + 4.6494s + 18.1563} \rightarrow T_3(s) = \frac{V_{o3}}{V_{o2}} = \frac{s^2C_2C_3 + G_1G_3}{s^2C_2C_4 + sC_2G_4 + G_2G_3} \tag{16.18c}$$

The components of the first- and second-order filters in Figure 16.8 are to be determined from these equations. The gain constants in the function were chosen to result in unity gain at dc. Comparing coefficients, we find with $\omega_n = 1/\tau_D \approx 2\pi$. 850 krad $s^{-1} \approx 5.341$ Mrad s^{-1}:

* Note that transmission zeros on the $j\omega$-axis can be added [3, Chapter 10.8] to a linear-phase network, such as a Bessel filter, without changing the phase because a zero factor, $(\omega_z^2 - \omega^2)$, is a purely real number on the $j\omega$-axis.

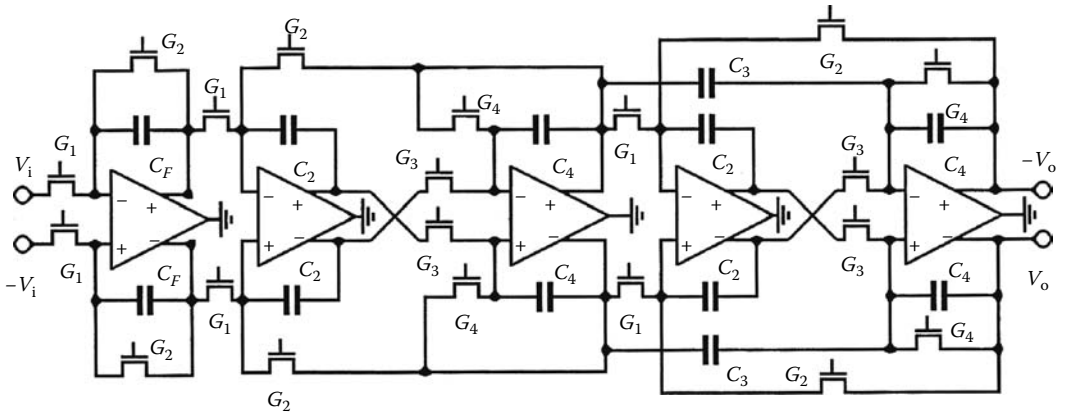


FIGURE 16.10 Circuit to realize the filter described by Equation 16.17. Note that for easy reference we have kept the subscripts on the elements in each section the same as in Figure 16.8.

From Equation 16.18a

$$G_1 = G_2 = 3.6467\omega_n C_F \tag{16.19a}$$

Choosing $C_F = 2$ pF gives $G_1 = G_2 = 38.952$ μS; also $C_1 = C_2 = 0$.

From Equation 16.18b

$$G_1 G_3 = G_2 G_3 = 14.2729\omega_n^2 C_2 C_4; \quad G_4 = 6.7040\omega_n C_4 \tag{16.19b}$$

Choosing $C_2 = C_4 = 2$ pF results in $G_1 = G_2 = G_3 = 40.354$ μS and $G_4 = 71.6$ μS; also, $C_1 = C_3 = C_5 = 0$.

From Equation 16.18c, $G_1 G_3 = 0.8325 \times 21.809\omega_n^2 C_2 C_4 = 18.1563\omega_n^2 C_2 C_4$;

$$G_2 G_3 = 18.1563\omega_n^2 C_2 C_4; \quad G_4 = 4.6494\omega_n C_4; \quad C_3 = 0.8325 C_4 \tag{16.19c}$$

Choosing $C_2 = 2$ pF and $C_4 = 10$ pF yields $C_3 = 8.325$ pF, $G_1 = G_2 = G_3 = 101.77$ μS and $G_4 = 248.32$ μS; also, $C_1 = C_5 = 0$. The remaining task is to convert the resistors into MOSFET devices. Assume the process provides transistors with $\mu C_{ox} = 120$ μA/V^2 and $V_T = 0.9$ V. For the choice of $V_C = 2$ V, the aspect ratios are then calculated from the above conductance values and from Equation 16.10 via

$$\frac{W_i}{L_i} = \frac{G_i(V_C)}{\mu C_{ox}(V_C - V_T)} = \frac{G_i(V_C)}{120\ \mu\text{A/V}^2 \times 1.1V} = \frac{G_i/\mu\text{S}}{132} \tag{16.20}$$

For instance, in the first-order section we find $W_1/L_1 = W_2/L_2 = 38.952/132 = 1/3.389$. The resulting circuit is shown in Figure 16.10.

16.2.3 Ladder Simulations

Using MOSFET-*C* integrators, the ladder simulation method for the MOSFET-*C* approach is entirely analogous to the active *RC* procedures discussed earlier in this book (see Chapter 15). The process is illustrated by a step-by-step generic example, which should guide the reader when implementing a specific design.

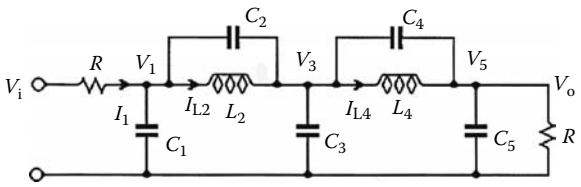


FIGURE 16.11 Fifth-order elliptic *LC* low-pass filter.

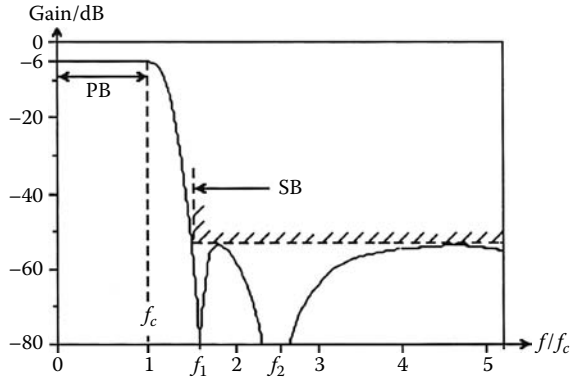


FIGURE 16.12 Transfer function magnitude of a fifth-order elliptic low-pass function.

Assume a fifth-order elliptic low-pass filter with the transfer function

$$H_{\text{ell}}(s) = \frac{\left(s^2 + \omega_1^2\right)\left(s^2 + \omega_2^2\right)}{(s + a)(s^2 + bs + c)(s^2 + ds + e)} \tag{16.21}$$

is prescribed, which with the help of readily available ladder synthesis software is realized by the *LC* ladder structure in Figure 16.11 with known component values. A plot of such a function is shown in Figure 16.12. The two transmission zeros f_1 and f_2 in the figure are obtained when L_2, C_2, and L_4, C_4, respectively, resonate. The *LC* active simulation proceeds by deriving the signal-flow graph equations [1, Chapter 6], [3, Chapter 15] or by writing the loop and node equations of the ladder along with the *V*–*I* relationships describing the functions of the elements:

$$I_1 = \frac{V_i - V_1}{R}, \quad I_{L2} = \frac{V_1 - V_3}{sL_2}, \quad I_{L4} = \frac{V_3 - V_5}{sL_4} \tag{16.22a}$$

$$V_1 = \frac{I_1 - [I_{L2} + sC_2(V_1 - V_3)]}{sC_1} \tag{16.22b}$$

$$V_3 = \frac{I_{L2} - I_{L4} + sC_2(V_1 - V_3) - sC_4(V_3 - V_5)}{sC_1} \tag{16.22c}$$

$$V_5 = V_o = \frac{I_{L4} + sC_4(V_3 - V_5)}{sC_5 + G} \tag{16.22d}$$

We recognize that these equations represent *integrations* of voltages into currents and currents into voltages. We also note that the currents through the capacitors C_2 and C_4 can be taken care of efficiently without resorting to integration: by connecting C_2 and C_4 directly to the voltage nodes V_1, V_3, and V_3, V_5, respectively, they conduct the currents as prescribed in Equation 16.22b through d. Next we reformat Equation 16.22a through d by eliminating I_1 from Equation 16.22a and b and rewriting Equation 16.22b through d such that V_1, V_3, and V_5, respectively, appear only on the left-hand side. The result is the new set of equations

$$I_{L2} = \frac{V_1 - V_3}{sL_2}; \quad I_{L4} = \frac{V_3 - V_5}{sL_4} \tag{16.23a}$$

$$V_1 = \frac{V_i G - I_{L2} + sC_2 V_3}{s(C_1 + C_2) + G} \tag{16.23b}$$

$$V_3 = \frac{I_{L2} - I_{L4} + sC_2 V_1 + sC_4 V_5}{s(C_2 + C_3 + C_4)} \tag{16.23c}$$

$$V_5 = V_o = \frac{I_{L4} + sC_4 V_3}{s(C_4 + C_5) + G} \tag{16.23d}$$

Recall that all signals in the MOSFET-*C* circuit are voltages, which in turn produce currents summed at the op-amp inputs, and that the integration constant must be *time* rather than *capacitance*. This is achieved by scaling the equations by a resistor R. We illustrate the process on Equation 16.23a and b:

$$I_{L4} = \frac{V_3 - V_5}{sL_4} \rightarrow \frac{I_{L4}}{G} = \frac{GV_3 - GV_5}{s(L_4 G)G} \tag{16.24a}$$

$$V_1 = \frac{V_i G - I_{L2} + sC_2 V_3}{s(C_1 + C_2) + G} \rightarrow V_1 = \frac{V_i G - (I_{L2}/G)G + sC_2 V_3}{s(C_1 + C_2) + G} \tag{16.24b}$$

Notice that $L_4 G^2$ has the unit of *farad*, i.e., it is a capacitor that, however, in Figures 16.13 and 16.14 will be labeled L_4 (and L_2, respectively) to help keep track of its origin. Similarly, I_{L2}/G and I_{L4}/G will be labeled V_{I2} and V_{I4}, respectively. The integrators can now be realized as in Figure 16.6b and then interconnected as the equations prescribe. Figure 16.13 illustrates the process for Equation 16.24. The MOSFET-*C* implementation of all appropriately scaled equations (Equation 16.23) leads to the

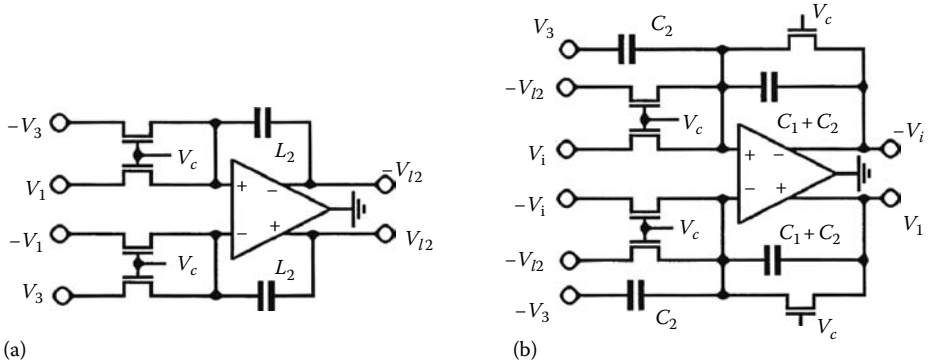


FIGURE 16.13 MOSFET-*C* implementation of (a) Equation 16.24a and (b) Equation 16.24b. All MOSFETs realize the value G.

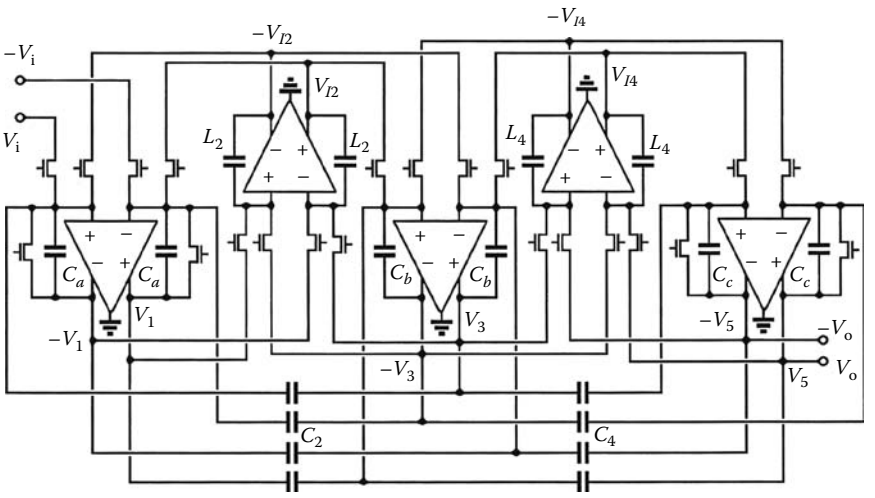


FIGURE 16.14 Operational simulation of the *LC* ladder of Figure 16.11 with MOSFET-*C* integrators; $C_a = C_1 + C_2$, $C_b = C_2 + C_3 + C_4$, $C_c = C_4 + C_5$.

fifth-order elliptic filter shown in Figure 16.14. As given in Equation 16.10, the aspect ratio of each MOSFET and the control voltage V_C are adjusted to realize the corresponding resistor values of the standard active *RC* implementation and all MOSFET gates are controlled by the same V_C for tuning purposes. Arrays of MOSFETs controlled by different values of V_C can be used to achieve programmable filter coefficients.

An often cited advantage of the MOSFET-*C* technique is the reduced sensitivity to parasitic capacitors, whereas the g_m-*C* approach discussed next must carefully account for parasitics by predistortion. Note from Figures 16.6b, 16.8, 16.10, and 16.14 that all capacitors and the MOSFET resistors are connected to voltage-driven nodes or to virtual ground so that parasitic capacitors to ground are of no consequence as long as amplifiers with sufficiently high gain and wide bandwidth are used. Fortunately, such amplifiers are being developed [7, paper 2-B.5] so that MOSFET-*C* circuits promise to become increasingly attractive in the future.

16.3 g_m-*C* Filters

At the time of this writing, the dominant active device used in the design of integrated continuous-time filters is the *transconductor* (g_m) or the *operational transconductance amplifier* (OTA) [8, Chapter 5], [3, Chapter 16]. Both names, g_m-*C* filters and OTA-*C* filters, are used in the literature; we will use the term g_m-*C* filter in this text. The main reasons for this prevalence appear to be the simple systematic design methods for g_m-*C* filters and, especially, the higher range of frequencies over which g_m-based filters can operate. Also, OTAs often have simpler circuitry (fewer elements) than op-amps. A transconductor is a voltage-to-current converter described by

$$I_{\text{out}} = g_m(s)V_{\text{in}} \tag{16.25}$$

where $g_m(s)$ is the frequency-dependent transconductance parameter with units of ampere/volt or siemens, abbreviated S. Typical values for g_m are tens to hundreds of microsiemens in CMOS and up to millisiemens in bipolar technology. A simplified small-signal equivalent circuit is shown in Figure 16.15. The dashed components in Figure 16.15 are parasitics that in an ideal OTA should be

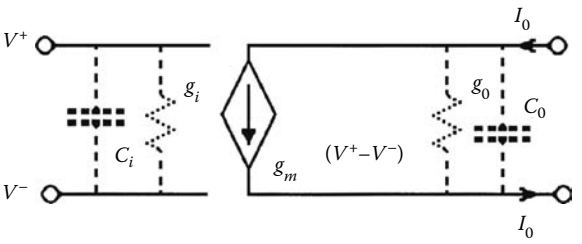


FIGURE 16.15 Small-signal equivalent circuit.

zero but that in practice must be accounted for. For common designs in CMOS technology, the input conductance g_i is zero; the input and output capacitances, c_i and c_o, are typically of the order of 0.05 pF or less and the output resistance $r_o = 1/g_o$ is in the range of 50 kΩ to 1 MΩ. The bandwidth of well-designed transconductors is so large that g_m in many cases can be regarded as constant, $g_m(s) = g_{m0}$, but for critical applications in high-frequency designs the transconductance pole and the resulting phase errors must be considered. A good model for these situations is

$$g_m(s) \approx g_{m0}e^{-s\tau} \approx \frac{g_{m0}}{1+s\tau} \approx g_{m0}(1-s\tau) \tag{16.26}$$

where $f = 1/(2\pi\tau)$ is the pole location (typically at several 100 MHz to 10 GHz) and the phase error $\Delta\phi = -\omega\tau$ is considered small, i.e., $\omega\tau \ll 1$. The three different approximations in Equation 16.26 for representing the frequency dependence are equivalent; the pole is used most often, the zero frequently results in simpler equations and algebra, and the phase may give better insight into the behavior of feedback loops.

The most commonly used circuit symbols are shown in Figure 16.16. Note that OTA designs with multiple differential inputs as in Figure 16.16c are readily available. They often lead to simpler filter designs with less silicon area and power consumption because only the OTA input stages must be duplicated, whereas the remaining parts of the OTA, such as output and common-mode feedback circuitry, can be shared. Essentially, if two OTAs with the same g_m value in a filter have a common output node (a frequent situation), the two OTAs can be merged into the circuit of Figure 16.16c, thus saving circuitry and power.

Filter design methods discussed in this section use only OTAs and capacitors: OTAs to provide gain and capacitors to provide integration. To establish time constants, resistors may also be required, but their function can be obtained from OTAs: ''resistors'' of value $1/g_m$ can be simulated by connecting the OTA output to its input with the polarities shown in Figure 16.17a and b. Inverting one pair of

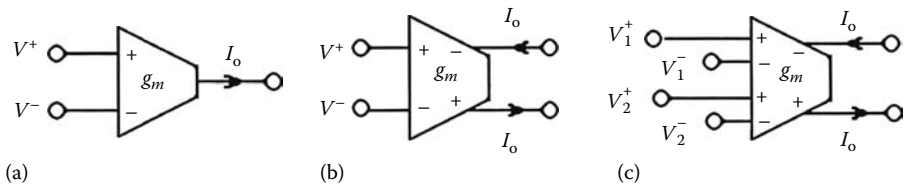


FIGURE 16.16 OTA symbols (a) differential input–single-ended output: $I_o = g_m(V^+ - V^-)$; (b) fully differential; and (c) with multiple inputs: $I_o = g_m\,[(V_1^+ - V_1^-) + (V_2^+ - V_2^-)]$.

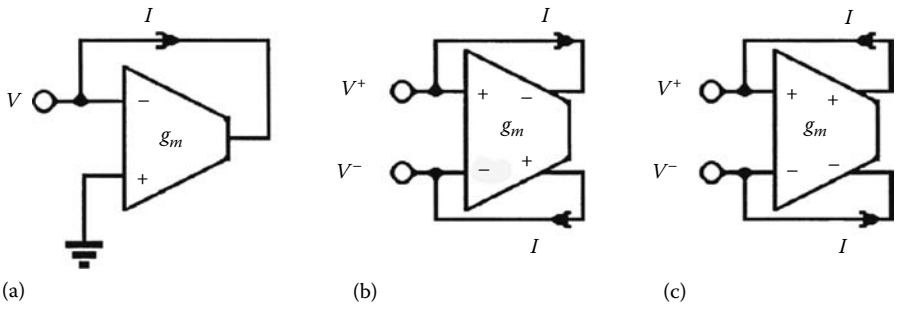


FIGURE 16.17 Simulated resistors: (a) positive single-ended of value $V/I = 1/g_m$; (b) positive differential resistor of value $(V^+ - V^-)/I = 1/g_m$; and (c) negative differential resistor $-1/g_m$.

terminals results in a negative resistor as in Figure 16.17c.* Since transconductors and capacitors can be used to build all components necessary for designing the filters, they are called transconductance-C or g_m-C filters. We discuss the remaining "composite" building blocks, integrators and gyrators† subsequently in Section 16.3.1.

In this section we will not go into the electronic circuit design methods for OTAs, but refer the reader to the literature, which contains a great number of useful transconductance designs in all current technologies. References [5,7] contain numerous papers that discuss practical transconductance circuits. The most popular designs currently use CMOS, but bipolar and BiCMOS are also widely employed, and GaAs has been proposed for applications at the highest frequencies or under unusually severe environmental conditions. Since transconductors are almost always used in open loop without local feedback, their input stages must handle the full amplitude of the signal to be processed. Typically, the OTA input stage is a differential pair with quite limited signal swing before nonlinearities become unacceptable. Thus, much design expertise has gone into developing linearization schemes for transconductance circuits. They have resulted in designs that can handle signals of the order of volts with nonlinearities of a fraction of 1%. Apart from simple source-degeneration techniques, the most commonly employed approaches use variations of the principle of taking the difference between the drain currents of two MOS devices in the saturation region but driven differentially, so that the difference current is linear in V_{gs}:

$$\begin{aligned} I_d^+ &= k(V_{gs} - V_T)^2 = k\left(V_{gs}^2 + V_T^2 - 2V_{gs}V_T\right) \\ I_d^- &= k(-V_{gs} - V_T)^2 = k\left(V_{gs}^2 + V_T^2 + 2V_{gs}V_T\right) \\ \Delta I_d &= I_d^+ - I_d^- = -4V_{gs}V_T \end{aligned} \tag{16.27}$$

Another approach reasons that the most linear (trans)conductance behavior should be obtainable from the current through a resistor. Thus, operating an MOS device in the resistive (triode) region,

$$I_d = k\left[(V_{gs} - V_T)V_{ds} - 0.5V_{ds}^2\right]$$

and taking the derivative with respect to V_{gs} for constant $V_{ds} = V_{DS}$ results in a perfectly linear transconductance,

* Such "negative resistors" are often used to cancel losses, for example to increase the dc gain of transconductors or the quality factors of filter stages. Specifically, a negative resistor can be employed to increase the quality factor of the inductor in Figures 16.21 and 16.22.

† A gyrator is a two-port circuit whose input impedance is inversely proportional to the load impedance: $Z_{in}(s) = r^2/Z_{load}(s)$. If $Z_{load} = 1/(sC)$, the input is inductive, $Z_{in}(s) = sr^2C = sL$. r is called the gyration resistance (see the discussion to follow).

$$g_m = \frac{dI_d}{dV_{gs}} = kV_{DS} \tag{16.28}$$

that furthermore can be adjusted (tuned) by varying a dc bias voltage (V_{DS}) as long as V_{DS} stays small enough for the transistor to remain in the triode region. The circuitry surrounding the triode-region MOS device must assure that V_{DS} remains constant and independent of the signal.

As mentioned, the literature contains numerous practical CMOS, bipolar, or biCMOS transconductance designs that require low power supply voltages (±1.5 V, or 0 to 3 V, or even less for low-power applications), and have acceptable signal swing (of the order of volts), with low nonlinearities (as low as a small fraction of 1%) and wide bandwidth (up to several hundred megahertz and even into the gigahertz range). Two further aspects of OTA design should be stressed at this point. First, since g_m-C filters often contain many transconductors, the designer ought to strive for simple OTA circuitry. It saves silicon real estate and at the same time often results in better frequency performance because of reduced parasitics at internal nodes. We point out though that there exists a trade-off between simple circuitry and large voltage swing: a wide linear signal range always requires special linearizing circuit techniques and, therefore, additional components. The second issue pertains to tuning. We mentioned earlier that continuous-time filters always require tuning steps to eliminate the effects of fabrication tolerances and component drifts. In IC technologies this implies that the circuit components must be electronically adjustable. Since (MOS) capacitors are generally fixed,* all tuning must be handled via the transconductance cells by changing one or more bias points. Usually two adjustments are needed: the magnitude of g_m must be varied to permit tuning the frequency parameters set by g_m/C-ratios and, as explained later, the phase $\Delta\phi$ must be varied to permit tuning the filters' quality factors. We discuss tuning methods in some detail in Section 16.4.

In the Section 16.3.1 we introduce the central building blocks from which g_m-C filters are constructed, the integrator and the gyrator. Just as we saw in the discussion of MOSFET-C circuits and in the earlier treatment of active RC filters, integrators are fundamental to the development of active filter structures, both for second-order sections and cascade designs, as well as for higher order LC ladder simulations. Gyrators along with capacitors are used to replace inductors in passive RLC filters so that many passive filter structures, such as LC ladders, can be directly translated into g_m-C form.

16.3.1 Basic Building Blocks

16.3.1.1 Integrators

Integrators are obtained readily by loading an OTA with a floating or a grounded capacitor as shown in Figure 16.18. Observe that the simpler technology of grounded capacitors requires four times the capacitor value and silicon area. Ideally, the integrator realizes the transfer function

$$\frac{V_o}{V_i} = -\frac{g_m}{sC} \tag{16.29a}$$

but notice that the function is sensitive to unavoidable parasitic capacitors as well as to the OTA output conductance g_o. Observe from Figure 16.19 that the output conductance is in parallel with the integrating capacitor C, and that the output capacitances C_o from the positive and negative output nodes of the OTA circuitry to ground add to the value of C. Furthermore, the designer should bear in mind that in IC technology floating capacitors have a substantial parasitic capacitance C_s (about 10% of the value of C) from the bottom plate to the substrate, i.e., to ac ground. To maintain symmetry, the designer may wish

* To implement small values of variable capacitor, MOS transistors connected as varactors can be used. The capacitor can be varied, e.g., via substrate bias.

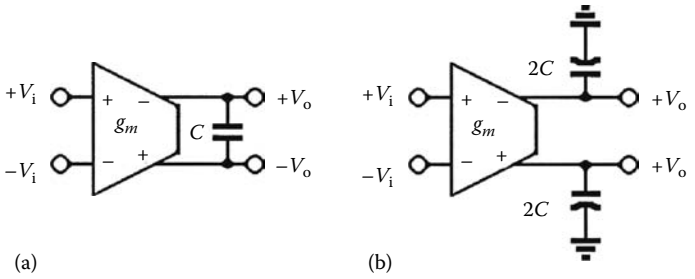


FIGURE 16.18 Integrator. The integrator capacitor may be floating (a) or grounded (b). Note that the grounded-capacitor realization requires four times the area.

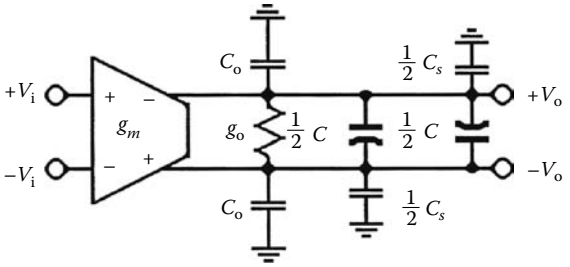


FIGURE 16.19 Parasitic capacitors associated with a g_m-C integrator.

to split the integrating capacitor into two halves connected such that the parasitic bottom plate capacitors $0.5C_s$ appear at the two OTA outputs. The situation is illustrated in Figure 16.19. Taking the parasitics into consideration, evidently, the integrator realizes

$$\frac{V_o}{V_i} = -\frac{g_m}{sC_{int} + g_o}\bigg|_{s=j\omega} = -\frac{g_m}{j\omega C_{int}(1 - j/Q_{int})} \tag{16.29b}$$

that is, it becomes lossy with a finite integrator quality factor

$$Q_{int} = \frac{\omega C_{int}}{g_o} \tag{16.30a}$$

and an effective integrating capacitor equal to

$$C_{int} = C + \frac{1}{2}\left(\frac{C_s}{2} + C_o\right) \tag{16.30b}$$

To maintain the correct integration constant as nominally designed, the circuit capacitor C should be predistorted to reflect the parasitics appearing at the integration nodes. The parasitics should be estimated as best as possible, for example, from a layout process file, and their values subtracted from the nominal value of C in the final layout. If grounded capacitors are used as in Figure 16.18b, the bottom plate should, of course, be connected to ground so that the substrate capacitances are connected between ground and the power supply. Thus, they are shorted out for the signals and play no role. Observe that the presence of parasitic capacitors tends to limit the high-frequency performance of these

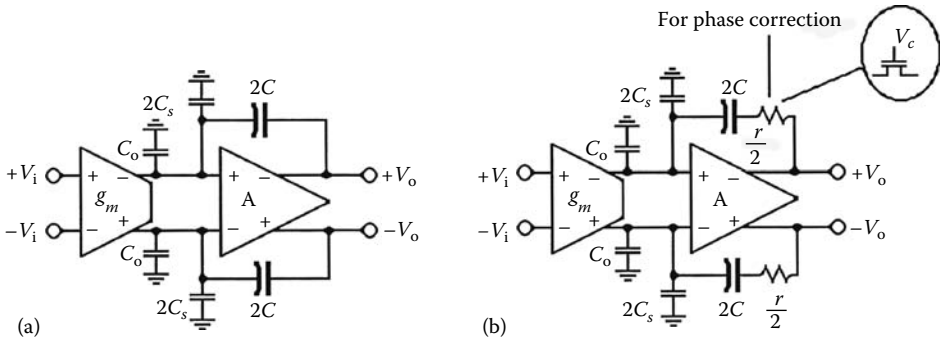


FIGURE 16.20 (a) g_m-C–op-amp integrator; (b) nominal correction for phase errors.

filters because high-frequency filters require large time constants, g_m/C, i.e., small capacitors.* The smallest capacitor C, however, must obviously be larger than the sum of all parasitics connected at the integrator output nodes to be able to absorb these parasitics.† Because the values of the parasitic capacitors can generally only be estimated, one typically chooses C to be three to five times larger than the expected parasitics to maintain some predictability in the design. The reader will notice that integrators with grounded capacitors, Figure 16.18b, have a small advantage in high-frequency circuits where parasitic capacitors become large relative to C. Because of the problem with parasitic capacitors, an alternative "g_m-C–op-amp" integrator is also used. It employs an op-amp to minimize their effects for the price of a second active device, increased noise, silicon area, and power consumption. Figure 16.20a shows the configuration. Notice that now the parasitic capacitors play no role because they are connected between ground and virtual ground (the op-amp inputs) so that they are never charged. A more careful analysis shows that the integrator realizes

$$\frac{V_o}{V_i} = \frac{g_m}{sC}\frac{1}{1+\frac{1}{A(s)}\left(1+\frac{C_p}{C}\right)} \approx \frac{g_m}{sC}\frac{1}{1+\frac{s}{\omega_t}\left(1+\frac{C_p}{C}\right)} \tag{16.31}$$

where $C_p = 0.5(C_o + 2C_s)$ represents the total parasitic capacitance at each of the op-amp input terminals. Evidently, the integrator has acquired a parasitic pole at

$$s = -\omega_t \frac{C}{C + C_p} \tag{16.32}$$

where we have modeled the amplifier gain as $A(s) \approx \omega_t/s$. The high-frequency performance is now limited by the op-amp behavior. It has been shown, though, that a low-gain wideband amplifier, essentially a second OTA with dc gain g_m/g_o, can be used for this application. Nevertheless, the second active device introduces parasitic poles (and zeros) whose effects must be carefully evaluated in practice. The dominant pole introduced by the op-amp may be canceled *nominally* by an rC phase lead as shown in Figure 16.20b. The circuit realizes

* Increasing g_m is generally not a satisfactory solution because it also increases the parasitics.

† Obviously, the highest operating frequencies are obtained for $C = 0$, that is, when a parasitic capacitor C_p is used as the integrating capacitor C. In that case, operating frequencies in the gigahertz range can be realized: $g_m/C \approx 2\pi \times 1$ GHz for $g_m = 100$ μS and $C_p \approx 15$ fF. Naturally, because of the uncertainty in the design values, an automatic tuning scheme is unavoidable.

$$\frac{V_o}{V_i} = \frac{g_m}{sC} \frac{1 + sCr}{1 - \frac{\omega}{\omega_t}\omega C_p r + \frac{s}{\omega_t}\left(1 + \frac{C_p}{C}\right)} \approx \frac{g_m}{sC} \frac{1 + sCr}{1 + \frac{s}{\omega_t}\left(1 + \frac{C_p}{C}\right)} \tag{16.33}$$

so that r should be chosen as

$$r = \frac{1}{\omega_t C}\left(1 + \frac{C_p}{C}\right) \tag{16.34}$$

to cancel the pole. The small resistor r may be a MOSFET in the triode region as indicated in Figure 16.20b so that $r(V_C)$ becomes variable for any necessary fine adjustments. Notice that the cancellation is only *nominal* because Equation 16.34 can never be satisfied exactly because of the uncertain values of ω_t and C_p.

16.3.1.2 Gyrators

A gyrator is defined by the equations

$$I_i = \frac{1}{r} V_o; \quad I_o = -\frac{1}{r} V_i \tag{16.35a}$$

where r is the so-called gyration resistance and the currents are defined as positive when flowing into the gyrator. Thus, the input impedance $Z_{in}(s)$ is inversely proportional to the load impedance $Z_{load}(s)$:

$$Z_{in}(s) = \frac{V_i}{I_i} = r^2 \frac{-I_o}{V_o} = r^2 \frac{1}{Z_{load}(s)} \tag{16.35b}$$

If a gyrator is loaded by a capacitor, $Z_{load}(s) = 1/(sC)$, the input impedance is proportional to frequency, i.e., it behaves like an inductor of value r^2C:

$$Z_{in}(s) = sr^2C = sL \tag{16.36}$$

Equation 16.35a indicates that a gyrator can be interpreted as a connection of an inverting and a noninverting transconductor of value $g_m = 1/r$ as shown in Figure 16.21a. This fact makes it very easy to build excellent gyrators with OTAs (see Figure 16.21b and c), whereas it has been found quite difficult to obtain good gyrators with op-amps. An exception is Antoniou's general impedance converter (GIC) [1, Chapter 5], [3, Chapter 14.1]. However, it is useful only at relatively moderate frequencies (up to about 5%–10% of the op-amp's gain-bandwidth product);* also, the circuit contains resistors that are not voltage-driven and, therefore, cannot readily be translated into a MOSFET-C equivalent as was discussed previously. The availability of good gyrators provides us with a convenient method for building high-frequency integrated g_m-C ladder filters, which is based on inductor-replacement to be discussed in Section 16.3.3.1.

Notice that the comments made earlier about the effects of parasitic capacitors also apply to inductor simulation: the parasitic input and output capacitors of the OTAs and the bottom-plate-to-substrate capacitances (Figure 16.19) add to the capacitor used to set the value of the simulated inductor. For instance, using the same notation as in Figure 16.19, the effective capacitor in Figure 16.21b equals

* As stated, Antoniou's GIC is really an *impedance converter*, not a gyrator (see Equation 15.63 and Figure 15.17). The GIC converts a load resistor into an inductive input impedance, rather than inverting a capacitor as described in Equation 16.35b.

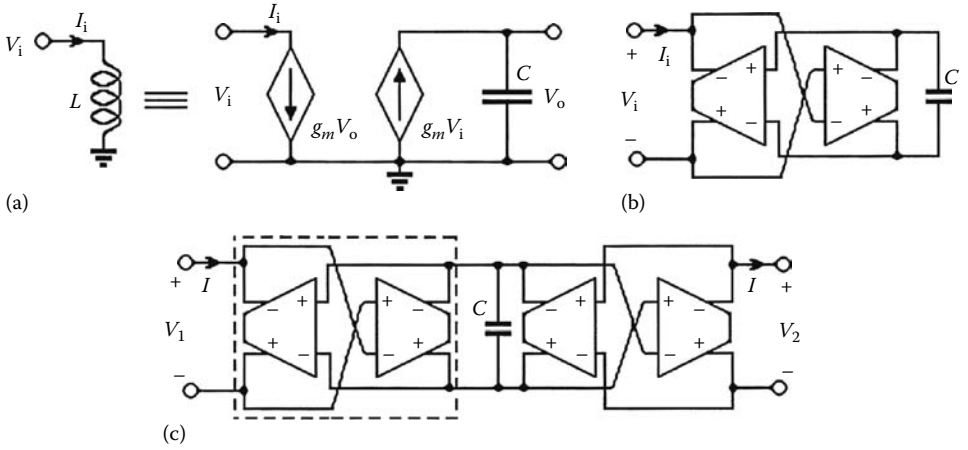


FIGURE 16.21 (a) Controlled-source realization of capacitively loaded gyrator to realize a grounded inductor; (b) differential g_m-C implementation; and (c) a floating inductor requires two gyrators.

$$C_{\text{eff}} = C + \frac{1}{2}\left(\frac{C_s}{2} + C_{\text{i}} + C_{\text{o}}\right) \tag{16.37}$$

where C_{i} is the parasitic capacitance at the input terminals of the OTA, see Figure 16.15, and C is assumed to be split as in Figure 16.19. Note also that a parasitic capacitor of value $C_{\text{in}} = C_{\text{i}} + C_{\text{o}}$ is measured across the inductor of Figure 16.21b so that the inductor

$$L_{\text{eff}} = \frac{C_{\text{eff}}}{g_m^2} \tag{16.38}$$

has a self-resonance frequency*

$$\omega_0 = \frac{1}{\sqrt{L_{\text{eff}} C_{\text{in}}}} \tag{16.39}$$

Finally, to complete the inductor model, recall that the OTAs have finite differential output conductances,† g_{o} (see Figure 16.15), which appear across the input and load terminals of the gyrator in Figure 16.21b. Consequently, the full inductive admittance realized by Figure 16.21b equals

$$Y_L(s) = g_{\text{o}} + sC_{\text{in}} + \frac{g_m^2}{sC_{\text{eff}} + g_{\text{o}}} \tag{16.40}$$

to yield the equivalent circuit in Figure 16.22 for the inductor L. The designer should keep this circuit in mind when using this method to develop a filter.

Evidently, the realized quality factor of the inductor equals $Q_L = \omega C_{\text{eff}}/g_{\text{o}}$. This means OTAs with large output resistance $r_{\text{o}} = 1/g_{\text{o}}$ are needed for high-quality inductor simulations.

* With modern submicron CMOS processes, the self-resonance frequency of electronic inductors can be as high as a few gigahertz with quality factors in the range of a few 100. See Ref. [9].

† As was mentioned, the input conductances can normally be neglected.

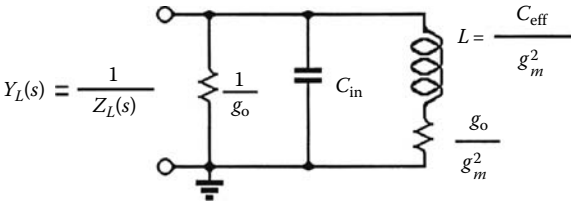


FIGURE 16.22 Passive equivalent circuit for the grounded inductor simulated in Figure 16.21b.

Next we discuss first- and second-order g_m-C sections used as building blocks for the cascade approach to high-order filter design. As was the case for the MOSFET-C method, we will see that g_m-C sections are constructed by interconnecting integrators.

16.3.1.3 First- and Second-Order Sections

Consider the integrator in Figure 16.18a; we make it lossy by loading it with the resistor in Figure 16.17b. Let the input signals also be fed through capacitors into the integrator output nodes as shown in Figure 16.23. The circuit is readily analyzed by writing Kirchhoff's current law at the output node to yield

$$\frac{V_o}{V_i} = -\frac{sC_1 + g_{m1}}{s(C_1 + C) + g_{m2}} \tag{16.41}$$

The circuit may realize any desired first-order function by choosing the appropriate values for the transconductances and the capacitors. For example, a low-pass can be obtained by setting $C_1 = 0$; a high-pass results from $g_{m1} = 0$ and possibly $C = 0$; and $C = 0$ and $g_{m1} = -g_{m2}$ results in an all-pass function.

A second-order block can be designed by interconnecting two integrators in a variety of feedback configurations. To keep the method more transparent, we show one such possibility in Figure 16.24 with single-ended outputs. A differential structure is obtained by mirroring the circuit at ground and duplicating the appropriate components as was demonstrated in connection with Figure 16.4. Let us disregard for now the dashed OTA with inputs V_a and V_b and apply an input signal V_i; writing the node equations for Figure 16.24 we obtain

$$g_{m3}(V_i - V_2) + g_{m4}(V_1 - V_3) = 0; \quad sC_1V_1 = -g_{m1}V_3; \quad sC_2V_2 = -g_{m2}V_1 \tag{16.42}$$

By eliminating two of the three voltages V_1, V_2, or V_3 from these equations, we obtain the bandpass, low-pass, and high-pass functions, respectively,

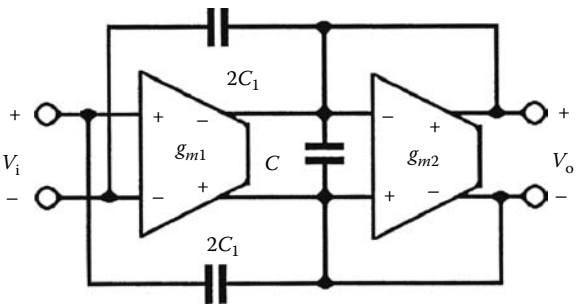


FIGURE 16.23 First-order g_m-C section.

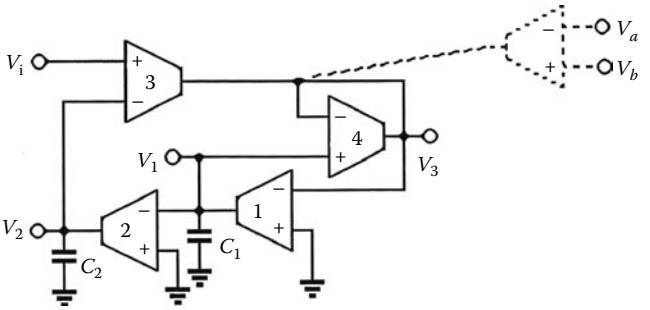


FIGURE 16.24 General biquadratic g_m-C section.

$$\frac{V_1}{V_i} = H_{BP}(s) = \frac{-sC_2g_{m1}g_{m3}}{s^2C_1C_2g_{m4} + sC_2g_{m1}g_{m4} + g_{m1}g_{m2}g_{m3}} \tag{16.43a}$$

$$\frac{V_2}{V_i} = H_{LP}(s) = \frac{g_{m1}g_{m2}g_{m3}}{s^2C_1C_2g_{m4} + sC_2g_{m1}g_{m4} + g_{m1}g_{m2}g_{m3}} \tag{16.43b}$$

$$\frac{V_3}{V_i} = H_{HP}(s) = \frac{s^2C_1C_2g_{m3}}{s^2C_1C_2g_{m4} + sC_2g_{m1}g_{m4} + g_{m1}g_{m2}g_{m3}} \tag{16.43c}$$

In *any* electrical network, one can generate different numerator polynomials, i.e., different transmission zeros, without disturbing the poles, i.e., the system polynomial, by applying an input *voltage* to any node that is lifted off *ground*, or by sending an input *current* into any *floating* node. The second of these possibilities is illustrated in Figure 16.24 in dashed form where a current $g_m(V_b - V_a)$ is sent into Node 3. We leave the analysis to the reader. We demonstrate the first possibility by applying V_i to the noninverting terminals of OTA_1 and OTA_2, which are grounded in Figure 16.24. The reader may show by routine analysis that lifting these two nodes off ground and then applying V_i to both of them (in addition to the original input) results in the complete biquadratic transfer function

$$\frac{V_3}{V_i} = \frac{s^2C_1C_2g_{m3} + sg_{m1}(C_2g_{m4} - C_1g_{m3}) + g_{m1}g_{m2}g_{m3}}{s^2C_1C_2g_{m4} + sC_2g_{m1}g_{m4} + g_{m1}g_{m2}g_{m3}} \tag{16.44}$$

with which, for example, a notch filter may be realized by setting $C_2g_{m4} = C_1g_{m3}$.

A great variety of second-order sections can be found in the literature, e.g., [3, Chapter 16.3]. Many are designed for specific transfer functions rather than the general circuit in Figure 16.24, and are often simpler and contain fewer OTAs. Readers are well advised to scan the available literature for the best circuit for their applications.

16.3.2 Cascade Realizations

Apart from modularity and simple design methods, the main advantage of the cascade approach is its generality: a cascade structure can realize a transfer function with arbitrary zero locations, whereas simulations of lossless *LC ladders* discussed below are restricted to $j\omega$-axis transmission zeros. Implementing a prescribed transfer function by the cascade method with g_m-C integrated filters follows the same principles that were discussed in Chapter 15 for discrete active *RC* filters and leads to the filter structure in Figure 16.9. The difference lies only in the final realization of the sections in monolithic form. We demonstrate the principle with the example of a high-frequency filter for the read/write channel of a

TABLE 16.2 Filter Parameters for Equation 16.45

	Pole Frequency (Normalized to 10^7 s^{-1})	Pole Quality Factor (Normalized to 10^7 s^{-1})	Zero
Biquad 1	$\omega_1 = 1.14762$	$Q_1 = 0.68110$	$\sigma_z = 0.95$
Biquad 2	$\omega_2 = 1.71796$	$Q_2 = 1.11409$	
Biquad 3	$\omega_3 = 2.31740$	$Q_3 = 2.02290$	
Section 4	$\sigma = 0.86133$	—	

magnetic disk storage system,* where the most critical specification is constant delay, i.e., linear phase. To this end, let us discuss the design of a seventh-order cascade low-pass with a constant delay approximated in the Chebyshev (equiripple) sense. The specifications call for a delay variation of maximally 1 ns over the passband and a bandwidth $f_c = 10$ MHz. The transfer function to be implemented is

$$H_7(s) = \frac{K_0\left(s^2 - \sigma_z^2\right)}{(s+\sigma)(s^2 + s\omega_1/Q_1 + \omega_1^2)(s^2 + s\omega_2/Q_2 + \omega_2^2)(s^2 + s\omega_3/Q_3 + \omega_3^2)} \tag{16.45}$$

with the required parameters given in Table 16.2. The purpose of the symmetrical pair of zeros at $\pm\sigma_z$ is magnitude equalization to effect a gain boost for pulse slimming. Note that these zeros do not affect the phase or the delay because the factor $(\omega^2 + \sigma_z^2)$ is real on the $j\omega$-axis. The fully differential second-order circuit chosen for the low-pass sections is shown in Figure 16.25a. Simple analysis shows that it realizes the function

$$H_{\mathrm{LP}}(s) = \frac{V_{\mathrm{LP}}}{V_{\mathrm{i}}} = \frac{g_{m1}g_{m2}}{s^2C_1C_2 + sC_1g_{m2} + g_{m1}g_{m2}} = \frac{\omega_0^2}{s^2 + s\omega_0/Q_0 + \omega_0^2} \tag{16.46}$$

where

$$\omega_0 = \sqrt{\frac{g_{m1}g_{m2}}{C_1C_2}} \tag{16.47a}$$

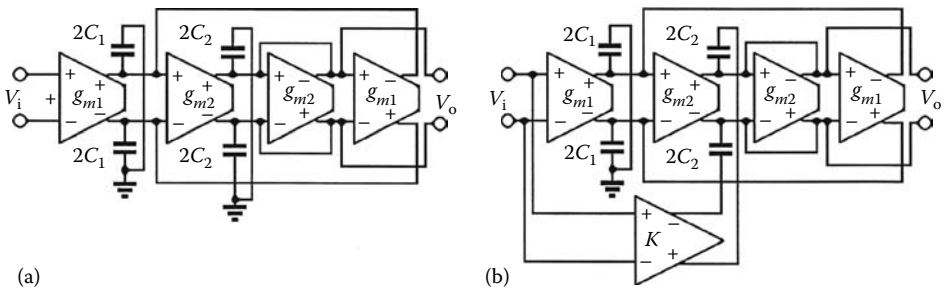


FIGURE 16.25 (a) Second-order low-pass g_m-C section; (b) an equalizer section.

* A similar commercially successful design in bipolar technology is discussed in Ref. [12].

and

$$Q_0 = \sqrt{\frac{g_{m1}C_2}{g_{m2}C_1}} \tag{16.47b}$$

are the pole frequency and pole quality factor, respectively. There remains the question of how to obtain the two real transmission zeros at $\pm\sigma_z$. For this purpose we recall that the zeros of a transfer function can be changed without destroying the poles by feeding the input signal or a fraction thereof into any of the ground nodes lifted off ground. For the situation at hand this can be accomplished by lifting the capacitors $2C_2$ in the low-pass off ground and feeding KV_i with the appropriate polarity into the terminals so generated. Figure 16.25b shows the resulting circuit, which may be analyzed to yield the transfer function

$$\frac{V_\mathrm{o}}{V_\mathrm{i}} = \frac{-Ks^2C_1C_2 + g_{m1}g_{m2}}{s^2C_1C_2 + sC_1g_{m2} + g_{m1}g_{m2}} \tag{16.48}$$

with the zero at $\sigma_\mathrm{z} = \pm\omega_0 K^{-1/2}$. Observe that the two transconductors labeled g_{m1} and g_{m2}, respectively, have common output terminals; they can, therefore, be merged into one double-input transconductor each as discussed in connection with Figure 16.16c. Further, we need the circuit in Figure 16.23 with $C_1 = 0$ to realize a first-order section, a lossy integrator with the function

$$\frac{V_\mathrm{o}}{V_\mathrm{i}} = -\frac{g_{m1}}{sC + g_{m2}} \tag{16.49}$$

Finally, we must determine the component values. To this end we compute from Equations 16.47 and 16.49 the relationships

$$\begin{gathered} C_1 = \frac{g_m}{\omega_i Q_i} = \frac{12.5}{\omega_i Q_i}\ \text{pF}; \quad C_2 = g_m\frac{Q_i}{\omega_i} = 12.5\frac{Q_i}{\omega_i}\ \text{pF}; \\ C = \frac{g_m}{\sigma} = \frac{12.5}{\sigma}\ \text{pF}; \quad K = \left(\frac{\omega_1}{\sigma_z}\right)^2 \end{gathered} \tag{16.50}$$

where we used that ω_i in Table 16.2 was normalized by a factor $10^7\ \mathrm{s}^{-1}$ and we assumed for simplicity that all OTAs have the same transconductance values,* $g_{m1} = g_{m2} = g_m = 125\ \mu\text{S}$. Lastly, to illustrate the need to account for parasitics, we observe that in Figure 16.25 the capacitor C_1 is paralleled by $2C_\mathrm{o}$ and $1C_\mathrm{i}$, and the capacitor C_2 by $2C_\mathrm{o}$ and $3C_\mathrm{i}$. The third input capacitor in parallel with C_2 is arrived at by the fact that each biquad must drive the next biquad in the cascade connection and there sees an OTA input. The capacitor C in Figure 16.23 is in parallel with $2C_\mathrm{o}$ and $1C_\mathrm{i}$. To arrive at numerical values we will assume $C_\mathrm{o} = 0.09$ pF and $C_\mathrm{i} = 0.04$ pF. Consequently, the capacitor values are obtained from Equation 16.50 as

$$C_1 = \frac{12.5}{\omega_i Q_i}\ \text{pF} - 0.22\ \text{pF}; \quad C_2 = 12.5\frac{Q_i}{\omega_i}\ \text{pF} - 0.3\ \text{pF}; \quad C = \frac{12.5}{\sigma}\ \text{pF} - 0.22\ \text{pF} \tag{16.51}$$

Table 16.3 contains the computed capacitor values and Figure 16.26 shows the final cascade block diagram with the equalizer section leading and the first-order low-pass at the end. The two control

* This assumption is for convenience of design and layout because it permits a given transconductance cell to be used throughout the circuit. However, using different transconductance values in different sections has the advantage that capacitor values may be equalized, since by Equation 17.50 the capacitors are proportional to g_m.

TABLE 16.3 Component Values of the Cascade Filter for Equation 16.45

Section	$i=1$	$i=2$	$i=3$	$i=4$
C_{1i}	15.77 pF	6.31 pF	2.45 pF	—
C_{2i}	7.12 pF	7.81 pF	10.61 pF	—
C	—	—	—	14.29 pF
K	1.459	—	—	—

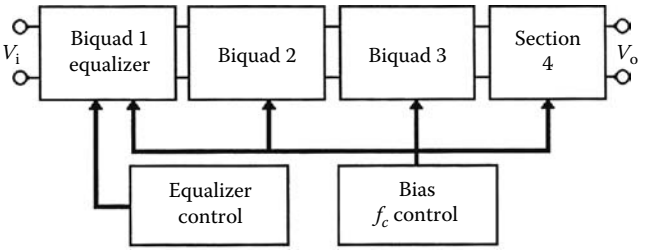


FIGURE 16.26 Structure of the seventh-order low-pass filter including control circuitry.

blocks are necessary to be able to tune the filter electronically: the gain K is varied via the *Equalizer control* block to set the position of the zeros and thereby the amount of gain boost, i.e., pulse slimming. Electronic tuning of the frequency parameter g_m/C is accomplished via adjusting the bias currents of the OTAs by the block labeled *Bias f_c control*. Thereby uncontrollable changes in the value g_m/C due to process variations or temperature can be accounted for. Details of such a control scheme are discussed in Section 16.4.

16.3.3 Ladder Simulations

As mentioned earlier, the main reason for using the popular *LC* ladder simulation method for filter design is the generally lower passband sensitivity of this topology to component tolerances [1, Chapter 3], [3, Chapter 13], see also Chapter 7. As before, the procedures are, in principle, identical to those discussed in connection with discrete circuits and are best illustrated with the help of a generic example. Let us consider again the classical ladder structure in Figure 16.11, which realizes the fifth-order elliptic low-pass characteristic Equation 16.21 and is described by Equation 16.22. Two methods are available to simulate the ladder. The first and most intuitive method replaces the inductors L_2 and L_4 by capacitively loaded gyrators (Figure 16.21). The second method recognizes that the inductors and the grounded capacitors in Figure 16.11 perform the function of integration. This signal-flow graph method is completely analogous to the process discussed earlier in Chapter 15 and for MOSFET-*C* filters, and will be presented in Section 16.3.3.2. We point out here that the element replacement and the signal-flow graph methods lead to the same circuitry [3, Chapter 16.4] so that either method can be used at the designer's convenience.

16.3.3.1 Element Replacement Methods

Replacing the inductors L_2 and L_4 in Figure 16.11 by capacitively loaded gyrators and using differential balanced circuitry leads to the circuit in Figure 16.27a. It is obtained by first converting the voltage source to a current source (Norton transformation), which also converts the series source resistor R into a shunt resistor. The first OTA in Figure 16.27a performs the source transformation, the second OTA is the grounded resistor. Since the two inductors are floating, the implementation of each requires two gyrators, i.e., four OTAs (see Figure 16.21c). Note that all OTAs are identical, and that all

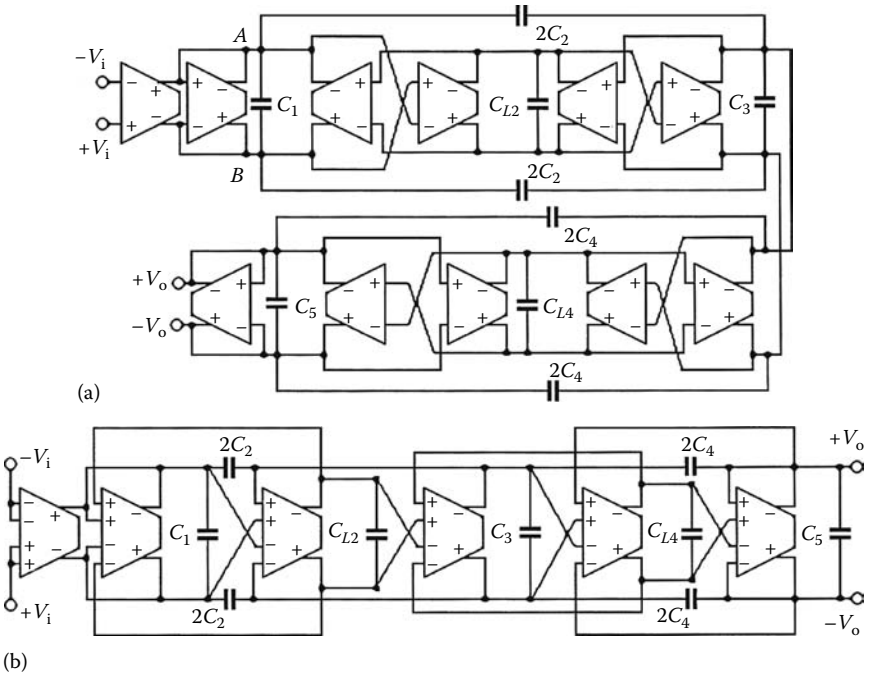


FIGURE 16.27 (a) Transconductor-*C* simulation by the element replacement method of the fifth-order elliptic low-pass ladder in Figure 16.11, including source and load resistors. All g_m-cells are identical. Note that the floating inductors require two gyrators for implementation. (b) The circuit with dual-input OTAs.

capacitors except C_2 and C_4 could be grounded; for example, instead of connecting C_1 *between* nodes *A* and *B* in Figure 16.27a, capacitors of value $2C_1$ could be connected from *both* nodes *A* and *B* to ground. Comparing the active circuit with the *LC* prototype readily identifies both structure* and components. The element values are obtained directly from the prototype *LC* ladder, e.g., from published tables, such as Ref. [2], or from appropriate synthesis software. Labeling for the moment the normalized components in the prototype *LC* ladder by the subscript *n*, i.e., R_n, $C_{i,n}$, and $L_{i,n}$, the transformation into the real component values with units of [F] and [H] is achieved by the following equations:

$$R = \frac{R_n}{g_m}; \quad C_i = C_{i,n}\frac{g_m}{\omega_c}; \quad L_i = L_{i,n}\frac{1}{g_m\omega_c} = \frac{C_{Li}}{g_m^2} \rightarrow C_{Li} = L_{i,n}\frac{g_m}{\omega_c} \tag{16.52}$$

where

g_m is the transconductance value chosen by the designer
ω_c is the normalizing frequency (usually the specified passband corner frequency)
R_n is in most prototype designs normalized to $R_n = 1$

Naturally, as discussed earlier, all capacitor values must be predistorted to account for the parasitic capacitors appearing at the capacitor nodes. For example, note that C_{L2} is paralleled by two OTA input and two OTA output capacitors, and for symmetrical layout (Figure 16.19), by $0.25C_s$. We see that the

* The number of floating capacitors is of course doubled because of the balanced differential structure of the active implementation.

element replacement method is a very straightforward design process; it has been found to work very well in practice.

Figure 16.27b shows the same circuit but realized with dual-input OTAs (Figure 16.16c). As was pointed out earlier, this merging of OTAs can always be used when two (or more) OTAs share a common output node. It results in simplified circuitry, and possibly reduced power consumption and chip area. In such cases, only the linearized input stages of the OTA must be duplicated, but bias, output, and common-mode rejection circuitry can be shared. Observe also that the input voltage is applied to both inputs of the first dual-input OTA in Figure 16.27b, thereby doubling its value of g_m. This multiplies the transfer function by a factor of two and eliminates the 6 dB loss inherent in the *LC* ladder (see Figure 16.12).

A small example will show the design process. For an antialiasing application we need to realize a third-order elliptic low-pass filter with $f_c = 10$ MHz bandwidth, 0.5 dB passband ripple, 17.5 MHz stopband edge, and 23 dB stopband attenuation. It leads to the transfer function [2]

$$H(s) = \frac{0.28163(s^2 + 3.2236)}{(s + 0.7732)(s^2 + 0.5016s + 1.1742)} \tag{16.53}$$

and the normalized element values $R_n = 1$, $C_{1n} = C_{3n} = 1.293$, $C_{2n} = 0.3705$, and $L_{2n} = 0.8373$. The topology is as in Figure 16.11 with $C_{4n} = L_{4n} = C_{5n} = 0$. Figure 16.28 shows the active circuit. Observe that we have realized each of the floating capacitors C_j as $0.5C_j + 0.5C_j$, $j = 1, 2, 3, L2$, with inverted bottom-plate connections. This design choice preserves symmetry in the balanced differential layout by placing the unavoidable substrate capacitors of value $\approx 0.1 \times (0.5C_{L2})$ *at each* of the upper and lower nodes of C_{L2} and $\approx 0.1 \times (0.5C_k + 0.5C_2)$ at the upper and lower nodes of C_k, $k = 1, 3$. Choosing the value of transconductance as $g_m \approx 180$ μS, using Equation 16.51, $\omega_c = 2\pi \times 10 \times 10^6$ s^{-1}, and observing the necessary predistortion for the *differential* parasitic OTA input and output capacitors, $C_i = 0.03$ pF and $C_o = 0.08$ pF, respectively, and for the bottom-plate capacitors C_s assumed to be 10% of the corresponding circuit capacitor value, results in

$$\begin{aligned}
C_1 &= C_{1n}\frac{g_m}{\omega_c} - 3C_o - 2C_i - 0.1\frac{1}{2}\left(C_2 + \frac{C_1}{2}\right) = 3.191 \text{ pF} \\
C_2 &= C_{2n}\frac{g_m}{\omega_c} = 1.061 \text{ pF} \\
C_3 &= C_{3n}\frac{g_m}{\omega_c} - 2(C_o + C_i) - 0.1\left(\frac{C_2}{2} + \frac{C_3}{4}\right) = 3.268 \text{ pF} \\
C_{L2} &= L_{2n}\frac{g_m}{\omega_c} - 2(C_o + C_i) - 0.1\frac{C_L}{4} = 2.120 \text{ pF}
\end{aligned} \tag{16.54}$$

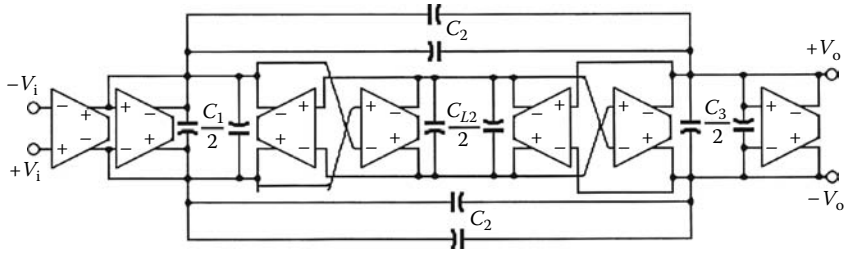


FIGURE 16.28 Realization of the third-order elliptic low-pass ladder of Equation 16.53. The capacitor values indicated refer to *each* of the pair of capacitors.

Notice that in a ladder structure all parasitic capacitors can be absorbed in the circuit capacitors. Consequently, no new parasitic poles or zeros are created that might destroy the transfer function shape. This is a further important advantage of the ladder-simulation method for g_m-C filters.

16.3.3.2 Signal-Flow Graph Methods

As we have seen earlier, the signal-flow graph (SFG) or operational-simulation method takes the circuit equations describing the ladder (Kirchhoff's laws and the I–V relationships for the elements) and realizes them directly via summers (for Kirchhoff's laws) and integrators (for inductors and grounded capacitors). The procedure was detailed in connection with Equations 16.22 through 16.24. Recall that Equation 16.23 represents *integrations* of voltages into currents and currents into voltages. As was the case for the op-amp–based active RC filters in Chapter 15 and the MOSFET-C design, all signals in the SFG g_m-C circuit are voltages. They are summed at the OTA inputs, then are multiplied by g_m to produce an output current that is integrated by a capacitor to produce a voltage as input for the OTA of the next stage. To reflect these facts in the relevant equations, we scale Equation 16.23 analogously to Equation 16.24 to obtain

$$\frac{I_{L2}}{g_m} = \frac{g_m(V_1 - V_3)}{s(L_{2n}g_m)g_m}; \quad \frac{I_{L4}}{g_m} = \frac{g_m(V_3 - V_5)}{s(L_{4n}g_m)g_m} \tag{16.55a}$$

$$V_1 = \frac{g_m V_\mathrm{i} - (I_{l2}/g_m)g_m + sC_{2n}V_3}{s(C_{1n} + C_{2n}) + g_m} \tag{16.55b}$$

$$V_3 = \frac{(I_{L2}/g_m)g_m - (I_{L4}/g_m)g_m + sC_{2n}V_1 + sC_{4n}V_5}{s(C_{2n} + C_{3n} + C_{4n})} \tag{16.55c}$$

$$V_5 = V_\mathrm{o} = \frac{(I_{L4}/g_m)g_m + sC_{4n}V_3}{s(C_{4n} + C_{5n}) + g_m} \tag{16.55d}$$

The scaling factor in this case is the design transconductance g_m. Note that $L_\mathrm{i}\, g_m^2$ in Equation 16.55a has units of capacitance and that source and load resistors in Equations 16.55b and d have the value $1/g_m$. Implementing these equations with lossless or lossy, as appropriate, integrators in fully differential form results in the circuit in Figure 16.29, where we used a signal notation similar to that in Figure 16.14 and chose all integrating capacitors grounded. Starting from a normalized LC prototype, the actual component values are obtained again via Equation 16.52. Note that the OTA at the input performs the voltage-to-current conversion (V_i to I_1) and that the last OTA both here and in Figure 16.27 implements the load resistor. The second OTA in Figure 16.27, realizing the source resistor, is saved in Figure 16.29 by sending the current I_1 directly into the integrating node (the capacitor C_1), as suggested by Equation 16.55b.* Also observe that circuit complexity in Figure 16.29 was kept low by resorting to OTAs with dual inputs. We note again that all transconductors in Figures 16.27through 16.29 are identical† so that a single optimized g_m-cell (an *analog gate*) can be used throughout the filter chip for an especially simple IC design process, analogous to that of a gate array. The inherent 6 dB loss of the LC ladder can also be eliminated, if desired, by lifting the two grounded inputs of the second OTA in Figure 16.29 off ground and connecting them to the input voltage $\pm V_\mathrm{i}/2$ as indicated by the dashed connections.

---

* Had we used this "resistor," the current into the integration node would have been realized as $g_m V_\mathrm{i} = g_m V_\mathrm{i} \times (1/g_m) \times g_m$, clearly a redundant method.

† This can generally be achieved in g_m-C ladder simulations; the only exception occurs for those LC ladders that require unequal terminating resistors, such as even-order Chebyshev filters; in that case, one of the transconductors will also be different.

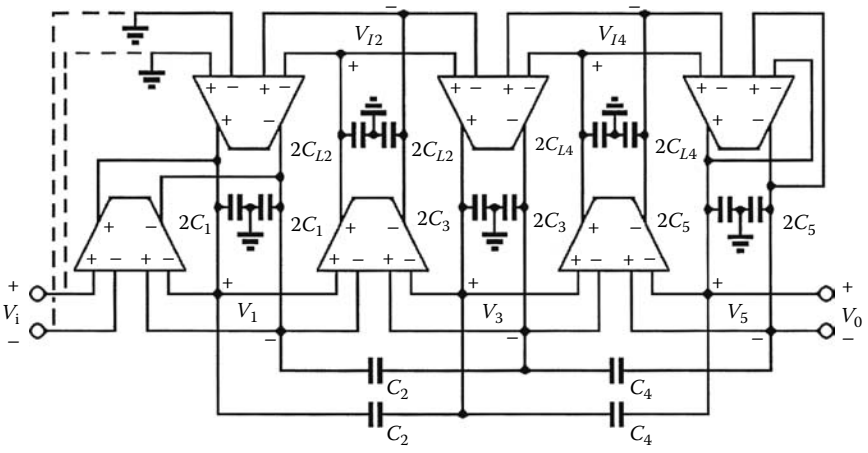


FIGURE 16.29 Signal-flow graph g_m-C realization of a fifth-order elliptic low-pass function. The circuit is an active simulation of the LC-ladder in Figure 16.11, including source and load resistors. All g_m-cells are identical.

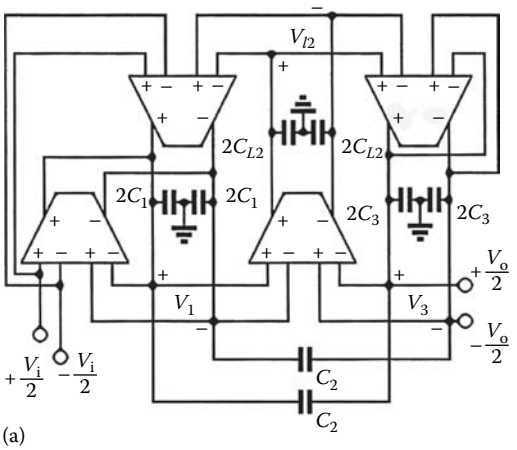
As a simple design example, consider again the realization of the third-order elliptic low-pass filter described by Equation 16.53. The nominal LC elements were given earlier; the circuit is described by Equation 16.55 with $C_{4n} = C_{5n} = L_{4n} = 0$, i.e.,

$$\frac{I_{L2}}{g_m} = \frac{g_m(V_1 - V_3)}{s(L_{2n}g_m)g_m} \tag{16.56a}$$

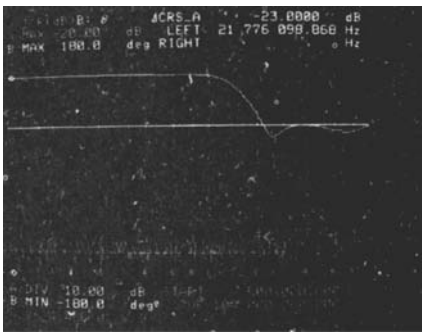
$$V_1 = \frac{2g_mV_i - (I_{L2}/g_m)g_m + sC_{2n}V_3}{s(C_{1n} + C_{2n}) + g_m} \tag{16.56b}$$

$$V_3 = V_o = \frac{(I_{L2}/g_m)g_m + sC_{2n}V_1}{s(C_{2n} + C_{3n}) + g_m} \tag{16.56c}$$

which leads to the circuit in Figure 16.30a, as the reader may readily verify. Note the connection of the input, which realizes the term $2g_mV_i$ in Equation 16.56b and eliminates the 6 dB loss as mentioned



(a)



(b)

FIGURE 16.30 (a) SFG g_m-C third-order elliptic low-pass filter; (b) experimental performance in 2-μm CMOS technology.

earlier. We observe again that the capacitors must be predistorted to account for parasitics as was discussed in connection with Equation 16.54; specifically, labeling as before the *differential* OTA input and output capacitors as C_i and C_o, respectively, we find

$$C_1 \approx C_{1,\text{nominal}} - 2C_o - 2C_i; \quad C_{L2} \approx C_{L2,\text{nominal}} - C_o - 2C_i; \quad C_3 \approx C_{3,\text{nominal}} - C_o - 2C_i$$

Figure 16.30b shows the experimental performance of the circuit fabricated in 2-μm CMOS technology with design-automation software that uses the OTAs from a design library as "analog gates," and *automatically* lays out the chip and predistorts the capacitors according to the process file. The transconductance value used is $g_m \approx 180$ μS, and the capacitors are approximately the same as the ones for the example in Figure 16.28. We will see in Section 16.5 that apart from differences due to layout parasitics, it is no coincidence that the element values are essentially the same as before. Notice that the filter meets all design specifications. The lower trace is the thermal noise at ≈70 dB below the signal; the measured total harmonic distortion was THD < 1% for a 2-$V_{p\text{-}p}$ input signal at 3 MHz.

16.4 Tuning

To obtain accurate filter performance with frequency-parameters set by *RC* products or C/g_m ratios, accurate *absolute* values of components are required. These must be realized by the IC process *and maintained during operation*. Although IC processing is very reliable in realizing accurate *ratios* of like components on a chip, the processing tolerances of *absolute values* must be expected to be of the order of 20%–50% or more. Component tolerances of this magnitude are generally far too large for an untuned filter to perform within specifications. Consequently, filters *must be tuned* to their desired performance by adjusting element values. Clearly, in fully integrated filters, where all components are on a silicon chip, tuning must be performed electronically by some suitably designed automatic control circuitry that is part of the total continuous-time filter system. Tuning implies *measuring* filter performance, *comparing* it with a known standard, *calculating* the errors, and *applying a correction* to the system to reduce the errors. An accurate *reference frequency*, e.g., a system clock (V_{REF} in Figure 16.31), is usually used as a convenient standard. From the filter's response to the signal at this known frequency, errors are detected and the appropriate correction signals, typically dc bias voltages, are applied via the control circuitry [1, Chapter 7], [3, Chapter 16.5]. Reference [7, pt. 6] contains many papers showing practical approaches to specific tuning problems.

Figure 16.31 shows a block diagram of this so-called Master–Slave architecture that is followed by most currently proposed designs. The Master–Slave system is used because the reference signal, V_{REF}, cannot be applied to the Main Filter simultaneously with the main signal, V_{in}, because of undesirable interactions (intermodulation) between V_{in} and V_{REF}. The approach is based on the premise that the filter must operate continuously, i.e., that V_{in} cannot be switched off occasionally to permit tuning in a signal-free

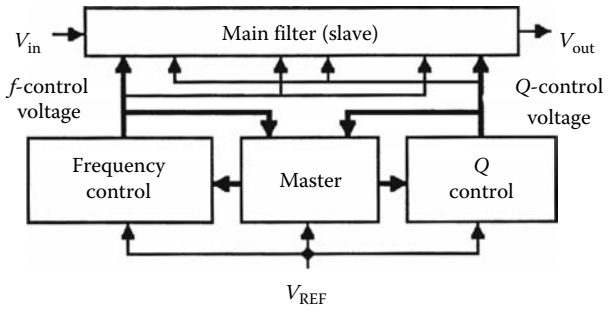


FIGURE 16.31 Block diagram of Master–Slave control system for integrated *c-t* filters.

environment. Therefore, V_{REF} is applied to a circuit, the "Master" filter, that is carefully designed to model the relevant behavior of the "Slave." Master–Slave design and layout require great care to avoid serious matching, noise, and crosstalk problems.

Figure 16.31 should help the reader understand the principle of the operation without having to consider the actual circuitry that is very implementation-specific, see Ref. [7, pt. 6] for design examples. The Main Filter (the "Slave") performs the required signal processing. Since the Master is designed to model the Slave, the behaviors of Master and Slave are assumed to match and track. Tuning is then accomplished by applying the generated correction or control signal simultaneously to both Master and Slave. The system contains a *Frequency-Control* block that detects frequency-parameter errors in the Master's response to the reference signal and generates a frequency-control voltage that is applied to the Master in a closed-loop control scheme such that any detected errors are minimized.* Since the Master is designed to be an accurate model of the Slave, their errors, too, can be assumed to match and track. Consequently, when the frequency-control voltage is applied at the appropriate locations to the Slave, it can be expected to correct any frequency errors in the main filter.

The purpose of the additional block in Figure 16.31, labeled *Q-Control*, is to tune the *shape* of the transfer characteristic. Once the frequency parameters are correct as tuned by the *f*-control loop, the transfer function shape is determined only by the quality factors Q_i of the poles and zeros. *Q*, as a ratio of two frequencies, is a *dimensionless* number and as such is determined by a *ratio* of like components (resistor, capacitor, and/or g_m-ratio; see e.g., Equation 16.47b). *At fairly low frequencies and moderate values of Q,* the quality factor is realizable quite accurately in an IC design. For high-frequency, high-*Q* designs, however, *Q* is found to be a very sensitive function of parasitic elements and phase shifts so that it is unreasonable to expect *Q* to turn out correctly without tuning. Therefore, including for generality a *Q*-Control block† permits automatic tuning of the transfer function shape by a scheme that is completely analogous to the *f*-control method, as illustrated in Figure 16.31.

With few exceptions, all currently proposed automatic tuning schemes follow the Master–Slave approach. A different concept proposed in Ref. [11] uses adaptive techniques to tune all poles *and* zeros of a filter function for improved tuning accuracy. Although the Master–Slave and the adaptive techniques work well, generally, the necessary circuitry has been found to be too large, noisy, and power hungry for many practical applications. Thus, alternative choices use a simple post-fabrication trim step‡ to eliminate the large fabrication tolerances, possibly together with careful design to make the electronic devices (OTAs) independent of temperature variations [12,13].

We also point out that at the highest frequencies, higher than, say, 100 MHz, circuit capacitors to realize the prescribed frequency parameters, g_m/C, become very small ($C \approx 0.15$ pF at 100 MHz and $g_m = 100$ μS). Filter behavior is then essentially determined by parasitics, and matching required between Master and Slave circuitry is unreliable. In that case, a form of direct tuning [7, paper 6.4] must be used, where the main filter is tuned during time periods when the signal is absent. The system is essentially as depicted in Figure 16.31, except that the Master is absent and V_{REF} is applied to the main filter. When tuning is complete (taking a few milliseconds), V_{REF} is turned off, V_{in} is again applied to the filter, and the generated dc control signals are held on a capacitor until the next tuning update.

---

* In the designs reported to date, frequency-control blocks built around some type of phase-locked loop scheme, using a multiplier or an EXOR gate as phase detector, have been found most successful, but magnitude-locking approaches similar to those employed for *Q* control have also been used. A largely digital approach to frequency and *Q* tuning is discussed in Ref. [10].

† Because magnitude errors can be shown in many cases to be proportional to errors in quality factor, the *Q*Control block is normally implemented around an amplitude-locking scheme, where the filter's magnitude response at a given frequency is locked to a known reference level.

‡ For example, an on-chip laser-trimmed resistor or an external resistor set to determine the bias for OTAs. It works for applications with very low quality factors or where no component drifts are expected during operation.

16.5 Discussion and Conclusions

In this chapter we discussed one of the fastest growing research and development areas in the topic of continuous-time filters, the field of fully integrated filters. Growing pressures to reduce costs and size, and improve reliability lead to increasingly larger parts of electronic systems being placed onto integrated circuits, a trend that *c-t* filters need to follow. As might have been expected, in most respects the methods for IC *c-t* filter design are identical to well-known active *RC* techniques, and follow directly the well-understood and proven standard active *RC* biquad-cascade or ladder-simulation methodologies. The difference lies in the final implementation, which the designer may adapt to any IC process and power supply level appropriate for the implementation of the system. Signal-to-noise ratios of the order of 65–80 dB and better, and distortion levels of less than 0.5% at signal levels of 1 V are obtainable. The two most prominent design approaches are the MOSFET-*C* and the g_m-*C* methods, both of which lead to easy and systematic designs and result in filters that have proven themselves in practice. At the time of this writing, g_m-*C* filters appear to have the edge in high-frequency performance: with OTA bandwidths reaching gigahertz frequencies even for CMOS technology, *c-t* filters can be designed readily for applications in the range of several hundred megahertz, i.e., to about 40%–60% of the active-device bandwidth. A second important difference that requires the designer's attention is that the IC filter must be automatically tunable. This implies electronically variable components, OTAs or MOSFET ''resistors,'' in the filter and some automatic control scheme for detecting errors and providing adjustments. Only area- and power-efficient designs, and simple low-noise circuitry for such tuning schemes will be acceptable and guarantee the ultimate commercial success of integrated filters. At the present time, many approaches have been proposed that the reader may modify or adapt to his/her system [7].

We point out again that the cascade design is the more general procedure because it permits the realization of arbitrary transmission zeros anywhere in the complex *s*-plane, as required, for example, in gain or phase equalizers. Lossless ladders, on the other hand, are more restrictive because their transmission zeros are constrained to lie on the $j\omega$ axis, but they have lower passband sensitivities to component tolerances than cascade filters. Since good gyrators are readily available for g_m-*C* filters—in contrast to MOSFET-*C* designs—two competing implementation methods appear to suggest themselves in the g_mC approach: the signal-flow graph and the element-substitution methods. As mentioned, there is in fact no discernible difference between the two methods; indeed, they lead to the same structures. For instance, the reader may readily verify that apart from a minor wiring change at the inputs, the circuits in Figures 16.27b and 16.29 are identical, as are the ones in Figures 16.28 (after redrawing it for dual-input OTAs) and 16.30. The wiring change is illustrated in Figure 16.32. Figure 16.32a shows

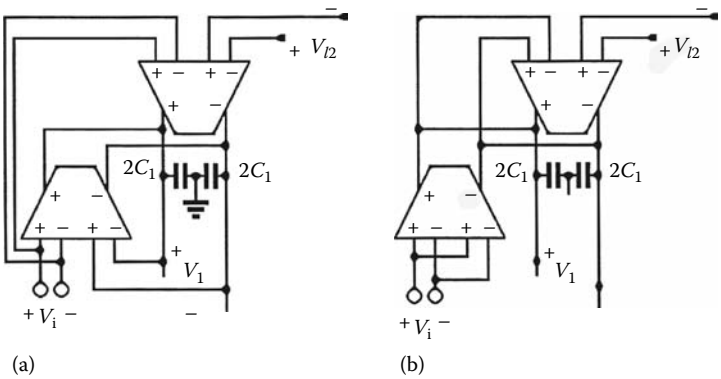


FIGURE 16.32 Excerpt from Figure 16.28 (a) to illustrate alternative wiring and (b) at the filter input 0.

the input section of Figure 16.29 (wired for 0-dB dc gain) and the wiring in Figure 16.32b is that of Figure 16.27b. Notice that both circuits realize

$$(sC_1 + g_m)V_1 = 2g_m V_i - g_m V_{I2} \tag{16.57}$$

The designer, therefore, may choose a ladder method based on his/her familiarity with the procedure, available tables, or prototype designs; the final circuits are the same.

A further item merits reemphasizing at this point: As the reader may verify from the examples presented, g_m-C ladder structures generally have at all circuit nodes a design capacitor that can be used to absorb parasitics by predistortion as was discussed earlier. It may be preferable, therefore, to avoid the g_m-C–op-amp integrator of Figure 16.20 with its increased noise level, power consumption, and with its associated parasitic poles (and zeros) and use parasitics-absorption as we have done in our examples. This method will *not* introduce any new parasitic critical frequencies into the filter and result in lower distortion of the transfer function shape and easier tuning. As we mentioned earlier, this feature is a substantial advantage of g_m-C ladder filters.

References

1. R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters: Passive, Active RC and Switched Capacitor*, Englewood Cliffs, NJ: Prentice-Hall, 1990.
2. A. I. Zverev, *Handbook of Filter Synthesis,* New York: Wiley, 1967.
3. R. Schaumann and M. Van Valkenburg, *Design of Analog Filters*, New York: Oxford University Press, 2001.
4. D. Li and Y. Tsividis, Active *LC* filters on silicon, *IEE Proc. Circuits Devices Syst*., 147(1), 49–56, Feb. 2000.
5. E. Sanchez-Sinencio and J. Silva-Martinez, CMOS transconductance amplifiers, architectures and active filters: a tutorial, *IEEE Proc. Circuits, Devices Syst*., 147(1), 3–12, Feb. 2000.
6. Y. Tsividis, M. Banu, and J. Khoury, Continuous-time MOSFET-C filters in VLSI, *IEEE Trans. Circuits Syst*., CAS-33, Special Issue on VLSI Analog and Digital Signal Processing, 125–140, 1986; see also *IEEE J. Solid-State Circuits*, SC-21, 15–30, 1986.
7. Y. Tsividis and J. A. Voorman, Eds., *Integrated Continuous-Time Filters: Principles, Design and Implementations*, New York: IEEE, 1993.
8. C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds., *Analogue IC Design: The Current-Mode Approach,* London: IEE—Peter Peregrinus, 1990.
9. A. Thanachayanont and A. Payne, CMOS floating active inductor and its applications to bandpass filter and oscillator designs, *IEE Proc. Circuits Devices Syst*., 147(1), 41–48, Feb. 2000.
10. A. I. Karsilayan and R. Schaumann, Mixed-mode automatic tuning scheme for high-Q continuous-time filters, *IEE Proc. Circuits Devices Syst*., 147(1), 57–64, Feb. 2000.
11. K. A. Kozma, D. A. Johns, and A. S. Sedra, Automatic tuning of continuous-time filters using an adaptive tuning technique, *IEEE Trans. Circuits Syst*., CAS-38, 1241–1248, 1991.
12. G. A. De Veirman and R. G. Yamasaki, Design of a bipolar 10-MHz continuous-time 0.05° equiripple linear phase filter, *IEEE J. Solid State Circuits*, SC-27, 324–331, March 1992.
13. C. A. Laber and P. R. Gray, A 20-MHz BiCMOS parasitic-insensitive continuous-time filter and second-order equalizer optimized for disk-drive read channels, *IEEE J. Solid State Circuits*, SC-28, 462–470, April 1993.

17

Switched-Capacitor Filters

17.1 Introduction **17**-1
17.2 Basic Building Blocks **17**-2
Voltage Gain Amplifiers • First-Order Blocks • Switched-Capacitor Biquadratic Sections
17.3 Effects of the Op-Amp Finite Parameters **17**-10
Finite Op-Amp DC Gain Effects • Finite Op-Amp Gain-Bandwidth Product Effects
17.4 Noise and Clock Feedthrough **17**-11
Noise due to the Op-Amp • Thermal Noise • 1/*f* Noise • Op-Amp Noise Contributions • Noise in a Switched-Capacitor Integrator • Noise due to the Switches • Clock Feedthrough • Channel Mobile Charge • Dynamic Range
17.5 Design Considerations for Low-Voltage Switched-Capacitor Circuits **17**-18
Low-Voltage Operational Amplifiers • Analog Switches
17.6 Fully Differential Filters **17**-21
17.7 Design Examples **17**-23
Biquadratic Filter • Sixth-Order Bandpass Ladder Filter • Programmable Switched-Capacitor Filter • Switched-Capacitor Filters with Reduced Area • Spectrum Analyzer Using Switched-Capacitor Techniques • Programmable Switched-Capacitor Oscillator
References **17**-35

Jose Silva-Martinez
Texas A&M University

Edgar Sánchez-Sinencio
Texas A&M University

17.1 Introduction

The need to have monolithic high-performance analog filters motivated circuit designers in the late 1970s to investigate alternatives to conventional active-RC filters. A practical alternative appeared in the form of switched-capacitor (SC) filters [1–3]. The original idea was to replace a resistor by an SC simulating the resistor. Thus, this equivalent resistor could be implemented with a capacitor and two switches operating with two-clock phases. SC filters consist of switches, capacitors, and op-amps. They are characterized by difference equations in contrast to differential equations for continuous-time filters. Simultaneously, the mathematical operator to handle sample-data systems such as SC circuits is the z-transform, while the Laplace transform is used for continuous-time circuits. Several key properties of SC circuits have made them very popular in industrial environments:

1. Time constants (RC products) from active-RC filters become capacitor ratios multiplied by the clock period T. That is

$$\mathrm{RC} \Rightarrow \frac{C}{C_\mathrm{R}} T = \frac{C}{C_\mathrm{R} f_\mathrm{c}}$$

   where f_c is the clock frequency used to drive the SC equivalent resistor.
2. Reduced silicon area, since the equivalent of large resistors can be simulated using small-size capacitors. Furthermore, positive and/or negative equivalent resistors can be easily implemented with SC techniques.
3. Above expression can be realized in real applications with a good accuracy of nearly 0.1%.
4. Typically, the load of an SC circuit is mainly capacitive; therefore the required op-amps do not require a low-impedance output stage. This allows the use of unbuffered cascode operational transconductance amplifiers, which is especially useful in high-speed applications.
5. SC filters can be implemented in digital circuit process technologies; metal–metal capacitors are quite often used in deep submicron technologies. Thus, useful mixed-mode signal circuits can be economically realized.
6. SC design technique has matured. In the audio range, SC design techniques have become the dominant design approach. Many circuits in communication applications and data converters use SC implementations.

In what follows, we will discuss basic building blocks involved in low- and high-order filters. Limitations and practical design considerations will be presented. Furthermore, due to the industrial push for lower power supply voltages, a brief discussion on low-voltage circuit design is included.

17.2 Basic Building Blocks

The basic building blocks involved in SC circuits are voltage gain amplifiers, integrators, second-order filters, and nonoverlapping clock generators. A key building block is the integrator. By means of a two-integrator loop a second-order (biquadratic) filter can be realized. Furthermore, a cascade connection of biquadratic filters yields higher order filters.

17.2.1 Voltage Gain Amplifiers

The gain amplifier is a basic building block in SC circuits. Both the peak gain of a second-order filter and the link between the resonators in a ladder filter are controlled by voltage amplifier stages rather than by integrators. Many other applications such as pipeline data converters require voltage gain stages as well. A voltage amplifier can be implemented by using two capacitors and an operational amplifier, as shown in Figure 17.1a. Ideally, the gain of this amplifier is given by $-C_\mathrm{S}/C_\mathrm{i}$. This topology is compact, versatile, and time continuous. Although this gain amplifier is quite simple, several second-order effects present in the op-amp make its design more complex. A major drawback of this topology is the lack of dc feedback.

For dc, the capacitors behave as open circuits; hence the operating point of the operational amplifier is not stabilized by the integrating capacitor C_i. In addition, the leakage current present at the input of the op-amp is integrated by the integrating capacitor, whose voltage eventually saturates the circuit. The leakage current I_leak in SCs circuits is a result of the diodes associated with the bottom plate of the capacitors and the switches (drain and source junctions). The leakage current is typically about 1 nA/cm^2. Analysis of Figure 17.1a considering the leakage current I_leak present at the inverting terminal of the amplifier yields

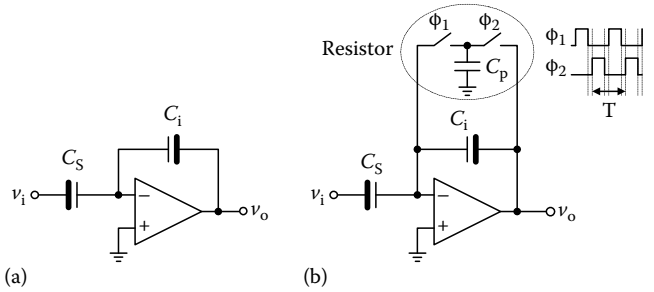


FIGURE 17.1 Voltage amplifiers. (a) Without dc feedback and (b) with dc feedback.

$$v_o(t) = v_o(t_o) - \frac{C_S}{C_i} V_i(t) - \frac{I_{leak}}{C_i}(t - t_o) \tag{17.1}$$

Note that the dc output voltage is defined by the input signal and the initial conditions at $t - t_o$. The leakage current present at the input of the op-amp is integrated by C_i and eventually saturates the circuit. To overcome this drawback, an SC resistor can be added as shown in Figure 17.1b. ϕ_1 and ϕ_2 are two nonoverlapping clock phases with nearly 50% duty cycle. The SC resistor gives a dc path for the leakage current but reduces further the low-frequency gain. A detailed analysis of this topology shows that the dc output voltage is equal to $-I_{leak}T/C_p$, with C_p the parasitic capacitor associated with the feedback path. Employing charge conservation analysis method for SC networks, it can also be shown that the z-domain transfer function of this topology is

$$H(z) = \frac{V_o(z)}{V_i(z)} = -\frac{C_S}{C_i} \frac{1 - z^{-1}}{1 - \left(1 - \frac{C_p}{C_i}\right) z^{-1}} \tag{17.2}$$

with $z = e^{j2\pi fT}$. For low frequencies, $z \approx 1$, the magnitude of the transfer function is very small, and only for high frequencies, $z \approx -1$, the circuit behaves as an inverting voltage amplifier.

An offset free voltage amplifier is shown in Figure 17.2. During the clock phase ϕ_2, the output voltage is equal to the op-amp offset voltage and it is sampled by both the integrating and sampling capacitors. Because of the sampling of the offset voltage during the previous clock phase, during the integrating clock phase the charge injected by the sampling capacitor is equal to $C_S V_i$, and the charge extracted from the integrating capacitor becomes equal to $C_i V_o$ if the offset voltage of the amplifier is the same in both clock phases. In this case, the offset voltage does not affect the charge recombination, and the z-domain transfer function during the clock phase ϕ_1 becomes

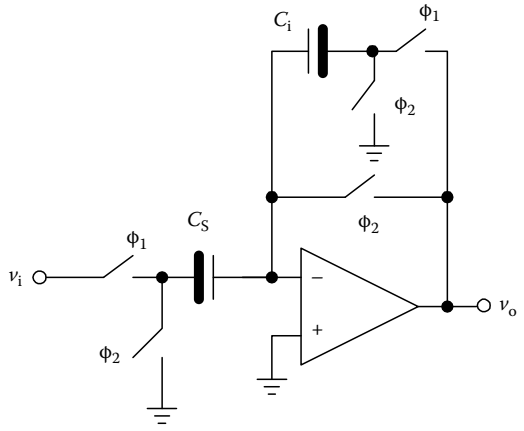


FIGURE 17.2 Voltage amplifier available during the ϕ_1 clock phase.

$$H(z) = -\frac{C_S}{C_i} \tag{17.3}$$

Equation 17.3 shows that ideally this topology is insensitive to the op-amp offset voltage, and does

not have any low-frequency limitation. The topology behaves as an inverting amplifier if the clock phases shown in Figure 17.2 are used. A noninverting amplifier is obtained if the clock phases associated with the sampling capacitor are interchanged.

Because during ϕ_2 the op-amp output is short circuited with the inverting input, this topology presents two drawbacks. First, the amplifier output is only available during the clock phase ϕ_1. This limitation could be important in complex applications wherein the output of the amplifier is required during both clock phases. The second disadvantage of this topology is the large excursion of the op-amp output voltage. During the first clock phase, the op-amp output voltage is equal to the offset voltage and in the next clock phase this voltage is equal to $-(C_S/C_i)V_i$. Hence an op-amp with large slew rate may be required; this may demand significant amount of power since the load capacitor is driving during this phase too.

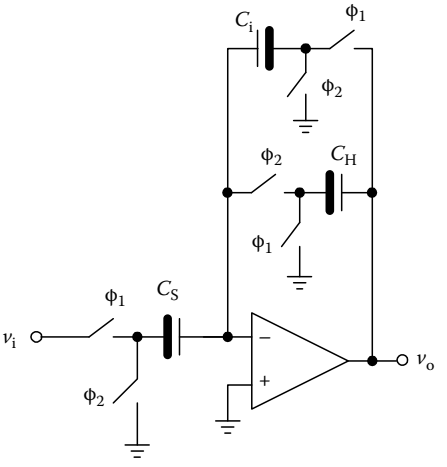


FIGURE 17.3 Voltage amplifier available during both clock phases.

Another interesting topology is shown in Figure 17.3. During ϕ_2, the op-amp output voltage is equal to the previous voltage plus the op-amp offset voltage plus V_o/A_V, where A_V is the open-loop dc gain of the op-amp. In this clock phase, both capacitors C_i and C_S are charged to the voltage at the inverting terminal of the op-amp. This voltage is approximately equal to the op-amp offset voltage plus V_o/A_V. During the next clock phase, the sampling capacitor is charged to $C_S(V_i - V_-)$, but because it was precharged to $-C_S V_-$, the injected charge to C_i is equal to $C_S V_i$. As a result of this charge cancellation, the op-amp output voltage is equal to $-(C_S/C_i)V_i$. Therefore, this topology has low sensitivity to the op-amp offset voltage and to the op-amp dc gain. A minor drawback of this topology is that the op-amp stays in open loop during the nonoverlapping clock phase transition times. This fact produces spikes during these time intervals. A solution for this is to connect a small capacitor between the op-amp output and the left-hand plate of C_S.

17.2.2 First-Order Blocks

The standard stray-insensitive integrators are shown in Figure 17.4. Note that in sampled data systems both input and output signals can be sampled at different clock periods. This yields different transfer functions; this property increases the flexibility of the architecture when used in complex systems. We will assume a two-phase nonoverlapping clock: an odd clock phase ϕ_1 and an even clock phase ϕ_2.

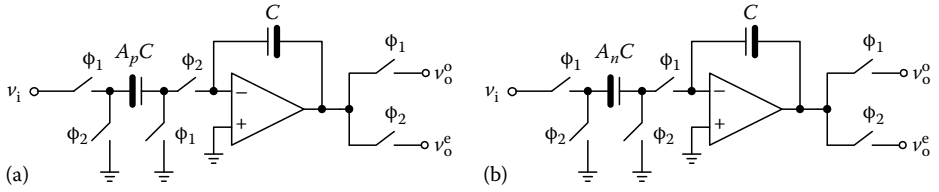


FIGURE 17.4 Conventional stray-insensitive SC integrators. (a) Noninverting and (b) inverting.

For the noninverting integrator the following transfer functions are often used

$$H^{oo}(z) = \frac{V_o^o(z)}{V_{in}^o(z)} = \frac{A_p z^{-1}}{1 - z^{-1}} = \frac{A_p}{z - 1} \tag{17.4a}$$

$$H^{oe}(z) = \frac{V_o^e(z)}{V_{in}^o(z)} = \frac{A_p z^{-1/2}}{1 - z^{-1}} = \frac{A_p z^{1/2}}{z - 1} \tag{17.4b}$$

where H^{xy} stands for the output defined at phase x while the input is sampled during phase y. For the inverting integrator

$$H^{oe}(z) = \frac{V_o^o(z)}{V_{in}^o(z)} = -\frac{A_n}{1 - z^{-1}} = -\frac{A_n z}{z - 1} \tag{17.5a}$$

$$H^{oe}(z) = \frac{V_o^e(z)}{V_{in}^o(z)} = -\frac{A_n z^{-1/2}}{1 - z^{-1}} = -\frac{A_n z^{1/2}}{z - 1} \tag{17.5b}$$

where z^{-1} represents a unit delay. A crude demonstration showing the integration nature of these SC integrators [1,8] in the s-domain is to consider high-sampling rate, that is, a clock frequency ($f_c = 1/T$) much higher than the operating signal frequencies. Let us consider Equation 17.4a, and assuming high-sampling rate ($\omega T \ll 1$) we can write a mapping from the z- to the s-domain:

$$z = e^{j2\pi fT} = e^{sT} \cong 1 + sT \tag{17.6}$$

$$H(s) = \frac{A_p}{z - 1}\bigg|_{z \cong 1 + sT} \cong \frac{1}{(T/A_p)s} \tag{17.7}$$

where s ($=j2\pi f$) is the complex frequency variable. This last expression corresponds to a continuous-time noninverting integrator with a time constant of $T/A_p = 1/f_c A_p$, that is, a capacitance ratio times the clock period.

In many applications the capacitor ratios associated with the integrators are very large, thus the total capacitance becomes excessive. This is particularly critical for biquadratic filters with high Q, where the ratio between the largest and smallest capacitance is proportional to the quality factor Q. A suitable inverting SC integrator for high-Q applications [11,13] is shown in Figure 17.5; the corresponding transfer function during ϕ_2 is

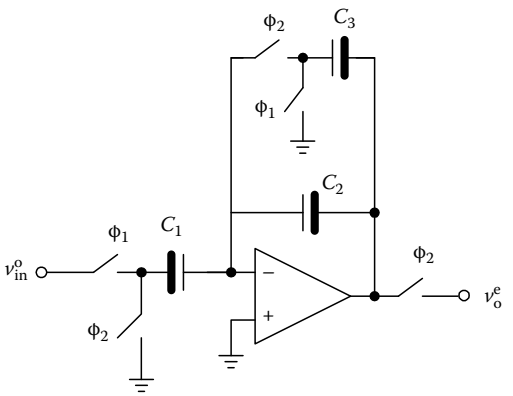


FIGURE 17.5 A SC integrator with reduced capacitance spread.

$$H^{oe}(z) = \frac{V_o^e(z)}{V_{in}^o(z)} = -\frac{C_1 C_3}{C_2(C_2 + C_3)}\left(\frac{z^{-1/2}}{1 - z^{-1}}\right) \tag{17.8}$$

This integrator is comparable in performance to the conventional of Figure 17.3, in terms of stray sensitivity and finite gain error. Note from Equation 17.8 that the transfer function is only defined during ϕ_2. During ϕ_1, the circuit behaves as a voltage amplifier plus the initial conditions of the previous clock phase, and in the following clock phase the same

amount of charge is extracted but the integrating capacitor is reduced due to the switches connected to C_1, thus relatively high-slew-rate op-amps could be required. A serious drawback could be the increased offset in comparison with the standard SC integrators. However, in typical two integrator loop filters, the other integrator can be chosen to be offset and low dc gain compensated as shown in Figure 17.6.

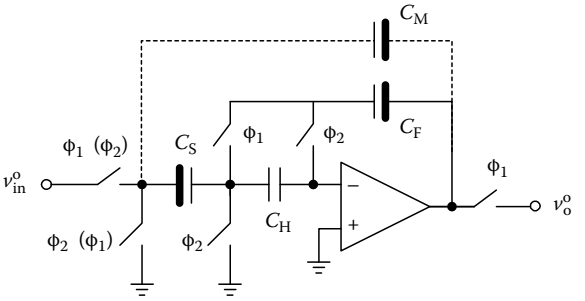


FIGURE 17.6 Offset and gain compensated integrator.

The SC integrator performs the integration during ϕ_1 by means of C_S and C_F and the hold capacitor C_H stores the offset voltage. The voltage across C_H compensates the offset voltage and the dc gain error of the op-amp. Note that the SC integrator of Figure 17.6 can operate as a noninverting integrator if the clocking in parenthesis is employed. C_M provides a time-continuous feedback around the op-amp. The transfer function for infinite op-amp gain is

$$H^{oo}(z) = \frac{V_o^o(z)}{V_{in}^o(z)} = -\frac{C_S}{C_F}\left(\frac{1}{1-z^{-1}}\right) \tag{17.9}$$

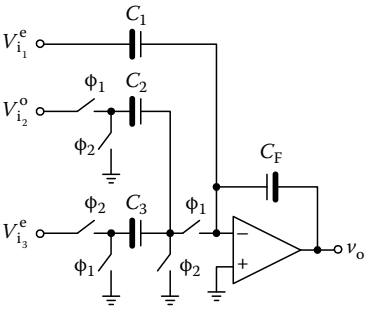


FIGURE 17.7 General form of a first-order building block.

Next we discuss a general form of a first-order building block (see Figure 17.7). Observe that some switches can be shared. The output voltage during ϕ_1 can be expressed as

$$V_o^o = -\frac{C_1}{C_F}V_{i_1}^e - \frac{C_2}{C_F}\left(\frac{1}{1-z^{-1}}\right)V_{i_2}^o + \frac{C_3}{C_F}\left(\frac{z^{-1/2}}{1-z^{-1}}\right)V_{i_3}^e \tag{17.10}$$

Observe that the capacitor C_3 and switches can be considered as the implementation of a negative resistor leading to a noninverting amplifier. Also note that $V_{i_2}^o$ could be V_o^e, this connection would make the integrator a lossy one. In that case Equation 17.10 can be written as

$$V_o^o\frac{\left(1+\frac{C_2}{C_F}\right)z-1}{z-1} = -\frac{C_1}{C_F}V_{i_1}^e + \frac{C_3}{C_F}\frac{z^{1/2}}{z-1}V_{i_3}^e, \quad \text{for } V_{i_2}^o = V_o^o \tag{17.11}$$

The building block of Figure 17.7 is the basis of higher order filters.

17.2.3 Switched-Capacitor Biquadratic Sections

The circuit shown in Figure 17.8 can implement any pair of poles and zeros in the z-domain. For $C_A = C_B = 1$ we can write

$$H^{ee}(z) = \frac{V_o^e(z)}{V_{in}^e(z)} = -\frac{(C_5+C_6)z^2 + (C_1C_2 - C_5 - 2C_6)z + C_6}{z^2 + (C_2C_3 + C_2C_4 - 2)z + (1 - C_2C_4)} \tag{17.12}$$

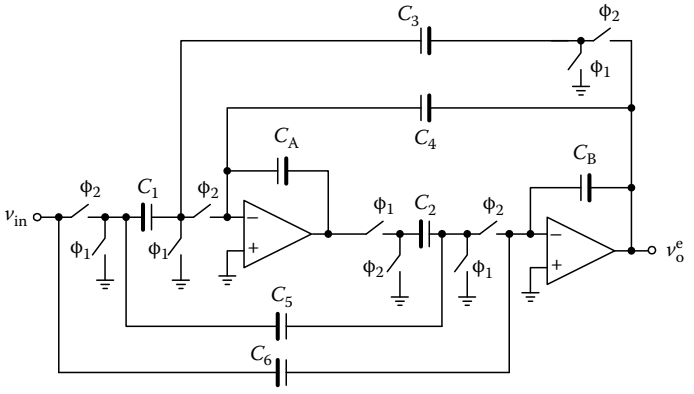


FIGURE 17.8 A SC biquadratic section [Martin-Sedra].

Simple design equations for a particular type of filter follows:

$$\text{Low-pass case: } C_5 = C_6 = 0 \tag{17.13a}$$

$$\text{High-pass case: } C_1 = C_5 = 0 \tag{17.13b}$$

$$\text{Bandpass case: } C_1 = C_6 = 0 \tag{17.13c}$$

Comparing the coefficients of the denominator of Equation 17.12 with the general z-domain expression $z^2 - (2r\cos\theta)z + r^2$ of a second-order system, we can obtain the following expressions:

$$C_2C_4 = 1 - r^2 \tag{17.14a}$$

$$C_2C_3 = 1 - 2r\cos\theta + r^2 \tag{17.14b}$$

Note that in this expression it has been assumed that the filter poles are complex conjugate located at $z = re^{\pm j\omega_{od}T}$. ω_{od} and Q are the poles frequency and the filter's quality factor, respectively. For equal voltage levels at the two integrator outputs, and assuming Q greater than 3, and high-sampling rate ($\theta = \omega_{od}T \ll 1$) we can write

$$C_2 = C_3 = \sqrt{1 + r^2 - 2r\cos\theta} \cong \sqrt{(1-r)^2 + r\theta^2} \cong \theta = \omega_{od}T \tag{17.14c}$$

$$C_4 = \frac{1 - r^2}{C_2} \cong \frac{1}{Q} \tag{17.14d}$$

where $\cos(\theta) \cong 1 - \theta^2/2$ and $r \cong 1$. The capacitance spread for high-sampling rate and high-Q unity DC gain low-pass filter can be expressed as

$$\frac{C_{max}}{C_{min}} = \max\left\{\frac{1}{C_2}, \frac{1}{C_4}\right\} = \max\left\{\frac{1}{\omega_{od}T}, Q\right\} \tag{17.15}$$

In some particular cases this capacitance spread becomes prohibited large. For such cases the SC integrators shown in Figures 17.5 and 17.6 can be used to replace the conventional building blocks.

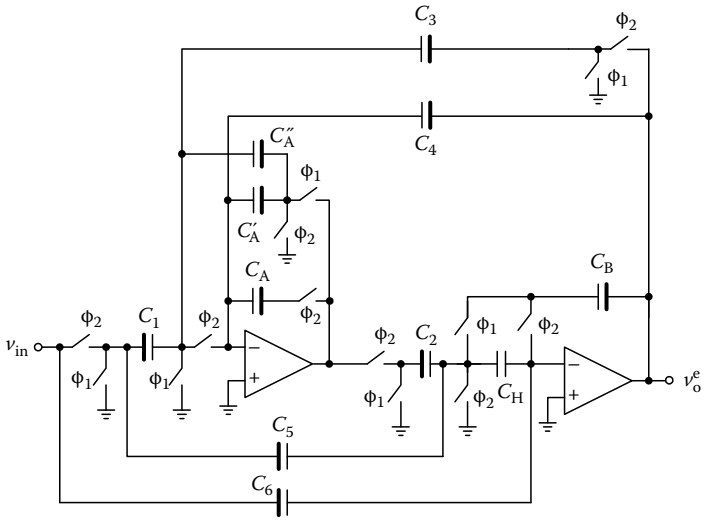


FIGURE 17.9 An improved capacitance area SC biquadratic section.

This combination yields the SC biquadratic section shown in Figure 17.9. This structure, besides offering a reduction of the total capacitance, also yields a reduction of the offset voltage of the op-amps. Note that the capacitors C_H does not play an important role in the design, and can be chosen to have a small value. The design equations for the zeros are similar to Equation 17.13a through c. For the poles, comparing with $z^2 - (2r\cos\theta)z + r^2$ and the analysis of Figure 17.9, we can write

$$\frac{C_2 C_3}{C_A + C_B} = 1 + r^2 - 2r\cos\theta \tag{17.16a}$$

$$\frac{C'_A C_2 C_4}{C_{A_1} C_A C_B} = 1 - r^2 \tag{17.16b}$$

where $C_{A1} = C'_A + C''_A$. Simple design equations can be obtained assuming high-sampling rate, large Q, and $C_2 = C_3 = C_4 = C'_A = C_h = 1$; then

$$C_A + C_B \cong \frac{1}{\omega_{od} T} \tag{17.17a}$$

$$C''_A \cong Q\omega_{od} T - 1 \tag{17.17b}$$

Another common situation is the use of SC filters at high frequencies; in such cases a structure with minimum gain-bandwidth product ($GB = \omega_u$) requirements is desirable. This structure is shown in Figure 17.10 and is often referred to as a decoupled structure. It is worthwhile to mention that two SC architectures can have ideally the same transfer function; however, with real op-amps their frequency (and time) response differs significantly. A rule of thumb to reduce the GB effects in SC filters is to minimize the direct connections between the output of one op-amp to the input of another op-amp. It is desirable to transfer the output of an op-amp to a grounded capacitor, and in the next clock phase, transfer the capacitor charge into the op-amp input. If required, the connection to the next op-amp has to be done during the hold phase of the previous op-amp. If the connection is done during the charge

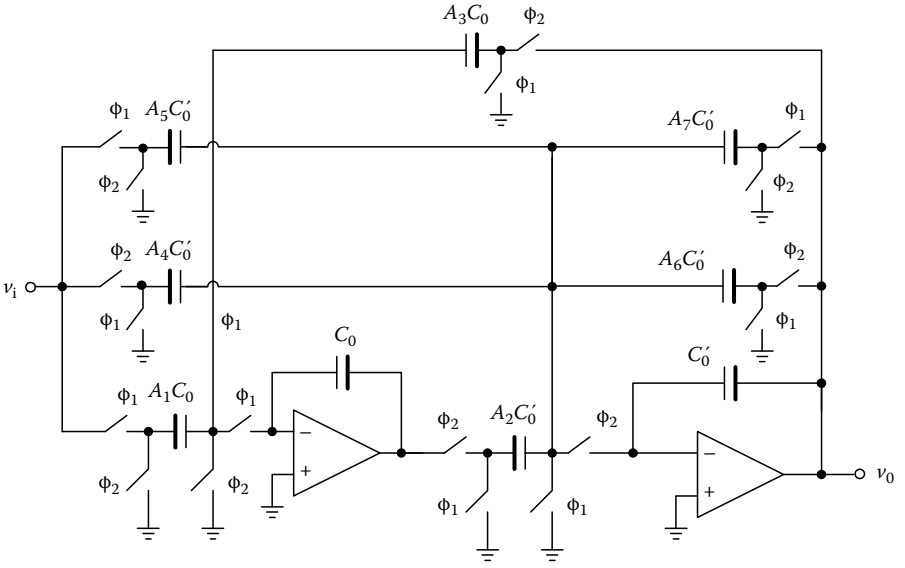


FIGURE 17.10 A decoupled SC biquadratic section.

injection phase, the output of the first integrator is a ramp (step response of the integrator), and the settling time for the next integrator increases further. More discussion on ω_u effects is given in Section 17.3.

If the input signal is sampled during ϕ_2 and held during ϕ_1, the ideal transfer function is given by

$$H^{ee}(z) = \frac{V_o^e(z)}{V_i^e(z)} = -\left(\frac{A_4}{1+A_6}\right)\left(\frac{z^2 - z\left(\frac{A_4+A_5-A_1A_2}{A_4}\right) + \frac{A_5}{A_4}}{z^2 - z\left(\frac{2+A_6+A_7-A_2A_3}{1+A_6}\right) + \frac{1+A_7}{1+A_6}}\right) \tag{17.18}$$

Analysis of this equation yields the following expressions:

$$r^2 = \frac{1+A_7}{1+A_6} \tag{17.19a}$$

$$2r\cos\theta = \frac{2+A_6+A_7-A_2A_3}{1+A_6} \tag{17.19b}$$

The capacitor A_7C_0' can be used as a design trade-off parameter to optimize the biquad performance. A simple set of design equations follows:

$$A_6 = \frac{(1+A_7)-r^2}{r^2} \tag{17.20a}$$

$$A_2 = A_3 = (1+A_7)\sqrt{\frac{1+r^2-2r\cos\theta}{r^2}} \tag{17.20b}$$

Under high-sampling rate and high-Q conditions, the following expressions can be obtained:

$$\omega_o \cong f_c\sqrt{\frac{A_2A_3}{1+A_6}} \tag{17.21a}$$

and

$$Q \cong \sqrt{\frac{A_2 A_3 (1 + A_6)}{A_6 - A_7}} \tag{17.21b}$$

A trade-off between Q-sensitivity and total capacitance is given by A_6 and A_7.

17.3 Effects of the Op-Amp Finite Parameters

17.3.1 Finite Op-Amp DC Gain Effects

The effect of finite op-amp dc voltage gain A_o in a lossless SC integrator is to transform a lossless integrator into a lossy one. This brings degradation in the transfer function both in magnitude and phase. Typically, the magnitude deviation due to the integrator amplitude variation is not critical. By contrast, the phase deviation from the ideal integrator has a very important influence on the overall performance. When the real SC integrators are used to build a two-integrator biquadratic filter, it can be proved that the actual quality factor becomes

$$Q_A = \frac{1}{\frac{1}{Q} + \frac{2}{A_o}} \cong \left(1 - \frac{2Q}{A_o}\right) Q \tag{17.22}$$

where Q and A_0 are the desired filter's quality factor and op-amp dc gain, respectively. The actual frequency of the poles suffers small deviations as well

$$\omega_{o_A} = \frac{A_o}{1 + A_o} \omega_o \tag{17.23}$$

From this equation we can conclude that the ω_o deviations are negligible if $A_o > 100$. However, the Q deviations can be significant depending on the Q and A_o values, e.g., Q_{error} is given by $-2Q/A_o$, which is more critical for high-Q applications.

17.3.2 Finite Op-Amp Gain-Bandwidth Product Effects

The op-amp bandwidth is very critical for high-frequency applications. The analysis is carried out when the op-amp voltage gain is modeled with one dominant pole, i.e.,

$$A_v(s) = \frac{A_o}{1 + s/\omega_p} = \frac{A_o \omega_p}{s + \omega_p} = \frac{\omega_u}{s + \omega_p} \cong \frac{\omega_u}{s} \tag{17.24}$$

where

A_o is the dc gain
ω_u is approximately the unity-gain bandwidth (GB)
ω_p is the op-amp bandwidth

The analysis taking into account $A_v(s)$ is rather cumbersome since the op-amp input–output characterization is a continuous-time system modeled by a first-order differential equation, and the rest of the SC circuit is characterized by discrete-time systems modeled by difference equations. It can be shown that the step response of a single SC circuit with an op-amp gain modeled by Equation 17.24 due to step inputs applied at $t = t_1$ is

$$V_o(t - t_1) = V_o(t_1) e^{-(t-t_1)\alpha\omega_u} + V_{od}\{1 - e^{-(t-t_1)\alpha\omega_u}\} \tag{17.25}$$

where

$V_o(t_1)$ and V_{od} are the initial and the final output voltages, respectively

α is a topology dependent voltage divider, $0 < \alpha \leq 1$, given by the following expression:

$$\alpha = \frac{\sum C_f}{\sum_i C_i} \tag{17.26}$$

where

C_f sum consists of all feedback capacitors connected directly between the op-amp output and the inverting input terminal

C_i sum comprises the capacitors connected to the negative op-amp terminal

Note that $\alpha\omega_u$ should be maximized for fast settling time. For the common case the SC circuit is driven by 50% duty cycle clock phases, the integrating time is close to $T/2$, hence the figure of merit to be maximized is $\alpha T\omega_u/2$. The output voltage will settle to its final value within an error of 0.67% after five equivalent time constants, hence a rule of thumb yielding reduced GB product effects requires

$$\frac{\alpha T\omega_u}{2} > 5 \tag{17.27a}$$

or

$$f_u > \frac{1.6}{\alpha} f_s \tag{17.27b}$$

For the multiple op-amp case the basic concept prevails. For a two-clock phase SC filters the system can be described by an $\mathbf{M} \times \mathbf{M}$ matrix $\mathbf{A}$ for each clock-phase, where $\mathbf{M}$ is the number of op-amps in the filter architecture. For a fixed clock frequency, $f_c = 1/T$, there are two components to be optimized: α and ω_u. effects biquadratic SC filter presented in the previous section (Figure 17.10) has the property that matrix $\mathbf{A}$ has been optimized. For illustration on how to determine the $\mathbf{A}$ matrix of an SC filter, let us consider the filter depicted in Figure 17.10; for each clock phases matrix $\mathbf{A}$ is given by

$$\mathbf{A}|_{\varphi_1} = \begin{bmatrix} \frac{1}{1+A_1+A_3} & 0 \\ 0 & 1 \end{bmatrix} \tag{17.28}$$

$$\mathbf{A}|_{\varphi_2} = \begin{bmatrix} 1 & \frac{A_2}{1+A_2+A_4+A_5+A_6+A_7} \\ 0 & \frac{1+A_8}{1+A_2+A_4+A_5+A_6+A_7} \end{bmatrix} \tag{17.29}$$

Thus, the designer should use the extra degrees of freedom to maximize the diagonal entries of $\mathbf{A}$ during all phases. Thus, the worst case for any entry of $\mathbf{A}$ determines the minimum value of ω. Another design consideration could be to maximize ω_u. This last consideration should be carefully addressed since very large ω_u values besides extra power consumption can cause excessive phase margin degradation or excessive noise if the capacitors are reduced. Therefore, a judiciously trade-off must be used in choosing the ω_u of each op-amp in the filter topology.

17.4 Noise and Clock Feedthrough

The lower range of signals that can be processed by the electronic devices is limited by several unwanted signals that appear at the output of the circuit. The RMS values of these electrical signals determine the

system noise level, and it represents the lowest limit for the incoming signals to be processed. In most of the cases the circuit cannot properly detect input signals smaller than the noise level.

The most critical noise sources are those due to (1) the employed elements (transistors, diodes, resistors, etc); (2) the noise induced by the clocks; (3) the harmonic distortion components generated due to the intrinsic nonlinear characteristics of the devices; and (4) the noise induced by the surrounding circuitry. In this section, types (1), (2), and (3) are considered. Using fully differential structures can further reduce the noise generated by the surrounding circuitry and coupled to the output of the SC circuit. These structures are partially treated in Section 17.5 but excellent references can be found in the literature [2,4,6,9,13].

17.4.1 Noise due to the Op-Amp

In an MOS transistor, the noise is generated by different mechanisms but there are two dominant noise sources: channel thermal noise and $1/f$ or flicker noise. A discussion of the nature of these noise sources follows.

17.4.2 Thermal Noise

The flow of the carriers due to the drain-source voltage takes places on the source-drain channel, mostly like in a typical resistor. Therefore, due to the random flow of the carriers, thermal noise is generated. For an MOS transistor biased in the linear region the spectral density of the thermal noise is approximately given by [1–4,12]

$$V_{\text{eqth}}^2 = 4kTR_{\text{on}} \tag{17.30}$$

where R_{on}, k, and T are the drain-source resistance of the transistor, the Boltzmann constant, and the temperature (in Kelvin degrees), respectively. In saturation region, the gate-referred spectral noise density can be calculated by the same expression but with R_{on} equal to δ/g_{m}, being g_{m} the small-signal transconductance of the transistor and δ a fitting parameter usually in the range of 0.7–1.

17.4.3 $1/f$ Noise

This type of noise is mainly due to the imperfections in the silicon–silicon oxide interface. The surface states and the traps in this interface randomly interfere with the charges flowing through the channel; hence the generated noise is strongly dependent of the technology. The $1/f$ noise (or flicker noise) is inversely proportional to the gate area because at larger areas more traps and surface states are present and some averaging occurs. The spectral density of the gate referred $1/f$ noise is commonly characterized as

$$V_{\text{eq}1/f}^2 = \frac{k_{\text{F}}}{WLf} \tag{17.31}$$

where the product WL, f, and k_{F} are the gate area of the transistor, the frequency in hertz, and the flicker constant, respectively. The spectral noise density of the MOS transistor is composed by both components, therefore the input referred spectral noise density of a transistor operated in its saturation region becomes

$$V_{\text{eq}}^2 = \frac{4\delta kT}{g_{\text{m}}} + \frac{k_{\text{F}}}{WLf} \tag{17.32}$$

17.4.4 Op-Amp Noise Contributions

In an op-amp, the output referred noise density is composed by the noise contribution of all transistors; hence the noise level is function of the op-amp architecture. A typical unbuffered folded cascode op-amp (folded cascode OTA) is shown in Figure 17.11. For the computation of the noise level, the contribution of each transistor has to be evaluated. Obtaining the OTA output current generated by the gate-referred noise of all the transistors can do this. For instance, the spectral density of the output-referred noise current due to M_1 is straightforwardly determined because the gate-referred noise is at the input of the OTA, leading to

$$i_{01}^2 = G_m^2 V_{eq1}^2 \tag{17.33}$$

where

G_m (equal to g_{m1} at low frequencies) is the OTA transconductance
V_{eq1}^2 is the input (gate) referred noise density of M_1

Similarly, the contributions of M_2 and M_5 to the spectral density of the output referred noise current are given by

$$\begin{aligned} i_{o2}^2 &= g_{m2}^2 v_{eq2}^2 \\ i_{o5}^2 &= g_{m5}^2 v_{eq5}^2 \end{aligned} \tag{17.34}$$

Usually the noise contributions of transistors M_3 and M_4 at medium frequencies are very small in comparison to the other components; this is a nice property of the cascode transistors. This is because their noise drain currents, due to the source degeneration implicit in the cascode transistors, are determined by the equivalent conductance associated with their sources instead of those by their transconductance. Since in a saturated MOS transistor the equivalent conductance is much smaller than the transistor transconductance, this noise drain current contribution can be neglected at low and medium frequencies; this is however not the case at very high frequencies where the parasitic capacitors play an important role. The noise contribution of M_6 is mainly common-mode noise evenly split in the

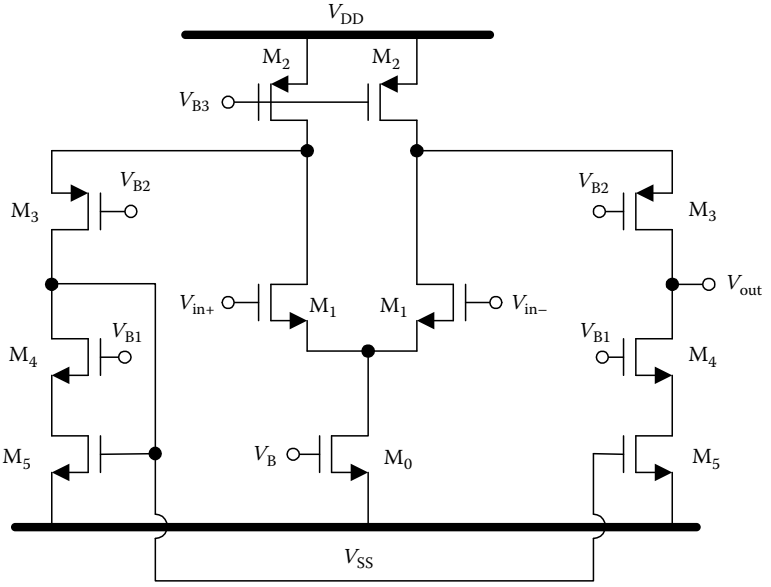


FIGURE 17.11 A folded cascode operational transconductance amplifier.

two arms by transistor M_1, therefore it is ideally canceled at the OTA output due to the current subtraction. At low frequencies, the spectral density of the total output referred noise current can be approximately calculated as

$$i_o^2 = 2\left(g_{m1}^2 V_{eq1}^2 + g_{m2}^2 v_{eq2}^2 + g_{m5}^2 v_{eq5}^2\right) \tag{17.35}$$

The factor 2 is the result of the pair of transistors M_1, M_2, and M_5. From this equation, the OTA input referred noise density is obtained by dividing the output noise current by the square of $G_m = g_{m1}$, yielding

$$v_{OTAin}^2 = 2v_{eq1}^2\left(1 + \frac{g_{m2}^2 v_{eq2}^2 + g_{m5}^2 v_{eq5}^2}{g_{m1}^2 v_{eq1}^2}\right) \tag{17.36}$$

According to this result, if g_{m1} is larger than g_{m2} and g_{m5}, the OTA input referred noise density is mainly determined by the OTA input stage. In that case and using Equation 17.32 and 17.36 yields

$$v_{OTAin}^2 \cong 2v_{eq1}^2 = \frac{2k_F}{W_1 L_1 f} + \frac{8\delta kT}{g_{m1}} = v_{eq/f1}^2 + 4kTR_{eqth} \tag{17.37}$$

where the factor 2 has been included in $v_{eq1/f}$ and R_{eqth}. In Equation 17.37, $v_{eq1/f}^2$ is the equivalent $1/f$ noise density and R_{eqth} is the equivalent resistance for noise, equal to $2\,\delta/g_{m1}$.

17.4.5 Noise in a Switched-Capacitor Integrator

In an SC lossless integrator, the output referred noise density component due to the OTA is frequency limited by the GB product of the OTA. In order to avoid misunderstandings, in this section f_u (the unity-gain frequency of the OTA in hertz) is used instead of ω_u (in radians per second). Since f_u must be higher than the clock frequency, high-frequency noise is folded back into the integrator baseband. In the case of the SC integrator and assuming that the flicker noise is not folded back, the output referred spectral noise density becomes

$$V_{oeq1}^2 = \left(v_{eq1/f}^2 + 4KTR_{eqth}\left(1 + \frac{2f_u}{f_c}\right)\right)|1 + H(z)|^2 \tag{17.38}$$

where the folding factor is equal to f_u/f_c, f_c is the clock frequency and $H(z)$ is the z-domain transfer function of the integrator. The factor $2f_u/f_c$ is the result of both positive and negative folding. Typically, the frequency range of the signal to be processed is around and below the unity-gain frequency of the integrator, therefore $|H(z)| > 1$ and Equation 17.38 can be approximated as

$$V_{oeq1}^2 = \left(v_{eq1/f}^2 + 4kTR_{eqth}\left(1 + \frac{2f_u}{f_c}\right)\right)|H(z)|^2 \tag{17.39}$$

17.4.6 Noise due to the Switches

In SC networks, the switches are implemented with single or complementary MOS transistors. These transistors are biased in the cutoff and ohmic region for open and close operations, respectively. In cutoff region, the drain-source resistance of the MOS transistor is very high, then the noise contribution of the

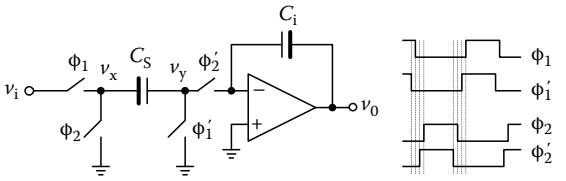


FIGURE 17.12 Typical switched-capacitor lossless integrator.

switch is confined to very low frequencies and it can be considered as dc offset. When the switch is in the on state, the transistor is biased in linear region and its spectral noise distribution is characterized by Equation 17.30. This noise contribution is the most fundamental limit for the signal-to-noise ratio of SC networks. The effects of these noise sources are better appreciated if an SC integrator is considered.

Let us consider the SC integrator of Figure 17.12 and assume that $\phi_1 = \phi_1'$ and $\phi_2 = \phi_2'$. The spectral noise density of the ϕ_1-driven switches are low-pass filtered by the on resistance of the switches R_{on} and the sampling capacitor C_s. The cutoff frequency of this continuous-time filter is given by $f_{on} = 1/(2\pi R_{on} C_s)$. Typically, f_{on} is higher than the clock frequency, therefore, the high-frequency noise is folded back into the baseband of the integrator when the SC integrator samples it. Taking into account the folding effects, the spectral noise density component of the switch-capacitor combination becomes

$$v_{eq2}^2 = 4kTR_{on}\left(1 + \frac{2f_{on}}{f_c}\right) \tag{17.40}$$

with R_{on} the switch resistance. Considering that the noise bandwidth is given by $2\pi f_{on} = 1/R_{on}C_S$, then the integrated noise level yields:

$$v_{eq-integrated}^2 = \frac{kT}{C_S}\left(1 + \frac{2f_{on}}{f_c}\right) \tag{17.41}$$

For the noise induced by the ϕ_2 driven switches, the situation is slightly different, but for practical purposes the same equation is used. According to Equation 17.41, the integrated noise power is inversely proportional to C_S. Therefore, for low-noise applications it is desirable to design the integrators with large capacitors. However, the costs for the noise reduction are larger silicon area and higher power consumption. This last result is because the slew rate of the OTA is inversely proportional to the load capacitance and in order to maintain the specifications it is necessary to increase the current drive capability of the OTA. Clearly, there is a trade-off between noise level, silicon area, and power consumption.

17.4.7 Clock Feedthrough

Another factor that limits the accuracy of SC networks is the charge induced by the clocking of the switches. These charges are induced by the gate-source capacitance, the gate-drain capacitance, and the charge stored in the channel. Furthermore, some of these charges are input signal dependent and introduce distortion in the circuit. Although these errors cannot be canceled, there are some techniques that reduce these effects.

The analysis of the clock feedthrough is not straightforward because it depends on the order of the clock phases, the relative delay of the clock phases, as well as the speed of the clock transition [13]. For instance, let us consider in Figure 17.12 the case when ϕ_1 goes down before ϕ_1'. This situation is shown in Figure 17.13a.

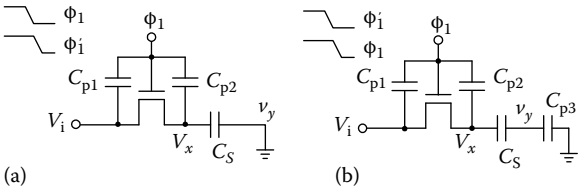


FIGURE 17.13 Charge induced due to the clocks. (a) If ϕ_1 goes down before ϕ_1' and (b) If ϕ_1' goes down before ϕ_1.

While C_{p1} is connected between two low-impedance nodes, C_{p2} is connected between ϕ_1, a low-impedance node, and v_x. For $\phi_1 > v_i + V_T$ the transistor is on and the current injected by C_{p2} is absorbed by the drain-source resistance; then v_x remains at a voltage equal to v_i. When the transistor is turned off, $\phi_1 < v_i + V_T$, charge conservation at node v_x leads to

$$v_x = v_i + \frac{C_{p2}}{C_S + C_{p2}}(\text{VSS} - v_i - V_T) \tag{17.42}$$

where VSS is the low level of ϕ_1 and ϕ_2. During the next clock phase, the charge of C_S is injected to C_i. Thus, C_{p2} induces a charge error proportional to $v_i C_{p2}/(C_S + C_{p2})$. In addition, an offset voltage proportional to VSS – V_T is also generated. Because the threshold voltage V_T is a nonlinear function of v_i, an additional error in the transfer function and harmonic distortion components appear at the output of the integrator. The same effect occurs when the clock phases ϕ_2 and ϕ_2' have a similar sequence.

Let us consider the case when ϕ_1' is opened before ϕ_1; the situation is shown in the Figure 17.13b. Before the transistors turn off, $v_x = v_i$ and $v_y = 0$. When the transistor is turned off, $\phi_1 < v_i + V_T$, the charge is recombined between C_S, C_{p2}, and C_{p3}. After the charge redistribution, the charge injected into C_{p3} is approximately given by

$$\Delta Q_{C_{p3}} = \frac{C_{p2}C_{p3}}{C_{p2} + C_{p3}}(\text{VSS} - v_i - V_T) \tag{17.43}$$

Notice that the current feedthrough generated by C_{p2} flows through both capacitors C_S and C_{p3}, hence $\Delta Q_{C_{p3}} = -\Delta Q_{C_s}$. During the next clock phase, both capacitors C_S and C_{p3} transfer the ideal charge $C_S v_i + \Delta Q_{C_s}$ and $\Delta Q_{C_{p3}}$, making the clock feedthrough induced error close to zero. The conclusion is that if the clock phase ϕ_1' is a bit delayed than ϕ_1 the clock-induced error is further reduced. This is also true for the clock phases ϕ_2 and ϕ_2'.

In Figure 17.12, the right-hand switch also introduces clock feedthrough, but unlike the clock feedthrough previously analyzed, this is input signal independent. When the clock phase ϕ_2 goes down, the gate-source overlap capacitor extracts from the summing node the following charge:

$$\Delta Q = C_{GS}(\text{VSS} - V_T) \tag{17.44}$$

In this case, V_T does not introduce distortion because v_y is almost at zero voltage for both clock phases. The main effect of this charge is to introduce an offset voltage. The same analysis reveals that the bottom right-hand switch introduces similar offset voltage. From the previous analysis it can be seen that using minimum transistors dimension can reduce the clock feedthrough. This implies minimum parasitic capacitors and minimum induced charge from the channel. If possible, the clock phases should be arranged for minimum clock feedthrough. The use of extra dummy switches driven by the complementary clock phase may alleviate the clock feedthrough issue, but its effectiveness must be carefully evaluated by postlayout simulations.

17.4.8 Channel Mobile Charge

Charge injection also occurs due to the mobile charge in the channel. If the transistor is biased in linear region, the channel mobile charge can be approximated by

$$Q_{ch} = C_{GS}(V_{GS} - V_T) \tag{17.45}$$

where C_{GC} is the gate-channel capacitor; see Figure 17.14.

When the switch is turned off this charge is released and part of it goes to the sampling capacitor. Fortunately, the previously discussed early clock phase technique for the reduction of clock feedthrough also reduces the effects of the channel mobile charge injection. The mobile charge injected to the sampling capacitor is a function of several parameters, e.g., input signal, falling rate of the clock, overlap capacitors, gate capacitor, threshold voltage, integrating capacitor, and the supply voltages.

The effects of the channel mobile charges are severe when the ϕ_1-driven transistor is opened before the ϕ_1'-driven transistor. In this case, the situation is similar to that shown in Figure 17.13a, in which one terminal of C_S is still grounded. While ϕ_1 is higher than $v_i + V_T$ the channel resistance is small and v_i absorbs most of the channel-released charge. For $\phi_1 < v_i + V_T$, the channel resistance increases further and a substantial amount of charge released by the channel will flow through C_S, introducing a charge error. If the clock phases are arranged as shown in Figure 17.13b, most of the charge released by the channel returns back to v_i. The main reason is because the equivalent capacitor seen at the right-hand side of the transistor is nearly equal to C_{p3}, if C_{p2} is neglected, making this a high-impedance node. Because this parasitic capacitor is smaller than the sampling capacitor, a small amount of extracted (or injected) charge will produce a huge variation on v_y, see Figure 17.13b, pushing back most of the mobile charges. Equation 17.45 shows that the mobile charge is a function of V_{GS} ($=V_{clock} - V_{in}$ for the sampling switch), hence the part of the charge injected into the sampling capacitor is signal dependent. This issue can be alleviated if the clock is correlated with the incoming signal such that VGS is maintained constant using charge pumps circuits; the technique proposed in Ref. [14] implements this concept.

17.4.9 Dynamic Range

The dynamic range is defined as the ratio of the maximum signal that the circuit can drive without significant distortion to the noise level. The maximum distortion tolerated by the circuit depends on the application, but −60 dB is commonly used. Since the linearity of the capacitor is good enough and if the harmonic distortion components introduced by the OTA input stage are small, the major limitation for

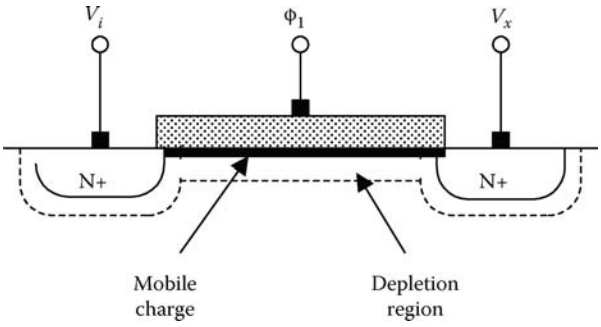


FIGURE 17.14 Cross section of the MOS transistor showing the mobile charge.

the linearity is determined by the output stage of the OTA. For the folded cascode OTA of Figure 17.11 this limit is

$$v_{\text{omax}} \cong V_{\text{R2}} + V_{\text{TP3}} \tag{17.46}$$

If the reference voltage V_{R2} is maximized, Equation 17.46 yields

$$V_{\text{omax}} \cong V_{\text{DD}} - 2V_{\text{DSATP}} \tag{17.47}$$

A similar expression can be obtained for the lowest limit. Assuming a symmetrical single-side output stage, from Equation 17.47, the maximum rms value of the OTA output voltage is given by

$$v_{\text{oRMS}} \cong \frac{V_{\text{DD}} - 2V_{\text{DSATP}}}{\sqrt{2}} \tag{17.48}$$

If the in-band noise, integrated up to $\omega = 1/R_{\text{int}}C_{\text{I}}$, is considered and if the most important term of Equation 17.41 is retained, the dynamic range of the single-ended SC integrator becomes

$$\text{DR} \cong \frac{(V_{\text{DD}} - 2V_{\text{DSATP}})}{\sqrt{2}\sqrt{2kT/C_{\text{S}}}} = \frac{(V_{\text{DD}} - 2V_{\text{DSATP}})}{2\sqrt{kT}}\sqrt{C_{\text{S}}} \tag{17.49a}$$

For this equation, the alias effects have been neglected and only the noise contribution of the switches driven by ϕ_1 and ϕ_2 are considered. At room temperature, this equation is reduced to the following expression:

$$\text{DR} \cong 7.9 \times 10^9 \sqrt{C_{\text{S}}}(V_{\text{DD}} - 2V_{\text{DSATP}}) \tag{17.49b}$$

According to this result, the dynamic range of the SC integrator is reduced when the power supplies are scaled down and minimum capacitors are employed. Clearly, there is a trade-off between power consumption, silicon area, and dynamic range. As an example, for the case of $C_{\text{S}} = 1.0$ pF and supply voltages of ± 1.5 V and $V_{\text{DSATP}} = 0.25$ V, the dynamic range of a single integrator is around 78 dB. For low-frequency applications, however, the dynamic range is lower due to the low-frequency flicker noise component. This is a very optimistic result since neither the op-amp noise not aliasing effects nor other noise sources were considered.

17.5 Design Considerations for Low-Voltage Switched-Capacitor Circuits

For the typical digital supply voltages, 0−5 V, SCs achieve dynamic ranges of the order of 80–100 dB. As long as the power supplies are reduced, the swing of the signal decreases and the switch resistance increases further. Both effects reduce the dynamic range of the SC networks. A discussion of these topics follows.

17.5.1 Low-Voltage Operational Amplifiers

The design techniques for low-voltage low-power amplifiers for SC circuits have been addressed by several authors [4,7–9,11–13]. The implementation of op-amps for low-voltage applications does not seem to be a fundamental limitation as long as the transistor threshold voltage is smaller than $(V_{\text{DD}} - V_{\text{SS}})/2$. This limitation will become clear in the design example presented in this section.

The design of the operational amplifier is strongly dependent on the application. For high-frequency circuits, the folded cascode op-amp is suitable, but the cascode transistors limit the swing of the signals at the output stage. If large output voltage swing is needed, complementary stages are desirable. To illustrate the design trade-offs involved in the design of a low-voltage OTA, let us consider the folded cascode OTA of Figure 17.11. For low-voltage applications and small signals, the transistors have to be biased with very low V_{DSAT} $(=V_{GS} - V_T)$.

For the case of supply voltages limited to ± 0.75 V and $V_T = 0.5$ V, $V_{DSAT1} + V_{DSAT6}$ must be lower than 0.25 V, otherwise the transistor M_6 goes to the triode region. For large signals, however, the variations of the input signal produce variations at the source voltage of M_1. It is well known that linear range of the differential pair is of the order of $\pm 1.4\ V_{DSAT1}$. Hence, if the noninverting input of the OTA is connected to the common-mode level V_{CM}, for proper operation of the OTA input stage (see Figure 17.11) it is desirable to satisfy

$$V_{CM} - V_T - \text{VSS} > 1.4V_{DSAT1} + V_{DSAT6} \tag{17.50}$$

It has to be taken into account that the threshold voltage of M_1 increases if an N-well process is used, due to the body effects. In critical applications, PMOS transistors fabricated in a different well with their source tied to their own well can be used. Equation 17.50 shows that it is beneficial to select a proper common-mode level V_{CM} to extend the linear range of the amplifier input stage, especially for low-voltage applications.

The dimensioning of the transistors and the bias conditions are directly related to the application. For instance, if the SC integrator must slew 1 V into 4 μs and the load capacitor is of the order of 10 pF, the OTA output current must be equal to or higher than 2.5 μA. Typically, for the folded cascode OTA the dc current of both output and input stages are the same. Therefore, the bias current for M_1, M_3, M_4, and M_5 can be equal to 2.5 μA. The bias current for M_2 and M_6 is 5 μA. If $V_{GS1} - V_{T1}$ is fixed at 100 mV the dimensions of M_1 can be computed according to the required small signal transconductance. Similarly, the dimension of the other transistors can be calculated, most of them designed to maximize the output swing and dc gain. A very important issue in the design of low-voltage amplifiers is the reference voltage. In the folded cascode of Figure 17.11, the values of the reference voltages V_{R1} and V_{R2} must be optimized for maximum output swing. OTA design techniques are fully covered in Refs. [11] and [13].

17.5.2 Analog Switches

For low-voltage applications, the highest voltage that can be processed is limited by the analog switches rather than by the op-amps. For a single NMOS transistor, the switch resistance is approximately given by

$$R_{DS} = \frac{1}{\mu_n C_{OX} \frac{W}{L}(V_{GS} - V_T)} \tag{17.51}$$

where μ_n and C_{OX} are technological parameters. According to Equation 17.51, the switch resistance increases further when V_{GS} approaches to V_T. This effect is shown in Figure 17.15 for the case $V_{DD} = -V_{SS} = 0.75$ V and $V_T = 0.5$ V. From this figure, the switch resistance is higher than 300 kΩ for input signals of 0.2 V. However, for a drain-source voltage higher than $V_{GS} - V_T$ the transistor saturates and does not behave as a switch anymore. This limitation clearly reduces further the dynamic range of the SC circuits.

A solution for this drawback is to generate the clocks from higher voltage supplies. A simplified diagram of a voltage doubler is depicted in Figure 17.16a. During the clock phase ϕ_1, the capacitor C_1 is charged to V_{DD} and during the next clock phase its negative plate is connected to V_{DD}. Hence, at the

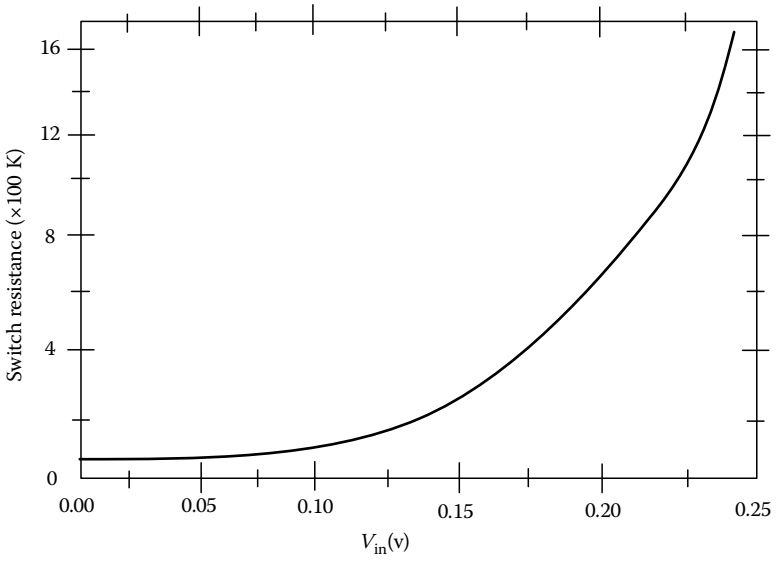


FIGURE 17.15 Typical switch resistance for an NMOS transistor.

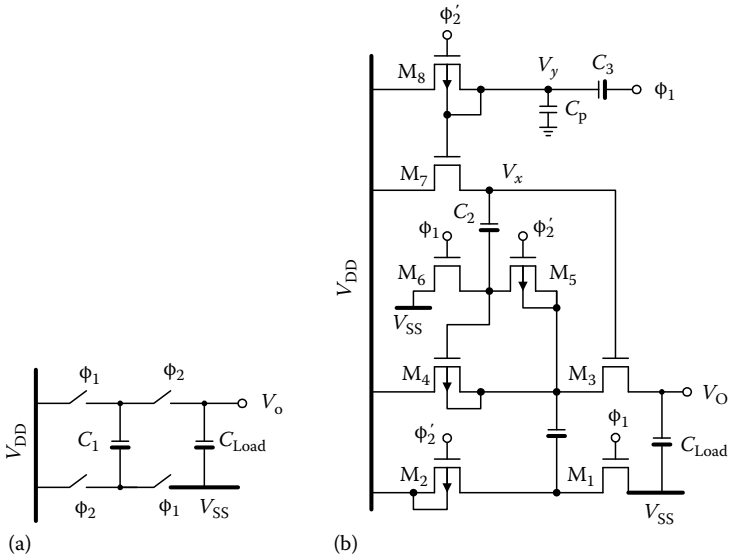


FIGURE 17.16 Voltage doubler. (a) Simplified diagram and (b) transistor level diagram.

beginning of ϕ_2, the voltage at the top plate of C_1 is equal to $2V_{DD} - V_{SS}$. After several clock cycles, if C_{LOAD} is not further discharged, the charge is recombined leading to an output voltage equal to $2V_{DD} - V_{SS}$. A practical implementation for an N-well process is shown in Figure 17.16b.

In this circuit, the transistors M_1, M_2, M_3, and M_4 behave as the switches S_1, S_2, S_3, and S_4 respectively, of Figure 17.16a. While for M_1 and M_2 the normal clocks are used, special clock phases are generated for M_3 and M_4 because they drive higher voltages. The circuit operates as follows.

During ϕ_1, M_8 is opened because ϕ_2' is high. The voltage at node v_y is higher than V_{DD} because the capacitors C_3 and C_p were charged to V_{DD} during the previous clock phase ϕ_2 and, when ϕ_1 goes up, charge is injected to the node through the capacitor C_3. Since v_y is higher than V_{DD}, M_7 is turned on. The bottom plate of C_2 is connected to V_{SS} by M_6 and C_2 is charged to $V_{DD}-V_{SS}$.

During ϕ_2, the refresh clock phase, the bottom plate of C_1 is connected to V_{DD} by the PMOS transistor M_2. Note that if an NMOS transistor is employed, the voltage at the bottom plate of C_1 is equal to $V_{DD}-V_T$, resulting in lower output voltage. During ϕ_2 if C_1 is not discharged, the voltage at its top plate is $2V_{DD} - V_{SS}$. The voltage at node v_x becomes close to $3V_{DD}-2V_{SS}$ volts, turning M_3 on and enabling the charge recombination of C_1 and C_{LOAD}. As a result, after several clock periods, the output voltage v_o becomes equal to $2V_{DD}-V_{SS}$. To avoid discharges the gate of M_4 is connected to the bottom plate of C_2. Thus, M_4 is turned off during the refresh phase. An efficient solution using boosted clock switches is described in Ref. [14]. Another technique using switched op-amps, described in Ref. [15], have been also successfully used.

17.6 Fully Differential Filters

An important issue when designing a high-frequency filter is the proper selection of the clock phases for the switches. It is desirable that the input to each integrator is as close as possible to a step function, and to minimize the loading capacitors during the integration clock phase. Making the OTAs to perform the hold and integrate functions at alternate phases ensure this. Unfortunately this approach cannot be implemented in the conventional single-ended topologies unless an additional inverter is added to the loop. Two backward integrators with a half-delay in each and by crossing the outputs of the fully differential integrators in order to have an odd number of inversions in the loops is one of the best configurations for high-frequency applications as depicted in Figure 17.17. The first op-amp may process the information

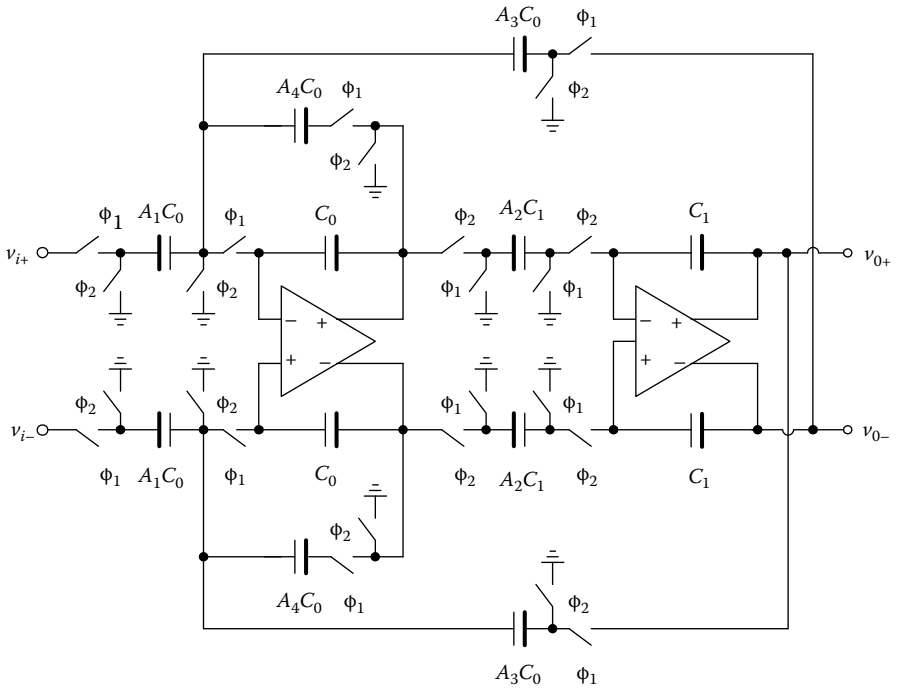


FIGURE 17.17 Fully decoupled, fully differential biquadratic filter.

during ϕ_1 and hold it during ϕ_2 while the second op-amp do the same functions during the complementary clock phases. In general, having the differential outputs available makes the system more flexible and gives several advantages while designing high-performance analog signal processors.

Fully differential amplifiers are widely used technique to reduce the effect of charge injection, clock feedthrough, and better rejection to common-mode signals such as substrate and power supply noise and an improved dynamic range when compared to single-ended circuits. As long as the noise present at the output of the amplifier is present in both outputs of the amplifier with the same amplitude and same phase, it will be rejected by the differential nature of the following stage. In addition, an additional loop that operates on the common-mode signals presents low output impedance for these signals, it fix the operating point of the differential outputs at the desired level and further minimizes the common-mode output fluctuations up to the unity-gain frequency of the loop. Their disadvantage comes from the fact that they require a common-mode feedback (CMFB) circuit.

The fully differential version of the folded cascode amplifier is depicted in Figure 17.18a. It requires the same amount of power as the single-ended version shown in Figure 17.11. Since the current mirror present in the single-ended amplifier for the conversion of the fully differential current of the differential pair into single-ended output is eliminated, the fully differential amplifier has a single internal pole located at the source terminal of M_3. Since several parasitic poles and phantom zeros are eliminated, the fully differential amplifier has inherently better phase margin than its single-ended counterpart. The circuit is fully symmetric and any noise injected through M_0 is evenly split by the differential pair and will appear at the two outputs as a pure common-mode signal. The common-mode noise present at the transistors M_1 is further attenuated by the small sensitivity of the differential pair to common-mode

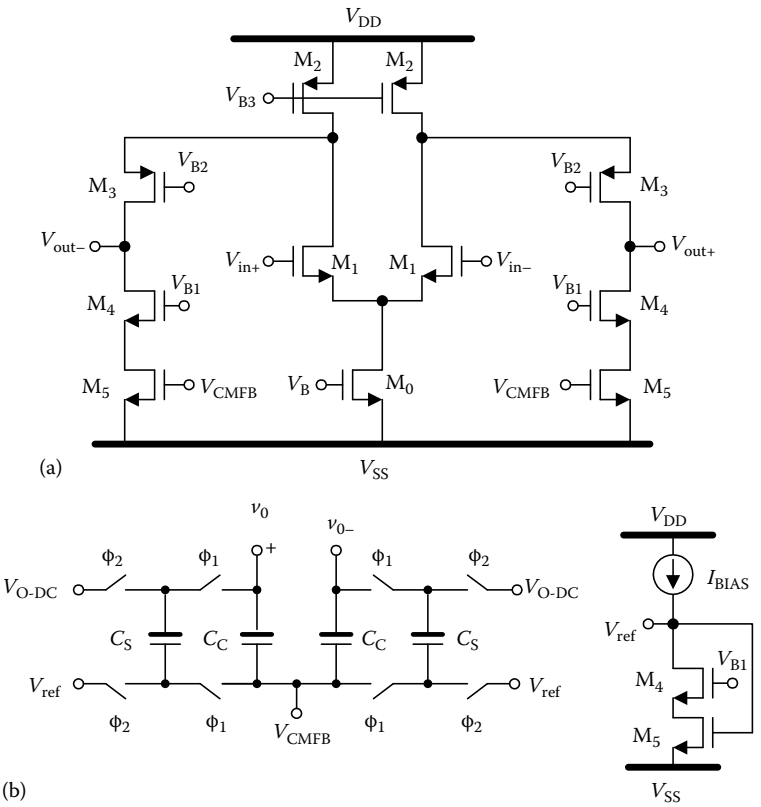


FIGURE 17.18 (a) Fully differential folded cascode amplifier and (b) switched-capacitor based CMFB circuit.

signals; the small common-mode current resulting at the drain of M_1 will be reflected at the output as a common-mode signal. Similarly, any V_{DD} noise generates a current noise through M_2 that appears as a common-mode noise at both outputs of the amplifier; notice that to minimize these components V_{B3} must be further attached to V_{DD}. V_{SS} noise generates current that appears at the output as common-mode noise, as well thanks to the differential nature of the topology. The gate voltage of transistors M_5 is used to accommodate the control of the CMFB loop; V_{B3} can also be used for that purpose.

The CMFB is a block that extracts the common-mode signal present at both outputs and compares its dc value with a reference voltage in a loop with enough gain for the common-mode signals. If the loop has enough gain, the error between the detected common-mode signal and the reference voltage is minimized forcing the common-mode output signals (DC operating point of each amplifier output) to be at the reference level. Several CMFB circuits have been proposed in the literature [4,11,12] but for SC applications it has been preferred to use SC-based CMFB (SC–CMFB) circuits [4] since they do not consume significant power and have better linearity when compared to their continuous-time counterpart. The CMFB circuits consist of a common-mode voltage detector (output is proportional to $\frac{v_{out+}+v_{out-}}{2}$ and a circuit that generates the common-mode error after comparing the common-mode output voltage and a proper and stable reference voltage. The efficient SC detector, shown in Figure 17.18b, uses two SC resistors (C_C and switches) that detect the common-mode output voltage and compare it with $V_{O\text{-}DC}$ [4,9,13]. Using conventional charge redistribution analysis and ignoring the effect of C_C, the voltage at node V_{CMFB} during ϕ_1 is given by

$$V_{CMFB} = \left(\frac{v_{out+} + v_{out-}}{2} - V_{O-DC}\right) + V_{ref} \tag{17.52}$$

The circuit makes the comparison of the common-mode output voltage and the desired output DC level $V_{O\text{-}DC}$. A remarkable advantage of this circuit is that in addition to the desired comparison, it translates the ideal voltage V_{ref} needed at the gate of M_5 to generate the DC current in transistors M_5. In steady state, V_{CMFB} will be very close to V_{ref}. Equation 17.52 is also valid for the common-mode noise; in the absence of noise in V_{ref} and $V_{O\text{-}DC}$, the common-mode output noise is compared with zero and for the frequencies where the CMFB loop is high, it is suppressed, thanks to the feedback. Hence it is desirable not only to have large low-frequency gain but wide-band CMFB loop to reject high-frequency noise as well.

Since the SC resistors and the parasitic capacitors generate a relatively low-frequency pole at V_{CMFB} node, the capacitors C_S are added to introduce a compensating zero that helps with the stabilization of the CMFB loop. One of the drawbacks of this topology is that the common detector loads significantly the output of the amplifier, reducing its unity-gain frequency and slew rate. The value of C_S is typically one-quarter to one-tenth of C_C. To prevent significant offset voltages due to charge injection from the switches, C_S is commonly chosen to be greater than 200 fF. Therefore, the capacitive loading on the amplifier due to the CMFB can go from 1 pF to 2 pF in most designs. As demonstrated in Ref. [16] this SC–CMFB suffers of poor rejection to the supply noise used as reference for the common-mode loop. This can be partially alleviated by combining passive common-mode detector and active signal comparator at the expenses of additional power consumption or using continuous-time CMFB systems.

17.7 Design Examples

17.7.1 Biquadratic Filter

In this section, a second-order bandpass filter (BPF) is designed. The specifications for this biquad are

Center frequency 1.6 kHz
Quality factor 16
Peak gain 10 dB
Clock frequency 8 KHz

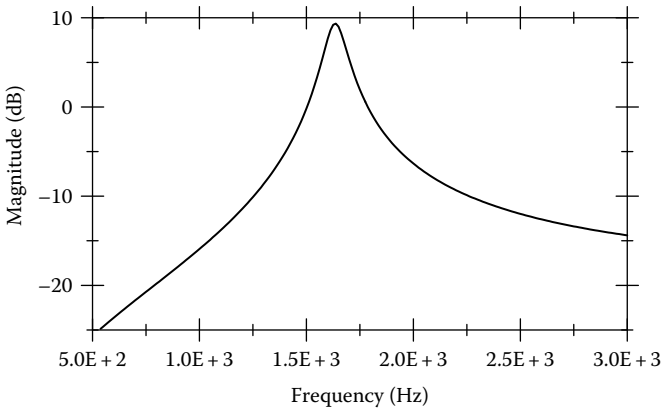


FIGURE 17.19 Frequency response of the second-order bandpass filter.

A transfer function in the z-domain that realizes this filter is given by the following expression:

$$H(z) = \frac{0.1953(z-1)z}{z^2 + 0.5455z + 0.9229} \tag{17.53}$$

The equivalent $H(s)$ is obtained and then mapped into the z-domain using the LDI transformation. This transfer function can be implemented by using the biquads presented in Section 17.2. For the biquad of Figure 17.10, and employing $A_1 = A_4 = A_7 = 0$, the circuit behaves as a BPF. Equating the terms of Equation 17.53 with the terms of Equation 17.19, the following equations are obtained:

$$\begin{aligned} A_6 &= \frac{1}{0.9229} - 1 \\ A_5 &= \frac{0.1953}{0.9229} \\ A_2A_3 &= 2 + A_6 - \frac{0.5455}{0.9229} \end{aligned} \tag{17.54}$$

Solving these equations, the following values are obtained: $A_6 = 0.0835$, $A_5 = 0.2116$, and $A_2A_3 = 1.4924$. A typical design procedure employs $A_2 = 1$. For this case, after node scaling the total capacitance is of the order of 32 unit capacitances. The frequency response of the filters is shown in Figure 17.19.

17.7.2 Sixth-Order Bandpass Ladder Filter

In this section, the design procedure for a sixth-order BPF based on an RLC prototype is considered. The ladder filters are very attractive because of their low passband sensitivity to tolerances in the filter components. Let us consider the following specifications:

Center frequency 100 kHz
Bandwidth 2.5 kHz
Passband ripple $<$0.5 dB
Clock frequency 2 MHz

The design starts with an RLC low-pass prototype, see Figure 17.20.

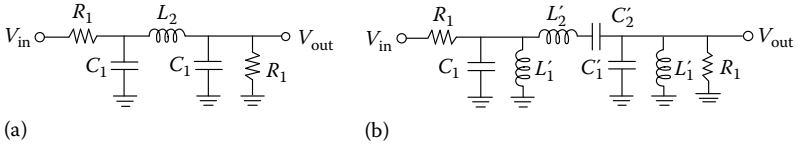


FIGURE 17.20 RLC prototypes. (a) Third-order low-pass filter and (b) sixth-order bandpass filter.

The values of the components for a prototype with 1 rad/s passband frequency can be obtained from tables or computed from well-known expressions. For this example, the passive components are $R_1 = 1\ \Omega$, $C_1 = 1.5963$ F, and $L_2 = 1.0967$ H. Using the low-pass to bandpass transformation, the capacitor C_1 is transformed into a parallel of a capacitor and an inductor. The values of the resulting components are

$$C_1' = \frac{C_1}{BW}$$
$$L_1' = \frac{1}{\omega_0^2 C'} \tag{17.55}$$

where ω_o and BW are the center frequency in radians per second and the filter bandwidth in radians per second, respectively. The inductor L_2 is transformed in a series of an inductor and a capacitor whose values become

$$L_2' = \frac{L_2}{\text{BW}}$$
$$C_2' = \frac{1}{\omega_0^2 L_2'} \tag{17.56}$$

The bandpass prototype is shown in Figure 17.20b. Before making the denormalizations it is desirable to transform the passive prototype to an active implementation. The grounded LC tank circuit can be simulated by the RC active implementation shown in Figure 17.21.

The value of the simulated inductance is

$$L_1' = R_1^2 C_1' \tag{17.57}$$

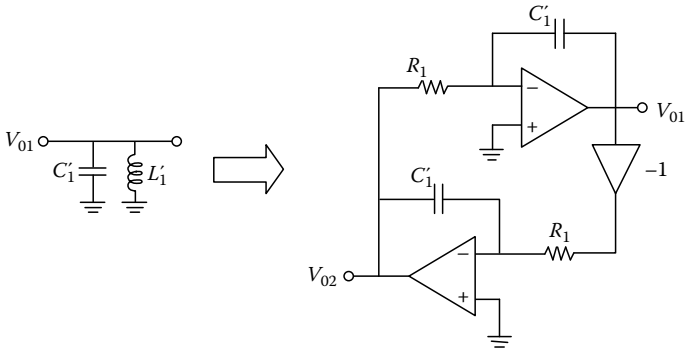


FIGURE 17.21 RC active implementation of a grounded LC tank circuit.

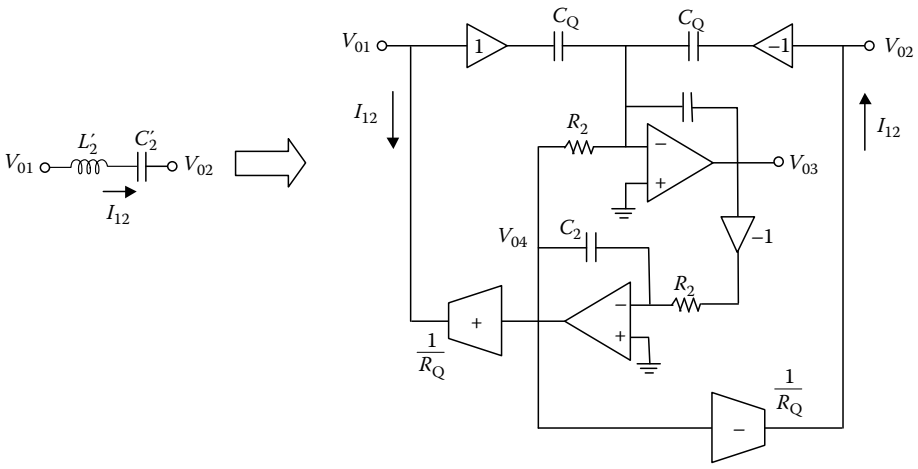


FIGURE 17.22 RC active implementation of a floating LC tank circuit.

It is important to note that the output of the active-RC circuit is a low-impedance node. Therefore, the inverting input of the op-amps must be used for current injection. Similarly, the circuit shown in Figure 17.22 can simulate a floating resonator. Obviously, this is an expensive implementation but typically several active elements can be shared in the final design. Using typical circuit analysis techniques it can be shown that i_{12} for the passive LC tank is related to the components by the following expression:

$$i_{12} = \frac{sC_2'}{1 + s^2 L_2' C_2'}(v_{o1} - v_{o2}) \tag{17.58}$$

and for the active implementation

$$i_{12} = \frac{\left(\frac{R_2}{R_Q} C_Q\right)s}{1 + s^2 R_2^2 C_2^2}(v_{o1} - v_{o2}) \tag{17.59}$$

Comparing Equations 17.57 and 17.58, the active and passive implementations are equivalents if the following constraints are satisfied:

$$C_2' = \frac{R_2}{R_Q} C_Q \tag{17.60}$$

$$L_2' C_2' = R_2^2 C_2^2 \tag{17.61}$$

Obviously, the resistors R_2 can be replaced by SC equivalents. Using these building blocks, the implementation of the sixth-order BPF is straightforward. The sixth-order SC BPF is shown in Figure 17.23. The implementation of the $L_1' C_1'$ tank circuit is straightforward from Figure 17.21. The resonant frequency of the SC resonator is determined by the capacitors θC_1. The node voltage v_{o1} is taken at the output of the first op-amp. The input resistor is implemented by the capacitor C_R and its associated switches. In Figure 17.23, the resonator associated with v_{o3} and v_{o4} and the capacitors C_Q implement the $L_2' C_2'$ floating tank circuit. The final step of the design is to compute the values of the components. A simplified design follows, but more detailed procedures are presented in Refs. [1,2,6].

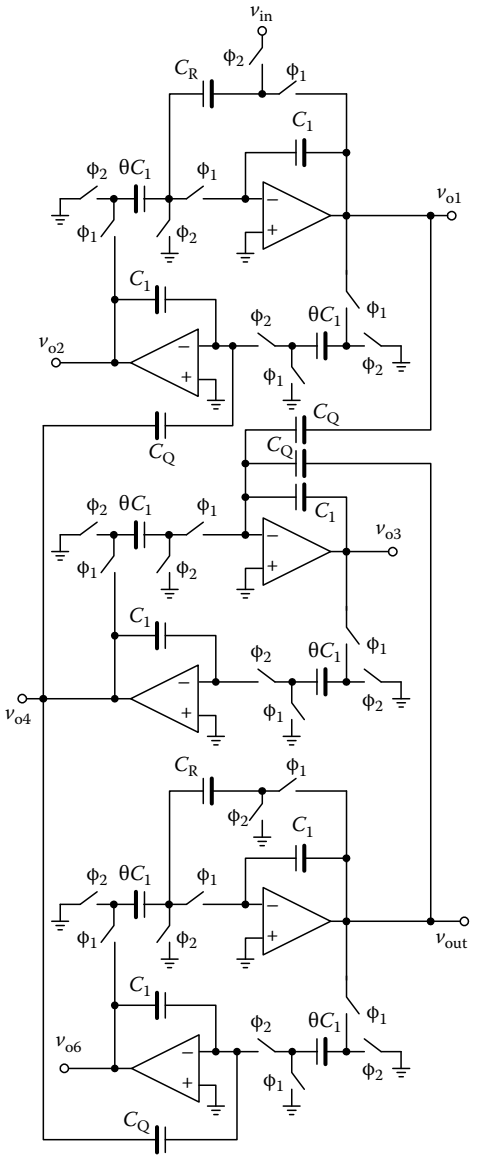


FIGURE 17.23 Sixth-order bandpass ladder filter.

The loops of the SC filter implementation are of the lossless discrete integrator (LDI) type and the resistors are of the serial type. Hence, if an LDI prewarping scheme is used the analog center frequency of the filter should be mapped to the desired value, mainly for high-Q filters wherein the interaction between resonators is very small. Since the resistors are not LDI, a distortion in the quality factor of the filter sections occurs, thus increasing the passband ripple. The LDI transformation relates the analog and discrete frequencies by the following expression [1,6]:

$$\omega_{\text{analog}} = 2f_c \sin\left(\frac{\omega_{\text{discrete}}}{2f_c}\right) \tag{17.62}$$

Applying the LDI transformation to the center frequency of the filter, the predistorted center frequency is equal to 99,056 kHz. Using this prewarped frequency and Equations 17.54 through 17.61, the following component values are obtained:

$$R_1 = 1\ \Omega$$
$$C_1' = 1.01623 \times 10^{-4}\ \text{F}$$
$$L_1' = 2.51316 \times 10^{-8}\ \text{H}$$
$$C_2' = 3.65803 \times 10^{-8}\ \text{F}$$
$$L_2' = 6.98181 \times 10^{-5}\ \text{H}$$

In addition, the continuous-time and SC resistors are approximately related by the following expression:

$$C_{\text{eq}} \cong \frac{1}{f_c R_{\text{cont}}} \tag{17.63}$$

In the SC filter, the transconductor $1/R_Q$ is implemented by C_Q/C_1R. Hence, the final values of the capacitors are

$C_R = 0.0738$ pF
$C_1 = 15.000$ pF
$\theta C_1 = 4.6900$ pF
$C_Q = 0.2833$ pF

While the filter center frequency of this design is accurate, the passband ripple is increased to around 0.8 dB instead of 0.5 dB. This is because the resistors used in the terminals of the filter are not implemented as LDI resistors. However, this effect can be partially corrected adjusting the resonant frequency of the second loop. If θC_1 is equal to 4.685 pF for the second resonator, the ripple is decreased to around 0.55 dB. The results for this case are shown in Figure 17.24.

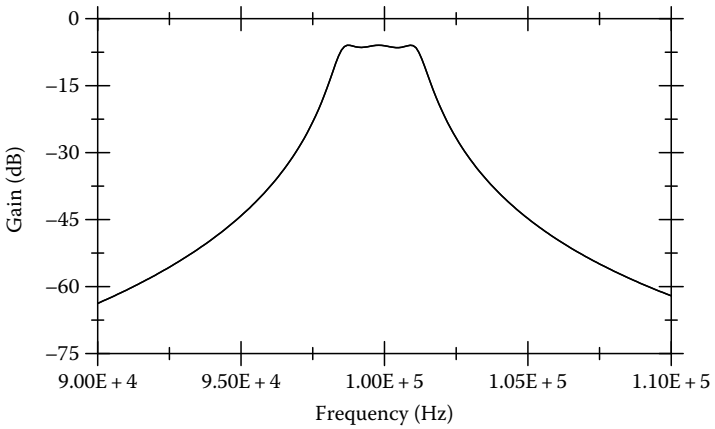


FIGURE 17.24 Frequency response for the sixth-order bandpass filter.

17.7.3 Programmable Switched-Capacitor Filter

In most of the programmable filters, the important parameters are the resonant frequencies, the pole-quality factor, and sometimes, the peak gain. Thus, programmable low-pass, bandpass, high-pass, and bump equalizers are typically designed. Nevertheless, in some applications it is more important to control the gain at the frequency bands instead of that at the resonant frequency, the quality factor, or the peak gain. A typical approach for the implementation of these systems is to employ a parallel of a low-pass, bandpass, and a high-pass filter with programmable peak gain. For a second-order system, the implementation of this approach needs at least six operational amplifiers, three capacitor banks, and the implementation of six poles. Therefore, the number of switches, the power dissipation, and the silicon area needed for these structures are considerable. Another approach follows in this section.

In order to independently control the low-, medium-, and high-frequency bands it is required to realize the following transfer function:

$$H(s) = \frac{K_1 s^2 + K_2 \mathrm{BW} s + K_3 \omega_o^2}{s^2 + \mathrm{BW} s + \omega_o^2} \tag{17.64}$$

where BW and ω_o are the bandwidth and the frequency of the poles, respectively. The control of the frequency bands is carried out by the parameters K_1, K_2, and K_3. From Equation 17.64 it is clear that the dc gain and the high-frequency gain are equal to K_3 and K_1, respectively. A disadvantage of this expression is the lack of good control in the medium frequencies. If K_1 is equal to K_3, the filter gain at the resonant frequency depends on K_2. However, for the general case it is affected by the parameters K_1 and K_3. In addition, the shape of the transfer function is not well behaved for moderate- and high-Q applications. The behavior of the transfer function is improved if the following equation is employed:

$$H(s) = K_1 \left(\frac{s^2 + K_2 K_3 \mathrm{BW} s + K_3 \omega_o^2}{s^2 + K_1 \mathrm{BW} s + K_1 \omega_o^2} \right) \tag{17.65}$$

For low and high frequencies the behavior of this equation is similar to that of Equation 17.64. At medium frequencies the behavior of the transfer function is better controlled by the parameter K_2 than

for the case of Equation 17.64. For the medium-frequency band, the gain is related to the product K_2K_3 instead of the absolute value of K_2. Therefore, the effect of parameter K_2 on the transfer function is related to K_3.

A block diagram representation of Equation 17.65 is presented in Figure 17.25. In this figure, the filter bandwidth BW is equal to ω_o/Q, with Q equal to the quality factor of the filter. The control parameters are the gain factors K_1, K_2, and K_3. The implementation of this block diagram can be carried out by using various techniques, e.g., OTA-C, MOSFET-C, or SC. An SC implementation is shown in Figure 17.26. For high-sampling rate $2\pi f_c \gg \omega_o$, the capacitors θC_I are related with the pole frequency by the equation $\theta = \omega_o T$. The capacitor banks control the gain of the three frequency bands. For the computation of the total capacitance, the capacitors can be associated in the following groups: the first group (K_3C_1, C_1), second group (θC_2, K_2C_2/Q, C_2), and third group ($\theta C_3/Q$, θC_3, C_3/K_1, C_3).

The versatility of the topology is shown in Figure 17.27. For these results, the following design parameters have been employed: clock frequency = 128 kHz, $\omega_o = 6.2832 * 350$ rad/s, and $Q = 0.5$. For Figure 17.26, the parameter K_3 is equal to 10, fixing the low-frequency gain at 20 dB. The medium-frequency band is controlled by the parameter K_2, which corresponds to the values 0.5 (attenuation of 6 dB), 1 (0 dB attenuation), and 2 (gain of 6 dB). The high-frequency gain is controlled by the parameter K_1, which corresponds to the values 1, 2.5, 5, and 10, giving a high-frequency gain of 0, 8, 14, and 20 dB, respectively.

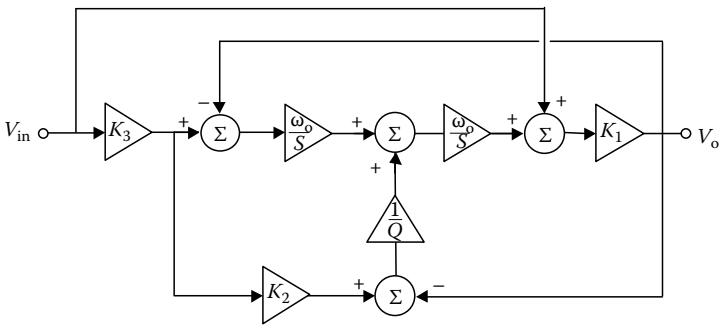


FIGURE 17.25 A flow diagram representation of Equation 17.65.

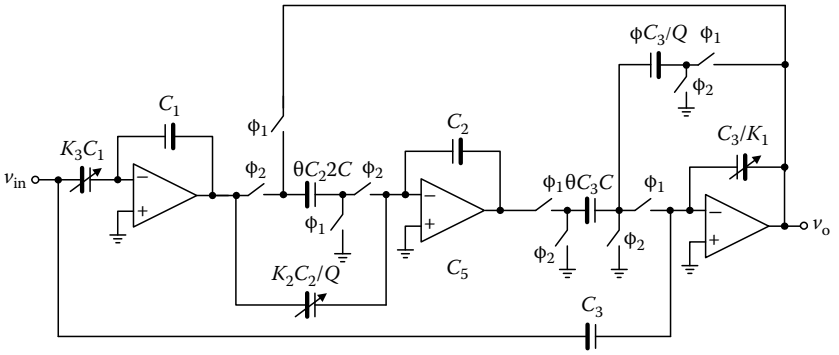


FIGURE 17.26 Switched-capacitor realization for the programmable filter.

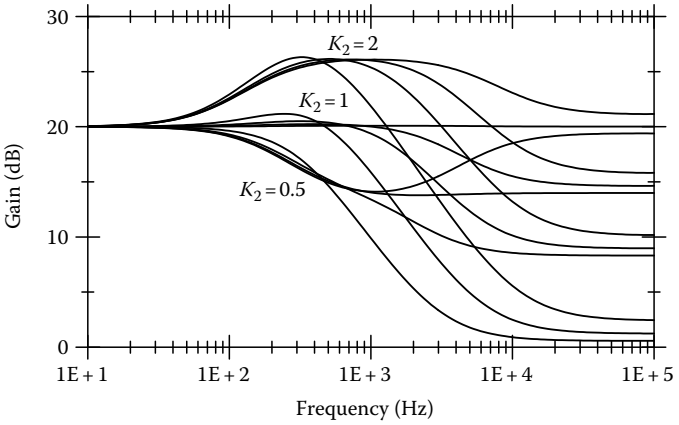


FIGURE 17.27 Effect of the parameter K_2 on the filter transfer function. For this plot $K_3 = 10$ and K_1 has been varied (1, 2.5, 5 and 10).

17.7.4 Switched-Capacitor Filters with Reduced Area

Most of the high-performance filters, sigma–delta modulators, and data converters are based on SC techniques [11,13]. Several architectures for narrow-band applications have been reported; many of them use several paths to relax the OTA specifications. In this approach two biquadratic filters connected in parallel running at $f_s/2$ each but acting in complementary clock phases lead to filters working at twice the clock speed. Gain-compensated single-stage OTAs and double sampling techniques are also available for the design of efficient narrow-band (highly selective) filters.

For high-Q filters, the scenario is even more complex because large capacitive spreads (proportional to Q) are required. Area efficient high-Q filters require special design strategies such as judiciously use of partial positive feedback and use of slower clocks in critical filter building blocks. The use of slower clocks allows us to reduce the capacitive spread as well. The resistance of an SC resistor driven by periodic clocks is approximately given by T/C $(=1/f_sC)$. If the signal is sampled once every N-periods, the equivalent resistance increases by a factor of N [17]. The basic idea of this approach is shown in Figure 17.28 for the case $N = 4$; the frequency of the clock phases ϕ_{11} and ϕ_{22} is $f_s/4$. Each time A_2C_0 is activated, small amount of charge is taken by the capacitor controlling the losses of the lossy integrator.

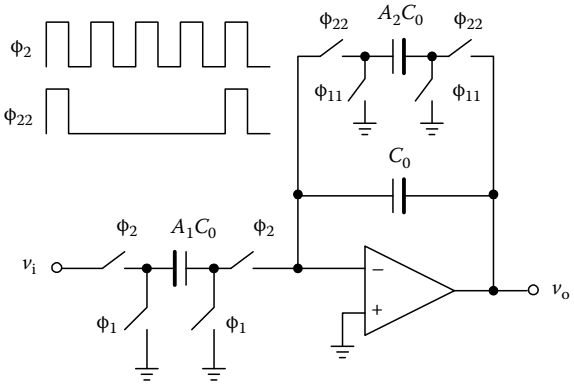


FIGURE 17.28 SC lossy integrator using a slower clock to reduce capacitor area.

If this technique is used for high-Q applications ($A_2 \ll 1$), those losses generate some spurious tones as well, but usually they are very small [17–18]. This approach results in lower capacitance spread without sacrificing sensitivity of critical filter parameters to component variations. The combination of several design techniques results in significant reduction in the total capacitance savings both silicon area and power consumption [18]. As a consequence of the use of slower clocks, additional alias components appear at integer multiples of f_s/N that may limit the filter's performance and increase the requirements on the antialias filter that precedes the SC network. It is critical to estimate the amount of alias components when using these techniques.

17.7.5 Spectrum Analyzer Using Switched-Capacitor Techniques

The growth in complexity and number of functions that can be integrated on a single chip have increased rapidly in the last years, making the testing a difficult task. Even though the majority of these systems are mainly digital, the analog section is always an important part of such mixed-mode architectures. Testing in analog circuits faces many problems; e.g., sweeping frequency and amplitude, measurement of magnitudes, and phase for a number of frequencies. Spectrum and network analyzers are routinely used for the characterization of mixed-mode systems. A cheap on-chip spectrum analyzer has been developed in Ref. [19]. Due to the flexibility of SC techniques, a gracious synchronization is ensured between the frequency of the input stimuli f_{in} and the center frequency of the circuits that perform the measurements, controlled by clocks ϕ_1 and ϕ_2.

A conceptual schematic diagram of the base-band SC network analyzer is shown in Figure 17.29. It consists of a digital frequency synthesizer, an SC sinewave generator, two VGAs, an SC BPF, and an analog-to-digital converter. The frequency synthesizer generates the master clock used as the sinusoidal generator sampling frequency as well as the nonoverlapping clock phases for the SC blocks. The sinewave generator, based on SC techniques, delivers a sinusoidal signal with a frequency of 1/16 of the master clock frequency. The amplitude of the signal coming from the sinusoidal generator is adjustable to provide the proper level to the stimuli. The output of the device under test (DUT) is bandpass filtered and conditioned by the second VGA to accommodate it to the proper input range of the ADC. The narrowband SC filter is a key building block of the spectrum-vector analyzer, its function is twofold: one, it can be centered at the center frequency f_0 to obtain the DUT transfer function; second, to select the proper frequency component (f_0, $2f_0$, $3f_0$,) for harmonic distortion characterization. The main advantage of this system is its inherent synchronization between the sinewave input signal and center frequency of the BPF.

The sinewave generator is based on SC circuits, with an oscillating frequency equal to $f_s/16$, where f_s is the clock frequency generated by the digital frequency synthesizer. Figure 17.30 shows the schematic diagram of a flexible sinewave generator using an SC circuit. It consists of a programmable gain amplifier whose preset gain stages correspond to the values of an ideal sampled and held sinewave. The SC circuit has four different gain stages, which generate a sinusoidal output with 16 steps per period generated from four capacitors and two reference voltages. The switch *PZ* sets the zero of the sinusoidal waveform.

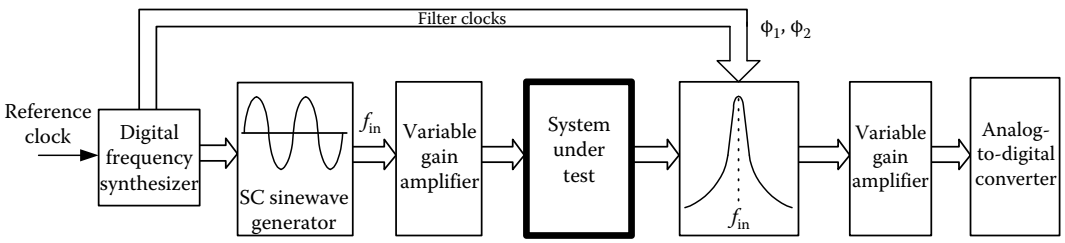


FIGURE 17.29 On-chip spectrum/vector analyzer block diagram using switched-capacitor techniques.

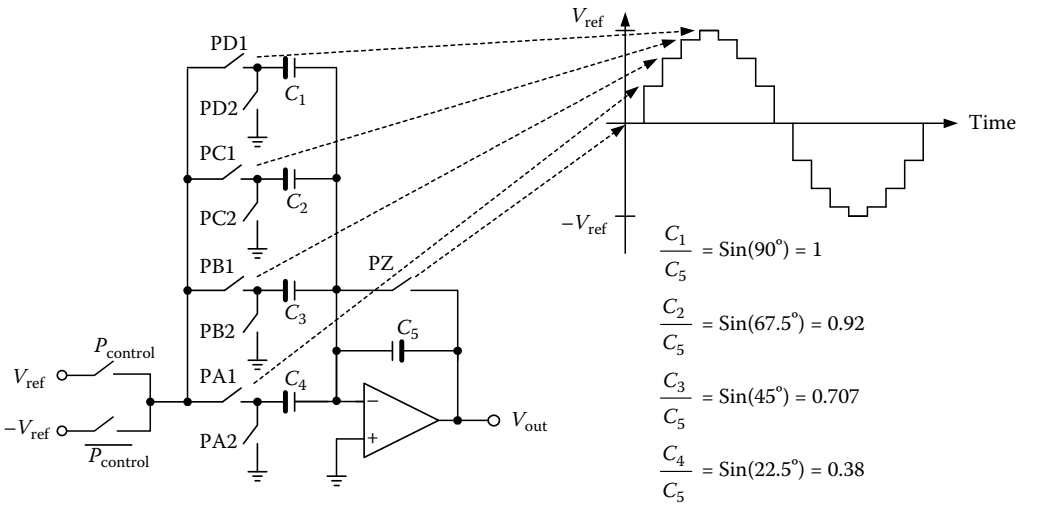


FIGURE 17.30 SC amplifier used to generate the discrete sinewave with low distortion.

The switches PA1 through PD1 are closed sequentially with each clock cycle to generate the first quarter-period of the sinusoidal waveform. Once the maximum value is obtained, the switches close in the opposite direction (from PD1 to PA1) in order to generate the second quarter-period. In the second zero crossing, $P_{control}$ switches from V_{ref} to $-V_{ref}$ so the lower half of the signal is generated. The capacitors are weighted such that the ideal values of a discrete sinewave signal are generated. The advantage of this implementation against direct digital synthesizers lies in the simplicity of the digital logic required to generate the required clock phases, resulting in a very compact implementation. The reset switch PZ also eliminates the accumulation of offset voltages. Since the peak value of the discrete sinewave signal is determined by V_{ref}, making it programmable allows obtaining an amplitude programmable sinewave generator. A modification of the structure allows generating the sinewave signal using a single reference voltage. Figure 17.31 shows the sinewave generator output for three different reference voltages: ±200 mV, ±100 mV, and ±50 mV. The oscillating frequency is 1 kHz, and the clock frequency is 16 MHz. The measured HD3 of this block is in the order of −51 dB for an input signal of 200 mV_{peak}.

17.7.6 Programmable Switched-Capacitor Oscillator

Sinewave oscillators are essential parts in many electronic systems and in a host of applications. Integrating the oscillator with the other circuit blocks on a single chip makes it easy and reliable to implement several applications including built-in testing. Among the various types of oscillator, a BPF-based oscillator is an attractive and practical implementation due to its many advantages such as the programmability of the oscillation frequency by means of changing the center frequency (f_0) of the BPF, and the fact that the oscillation amplitude can be controlled with the help of a comparator. In base-band applications, BPF can be implemented with conventional SC design techniques that are preferred because of its accuracy, simple implementation, and reduced sensitivity to process and temperature variations. A block diagram of a BPF-based SC oscillator is shown in Figure 17.32 [20].

For highly linear oscillator, conventional approaches to minimize the frequency harmonics of the SC BPF-based oscillator requires a high-quality-factor (Q-factor) BPF, which involves high capacitor spread and, hence, leads to large silicon area. Also, improving the linearity by increasing Q-factor is not that efficient considering the fact that the nth-order harmonic distortion (HD_n) can be approximated as

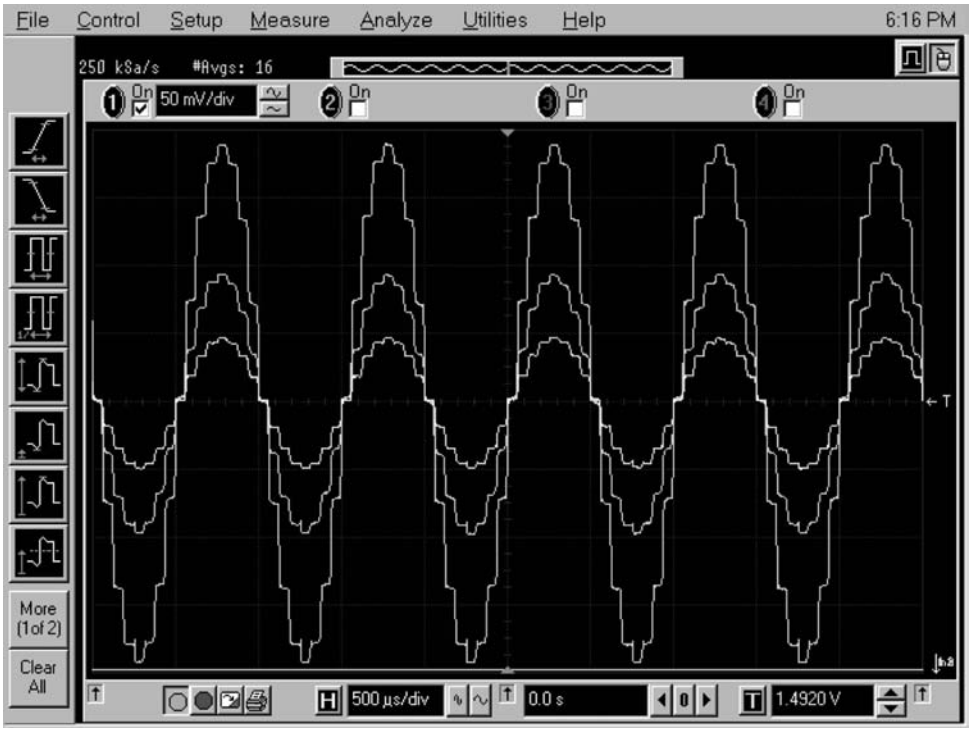


FIGURE 17.31 Measured output signal of the sinewave generator.

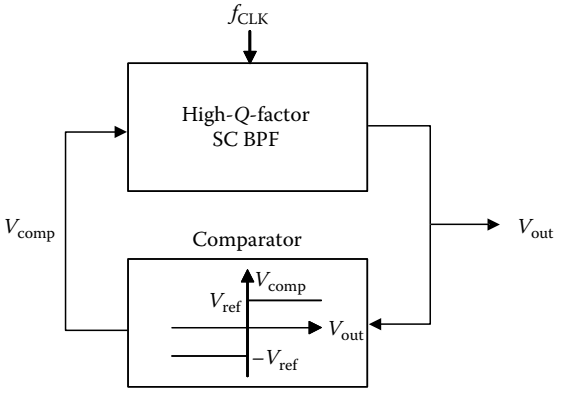


FIGURE 17.32 Block diagram of conventional BPF-based SC oscillator.

$1/(n^2Q)$, where Q is a Q-factor of BPF in second-order BPF-based oscillator. If HD3 of -60 dB is needed, then the required Q-factor becomes more than 100 that may not be practical for an IC realization.

The linearity can be significantly improved without requiring a high-Q-factor BPF if a technique based on nonlinear shaping of the frequency spectrum is used. For this purpose, a comparator must exhibit multilevel outputs. The four-level square wave with a certain condition (as depicted for $f(t)$ in Figure 17.33) rejects the third-order harmonic component. Implementation of a multilevel comparator

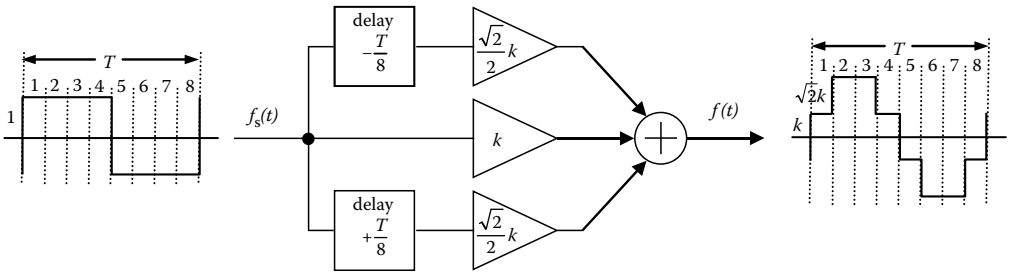


FIGURE 17.33 Conceptual diagram of a four-level quasi-sinusoidal wave generator.

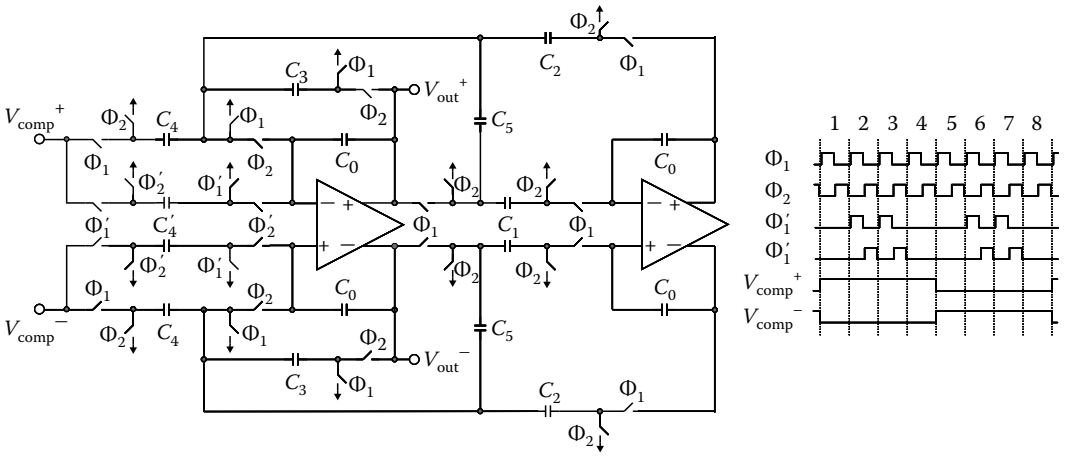


FIGURE 17.34 SC BPF implementation with an embedded FIR filter.

can be explained as adding finite-impulse-response (FIR) filter to a conventional comparator as shown in Figure 17.33. In this figure, $f_s(t)$ is an output of a conventional comparator and $f(t)$ is an output of FIR filter. A transfer function of an FIR filter in discrete-time domain, assuming that $T/8$ is one sampling period, can be expressed

$$H(z) = \frac{F(z)}{F_s(z)} = \frac{k}{\sqrt{2}} z(1 + \sqrt{2}z^{-1} + z^{-2}) \tag{17.66}$$

Equation 17.66 shows zeros at $z = 0$, $e^{\pm j3\pi/4}$. Note that $z = e^{\pm j3\pi/4}$ corresponds to the third-order harmonic frequency since $8/T$ is used for a sampling frequency. FIR filter can be easily embedded into SC BPF with the minimum cost, because BPF readily has an adder and a multiplier at its input. Figure 17.34 shows the BPF with the embedded FIR filter. In Figure 17.34, C_4' should be $\sqrt{2}$ C_4 for the cancellation of third-order harmonic, and all other capacitors should be determined by the requirements as of a conventional BPF.

The SC BPF-based oscillator was fabricated using CMOS 0.35 μm technology. BPF has a center frequency of $f_0 = 10$ MHz, a Q-factor of 10, and a master clock frequency of $f_C = 80$ MHz. The measured frequency spectrums for both oscillators are shown in Figure 17.35. Over −54 dB of HD3 for the SC BPF-based with FIR filter is achieved, which is 20 dB smaller than the conventional oscillator. HD3 improvement of the proposed oscillator is mainly determined by the accuracy of the multiplying factor

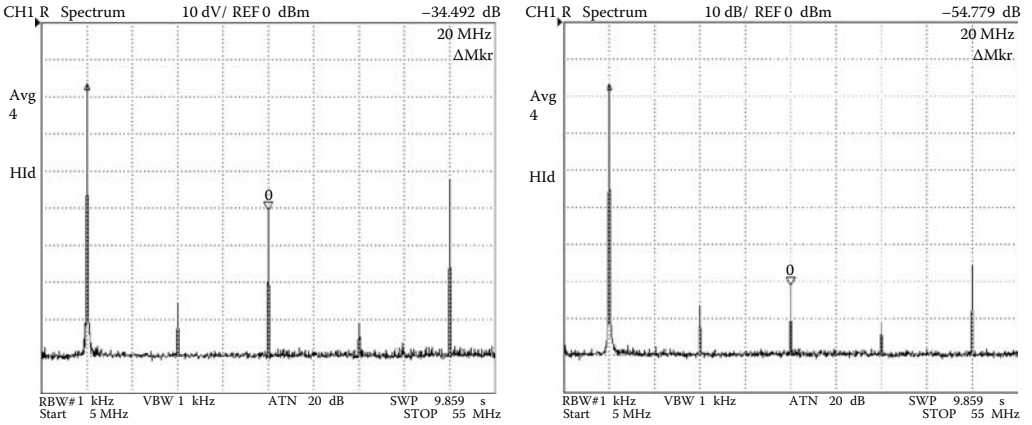


FIGURE 17.35 Measured frequency spectrum of the SC oscillators.

($\sqrt{2}$ in this case). In practical implementation, the ratio of 4:3 (=1.33) can be used for approximation of $\sqrt{2}$ (=1.4142), which limits the amount of harmonic cancellation. The implementation of the FIR filter requires minimum additional area (10% of the total area).

References

1. P. E. Allen and E. Sánchez-Sinencio, *Switched-Capacitor Circuits*. New York: Van Nostrand, 1984.
2. R. Unbehauen and A. Cichocki, *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems*. Berlin: Springer-Verlag, 1989.
3. R. W. Brodersen, P. R. Gray, and D. A. Hodges, MOS switched-capacitor filters, *Proceedings on IEEE*, 657, 61–75, January 1979.
4. R. Castello and P. R. Gray, A high-performance micropower switched-capacitor filter, *IEEE Journal on Solid-State Circuits*, SC-20, 1122–1132, December 1985.
5. E. Sánchez-Sinencio, J. Silva-Martinez, and R. L. Geiger, Biquadratic SC filters with small GB effects, *IEEE Transactions on Circuits Systems*, 31, 876–884, October 1984.
6. R. Gregorian and G. Temes, *Analog MOS Integrated Circuits*, New York: Wiley, 1986.
7. E. Vittoz, Very low power circuit design: Fundamentals and limits, in *Proc. IEEE/ISCAS'93*, Chicago, IL, May 1993, 1451–1453.
8. R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, New York: McGraw-Hill, 1990.
9. R. Castello and P. R. Gray, Performance limitations in switched-capacitor filters, *IEEE Transactions Circuits Systems*, CAS-32, 865–876, September 1985.
10. K. Nagaraj, A parasitic insensitive area-efficient approach for realizing very large time constants in switched-capacitor filters, *IEEE Transactions on Circuits Systems*, 36, 1210–1216, September 1989.
11. E. Sánchez-Sinencio and A. Andreou, editors, *Low-Voltage/Low-Power Integrated Circuits and Systems*, New York: IEEE Press, 1999.
12. P. R. Gray and R. Meyer, MOS operational amplifier design—a tutorial overview, *IEEE Journal on Solid-State Circuits*, 17, 6, 969–982, December 1982.
13. D. A. Johns and K. W. Martin, *Analog Integrated Circuit Design*, John Wiley & Sons, New York, 1997.
14. A. M. Abo and P. R. Gray, A 1.5 V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter, *IEEE Journal on Solid-State Circuits*, 34, 5, 599–606, May 1999.
15. J. Crols and M. Steyaert, Switched-opamp: An approach to realize full CMOS switched-capacitor circuits at very low power supply voltages, *IEEE Journal on Solid-State Circuits*, 29, 8, 936–942, August 1994.

16. D. Hernandez-Garduno and J. Silva-Martinez, Continuous-time common-mode feedback for high-speed switched-capacitor networks, *IEEE Journal of Solid-State Circuits*, 40, 1610–1617, August 2005.
17. J. L. Ausin et al., Switched-capacitor circuits with periodical nonuniform individual sampling, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 50, 8, 404–414. August 2003.
18. J. Adut, M. Rocha-Perez, and J. Silva-Martinez, A 6th order broadband 10.7 MHz SC ladder filter, *IEEE Transactions on Circuits and Systems, part I*. 53, 1625–1635, August 2006.
19. M. Mendez-Rivera, J. Silva-Martinez, E. Sanchez-Sinencio, and A. Valdes-Garcia, An on-chip spectrum analyzer for analog built-in testing, *Journal of Electronic Testing: Theory and Applications*, 21, 205–219, February 2005.
20. S. W. Park, José L. Ausín, F. Bahmani, and E. Sánchez-Sinencio, Nonlinear shaping SC oscillator with enhanced linearity, *IEEE Journal of Solid-State Circuits*, 42, 11, 2421–2431, November 2007.

III

Digital Filters

Rashid Ansari
University of Illinois at Chicago

A. Enis Cetin
Bilkent University

18 **FIR Filters** *M. H. Er, Andreas Antoniou, Yong Ching Lim, and Tapio Saramäki* **18**-1
Properties of FIR Filters • Windowing Techniques • References • Design of FIR Filters by Optimization • References • Design of Computationally Efficient FIR Filters Using Periodic Subfilters as Building Blocks • References

19 **IIR Filters** *Sawasd Tantaratana, Chalie Charoenlarpnopparut, Phakphoom Boonyanant, and Yong Ching Lim* **19**-1
Properties of IIR Filters • References • Design of IIR Filters • References • Lattice Digital Filters • References

20 **Finite Wordlength Effects** *Bruce W. Bomar* **20**-1
Number Representation • Fixed-Point Quantization Errors • Floating-Point Quantization Errors • Roundoff Noise • Limit Cycles • Overflow Oscillations • Coefficient Quantization Error • Realization Considerations • References

21 **VLSI Implementation of Digital Filters** *Joseph B. Evans and Timothy R. Newman* **21**-1
Introduction • General-Purpose Processors • Special-Purpose Implementations • Programmable Logic Implementations • References • Further Information

22 **Two-Dimensional FIR Filters** *Rashid Ansari and A. Enis Cetin* **22**-1
Introduction • Preliminary Design Considerations • General Design Methods for Arbitrary Specifications • Special Design Procedure for Restricted Classes • 2-D FIR Filter Implementation • Two-Dimensional Filter Banks • Acknowledgments • References • Further Information

23 **Two-Dimensional IIR Filters** *A. G. Constantinides and Xiaojian Xu* **23**-1
Introduction • Transformation Techniques • Design of Separable Product Filters • Design of Circularly Symmetric Filters • Design of 2-D IIR Filters by Optimization • References

24 1-D Multirate Filter Banks *Nick G. Kingsbury and David B. H. Tay* **24**-1
Introduction: Why Use Filter Banks? • 2-Band Filter Bank • Multirate Filtering • Binary Filter Trees • Wavelets and Scaling Functions • Good FIR Filters and Wavelets • IIR Filter Banks • Polyphase Representations and the Lifting Scheme • Nonlinear Filter Banks • *M*-Band Filter Banks • Hilbert Pairs of Filter Banks (the Dual Tree) • References

25 Directional Filter Banks *Jose Gerardo Rosiles and Mark J. T. Smith* **25**-1
Introduction • Basic Theory of 2-D Multirate Systems • 2-D Maximally Decimated Filter Banks • Bamberger Directional Filter Bank • Design of 2-D Two-Channel Fan Filter Banks • Undecimated Directional Filter Bank • Bamberger Pyramids • Applications • Closing Remarks • References

26 Nonlinear Filtering Using Statistical Signal Models *Kenneth E. Barner, Tuncer C. Aysal, and Gonzalo R. Arce* **26**-1
Introduction • *M*-Estimation • *LM*-Estimation • Applications of Robust Nonlinear Estimators/Filters • Concluding Remarks and Future Directions • References

27 Nonlinear Filtering for Image Denoising *Nasir M. Rajpoot, Zhen Yao, and Roland G. Wilson* **27**-1
Filtering for Image Denoising • Polar Cosine Transform • Planelets • 3-D Wavelet Packets • Discussion and Conclusion • References

28 Video Demosaicking Filters *Bahadir K. Gunturk and Yucel Altunbasak* **28**-1
Introduction • Imaging Model • Demosaicking Methods • Demosaicking of Video and Super-Resolution Reconstruction • Related Research Problems • Evaluation of Demosaicking Algorithms • Conclusions and Future Directions • Acknowledgments • References

18

FIR Filters

M. H. Er
Nanyang Technological University

Andreas Antoniou
University of Victoria

Yong Ching Lim
National University of Singapore

Tapio Saramäki
Tampere University of Technology

18.1 Properties of FIR Filters **18**-1
Linear Phase Property • Frequency Response of Linear Phase FIR Filters • Locations of Zeros of Linear Phase FIR Filters
18.2 Windowing Techniques **18**-6
References **18**-15
18.3 Design of FIR Filters by Optimization **18**-15
Equiripple FIR Filters • Problem Formulation
References **18**-37
18.4 Design of Computationally Efficient FIR Filters Using Periodic Subfilters as Building Blocks **18**-38
Frequency-Response Masking Approach • Multistage Frequency-Response Masking Approach • Design of Narrowband Filters • Design of Wideband Filters • Generalized Designs
References **18**-60

18.1 Properties of FIR Filters

M.H. Er

18.1.1 Linear Phase Property

The finite-impulse response (FIR) filter is characterized by a unit-sample response that has a finite duration. One of the advantages of FIR filters compared to their infinite-impulse response (IIR) counterparts is that FIR filters can be designed with exactly linear phase. Linear phase response is important for applications where phase distortion due to n onlinear phase can degrade performance, such as in data transmission and television applications.

A FIR causal filter can be characterized by the transfer function [1]

$$H(z) = \sum_{n=0}^{N-1} h(nT) z^{-n} \tag{18.1}$$

where

$h(nT)$ is the impulse response of the filter
N is the filter length
T is the sampling interval

Using the relationship that

$$H(z) = \frac{Y(z)}{X(z)} \tag{18.2}$$

the difference equation of a FIR filter can be obtained by taking the inverse Z-transform of Equation 18.1, that is

$$y(iT) = \sum_{n=0}^{N-1} h(nT)X(iT - nT) \tag{18.3}$$

which says that the current output of a FIR causal filter is the weighted sum of the current and past inputs. The weighting coefficients are given by the impulse response of the filter.

From Equation 18.1, the frequency response can be obtained by replacing $z = e^{j\omega T}$ as

$$H(e^{j\omega T}) = \sum_{n=0}^{N-1} h(nT)e^{-j\omega nT} = M(\omega)e^{j\phi(\omega)} \tag{18.4}$$

where $M(\omega)$ and $\phi(\omega)$ are the magnitude and phase responses, respectively, defined as

$$M(\omega) = \left|H(e^{j\omega T})\right| \tag{18.5a}$$

$$\phi(\omega) = \arg\ H(e^{j\omega T}) \tag{18.5b}$$

The phase delay and group (time) delay functions of a filter are defined as

$$\tau_p = -\frac{\phi(\omega)}{\omega} \tag{18.6}$$

and

$$\tau_g = -\frac{d\phi(\omega)}{d\omega} \tag{18.7}$$

respectively. Filters for which τ_p and τ_g are independent of frequency are referred to as constant time delay or linear phase filters. Hence, the phase response of a linear phase filter is given by

$$\phi(\omega) = -\tau\omega, \quad -\pi < \omega < \pi \tag{18.8}$$

where τ is a constant phase delay in samples.

From Equations 18.4, 18.5b and 18.8, the phase response can be expressed as

$$\phi(\omega) = -\tau\omega = \tan^{-1}\frac{-\sum_{n=0}^{N-1} h(nT)\sin(\omega nT)}{\sum_{n=0}^{N-1} h(nT)\cos(\omega nT)} \tag{18.9}$$

Consequently,

$$\tan(\omega\tau) = \frac{\sum_{n=0}^{N-1} h(nT)\sin(\omega nT)}{\sum_{n=0}^{N-1} h(nT)\cos(\omega nT)} \tag{18.10}$$

Using the definition $\tan(\omega\tau) = \sin(\omega\tau)/\cos(\omega\tau)$, Equation 18.10 can be reexpressed as

$$\sum_{n=0}^{N-1} h(nT)\sin(\omega\tau - \omega nT) = 0 \tag{18.11}$$

It can be shown [1] that a solution to Equation 18.11 is given by

$$\tau = \frac{(N-1)T}{2} \tag{18.12}$$

and

$$h(nT) = h[(N-1-n)T], \quad 0 \le n \le N-1 \tag{18.13}$$

Hence, FIR filters can be designed to have constant phase and group delays if the conditions of Equations 18.12 and 18.13 are satisfied. The symmetry property of Equation 18.13 can also lead to efficient filter realizations.

In applications where only constant group delay is needed, the phase response can have the form

$$\phi(\omega) = \phi_0 - \tau\omega \tag{18.14}$$

where ϕ_0 is a constant. With $\phi_0 = \pm\pi/2$, it can be shown [1] that the impulse response is of the form

$$h(nT) = -h[(N-1-n)T], \quad 0 \le n \le N-1 \tag{18.15}$$

In this case, the impulse response exhibits antisymmetrical property.

18.1.2 Frequency Response of Linear Phase FIR Filters

The frequency response of a causal linear phase FIR filter can be simplified to some simple forms using Equations 18.13 and 18.15 and the values of N as follows:

(1) *Symmetric impulse response and N = odd.* In this case,

$$H(e^{j\omega T}) = \sum_{n=0}^{(N-3)/2} h(nT)e^{-j\omega nT} + h\left[\frac{(N-1)T}{2}\right]e^{-j\omega(N-1)T/2} + \sum_{n=(N+1)/2}^{N-1} h(nT)e^{-j\omega nT} \tag{18.16}$$

Using Equation 18.13, letting $m = N-1-n$ and changing the limits of summation, and finally letting $m = n$, the last summation in Equation 18.16 can be reexpressed as

$$\sum_{n=(N+1)/2}^{N-1} h(nT)e^{-j\omega nT} = \sum_{n=0}^{(N-3)/2} h(nT)e^{-j\omega(N-1-n)T} \tag{18.17}$$

Substituting Equation 18.17 into Equation 18.16, one obtains

$$H(e^{j\omega T}) = \sum_{n=0}^{(N-3)/2} h(nT)\left[e^{-j\omega nT} + e^{-j\omega(N-1-n)T}\right] + h\left[\frac{(N-1)T}{2}\right] e^{-j\omega(N-1)T/2} \tag{18.18}$$

Factoring $e^{-j\omega(N-1)T/2}$ in Equation 18.18 and letting $k = (N-1)/2 - n$, Equation 18.18 can be reexpressed as

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{(N-1)/2} h\left[\left(\frac{N-1}{2} - k\right)T\right](e^{j\omega kT} + e^{-j\omega kT}) + h\left[\frac{(N-1)T}{2}\right]\right\} \tag{18.19}$$

Using the property that $e^{j\theta} + e^{-j\theta} = 2\cos\theta$, Equation 18.19 can be simplified to the form

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{(N-1)2} 2h\left[\left(\frac{N-1}{2} - k\right)T\right]\cos(\omega kT) + h\left[\frac{(N-1)T}{2}\right]\right\} \tag{18.20}$$

Letting $a(o) = h[(N-1)T/2]$ and $a(k) = 2h[((N-1)/2 - k)T]$, Equation 18.20 can be simplified further to

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left[\sum_{k=0}^{(N-1)/2} a(k)\cos(\omega kT)\right] \tag{18.21}$$

(2) *Symmetric impulse response and N = even*. For this case, the frequency response takes the form

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left\{\sum_{k=0}^{N/2-1} 2h(kT)\cos\left[\omega\left(\frac{N}{2} - k - \frac{1}{2}\right)T\right]\right\} \tag{18.22}$$

Letting $b(k) = 2h[N/2 - k)T]$, $k = 1, 2, \ldots, N/2$, Equation 18.22 can be expressed as

$$H(e^{j\omega T}) = e^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{N/2} b(k)\cos\left[\omega\left(k - \frac{1}{2}\right)T\right]\right\} \tag{18.23}$$

An interesting feature of this frequency response is that $H(e^{j\omega T})$ is always equal to zero for $\omega = \pi$, independent of $b(k)$. This implies that high-pass filter characteristics cannot be realized with this type of filter.

(3) *Antisymmetric impulse response and N = odd*. For this case, the derivation of the frequency response is the same as that in (1) except that the cosine summations are replaced by the sine summations multiplied by j because of Equation 18.15. Hence, the frequency response is given by

$$H(e^{j\omega T}) = je^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{(N-1)/2} 2h\left[\left(\frac{N-1}{2} - k\right)T\right]\sin(\omega kT) + h\left[\frac{(N-1)T}{2}\right]\right\} \tag{18.24}$$

It should be noted that for odd values of N, Equation 18.15 requires that $h[(N-1)T/2] = 0$. Letting $c(k) = 2h[((N-1)/2 - k)T]$, $k = 1, 2, \ldots, (N-1)/2$, Equation 18.24 becomes

TABLE 18.1 Frequency Response of Linear Phase FIR Filters

$h(nT)$	N	$H(e^{j\omega T})$
Symmetrical	Odd	$e^{-j\omega(N-1)T/2}\sum_{k=0}^{(N-1)/2} a(k)\cos(\omega kT)$
	Even	$e^{-j\omega(N-1)T/2}\sum_{k=1}^{N/2} b(k)\cos\left[\omega\left(k-\frac{1}{2}\right)T\right]$
Antisymmetrical	Odd	$je^{-j\omega(N-1)T/2}\sum_{k=1}^{(N-1)/2} c(k)\sin(\omega kT)$
	Even	$je^{-j\omega(N-1)T/2}\sum_{k=1}^{N/2} d(k)\sin\left[\omega\left(k-\frac{1}{2}\right)T\right]$

Where $a(o)=h\left[\frac{(N-1)T}{2}\right]$, $a(k)=c(k)=2h\left[\left(\frac{N-1}{2}-k\right)T\right]$, $b(k)=d(k)=2h\left[\left(\frac{N}{2}-k\right)T\right]$.

$$H(e^{j\omega T})=je^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{(N-1)/2} c(k)\sin(\omega kT)\right\} \tag{18.25}$$

A notable feature of this frequency response is that at frequencies $\omega=0$ and $\omega=\pi$, the frequency response is always zero, independent of $c(k)$.

(4) *Antisymmetric impulse response and N = even*. For this case, the frequency response is the same as that in (2) except the cosine summations become sine summations multiplied by j as follows:

$$H(e^{j\omega T})=je^{-j\omega(N-1)T/2}\left\{\sum_{k=0}^{N/2-1} 2h(kT)\sin\left[\omega\left(\frac{N}{2}-k-\frac{1}{2}\right)T\right]\right\} \tag{18.26}$$

Letting $d(k)=2h[N/2-k)T]$, $k=1, 2, \ldots, N/2$, Equation 18.26 becomes

$$H(e^{j\omega T})=je^{-j\omega(N-1)T/2}\left\{\sum_{k=1}^{N/2} d(k)\sin\left[\omega\left(k-\frac{1}{2}\right)T\right]\right\} \tag{18.27}$$

In this case, the frequency response is zero at $\omega=0$, independent of $d(k)$.

In summary, the frequency responses of the four possible types of FIR filters with linear phase are given in Table 18.1.

18.1.3 Locations of Zeros of Linear Phase FIR Filters

The symmetric and antisymmetric conditions of the impulse response given by Equations 18.13 and 18.15 impose certain constraints on the zeros of the transfer function $H(z)$ [2]. For the case where N is an odd value, $H(z)$ can be written as

$$H(z)=z^{-(N-1)/2}\sum_{k=0}^{(N-1)/2}\frac{a(k)}{2}(z^k\pm z^{-k}) \tag{18.28}$$

where the $\pm$ sign corresponds to symmetry and antisymmetry in the impulse response respectively, and $a(o)$ and $a(k)$ are defined in Table 18.1.

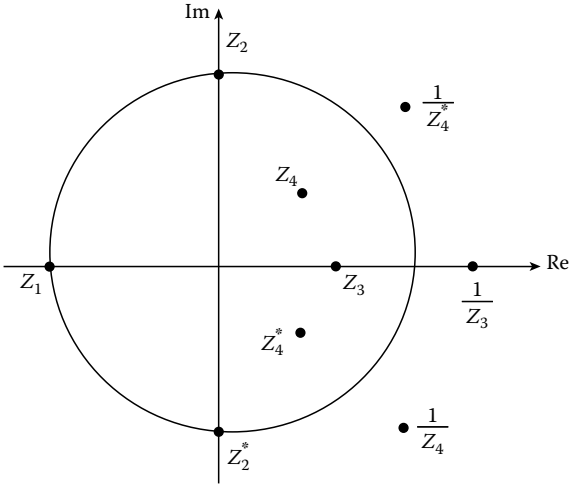


FIGURE 18.1 Typical zero positions of linear phase filters.

Substituting z^{-1} for z in Equation 18.28, one obtains

$$H(z^{-1}) = z^{(N-1)/2} \sum_{k=0}^{(N-1)/2} \frac{a(k)}{2}(z^{-k} \pm z^{k}) \tag{18.29}$$

It follows from Equations 18.28 and 18.29 that

$$H(z^{-1}) = \pm z^{(N-1)} H(z) \tag{18.30}$$

Equation 18.30 shows that $H(z)$ and $H(z^{-1})$ are identical to within a delay of $(N-1)$ samples and a multiplier of ± 1. Thus, the zeros of $H(z^{-1})$ are identical to the zeros of $H(z)$. Therefore, if $z_i = r_i e^{j\phi i}$ is a zero of $H(z)$, then $z_i^{-1} = (1/r_i)e^{-j\phi i}$ must also be a zero of $H(z)$. This has the following implications on the zero locations:

1. If $r_i = 1$ and $\phi_i = 0$ or π, then the zeros lie at either $z = +1$ or $z = -1$. In these cases, the zero is its own complex conjugate.
2. If $r_i = 1$ and $\phi_i \neq 0$ or π, then the zeros of $H(z)$ that are on the unit circle are also zeros of $H(z^{-1})$ that are on the unit circle. Hence, the zeros occur in complex conjugate pairs on the unit circle.
3. If $r_i \neq 1$ and $\phi_i = 0$ or π, then the zeros are real and occur in reciprocal pairs on the unit circle.
4. If $r_i \neq 1$ and $\phi_i \neq 0$ or π, then the zeros occur in quadruplets with complex conjugate reciprocal pairs off the unit circle.

Figure 18.1 shows the possible types of zeros for linear phase FIR filters.

18.2 Windowing Techniques

M.H. Er

Windowing is one of the earliest techniques for designing FIR filters [3,4]. The technique is simple because the filter coefficients can be obtained in closed form without the need for solving complex optimization problems as in some other sophisticated FIR design techniques. Hence, the design time is very short and the technique remains an attractive tool for FIR filter design.

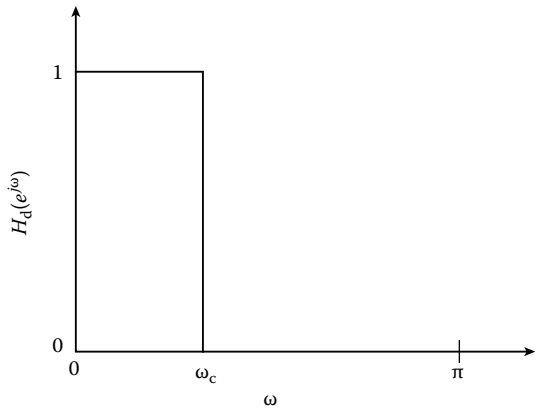


FIGURE 18.2 An ideal low-pass filter specification.

To understand the windowing technique, first consider the process of obtaining a finite-length impulse response by truncating an infinite-duration impulse response sequence. Suppose $H_d(e^{j\omega T})$ is an ideal desired low-pass response with cutoff frequency ω_c. As the frequency response of an FIR filter is a periodic function, it can be expressed as a Fourier series as

$$H_d(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h_d(nT)e^{-j\omega nT} \tag{18.31}$$

where

$$h_d(nT) = \frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} H_d(e^{j\omega T})e^{j\omega nT}d\omega \tag{18.32}$$

In general, $H_d(e^{j\omega T})$ is piecewise constant with a certain passband and stopband and with discontinuities at the boundaries between bands. Hence, the impulse sequence $h_d(nT)$ is of infinite duration. For example, for the ideal low-pass response shown in Figure 18.2, the corresponding impulse response sequence is given by

$$h_d(nT) = \frac{\omega_c T}{\pi}\left(\frac{\sin \omega_c nT}{\omega_c nT}\right), \quad -\infty \le n \le \infty \tag{18.33}$$

It is clear that Equation 18.33 is a noncausal IIR filter. Also it is unstable and therefore unrealizable.

The rectangular window. One way to obtain a finite-duration causal impulse response is to simply truncate $h_d(nT)$ and introduce sufficient delay to obtain a causal impulse response, i.e., define

$$h(nT) = \begin{cases} h'_d(nT) & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.34}$$

where $h'_d(nT)$ is a delay version of $h_d(nT)$.

This can be represented as the product of the desired impulse response and a finite-duration window $w_r(nT)$, i.e.,

$$h(nT) = h'_d(nT)w_r(nT) \tag{18.35}$$

where $w_r(nT)$ is the rectangular window function defined as

$$w_r(nT) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.36}$$

Let $\theta = \omega T$ and using the fact that multiplication of two discrete-time sequences corresponds to a convolution of their Fourier transforms. Hence,

$$H(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H'_{\mathrm{d}}(e^{j\theta})W_{\mathrm{r}}(e^{j(\omega-\theta)})\mathrm{d}\theta \tag{18.37}$$

where $W_{\mathrm{r}}(e^{j\theta})$ is the spectrum of the rectangular window.

Since the two functions in the integral are periodic, a circular convolution results and the limits of integration are taken over one period. Thus the frequency response $H(e^{j\omega})$ will be a "smeared" version of the desired response $H'_{\mathrm{d}}(e^{j\omega})$ and the discontinuities in the desired frequency response become transition bands of $H(e^{j\omega})$. To understand this, it is instructive to examine the frequency response for the causal rectangular window, that is,

$$\begin{aligned} W_{\mathrm{r}}(e^{j\omega T}) &= \sum_{n=0}^{N-1} e^{-j\omega nT} \\ &= e^{-j\omega(N-1)T/2}\frac{\sin(\omega NT/2)}{\sin(\omega T/2)} \end{aligned} \tag{18.38}$$

The spectrum $W_{\mathrm{r}}(e^{j\omega T})$ for $N = 31$ is shown in Figure 18.3. The spectrum $W_{\mathrm{r}}(e^{j\omega T})$ has two features that are worth noting, the mainlobe width and the sidelobe amplitude. The mainlobe width is defined as the distance between the two points closest to $\omega = 0$, where $W_{\mathrm{r}}(e^{j\omega T})$ is zero. For a rectangular window, the mainlobe width is equal to $4\pi/N$. The maximum sidelobe amplitude for $W_{\mathrm{r}}(e^{j\omega T})$ is equal to approximately -13 dB relative to the maximum value at $\omega = 0$.

Figure 18.4 shows the log–magnitude response of applying a 31-point rectangular window to approximate an ideal low-pass filter with a cutoff frequency equal to $\pi/4$. It can be seen that the sharp transition in the ideal response at $\omega = \omega_{\mathrm{c}}$ has been converted into a gradual transition. Also, in the passband a series of overshoots and undershoots occur, and in the stopband, where the desired response is zero, the FIR filter has a nonzero response. These are the results of the convolution between $W_{\mathrm{r}}(e^{j\omega T})$ and $H_{\mathrm{d}}(e^{j\omega T})$. The mainlobe of $W_{\mathrm{r}}(e^{j\omega T})$ causes the smearing of the desired response and the sidelobes of $W_{\mathrm{r}}(e^{j\omega T})$ appear as overshoots and undershoots to the desired response. It is interesting to note that there will

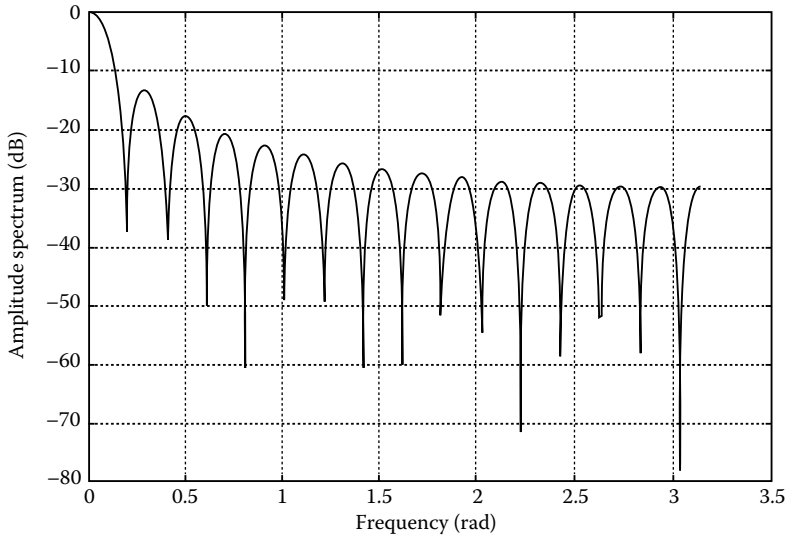


FIGURE 18.3 Fourier transform of the rectangular window.

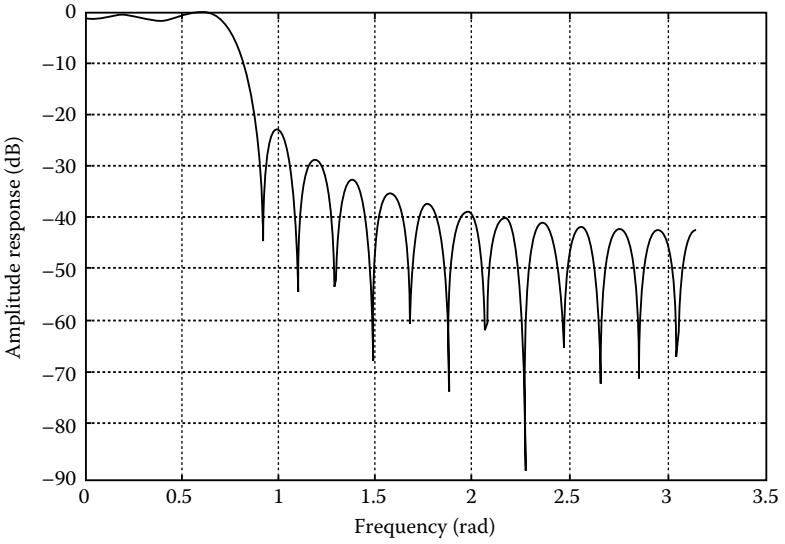


FIGURE 18.4 Magnitude response of low-pass FIR filter design using a 31-point rectangular window.

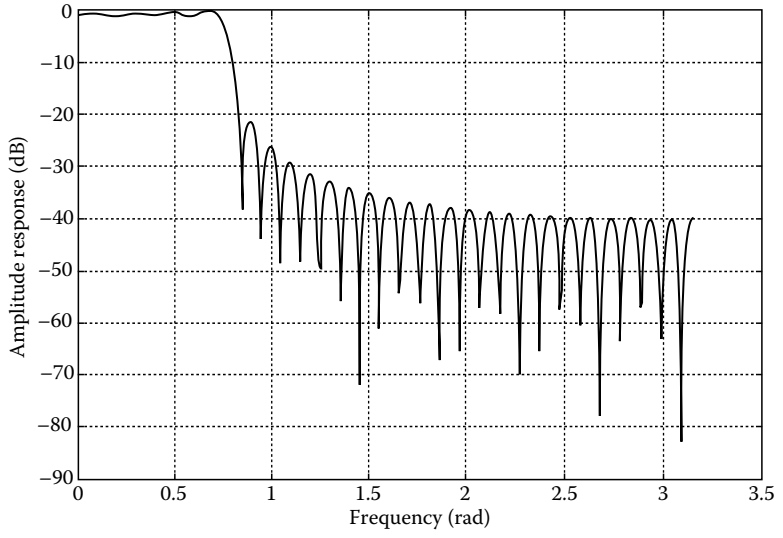


FIGURE 18.5 Magnitude response of low-pass FIR filter design using a 61-point rectangular window.

always be oscillations in the function $H(e^{j\omega T})$ in the vicinity of the steep transitions in $H'_{\mathrm{d}}(e^{j\omega T})$, no matter how large the value of N as shown in Figure 18.5 for $N = 61$. This result is known as the Gibbs phenomenon [2] in the theory of Fourier series.

To reduce the oscillations in $H(e^{j\omega T})$, other window functions having spectra exhibiting smaller sidelobes must be used. To understand how the form of windows should be selected, it is observed that the sidelobes of the rectangular window represent the high-frequency components and are due to the sharp transitions from one to zero at the edges of the window. Therefore, the amplitudes of these sidelobes can be reduced by replacing the sharp transitions by more gradual ones. Some of the most frequently used window functions are described below.

The Bartlett window. The Bartlett window, also known as the triangular window, is defined as

$$w_t(nT) = \begin{cases} \frac{2n}{N-1} & 0 \le n \le \frac{N-1}{2} \\ 2 - \frac{2n}{N-1} & \frac{N-1}{2} \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.39}$$

The spectrum $W_t(e^{j\omega T})$ is shown in Figure 18.6. As expected, the sidelobe level is smaller than that of the rectangular window, being reduced from -13 to -25 dB relative to the maximum. However, the mainlobe width is now $8\pi/N$, twice that of the rectangular window. Hence, there is a trade-off between mainlobe width and sidelobe level.

Figure 18.7 illustrates the FIR low-pass magnitude response obtained by using the Bartlett window. Comparing Figure 18.7 to Figure 18.4, it is observed that the Bartlett window produces a smoother magnitude response.

The Hanning window. The Hanning window, also known as the raised-cosine window, is given by

$$w_c(nT) = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right] & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.40}$$

The amplitude spectrum of this window is shown in Figure 18.8 for $N = 31$. The magnitude of the first sidelobe level is -31 dB, down with respect to the peak value at $\omega = 0$. Comparing to the triangular window, there is an improvement of 6 dB. Since the mainlobe widths of both windows are the same, the Hanning window is preferred over the triangular one.

The amplitude response of the FIR low-pass filter with $\omega_c = \pi/4$ produced by applying the Hanning window with $N = 31$ is shown in Figure 18.9. The largest peak in the stopband is now reduced to -44 dB relative to the passband level.

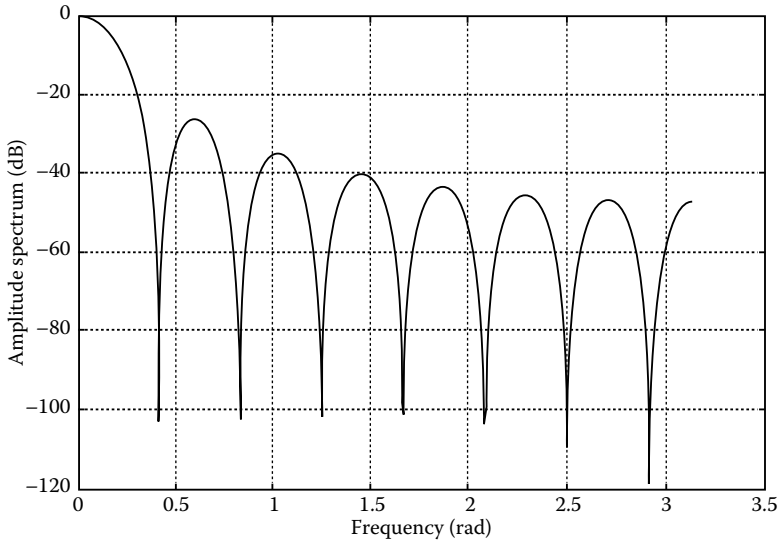


FIGURE 18.6 Fourier transform of the Bartlett window.

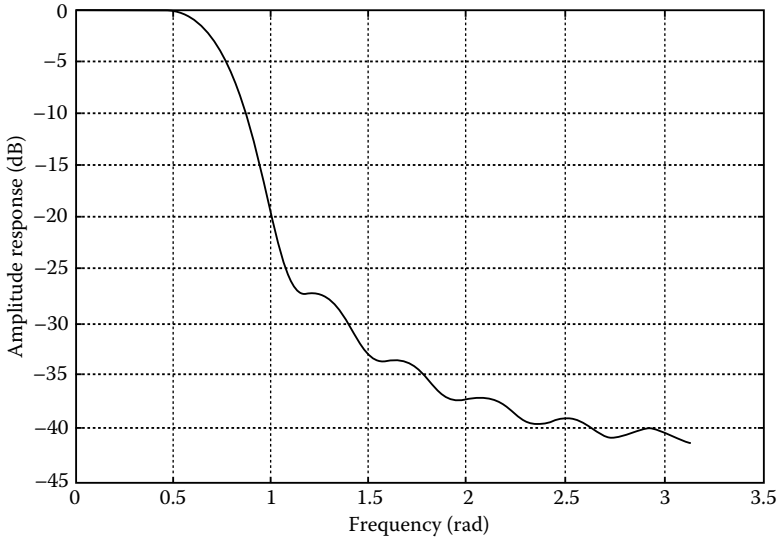


FIGURE 18.7 Magnitude response of low-pass FIR filter design using a 31-point Bartlett window.

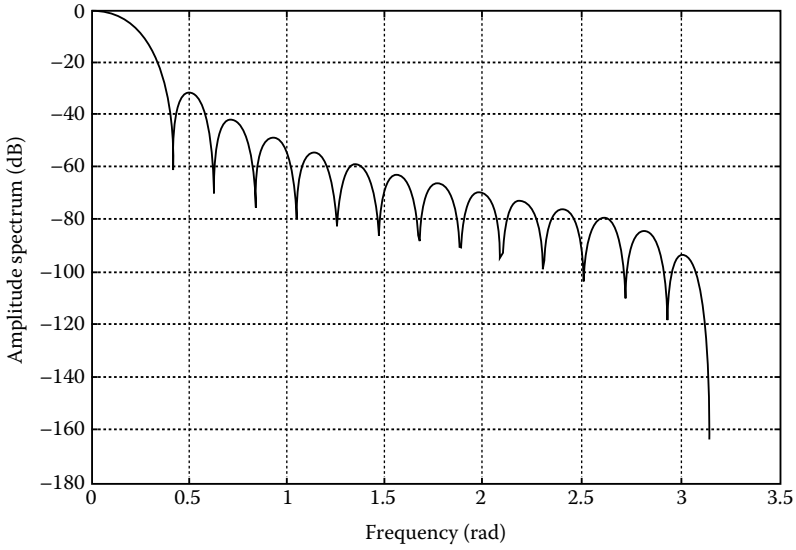


FIGURE 18.8 Fourier transform of the Hanning window.

The Hamming window. The Hamming window is given by

$$w_{\mathrm{h}}(nT) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.41}$$

Figure 18.10 shows the amplitude spectrum of the Hamming window, The magnitude of the highest sidelobe is about −41 dB, a reduction of 10 dB relative to the Hanning window. This reduction is achieved at the expense of higher sidelobes at the higher frequencies.

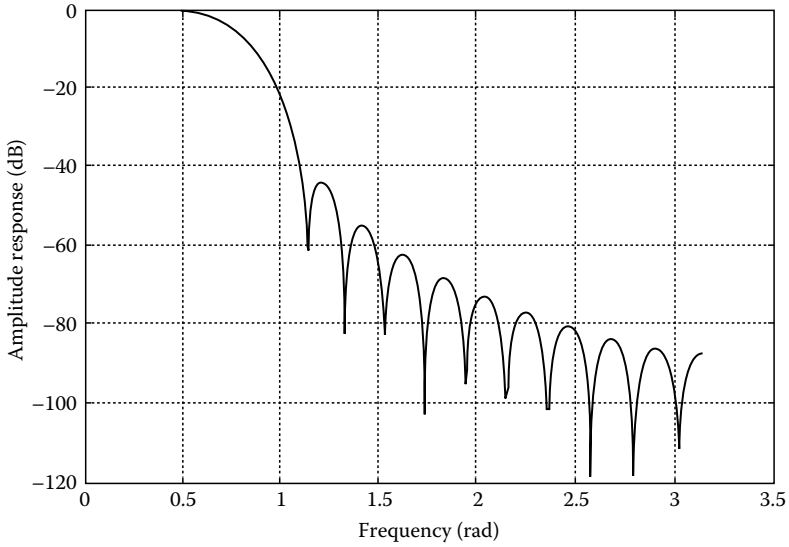


FIGURE 18.9 Magnitude response of low-pass FIR filter design using a 31-point Hanning window.

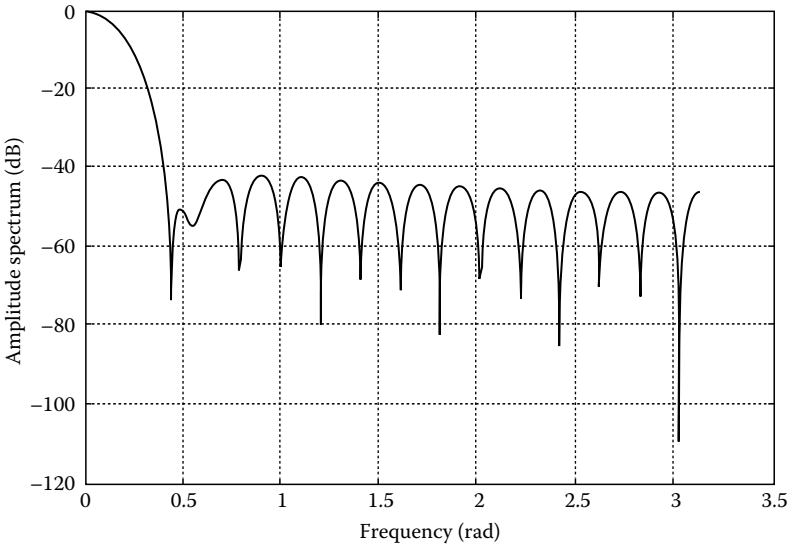


FIGURE 18.10 Fourier transform of the Hamming window.

Figure 18.11 illustrates the amplitude response of the FIR low-pass filter with $\omega_c = \pi/4$ obtained by applying the Hamming window for $N = 31$. The first sidelobe peak is -51 dB, a -7 dB improvement with respect to that using the Hanning window. However, it is noted that as frequency increases, the stopband attenuation does not increase as much as that produced by using the Hanning window.

The Blackman window. The Blackman window is given by

$$w_b(nT) = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.42}$$

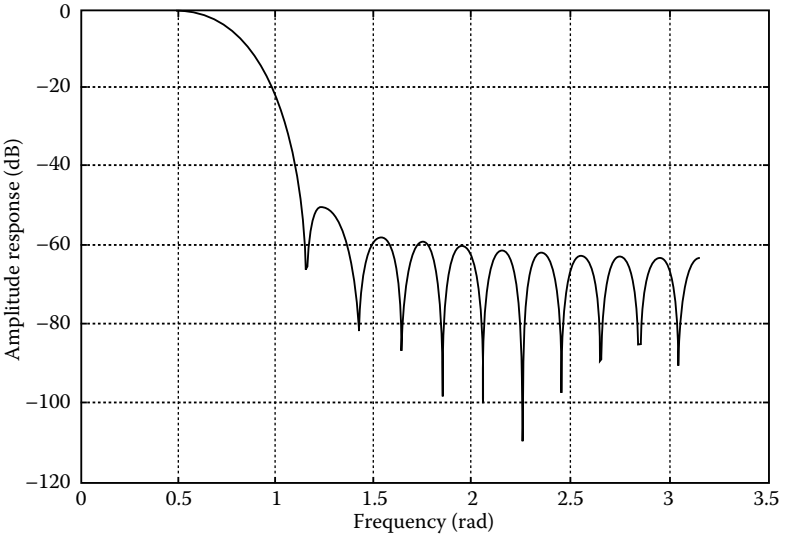


FIGURE 18.11 Magnitude response of low-pass FIR filter design using a 31-point Hamming window.

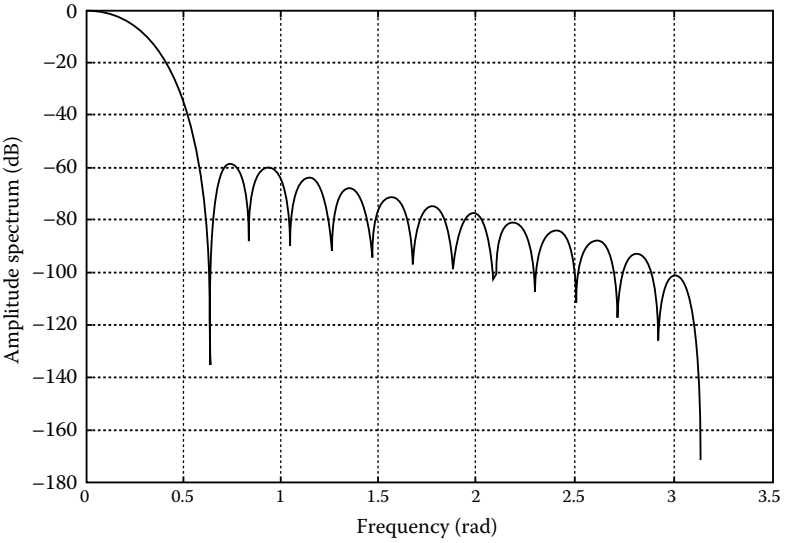


FIGURE 18.12 Fourier transform of the Blackman window.

The amplitude response of the Blackman window is shown in Figure 18.12. It can be seen that it has the highest sidelobe level, down −57 dB from the mainlobe peak. However, the mainlobe width has increased to $12\pi/N$.

The amplitude response of the FIR low-pass filter obtained when applying the Blackman window is shown in Figure 18.13. The minimum attenuation in the stopband is about −74 dB, but it occurs for $\omega > \pi/2$.

The Kaiser window. For the foregoing windows, the width of the mainlobe is inversely proportional to N. However, the minimum stopband attenuation is independent of the window length and is a function of

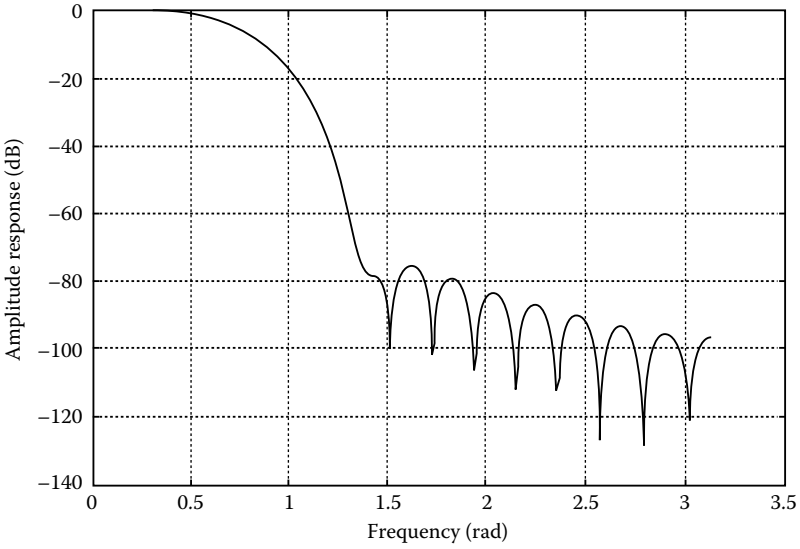


FIGURE 18.13 Magnitude response of low-pass FIR filter design using a 31-point Blackman window.

the selected window. Hence, to meet a desired stopband attenuation, the designer is forced to select a window that meets the design specifications. It is worth noting that windows with low sidelobe levels have broader mainlobe widths, hence requiring an increase in the order of the filter N to achieve the desired transition width.

In 1974, Kaiser [4] introduced a new window, now known as the Kaiser window, based on discrete-time approximations of the prolate spheroidal wave functions. This window has a variable parameter β, which can be varied to control the sidelobe level with respect to the mainlobe peak. As in other windows,

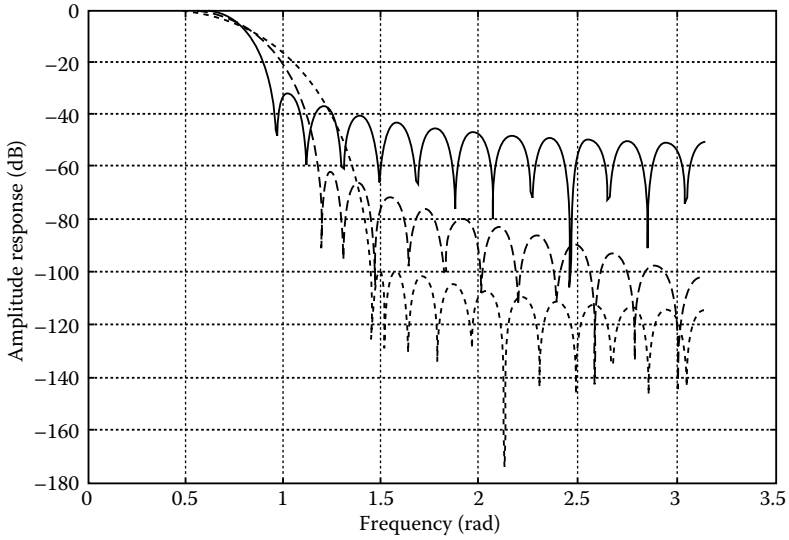


FIGURE 18.14 Magnitude responses of low-pass FIR filter design using a 31-point Kaiser window with $\beta = 1$ (solid line), $\beta = 6$ (dash line), and $\beta = 10$ (dotted line).

TABLE 18.2 Spectral Properties of N-Point Windows

Window	Mainlobe Width	Peak Amplitude of Sidelobe (dB)
Rectangular	$4\pi/N$	−13
Bartlett	$8\pi/N$	−25
Hanning	$8\pi/N$	−31
Hamming	$8\pi/N$	−41
Blackman	$12\pi/N$	−57

the mainlobe width can be adjusted by changing the length of the window, which in turn adjusts the transition width of the filter. Therefore, FIR filters can be efficiently designed using the Kaiser window.

The Kaiser windows are defined by

$$w_k(nT) = \begin{cases} \dfrac{I_0\left(2\beta\sqrt{\frac{n}{N-1}-\left(\frac{n}{N-1}\right)^2}\right)}{I_0(\beta)} & 0 \le n \le N-1 \\ 0 & \text{elsewhere} \end{cases} \tag{18.43}$$

where $I_0(x)$ is the modified zeroth-order Bessel function of the first kind. Kaiser has shown that these windows are nearly optimum in the sense of having the largest energy in the mainlobe for a given peak sidelobe amplitude.

To give an impression of the results that can be obtained by using Kaiser windows, Figure 18.14 shows the amplitude responses of an FIR low-pass filter design with $N = 31$ and $\omega_c = \pi/4$ and with $\beta = 1$, $\beta = 6$, and $\beta = 10$. As the value of β increases, the stopband attenuation of the low-pass filter increases and the transition band widens. Proper choice of N then leads to the final design.

The windows discussed above are compared in terms of their mainlobe width and the maximum sidelobe level in Table 18.2.

References

1. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
2. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
3. L. R. Rabiner, Techniques for designing finite duration impulse response digital filters, *IEEE Trans. Commun. Technol.*, COM-19, 188–195, April 1971.
4. J. F. Kaiser, Nonrecursive digital filter design using the I_O-sinh window function, in *Selected Papers in Digital Signal Processing, II*, New York: IEEE, 1976.

18.3 Design of FIR Filters by Optimization

Andreas Antoniou

18.3.1 Equiripple FIR Filters

The design of FIR filters can be accomplished either through noniterative or iterative methods. Noniterative methods entail the use of a small set of closed-form formulas and are, as a consequence, simple to apply. A frequently used method of this class is through the use of the Fourier series in conjunction

with window functions. Iterative methods are based on the application of optimization techniques. These are characterized by a substantial increase in the computational complexity, but often lead to designs that are optimal in some respect.

This section deals with an iterative method for the design of FIR filters known as the *weighted-Chebyshev* method. In this approach, an error function is formulated for the desired filter in terms of a linear combination of cosine functions and is then minimized by using a very efficient multivariable optimization algorithm known as the *Remez exchange algorithm*. When convergence is achieved, the error function becomes equiripple, as in other Chebyshev solutions. The amplitude of the error in different frequency bands of interest is controlled by applying weighting to the error function.

The weighted-Chebyshev method is very flexible and can be used to obtain optimal solutions for most types of FIR filters, e.g., digital differentiators, Hilbert transformers, and low-pass, high-pass, bandpass, bandstop, and multiband filters with piecewise-constant amplitude responses. Furthermore, it can be used to design filters with arbitrary amplitude responses. Consequently, it is widely used. In common with other optimization methods, the weighted-Chebyshev method requires a large amount of computation; however, as the cost of computation is becoming progressively cheaper with time, this disadvantage is not a serious one.

The underlying principles of the weighted-Chebyshev method were proposed during the early 1970s [1–3] and a series of developments soon after [4–8] led to the well-known computer program of McClellan et al. [9]. Some more recent enhancements to the method are reported in Refs. [10,11]. A detailed treatment of the subject can be found in Ref. [12].

18.3.2 Problem Formulation

An FIR filter with a symmetrical impulse response and odd length N can be represented by the transfer function

$$H(z) = \sum_{n=0}^{N-1} h(nT)z^{-n}$$

If we assume a sampling rate $\omega_s = 2\pi$, we have $T = 2\pi/\omega_s = 1$ s, and hence the frequency response of the filter can be expressed as

$$H(e^{j\omega T}) = e^{-jc\omega}P_c(\omega)$$

where

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega \tag{18.44}$$

with

$$\begin{aligned} a_0 &= h(c) \\ a_k &= 2h(c-k) \quad \text{for } k = 1, 2, \ldots, c \\ c &= (N-1)/2 \end{aligned}$$

For a desired frequency response $e^{-jc\omega}D(\omega)$ and a specified weighting function $W(\omega)$, an error function $E(\omega)$ can be constructed as

$$E(\omega) = W(\omega)[D(\omega) - P_c(\omega)] \tag{18.45}$$

If it were possible to minimize the magnitude of the above error such that

$$|E(\omega)| \leq \delta_p$$

with respect to some compact subset of the frequency interval $[0, \pi]$, say Ω, a filter would be obtained in which

$$|E_0(\omega)| = |D(\omega) - P_c(\omega)| \leq \frac{\delta_p}{|W(\omega)|} \quad \text{for } \omega \in \Omega \tag{18.46}$$

In an equiripple filter, the magnitude of the error oscillates uniformly between zero and some maximum in each passband and stopband. In a low-pass equiripple filter, the amplitude response assumes the form depicted in Figure 18.15, where δ_p and δ_a are the amplitudes of the passband and stopband ripples, and ω_p and ω_a are the passband and stopband edges, respectively. Hence, we require

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ 0 & \text{for } \omega_a \leq \omega \leq \pi \end{cases}$$

with

$$|E_0(\omega)| \leq \begin{cases} \delta_p & \text{for } 0 \leq \omega \leq \omega_p \\ \delta_a & \text{for } \omega_a \leq \omega \leq \pi \end{cases} \tag{18.47}$$

Therefore, from Equations 18.46 and 18.47 we deduce

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \leq \omega \leq \omega_p \\ \delta_p/\delta_a & \text{for } \omega_a \leq \omega \leq \pi \end{cases}$$

Similarly, for high-pass filters, we obtain

$$D(\omega) = \begin{cases} 0 & \text{for } 0 \leq \omega \leq \omega_a \\ 1 & \text{for } \omega_p \leq \omega \leq \pi \end{cases}$$

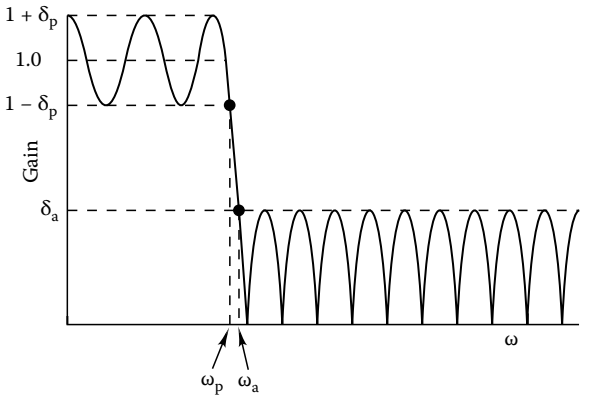


FIGURE 18.15 Amplitude response of equiripple low-pass filter. (Reproduced from Antoniou, A. *Digital Filters: Analysis, Design, and Applications*, McGraw-Hill, New York, 1993. With permission.)

and

$$W(\omega) = \begin{cases} \delta_p/\delta_a & \text{for } 0 \le \omega \le \omega_a \\ 1 & \text{for } \omega_p \le \omega \le \pi \end{cases}$$

Bandpass and bandstop filters. The above formulation can be easily extended to other types of filters. For bandpass filters, we have

$$D(\omega) = \begin{cases} 0 & \text{for } 0 \le \omega \le \omega_{a1} \\ 1 & \text{for } \omega_{p1} \le \omega \le \omega_{p2} \\ 0 & \text{for } \omega_{a2} \le \omega \le \pi \end{cases}$$

and

$$W(\omega) = \begin{cases} \delta_p/\delta_a & \text{for } 0 \le \omega \le \omega_{a1} \\ 1 & \text{for } \omega_{p1} \le \omega \le \omega_{p2} \\ \delta_p/\delta_a & \text{for } \omega_{a2} \le \omega \le \pi \end{cases}$$

where δ_p and δ_a are the amplitudes of the passband and stopband ripples, respectively, ω_{p1} and ω_{p2} are the passband edges, and ω_{a1} and ω_{a2} are the stopband edges, as depicted in Figure 18.16. On the other hand, for bandstop filters

$$D(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_{p1} \\ 0 & \text{for } \omega_{a1} \le \omega \le \omega_{a2} \\ 1 & \text{for } \omega_{p2} \le \omega \le \pi \end{cases}$$

and

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_{p1} \\ \delta_p/\delta_a & \text{for } \omega_{a1} \le \omega \le \omega_{a2} \\ 1 & \text{for } \omega_{p2} \le \omega \le \pi \end{cases}$$

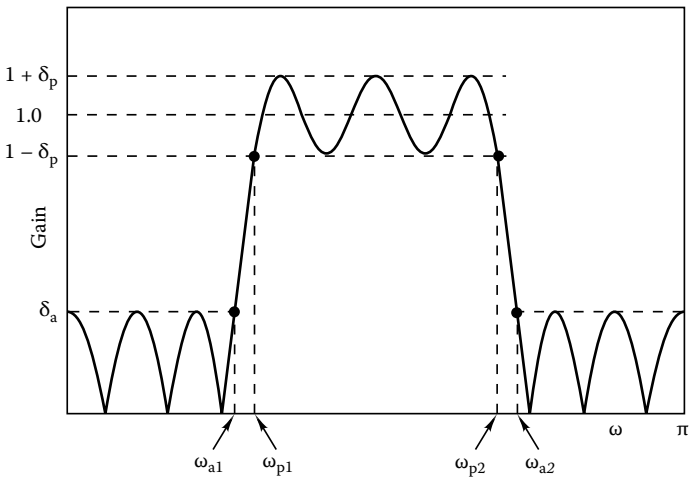


FIGURE 18.16 Amplitude response of equiripple bandpass filter. (Reproduced from Antoniou, A. *Digital Filters: Analysis, Design, and Applications*, McGraw-Hill, New York, 1993. With permission.)

The alternation theorem. An effective approach for the design of equiripple filters is to solve the minimax problem

$$\underset{x}{\text{minimize}}\left\{\max_{\omega}|E(\omega)|\right\} \tag{18.48}$$

where

$$x = [\,a_0 \quad a_1 \quad \cdots \quad a_c\,]^{\mathrm{T}}$$

is a column vector whose elements are the coefficients of the transfer function of the filter, which happen to be the values of the impulse response. The solution of this problem exists by virtue of the so-called alternation theorem [13], which is as follows.

THEOREM 18.1

If $P_c(\omega)$ is a linear combination of $r = c + 1$ cosine functions of the form

$$P_c(\omega) = \sum_{k=0}^{c} a_k \cos k\omega$$

then a necessary and sufficient condition that $P_c(\omega)$ be the unique, best, weighted-Chebyshev approximation to a continuous function $D(\omega)$ on Ω, where Ω is a compact subset of the frequency interval $[0, \pi]$, is that the weighted error function $E(\omega)$ exhibit at least $r + 1$ extremal frequencies in Ω, i.e., there must exist at least $r + 1$ points $\hat{\omega}$ in Ω such that

$$\begin{aligned} &\hat{\omega}_0 < \hat{\omega}_1 < \cdots < \hat{\omega}_r \\ &E(\hat{\omega}_i) = -E(\hat{\omega}_{i+1}) \quad \text{for } i = 0,\ 1,\ \ldots,\ r-1 \end{aligned}$$

and

$$|E(\hat{\omega}_i)| = \max_{\omega\in\Omega}|E(\omega)| \quad \text{for } i = 0,\ 1,\ \ldots,\ r$$

From the alternation theorem and Equation 18.45 we can write

$$E(\hat{\omega}_i) = W(\hat{\omega}_i)[D(\hat{\omega}_i) - P_c(\hat{\omega}_i)] = (-1)^i\delta \tag{18.49}$$

for $i = 0, 1, \ldots, r$, where δ is a constant. This system of equations can be put in matrix form as

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos c\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos c\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos c\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_c \\ \delta \end{bmatrix} = \begin{bmatrix} D(\hat{\omega}_0) \\ D(\hat{\omega}_1) \\ \vdots \\ D(\hat{\omega}_r) \end{bmatrix} \tag{18.50}$$

If the extremal frequencies (or extremals for short) were known, coefficients a_k and, in turn, the frequency response of the filter could be computed using Equation 18.44. The solution of this system exists since the above $(r+1)\times(r+1)$ matrix is nonsingular [13].

18.3.2.1 Remez Exchange Algorithm

The Remez exchange algorithm is an *iterative multivariable* algorithm, which is naturally suited for the solution of the minimax problem stated in Equation 18.48. It is based on the second optimization method of Remez [14] and involves the following basic steps.

ALGORITHM 18.1: Basic Remez Exchange Algorithm

---

1. Initialize extremals $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_r$ ensuring that an extremal is assigned at each band edge.
2. Locate the frequencies $\hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_\rho$ at which $|E(\omega)|$ is maximum and $|E(\hat{\omega}_i)| \geq \delta$. These frequencies are *potential* extremals for the next iteration.
3. Compute the convergence parameter

$$Q = \frac{\max\left|E(\hat{\omega}_i)\right| - \min\left|E(\hat{\omega}_i)\right|}{\max\left|E(\hat{\omega}_i)\right|}$$

   where $i = 0, 1, \ldots, \rho$.
4. Reject $\rho - r$ *superfluous* potential extremals $\hat{\omega}_i$ according to an appropriate rejection criterion and renumber the remaining $\hat{\omega}_i$ sequentially; then set $\hat{\omega}_i = \hat{\omega}_i$ for $i = 0, 1, \ldots, r$.
5. If $Q > \varepsilon$, where ε is a convergence tolerance (say $\varepsilon = 0.01$), repeat from step 2; otherwise continue to step 6.
6. Compute $P_c(\omega)$ using the last set of extremals; then deduce $h(n)$, the impulse response of the required filter, and stop.

The amount of computation required by the algorithm tends to depend quite heavily on the initialization scheme used in step 1, on the search method used for the location of the maxima of the error function in step 2, and on the criterion used to reject superfluous frequencies $\hat{\omega}_i$ in step 4.

Initialization of extremals. The simplest scheme for the initialization of extremals $\hat{\omega}_i$ for $i = 0, 1, \ldots, r$ is to assume that they are uniformly spaced in the frequency bands of interest. If there are J distinct bands in the required filter of widths $B_1, B_2, \ldots, B_J$ and extremals are to be located at the left-hand and right-hand band edges of each band, the sum of these bandwidths should be divided into $r + 1 - J$ intervals. Thus the average interval between adjacent extremals is

$$W_0 = \frac{1}{r + 1 - J} \sum_{j=1}^{J} B_j$$

Since the quantities B_j/W_0 need not be integers, the use of W_0 for the generation of the extremals will almost always result in a fractional interval in each band. This problem can be avoided by rounding the number of intervals B_j/W_0 to the nearest integer and then readjusting the frequency interval for the corresponding band accordingly. This can be achieved by letting the number of intervals in bands j and J be

$$m_j = \text{Int}\left(\frac{B_j}{W_0} + 0.5\right) \quad \text{for } j = 1, 2, \ldots, J - 1$$

and

$$m_J = r - \sum_{j=1}^{J-1} (m_j + 1)$$

respectively, and then recalculating the frequency intervals for the various bands as

$$W_j = \frac{B_j}{m_j} \quad \text{for } j = 1, 2, \ldots, J$$

A more sophisticated initialization scheme, which was found to give good results, is described in Ref. [15].

Location of maxima of the error function. The frequencies, $\hat{\omega}_i$, which *must include maxima at band edges if* $|E(\hat{\omega}_i)| \geq |\delta|$, can be located by simply evaluating $|E(\omega)|$ over a dense set of frequencies. A reasonable number of frequency points that yield sufficient accuracy in the determination of the frequencies $\hat{\omega}_i$ are $8(N+1)$. This corresponds to about 16 frequency points per ripple of $|E(\omega)|$. A suitable frequency interval for the jth band is $w_j = W_j/S$ with $S = 16$.

The above *exhaustive* search can be implemented in terms of Algorithm 18.2 below, where ω_{Lj} and ω_{Rj} are the left-hand and right-hand edges in band j; W_j is the interval between adjacent extremals and m_j is the number of intervals W_j in band j; w_j is the interval between successive samples of $|E(\omega)|$ in interval W_j and S is the number of intervals w_j in each interval W_j; N_j is the total number of intervals w_j in band j; and J is the number of bands.

ALGORITHM 18.2: Exhaustive Step-by-Step Search

1. Set $N_j = m_j S$, $w_j = B_j/N_j$, and $e = 0$.
2. For each of bands $1, 2, \ldots, j, \ldots, J$ do the following. For each of frequencies $\omega_{1j} = \omega_{Lj}$, $\omega_{2j} = \omega_{Lj} + \omega_j, \ldots, \omega_{ij} = \omega_{LJ} + (i-1)\omega_j, \ldots, \omega_{(N_j+1)j} = \omega_{Rj}$, set $\hat{\omega}_e = \omega_{ij}$ and $e = e + 1$ provided that $|E(\omega_{ij})| \geq |\delta|$ and one of the following conditions holds:
   (a) Case $\omega_{ij} = \omega_{Lj}$: if $|E(\omega_{ij})|$ is maximum at $\omega_{ij} = \omega_{Lj}$ (i.e., $|E(\omega_{Lj})| > |E(\omega_{Lj} + \varepsilon)|$)
   (b) Case $\omega_{Lj} < \omega_{ij} < \omega_{Rj}$: if $|E(\omega)|$ is maximum at $\omega = \omega_{ij}$ (i.e., $|E(\omega_{ij} - \omega_j)| < |E(\omega_{ij})| > |E(\omega_{ij} + \omega_j)|$)
   (c) Case $\omega_{ij} = \omega_{Rj}$: if $|E(\omega_{ij})|$ is maximum at $\omega_{ij} = \omega_{Rj}$ (i.e., $|E(\omega_{Rj})| > |E(\omega_{Rj} - \varepsilon)|$)

The parameter ε in steps 2(a) and 2(c) is a small positive constant and a value $10^{-2} w_j$ yields satisfactory results.

In practice, $|E(\omega)|$ is maximum at an interior left-hand band edge* if its first derivative at the band edge is negative, and a mirror-image situation applies at an interior right-hand band edge. In such cases, $|E(\omega)|$ has a zero immediately to the right or left of the band edge and the inequality in step 2(a) or 2(c) may sometimes fail to identify a maximum. However, the problem can be avoided by using the inequality $|E(\omega_{Lj} - \varepsilon)| > |E(\omega_{Lj})|$ in step 2(a) and $|E(\omega_{Rj})| < |E(\omega_{Rj} + \varepsilon)|$ in step 2(c) for interior band edges.

In rare circumstances, a maximum of $|E(\omega)|$ may occur between a band edge and the first sample point. Such a maximum may be missed by Algorithm 18.2, but the problem can be easily identified since the number of potential extremals will then be less than the minimum. The remedy is to check the number of potential extremals at the end of each iteration and if it is found to be less than $r + 1$, the density of sample points, i.e., S is doubled and the iteration is repeated. If the problem persists, the process is repeated until the required number of potential extremals is obtained. If a value of S equal to or less than 256 does not resolve the problem, the loss of potential extremals is most likely due to some other reason.

An important precaution in the implementation of the preceding search method is to ensure that extremals belong to the dense set of frequency points to avoid numerical ill-conditioning in the computation of $E(\omega)$ (see Equations 18.49 and 18.51). In addition, the condition $|E(\omega_{ij})| \leq |\delta|$ should

* An interior band edge is one in the range $0 < \omega < \pi$, i.e., not at $\omega = 0$ or π.

be replaced by $|E(\omega_{ij})| > |\delta| - \varepsilon_1$, where ε_1 is a small positive constant, say 10^{-6}, to ensure that no maxima are missed owing to roundoff errors.

The search method is very reliable and its use in Algorithm 18.1 leads to a *robust* algorithm since the entire frequency axis is searched using a dense set of frequency points. Its disadvantage is that it requires a considerable amount of computation and is, therefore, inefficient.

A more efficient version of Algorithm 18.2 is obtained by maintaining all the interior band edges as extremals throughout the optimization independently of the behavior of the error function at the band edges. However, the algorithm obtained tends to be somewhat less robust, i.e., it tends to fail more frequently than Algorithm 18.2.

Computation of $|E(\omega)|$ *and* $P_c(\omega)$. In steps 2 and 6 of the basic Remez algorithm (Algorithm 18.1), $|E(\omega)|$ and $P_c(\omega)$ need to be evaluated. This can be done by determining coefficients a_k by inverting the matrix in Equation 18.50. However, this approach is inefficient and may be subject to numerical ill-conditioning, in particular, if δ is small and N is large. An alternative and more efficient approach is to deduce δ analytically and then interpolate $P_c(\omega)$ on the r frequency points using the *barycentric* form of the *Lagrange interpolation* formula. The necessary formulation is as follows.

Parameter δ can be deduced as

$$\delta = \frac{\sum_{k=0}^{r} \alpha_k D(\hat{\omega}_k)}{\sum_{k=0}^{r} \frac{(-1)^k \alpha_k}{W(\hat{\omega}_k)}}$$

and $P_c(\omega)$ is given by

$$P_c(\omega) = \begin{cases} C_k & \text{for } \omega = \hat{\omega}_0, \hat{\omega}_1, \ldots, \hat{\omega}_{r-1} \\ \frac{\sum_{k=0}^{r-1} \frac{\beta_k C_k}{x - x_k}}{\sum_{k=0}^{r-1} \frac{\beta_k}{x - x_k}} & \text{otherwise} \end{cases} \tag{18.51}$$

where

$$\alpha_k = \prod_{i=0, i\neq k}^{r} \frac{1}{x_k - x_i}$$

$$C_k = D(\hat{\omega}_k) - (-1)^k \frac{\delta}{W(\hat{\omega}_k)}$$

$$\beta_k = \prod_{i=0, i\neq k}^{r-1} \frac{1}{x_k - x_i}$$

with

$$x = \cos\omega \quad \text{and} \quad x_i = \cos\hat{\omega}_i \quad \text{for } i = 0, 1, 2, \ldots, r$$

In step 2 of the Remez algorithm, $|E(\omega)|$ often needs to be evaluated at a frequency that was an extremal during the previous iteration. For these cases, the magnitude of the error function is simply $|\delta|$ according to Equation 18.49, and need not be evaluated. This would reduce the amount of computation to some extent.

An alternative formulation that simplifies the implementation of the Remez exchange algorithm can be found in Ref. [12].

Rejection of superfluous potential extremals. The solution of Equation 18.50 can be obtained only if *precisely* $r+1$ extremals are available. By differentiating $E(\omega)$, one can show that in a filter with one frequency band of interest (e.g., a digital differentiator) the number of maxima in $|E(\omega)|$ (potential extremals in step 2 of Algorithm 18.1) can be as high as $r+1$. In the weighted-Chebyshev method, band edges at which $|E(\omega)|$ is maximum or $|E(\omega)| \geq |\delta|$ are treated as potential extremals (see Algorithm 18.2). Therefore, whenever the number of frequency bands is increased by one, the number of potential extremals is increased by 2, i.e., for a filter with J bands there can be as many as $r+2J-1$ frequencies $\hat{\omega}_i$ and a maximum of $2J-2$ superfluous $\hat{\omega}_i$ may occur. This problem is overcome by rejecting $\rho - r$ of the potential extremals $\hat{\omega}_i$, if $\rho > r$, in step 4 of the algorithm.

A simple rejection scheme is to reject the $\rho - r$ frequencies $\hat{\omega}_i$ that yield the lowest $|E(\hat{\omega}_i)|$ and then renumber the remaining $\hat{\omega}_i$ from 0 to r [8]. This strategy is based on the well-known fact that the magnitude of the error in a given band is inversely related to the density of extremals in that band, i.e., a low density of extremals results in a large error and a high density results in a small error. Conversely, a low band error is indicative of a high density of extremals, and rejecting superfluous $\hat{\omega}_i$ in such a band is the appropriate course of action.

A problem with the scheme just described is that whenever a frequency remains an extremal in two successive iterations, $|E(\omega)|$ assumes the value of $|\delta|$ in the second iteration by virtue of Equation 18.49. In practice, there are almost always several frequencies that remain extremals from one iteration to the next, and the value of $|E(\omega)|$ at these frequencies will be the same. Consequently, the rejection of potential extremals on the basis of the magnitude of the error can become arbitrary and may lead to the rejection of potential extremals in bands where the density of extremals is low. This tends to increase the number of iterations, and it may even prevent the algorithm from converging on occasion. This problem can to some extent be alleviated by rejecting only potential extremals that are not band edges.

An alternative rejection scheme based on the aforementioned strategy, which gives excellent results for two-band and three-band filters, involves ranking the frequency bands in the order of lowest average band error, dropping the band with the highest average error from the list, and then rejecting potential extremals, one per band, in a cyclic manner starting with the band with the lowest average error [11]. The steps involved are as follows.

ALGORITHM 18.3: Rejection of Superfluous Potential Externals

1. Compute the average band errors

$$E_j = \frac{1}{\nu_j} \sum_{\hat{\omega}_i \in \Omega_j} \left| E(\hat{\omega}_i) \right| \quad \text{for } j = 1, 2, \ldots, J$$

   where Ω_j is the set of potential externals in band j given by

$$\Omega_j = \left\{ \hat{\omega}_j : \omega_{\mathrm{L}j} \leq \hat{\omega}_j \leq \omega_{\mathrm{R}j} \right\}$$

   ν_j is the number of potential externals in band j, and J is the number of bands.
2. Rank the J bands in the order of lowest average error and let $l_1, l_2, \ldots, l_{\mathrm{J}}$ be the ranked list obtained, i.e., l_1 and l_{J} are the bands with the lowest and highest average errors, respectively.
3. Reject one $\hat{\omega}_i$ in each of bands $l_1, l_2, \ldots, l_{\mathrm{J}-1}, l_1, l_2, \ldots, l_1$ until $\rho - r$ superfluous $\hat{\omega}_i$ are rejected. In each case, reject the $\hat{\omega}_i$, other than a band edge, that yields the lowest $|E(\hat{\omega}_i)|$ in the band.

For example, if $J = 3$, $\rho - r = 3$, and the average errors for bands 1, 2, and 3 are, respectively, 0.05, 0.08, and 0.02, then $\hat{\omega}_i$ are rejected in bands 3, 1, and 3. Note that potential extremals are not rejected in band 2, which is the band of highest average error.

Computation of impulse response. The impulse response in step 6 of Algorithm 18.1 can be determined by noting that function $P_c(\omega)$ is the frequency response of a noncausal version of the required filter. The impulse response of this filter, represented by $h_0(n)$ for $-c \leq n \leq c$, can be determined by computing $P_c(k\Omega)$ for $k = 0, 1, 2, \ldots, c$, where $\Omega = 2\pi/N$, and then using the *inverse discrete Fourier transform*. It can be shown that

$$h_0(n) = h_0(-n) = \frac{1}{N}\left[P_c(0) + \sum_{k=1}^{c} 2P_c(k\Omega)\cos\left(\frac{2\pi kn}{N}\right)\right]$$

for $n = 0, 1, 2, \ldots, c$. Therefore, the impulse response of the required causal filter is given by

$$h(n) = h_0(n - c)$$

for $n = 0, 1, 2, \ldots, N - 1$.

18.3.2.2 Improved Search Methods

For a filter of length N, with the number of intervals w_j in each internal W_j equal to S, the exhaustive step-by-step search described (Algorithm 18.2) requires about $S \times (N + 1)/2$ function evaluations, where each function evaluation entails $N - 1$ additions, $(N + 1)/2$ multiplications, and $(N + 1)/2$ divisions (see Equation 18.51).

A Remez optimization usually requires 4 to 8 iterations for low-pass or high-pass filters, 4 to 10 iterations for bandpass filters, and 4 to 12 iterations for bandstop filters. Further, if prescribed specifications are to be achieved and the appropriate value of N is unknown, typically two to four Remez optimizations have to be performed.* Thus, if $N = 101$, $S = 16$, number of Remez optimizations $= 4$, iterations per optimization $= 6$, the design would entail 24 iterations, 19,200 function evaluations, 1.92×10^6 additions, 0.979×10^6 multiplications, and 0.979×10^6 divisions. This is in addition to the computation required for the evaluation of δ and coefficients α_k, C_k, and β_k once per iteration. In effect, the amount of computation required to complete a design is quite substantial.

The bulk of the computation in Algorithm 18.2 is carried out to locate the maxima of $|E(\omega)|$ and the large amount of computation is a consequence of the exhaustive character of the search. Therefore, any attempt to reduce the computational complexity of the Remez exchange algorithm must of necessity involve a more efficient search for the maxima of $|E(\omega)|$.

The error function in the weighted-Chebyshev method is well behaved in practice, and is normally unimodal between successive zeros, as can be seen in Figures 18.15 and 18.16. Hence, the maxima of $|E(\omega)|$ can be located through more sophisticated search methods that utilize gradient information. Two such methods are the so-called *selective step-by-step search* and *cubic-interpolation search* reported in Refs. [10,11]. Collectively, the two search methods can reduce the amount of computation to about one-fifth the amount required by the exhaustive search.

Selective step-by-step search. The underlying principle in the development of the selective step-by-step search is that normally there is strict alternation between the maxima and the zeros of $|E(\omega)|$. In a given iteration, the maxima of $|E(\omega)|$ are either old maxima from the previous iteration that have moved or new maxima introduced at band edges. New interior maxima may also arise, in theory, but such occurrences are quite rare in practice.

* See Section 18.3.2.3.

The selective step-by-step search involves three distinct parts as follows:

(1) Maxima that correspond to previous maxima are located by searching in the neighborhoods of the most recent set of extremals in a step-by-step fashion using the first derivative of $|E(\omega)|$. If the first derivative is positive at an extremal, the search is carried out to the right of the extremal; otherwise, the search is carried out to the left of the extremal.
(2) New maxima at band edges can be located by noting the circumstances under which new maxima can arise. These are as follows:
   a. To the right of $\omega = 0$ (first band), if there is an extremal and $|E(\omega)|$ has a minimum at $\omega = 0$
   b. To the left of $\omega = \pi$ (last band), if there is an extremal and $|E(\omega)|$ has a minimum at $\omega = \pi$
   c. At $\omega = 0$, if there is no extremal at $\omega = 0$
   d. At $\omega = \pi$, if there is no extremal at $\omega = \pi$
   e. To the right of an interior left-hand edge
   f. To the left of an interior right-hand edge
   g. At $\omega = \omega_{Lj}$, if there is no extremal at $\omega = \omega_{Lj}$
   h. At $\omega = \omega_{Rj}$, if there is no extremal at $\omega = \omega_{Rj}$
(3) New interior maxima, which cannot be located by the checks in (1) and (2) can be found by noting the presence of large gaps in the set of potential extremals identified in (1) and (2). If the difference between two consecutive potential extremals exceeds 1.5 to 2 times the initial interval between extremals (i.e., W_j), then the interval is checked for additional maxima.

If a selective step-by-step search based on the above principles is used in Algorithm 18.1, then at the start of the optimization the distance between a typical extremal $\hat{\omega}_i$ and the nearby maximum point $\hat{\omega}_i$ will be less than half the period of the corresponding ripple of $|E(\omega)|$, owing to the relative symmetry of the ripples of the error function. In effect, during the first iteration less than half of the combined width of the different bands needs to be searched. Thus the number of function evaluations required would be reduced from about 16 to less than 8 per extremal in practice. This will reduce the number of function evaluations by more than 50% relative to that required by the exhaustive search of Algorithm 18.1 without degrading the accuracy of the optimization in any way. As the optimization progresses and the solution is approached, extremal $\hat{\omega}_i$ and maximum point $\hat{\omega}_i$ tend to coincide and, therefore, the cumulative length of the frequency range that has to be searched is progressively reduced, thereby resulting in further economies in the number of function evaluations. In the last iteration, only two or three function evaluations are needed (including derivatives) per ripple. As a result, the total number of function evaluations can be reduced by 65%–70% relative to that required by the exhaustive search [10,11].

Cubic-interpolation search. The maxima in item (1) of the above method can also be found through the use of one stage of polynomial interpolation. Either quadratic or cubic interpolation can be used. In these methods, a polynomial approximation is obtained for the magnitude of the error function in the neighborhood of a given extremal and the location of the maximum is determined by finding the point at which the first derivative is zero. Although cubic interpolation entails a more complicated formulation than quadratic interpolation, it leads to improved accuracy, which tends to translate into improved efficiency.

Several choices are possible in setting up a cubic-interpolation search for the problem at hand. One that was found to work well in practice entails evaluating $|E(\omega)|$ at three frequency points and its derivative at one point. Choosing the extremal itself as one of the points reduces the computation further since the value of $|E(\omega)|$ is known to be $|\delta|$ from the previous iteration. Thus this scheme entails three function evaluations per extremal.

The computational complexity of the cubic-interpolation search described remains constant from iteration to iteration since the number of function evaluations required to perform an interpolation is constant. At the start of the optimization, the cubic-interpolation search is more efficient than the selective step-by-step search. However, as the solution is approached the number of function evaluations

required by the selective search is progressively reduced, as was stated earlier, and at some point the selective search becomes more efficient. A prudent strategy under the circumstances is to use the cubic-interpolation search at the start of the optimization and switch over to the selective step-by-step search when some suitable criterion is satisfied. Extensive experimentation has shown that computational advantage can be gained by using the cubic-interpolation search if parameter Q (see Algorithm 18.1) is greater than about 0.65 and the selective search otherwise. The use of the cubic-interpolation search along with the selective step-by-step search of the preceding section can reduce the number of function evaluations by 70%–85% relative to that required by the exhaustive search [10,11].

More information, including the necessary formulation as well as a practical and efficient implementation of the Remez exchange algorithm in terms of the above search methods, can be found in Ref. [12].

Example 18.1

The Remez algorithm was used with (1) the exhaustive search, (2) the selective step-by-step search, and (3) the selective search in conjunction with the cubic-interpolation search to design an FIR equiripple high-pass filter satisfying the following specifications:

Filter length N: 23
Passband edge $\boldsymbol{\omega}_p$: 2.0 rad/s
Stopband edge $\boldsymbol{\omega}_a$: 1.0 rad/s
Ratio $\boldsymbol{\delta}_p/\boldsymbol{\delta}_a$: 15.0
Sampling frequency $\boldsymbol{\omega}_s$: $2\boldsymbol{\pi}$ rad/s

The progress of the design is illustrated in Table 18.3. As can be seen, the exhaustive and selective search methods required six iterations each, whereas the selective search in conjunction with cubic interpolation required seven iterations. However, the number of function evaluations (evaluations of $P_c(\boldsymbol{\omega})$ using Equation 18.51 plus evaluations of its first or second derivative) decreased from 1013 in the first method to 350 in the second method to 259 in the third method. In the Remez algorithm, approximately 80%–90% of the computational effort involves function evaluations. In effect, relative to that required by the exhaustive search, the use of the selective step-by-step search reduced the amount of computation by about 65.4%, and the use of the selective step-by-step search in conjunction with the cubic-interpolation search reduced the amount of computation by about 74.4%.

The three methods resulted in approximately the same impulse responses, as can be seen in Table 18.4, and the passband ripple and minimum stopband attenuation obtained in each case were 0.043 and 75.7 dB, respectively. The amplitude response of the filter is illustrated in Figure 18.17.

TABLE 18.3 Progress in Design of High-Pass Filter (Example 18.1)

Iteration Number	Exhaustive Search		Selective Search		Selective Search with Cubic Interpolation	
	Q	FEs	Q	FEs	Q	FEs
1	0.9912	169	0.9912	93	0.9912	66
2	0.9207	168	0.9207	86	0.9406	44
3	0.9480	169	0.9480	55	0.8830	42
			$\hat{\omega}_{32}$ rejected			
4	0.7249	169	0.7249	62	0.6952	31
5	0.0923	169	0.0923	31	0.1417	31
6	0.0017	169	0.0017	23	0.0102	23
7	—	—	—	—	0.0000	22
Total FE's		1013		350		259

TABLE 18.4 Impulse Response of High-Pass Filter (Example 18.1)

	$h_0(n) = h_0(-n)$	
n	Exhaustive or Selective Search	Selective Search with Cubic Interpolation
0	5.034954×10^{-1}	5.035077×10^{-1}
1	-3.123538×10^{-1}	-3.123535×10^{-1}
2	-3.085731×10^{-3}	-3.097829×10^{-3}
3	8.932914×10^{-2}	8.932911×10^{-2}
4	2.053235×10^{-3}	2.063564×10^{-3}
5	-3.898118×10^{-2}	-3.898177×10^{-2}
6	-8.467375×10^{-4}	-8.540079×10^{-4}
7	1.660800×10^{-2}	1.660858×10^{-2}
8	5.401585×10^{-5}	5.892008×10^{-5}
9	-6.100465×10^{-3}	-6.101979×10^{-3}
10	7.298411×10^{-4}	7.281192×10^{-4}
11	9.275654×10^{-4}	9.285482×10^{-4}

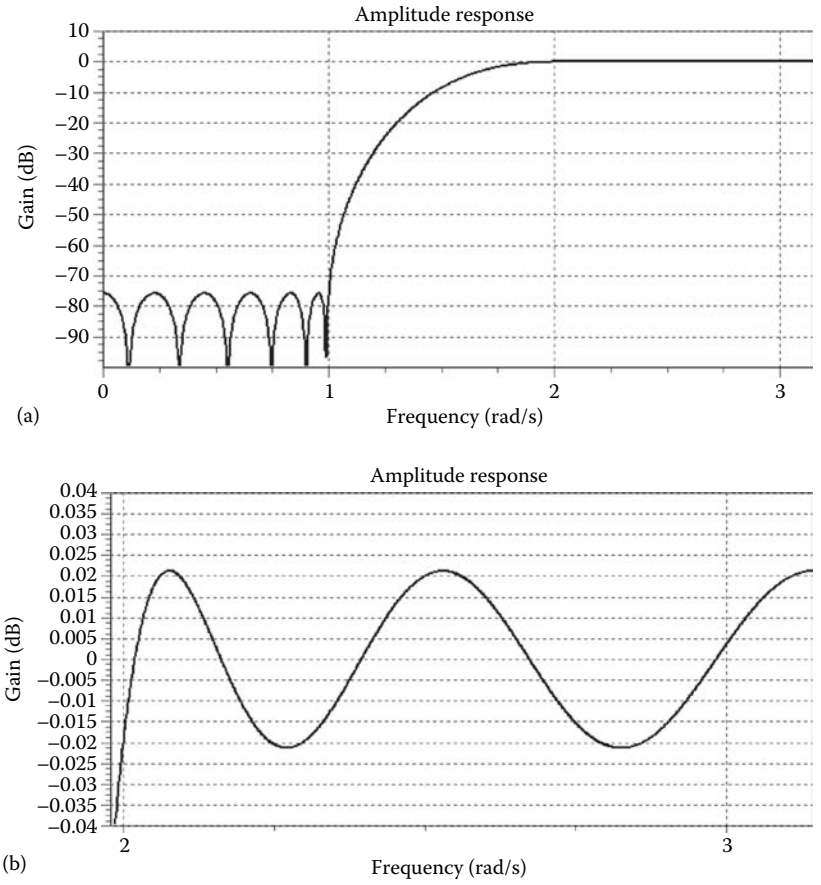


FIGURE 18.17 Amplitude response of equiripple high-pass filter (Example 18.1): (a) baseband and (b) passband.

Example 18.2

In this example, the Remez algorithm was used with (1) the exhaustive search, (2) the selective step-by-step search, and (3) the selective step-by-step search in conjunction with the cubic-interpolation search to design an FIR equiripple bandstop filter satisfying the following specifications:

Filter length N: 29
Lower passband edge ω_{p1}: 0.8 rad/s
Upper passband edge ω_{p2}: 2.1 rad/s
Lower stopband edge ω_{a1}: 1.1 rad/s
Upper stopband edge ω_{a2}: 1.8 rad/s
Ratio δ_{p1}/δ_a: 5.0
Ratio δ_{p1}/δ_{p2}: 2.0
Sampling frequency ω_s: 2π rad/s

The progress of the design is illustrated in Table 18.5. In this example, each of the three methods required four iterations. The number of function evaluations decreased from 804 in the first method to 190 in the second method to 131 in the third method. In effect, the use of the selective step-by-step search reduced the amount of computation by about 76.4%, and the use of the

TABLE 18.5 Progress in Design of Bandstop Filter (Example 18.2)

Iteration Number	Exhaustive Search		Selective Search		Selective Search with Cubic Interpolation	
	Q	FEs	Q	FEs	Q	FEs
1	0.6836	201	0.6836	79	0.6940	36
2	0.3138	201	0.3138	51	0.2378	36
3	0.0804	201	0.0804	34	0.0675	32
4	0.0000	201	0.0000	26	0.0007	27
Total FE's		804		190		131

TABLE 18.6 Impulse Response of Bandstop Filter (Example 18.2)

	$h_0(n) = h_0(-n)$	
n	Exhaustive or Selective Search	Selective Search with Cubic Interpolation
0	6.656629×10^{-1}	6.656478×10^{-1}
1	-4.187327×10^{-2}	-4.186510×10^{-2}
2	2.635370×10^{-1}	2.635297×10^{-1}
3	8.005521×10^{-2}	8.005307×10^{-2}
4	-1.131284×10^{-1}	-1.131056×10^{-1}
5	-3.691645×10^{-2}	-3.691932×10^{-2}
6	-9.914085×10^{-4}	-1.013621×10^{-3}
7	-3.018917×10^{-2}	-3.017017×10^{-2}
8	2.931776×10^{-2}	2.930006×10^{-2}
9	5.022490×10^{-2}	5.022450×10^{-2}
10	-9.715988×10^{-3}	-9.687345×10^{-3}
11	-2.550790×10^{-2}	-2.553543×10^{-2}
12	-4.023265×10^{-4}	-3.827029×10^{-4}
13	-3.410741×10^{-2}	-3.412007×10^{-2}
14	-1.421939×10^{-2}	-1.424189×10^{-2}

selective step-by-step search in conjunction with the cubic-interpolation search reduced the amount of computation by about 83.7%, relative to that required by the exhaustive search.

The three methods resulted in approximately the same impulse responses, as can be seen in Table 18.6. The amplitude response of the filter is illustrated in Figure 18.18; the passband ripples obtained for the two passbands were 1.78 and 0.89 dB, respectively, and the minimum stopband attenuation was 33.79 dB.

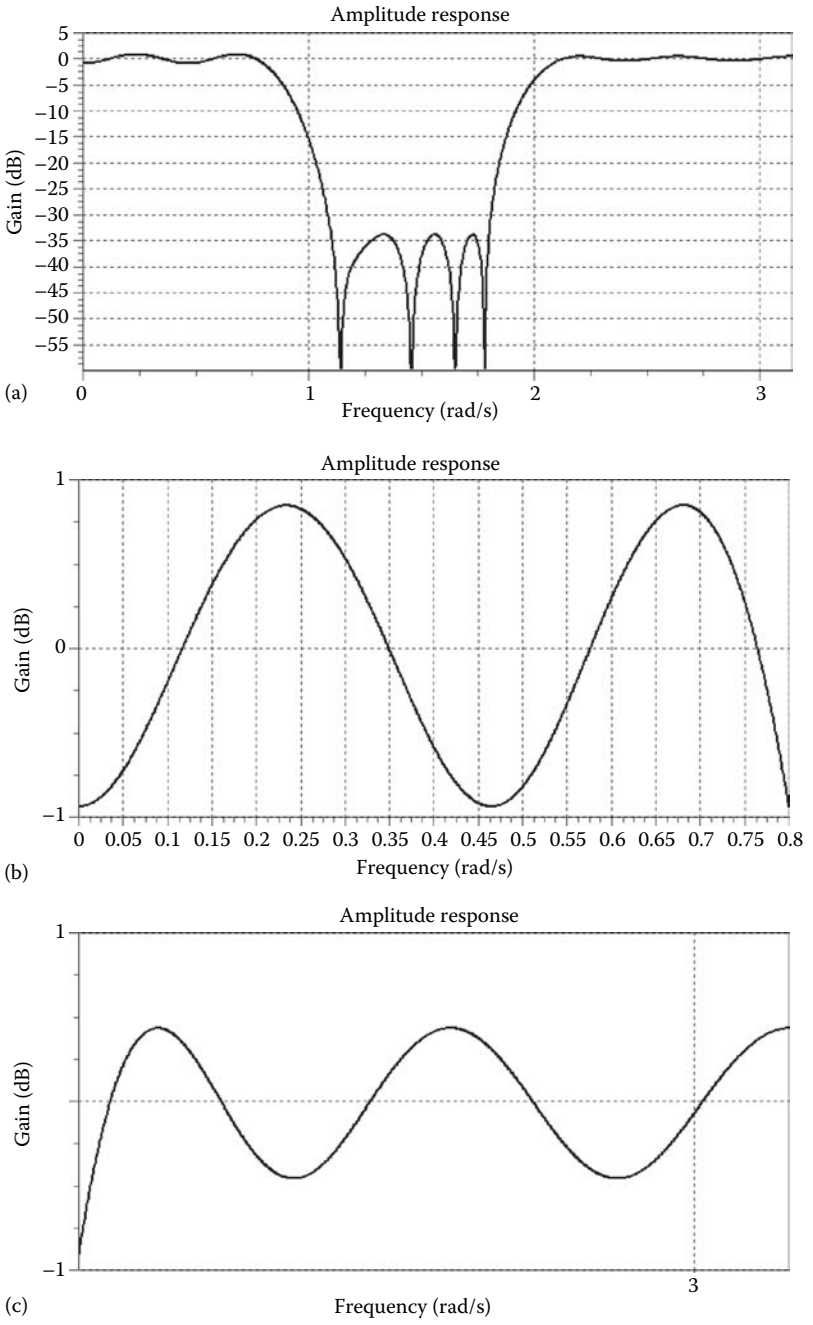


FIGURE 18.18 Amplitude response of equiripple bandstop filter (Example 18.2): (a) baseband, (b) lower passband, and (c) upper passband.

18.3.2.3 Prescribed Specifications

Given a filter length N, a set of passband and stopband edges, and a ratio δ_p/δ_a, an FIR filter with approximately piecewise-constant amplitude-response specifications can be readily designed. While the filter obtained will have passband and stopband edges at the correct locations and the ratio δ_p/δ_a will be as required, the amplitudes of the passband and stopband ripples are highly unlikely to be precisely as specified. An acceptable design can be obtained by predicting the value of N on the basis of the required specifications and then designing filters for increasing or decreasing values of N until the lowest value of N that satisfies the specifications is found.

A reasonably accurate *empirical* formula for the prediction of N for the case of low-pass and high-pass filters, due to Hermann et al. [16], is

$$N = \text{Int}\left[\frac{(D - FB^2)}{B} + 1.5\right] \tag{18.52}$$

where

$$\begin{aligned} B &= \left|\omega_a - \omega_p\right|/2\pi \\ D &= \left[0.005309(\log\delta_p)^2 + 0.07114\log\delta_p - 0.4761\right]\log\delta_a \\ &\quad - \left[0.00266(\log\delta_p)^2 + 0.5941\log\delta_p + 0.4278\right] \\ F &= 0.51244(\log\delta_p - \log\delta_a) + 11.012 \end{aligned}$$

Interestingly this formula can also be used to predict the filter length in the design of bandpass, bandstop, and multiband filters in general. In these filters, a value of N is computed for each transition band between a passband and stopband or a stopband and passband using Equation 18.52 and the largest value of N so obtained is taken to be the predicted filter length. *Prescribed specifications* can be achieved by using the following design algorithm.

ALGORITHM 18.4: Design of Filters Satisfying Prescribed Specifications

1. Compute N using Equation 18.52; if N is even, set $N = N + 1$.
2. Design a filter of length N and determine the minimum value of δ, say $\breve{\delta}$.
   A. If $\breve{\delta} > \delta_p$, then do the following:
      i. Set $N = N + 2$, design a filter of length N, and find $\breve{\delta}$.
      ii. If $\breve{\delta} \leq \delta_p$, then go to step 3; else, go to step 2(A)(i).
   B. If $\breve{\delta} < \delta_p$, then do the following:
      i. Set $N = N - 2$, design a filter of length N, and find $\breve{\delta}$.
      ii. If $\breve{\delta} > \delta_p$ then go to step 4; else, go to step 2(B)(i).
3. Use the last set of extremals and the corresponding value of N to obtain the impulse response of the required filter and stop.
4. Use the last but one set of extremals and the corresponding value of N to obtain the impulse response of the required filter and stop.

Example 18.3

Algorithm 18.4 was used to design an FIR equiripple bandpass filter that would satisfy the following specifications:

Odd filter length
Maximum passband ripple A_p: 0.5 dB
Minimum stopband attenuation A_{a1}: 50.0 dB
Minimum stopband attenuation A_{a2}: 30.0 dB
Lower passband edge ω_{p1}: 1.2 rad/s
Upper passband edge ω_{p2}: 1.8 rad/s
Lower stopband edge ω_{a1}: 0.9 rad/s
Upper stopband edge ω_{a2}: 2.1 rad/s
Sampling frequency ω_s: 2π rad/s

The progress of the design is illustrated in Table 18.7. As can be seen, a filter of length 41 was initially predicted, which was found to have a passband ripple of 0.47 dB, a minimum stopband attenuation of 50.4 in the lower stopband, and 30.4 dB in the upper stopband, i.e., the required specifications were satisfied. Then a filter length of 39 was tried and found to violate the specifications. Hence the first design is the required filter. The impulse response is given in Table 18.8. The corresponding amplitude response is depicted in Figure 18.19.

18.3.2.4 Generalization

There are four types of constant-delay FIR filters. The impulse response can be *symmetrical* or *antisymmetrical*, and the filter length can be *odd* or *even*. In the preceding sections, we considered the design of filters with symmetrical impulse response and odd length. In this section, we show that the Remez algorithm can also be applied for the design of other types of filters.

Antisymmetrical impulse response and odd filter length. Assuming that $\omega_s = 2\pi$, the frequency response of an FIR filter with *antisymmetrical* impulse and *odd* length can be expressed as

$$H(e^{j\omega T}) = e^{-jc\omega} jP'_c(\omega)$$

TABLE 18.7 Progress in Design of Bandpass Filter (Example 18.3)

n	Iterations	FEs	A_p, dB	A_{a1}, dB	A_{a2}, dB
41	8	550	0.47	50.4	30.4
39	7	527	0.67	47.5	27.5

TABLE 18.8 Impulse Response of Bandpass Filter (Example 18.3)

n	$h_0(n) = h_0(-n)$	n	$h_0(n) = h_0(-n)$
0	2.761666×10^{-1}	11	2.726816×10^{-2}
1	1.660224×10^{-2}	12	-2.663859×10^{-2}
2	-2.389235×10^{-1}	13	-1.318252×10^{-2}
3	-3.689501×10^{-2}	14	6.312944×10^{-3}
4	1.473038×10^{-1}	15	-5.820976×10^{-3}
5	2.928852×10^{-2}	16	5.827957×10^{-3}
6	-4.770552×10^{-2}	17	1.528658×10^{-2}
7	-2.008131×10^{-3}	18	-8.288708×10^{-3}
8	-1.875082×10^{-2}	19	-1.616904×10^{-2}
9	-2.262965×10^{-2}	20	1.092728×10^{-2}
10	3.860990×10^{-2}	—	—

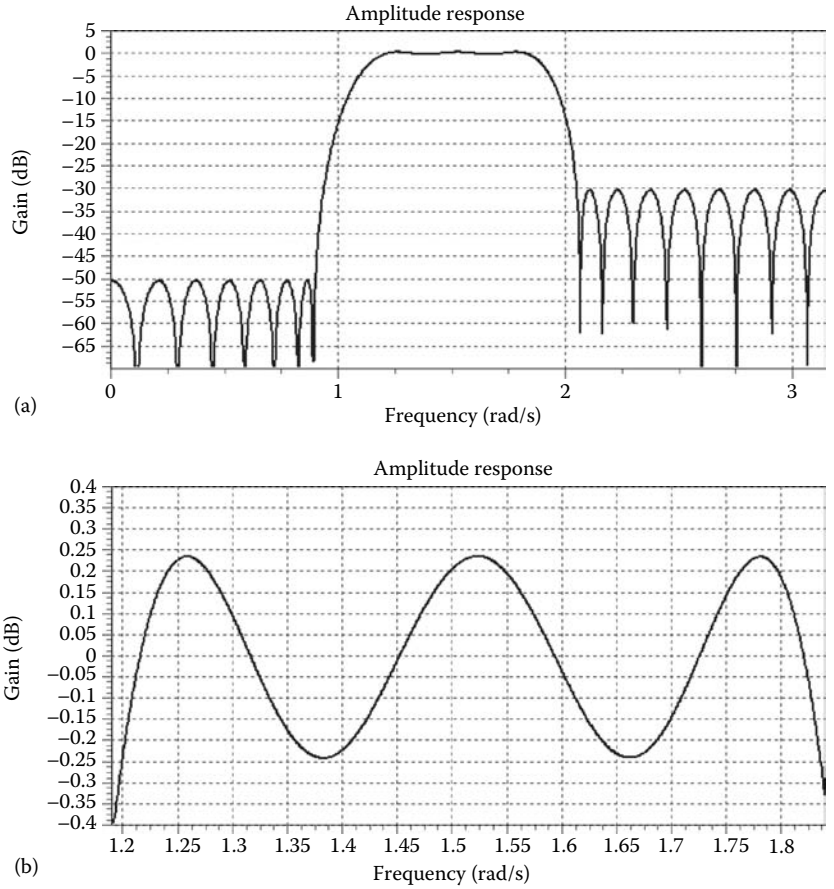


FIGURE 18.19 Amplitude response of equiripple bandpass filter (Example 18.3): (a) baseband and (b) passband.

where

$$\begin{aligned} P_c'(\omega) &= \sum_{k=1}^{c} a_k \sin k\omega \\ a_k &= 2h(c-k) \quad \text{for } k = 1, 2, \ldots, c \\ c &= (N-1)/2 \end{aligned} \tag{18.53}$$

A filter with a desired frequency response $e^{-jc\omega}\, jD(\omega)$ can be designed by constructing the error function

$$E(\omega) = W(\omega)[D(\omega) - P_c'(\omega)] \tag{18.54}$$

and then minimizing $|E(\omega)|$ with respect to some compact subset of the frequency interval $[0, \pi]$. From Equation 18.53, $P_c'(\omega)$ can be expressed as [6]

$$P_c'(\omega) = (\sin \omega)\ P_{c-1}(\omega) \tag{18.55}$$

where

$$P_{c-1}(\omega) = \sum_{k=0}^{c-1} \tilde{c}_k \cos k\omega$$

and

$$\begin{aligned} a_1 &= \tilde{c}_0 - \frac{1}{2}\tilde{c}_2 \\ a_k &= \frac{1}{2}(\tilde{c}_{k-1} - \tilde{c}_{k+1}) \quad \text{for } k = 2, 3, \ldots, c-2 \\ a_{c-1} &= \frac{1}{2}\tilde{c}_{c-2} \\ a_c &= \frac{1}{2}\tilde{c}_{c-1} \end{aligned}$$

Hence Equation 18.54 can be put in the form

$$E(\omega) = \tilde{W}(\omega)[\tilde{D}(\omega) - \tilde{P}(\omega)] \tag{18.56}$$

where

$$\begin{aligned} \tilde{W}(\omega) &= Q(\omega)W(\omega) \\ \tilde{D}(\omega) &= D(\omega)/Q(\omega) \\ \tilde{P}(\omega) &= P_{c-1}(\omega) \\ Q(\omega) &= \sin\omega \end{aligned}$$

Evidently, Equation 18.56 is of the same form as Equation 18.45, and upon proceeding as in Section 18.3.2, one can obtain the system of equations

$$\begin{bmatrix} 1 & \cos\hat{\omega}_0 & \cos 2\hat{\omega}_0 & \cdots & \cos(c-1)\hat{\omega}_0 & \frac{1}{W(\hat{\omega}_0)} \\ 1 & \cos\hat{\omega}_1 & \cos 2\hat{\omega}_1 & \cdots & \cos(c-1)\hat{\omega}_1 & \frac{-1}{W(\hat{\omega}_1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\hat{\omega}_r & \cos 2\hat{\omega}_r & \cdots & \cos(c-1)\hat{\omega}_r & \frac{(-1)^r}{W(\hat{\omega}_r)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{c-1} \\ \delta \end{bmatrix} = \begin{bmatrix} \tilde{D}(\hat{\omega}_0) \\ \tilde{D}(\hat{\omega}_1) \\ \vdots \\ \tilde{D}(\hat{\omega}_r) \end{bmatrix}$$

where $r = c$ is the number of cosine functions in $P_{c-1}(\omega)$. The above system is the same as that in Equation 18.50 except that the number of extremals has been reduced from $c+2$ to $c+1$; therefore, the application of the Remez algorithm follows the methodology detailed in Sections 18.3.2 and 18.3.2.1 formulation and the Remez exchange algorithm.

The use of Algorithm 18.1 yields the optimum $P_{c-1}(\omega)$ and from Equation 18.55, the cosine function $P'_c(\omega)$ can be formed. Now $jP'_c(\omega)$ is the frequency response of a noncausal version of the required filter. The impulse response of this filter can be obtained as

$$h_0(n) = -h_0(-n) = -\frac{1}{N}\left[\sum_{k=1}^{c} 2P'_c(k\Omega)\ \sin\left(\frac{2\pi kn}{N}\right)\right]$$

for $n = 0, 1, 2, \ldots, c$, where $\Omega = 2\pi/N$, by using the inverse discrete Fourier transform. The impulse response of the corresponding causal filter is given by

$$h(n) = h_0(n-c)$$

for $n = 0, 1, 2, \ldots, N-1$.

The Remez algorithm can also be applied for the design of filters with symmetrical or antisymmetrical impulse response and even N. However, these filters are used less frequently. The reader is referred to Refs. [6,12] for more details.

18.3.2.5 Digital Differentiators

The Remez algorithm can be easily applied for the design of *equiripple digital differentiators*. The ideal frequency response of a causal differentiator is of the form $e^{-jc\omega}\, jD(\omega)$ where

$$D(\omega) = \omega \quad \text{for } 0 < |\omega| < \pi \tag{18.57}$$

and

$$c = (N-1)/2$$

Since the frequency response is antisymmetrical, differentiators can be designed in terms of filters with antisymmetrical impulse response of either odd or even length.

Problem formulation. Assuming odd filter length, Equations 18.54 and 18.57 give the error function

$$E(\omega) = W(\omega)\left[\omega - P'_c(\omega)\right] \quad \text{for } 0 < \omega \le \omega_p$$

where ω_p is the required bandwidth. Equiripple absolute or relative error may be required, depending on the application at hand. Hence, $W(\omega)$ can be chosen to be either unity or $1/\omega$. In the latter case, which is the more meaningful of the two in practice, $E(\omega)$ can be expressed as

$$E(\omega) = 1 - \frac{1}{\omega} P'_c(\omega) \quad \text{for } 0 < \omega \le \omega_p$$

and from Equation 18.55

$$E(\omega) = 1 - \frac{\sin\omega}{\omega} P_{c-1}(\omega) \quad \text{for } 0 < \omega \le \omega_p \tag{18.58}$$

Therefore, the error function can be expressed as in Equation 18.56 with

$$\tilde{W}(\omega) = \frac{1}{\tilde{D}(\omega)} = \frac{\sin\omega}{\omega}$$
$$\tilde{P}(\omega) = P_{c-1}(\omega)$$

Prescribed specifications. A digital differentiator is fully specified by the constraint

$$|E(\omega)| \le \delta_p \quad \text{for } 0 < \omega \le \omega_p$$

where

δ_p is the maximum passband error
ω_p is the bandwidth of the differentiator

The differentiator length N that will just satisfy the required specifications is not normally known *a priori* and, although it may be determined on a hit-or-miss basis, a large number of designs may need to be carried out. In filters with approximately piecewise-constant amplitude responses, N can be predicted using the empirical formula of Equation 18.52. In the case of differentiators, N can be predicted by noting

a useful property of digital differentiators. If δ and δ_1 are the maximum passband errors in differentiators of lengths N and N_1, respectively, then the quantity $\ln(\delta/\delta_1)$ is *approximately linear* with respect to $N - N_1$ for a wide range of values of N_1 and ω_p. Assuming linearity, we can show that [17]

$$N = N_1 + \frac{\ln(\delta/\delta_1)}{\ln(\delta_2/\delta_1)}(N_2 - N_1) \tag{18.59}$$

where δ_2 is the maximum passband error in a differentiator of length N_2.

By designing two low-order differentiators, a fairly accurate prediction of the required value of N can be obtained by using Equation 18.59. Once a filter order is predicted a series of differentiators can be designed with increasing or decreasing N until a design that just satisfies the specifications is obtained.

Example 18.4

The selective step-by-step search with cubic interpolation was used in Algorithm 18.4 to design a digital differentiator that should satisfy the following specifications:

Odd differentiator length
Bandwidth ω_p: 2.75 rad/s
Maximum passband ripple δ_p: 1.0×10^{-5}
Sampling frequency ω_s: 2π rad/s

The progress of the design is illustrated in Table 18.9. First, differentiators of lengths 21 and 23 were designed and the required N to satisfy the specifications was predicted to be 55 using Equation 18.59. This differentiator length was found to satisfy the specifications, and a design for length 53 was then carried out. The second design was found to violate the specifications and hence the first design is the required differentiator. The impulse response of this differentiator is given in Table 18.10. The amplitude response and passband relative error of the differentiator are plotted in Figure 18.20a and b.

18.3.2.6 Arbitrary Amplitude Responses

Very frequently FIR filters are required whose amplitude responses cannot be described by analytical functions. For example, in the design of two-dimensional filters through the singular-value decomposition [18,19], the required two-dimensional filter is obtained by designing a set of one-dimensional digital filters whose amplitude responses turn out to have arbitrary shapes. In these applications, the desired amplitude response $D(\omega)$ is specified in terms of a table that lists a prescribed set of frequencies and the corresponding values of the required filter gain. Filters of this class can be readily designed by employing some interpolation scheme that can be used to evaluate $D(\omega)$ and its first derivative with respect to ω at any ω. A suitable scheme is to fit a set of third-order polynomials to the prescribed amplitude response.

18.3.2.7 Multiband Filters

The algorithms presented in the previous sections can also be used to design *multiband* filters. While there is no theoretical upper limit on the number of bands, in practice, the design tends to become more and more difficult as the number of bands is increased. The reason is that the difference between the number of possible maxima in the error function and the number of extremals increases linearly

TABLE 18.9 Progress in Design of Digital Differentiator (Example 18.4)

N	Iterations	FE's	δ_p
21	4	145	1.075×10^{-2}
23	4	162	6.950×10^{-3}
55	7	815	8.309×10^{-6}
53	7	757	1.250×10^{-5}

TABLE 18.10 Impulse Response of Digital Differentiator (Example 18.4)

n	$h(n) = -h_0(-n)$	n	$h_0(n) = -h_0(-n)$
0	0.0	14	1.762268×10^{-2}
1	-9.933416×10^{-1}	15	-1.313097×10^{-2}
2	4.868036×10^{-1}	16	9.615295×10^{-3}
3	-3.138353×10^{-1}	17	-6.902518×10^{-3}
4	2.245441×10^{-1}	18	4.844090×10^{-3}
5	-1.690252×10^{-1}	19	-3.312235×10^{-3}
6	1.306918×10^{-1}	20	2.197502×10^{-3}
7	-1.024631×10^{-1}	21	-1.407064×10^{-3}
8	8.081083×10^{-2}	22	8.632670×10^{-4}
9	-6.377426×10^{-2}	23	-5.023168×10^{-4}
10	5.016708×10^{-2}	24	2.729367×10^{-4}
11	-3.921782×10^{-2}	25	-1.349790×10^{-4}
12	3.039137×10^{-2}	26	5.859128×10^{-5}
13	-2.329439×10^{-2}	27	-1.634535×10^{-5}

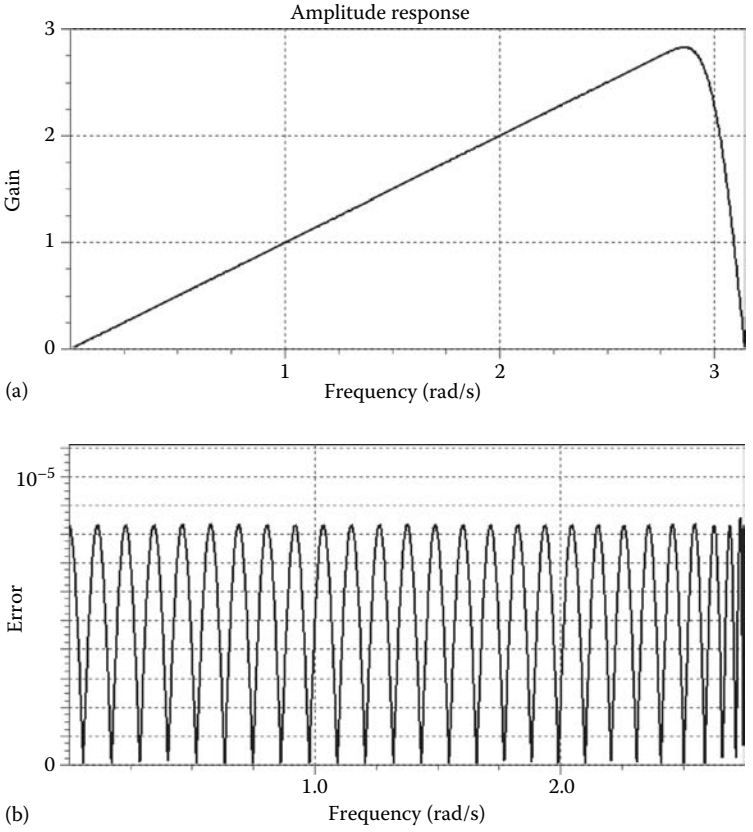


FIGURE 18.20 Design of digital differentiator (Example 18.4): (a) amplitude response and (b) passband relative error.

with the number of bands, e.g., if the number of bands is 8, then the difference is 14. As a consequence, the number of potential extremals that need to be rejected is large and the available rejection techniques become inefficient. The end result is that the number of iterations is increased quite significantly, and convergence is slow and sometimes impossible.

In mathematical terms, the above difficulty is attributed to the fact that, in the weighted-Chebyshev methods considered here, the approximating polynomial becomes seriously *underdetermined* if the number of bands exceeds three. The problem can be overcome by using the generalized Remez method described in Ref. [15]. This approach was found to yield better results for filters with more than four or five bands.

References

1. O. Hermann, Design of nonrecursive digital filters with linear phase, *Electron. Lett*., 6, 182–184, May 1970.
2. E. Hofstetter, A. Oppenheim, and J. Siegel, A new technique for the design of non-recursive digital filters, *5th Annu. Princeton Conf. Informat. Sci. Syst*., 64–72, March 1971.
3. T. W. Parks and J. H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, *IEEE Trans. Circuit Theory*, CT-19, 189–194, March 1972.
4. T. W. Parks and J. H. McClellan, A program for the design of linear phase finite impulse response digital filters, *IEEE Trans. Audio Electroacoust*., AU-20, 195–199, August 1972.
5. L. R. Rabiner and O. Herrmann, On the design of optimum FIR low-pass filters with even impulse response duration, *IEEE Trans. Audio Electroacoust*., AU-21, 329–336, August 1973.
6. J. H. McClellan and T. W. Parks, A unified approach to the design of optimum FIR linear-phase digital filters, *IEEE Trans. Circuit Theory*, CT-20, 697–701, November 1973.
7. J. H. McClellan, T. W. Parks, and L. R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust*, AU-21, 506–526, December 1973.
8. L. R. Rabiner, J. H. McClellan, and T. W. Parks, FIR digital filter design techniques using weighted Chebyshev approximation, *Proc. IEEE*, 63, 595–610, April 1975.
9. J. H. McClellan, T. W. Parks, and L. R. Rabiner, FIR linear phase filter design program, in *Programs for Digital Signal Processing*, New York: IEEE, 1979, pp. 5.1-1–5.1-13.
10. A. Antoniou, Accelerated procedure for the design of equiripple nonrecursive digital filters, *IEE Proc., Pt. G*, 129, 1–10, February 1982 (see *IEE Proc., Pt. G,* vol.129, p.107, June 1982 for errata).
11. A. Antoniou, New improved method for the design of weighted-Chebyshev, nonrecursive, digital filters, *IEEE Trans. Circuits Syst*., CAS-30, 740–750, October 1983.
12. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, New York: McGraw-Hill, 2005.
13. E. W. Cheney, *Introduction to Approximation Theory,* New York: McGraw-Hill, 1966, 72–100.
14. E. Ya. Remez, *General Computational Methods for Tchebycheff Approximation*, Kiev: Atomic Energy Comm., 1957, Translation 4491, 1–85, 1957.
15. D. J. Shpak and A. Antoniou, A generalized Reméz method for the design of FIR digital filters, *IEEE Trans. Circuits Syst*., 37, 161–174, February 1990.
16. O. Herrmann, L. R. Rabiner, and D. S. K. Chan, Practical design rules for optimum finite impulse response low-pass digital filters, *Bell Syst. Tech. J*., 52, 769–799, July–August 1973.
17. A. Antoniou and C. Charalambous, Improved design method for Kaiser differentiators and comparison with equiripple method, *IEE Proc., Pt. E*, 128, 190–196, September 1981.
18. A. Antoniou and W. -S. Lu, Design of two-dimensional digital filters by using the singular value decomposition, *IEEE Trans. Circuits Syst.,* CAS-34, 1191–1198, October 1987.
19. W. -S. Lu, H. -P. Wang, and A. Antoniou, Design of two-dimensional FIR digital filters using the singular-value decomposition, *IEEE Trans. Circuits Syst*., CAS-37, 35–46, January 1990.

18.4 Design of Computationally Efficient FIR Filters Using Periodic Subfilters as Building Blocks

Tapio Saramäki

For many digital signal processing applications, FIR filters are preferred over their IIR counterparts as the former can be designed with exactly linear phase and they are free of stability problems and limit cycle oscillations. The major drawback of FIR filters is that they require, especially in applications demanding narrow transition bands, considerably more arithmetic operations and hardware components than do comparable IIR filters. Ignoring the correction term for very low-order filters, the minimum order of an optimum linear-phase low-pass FIR filter can be approximated [1] by

$$N \approx \Phi(\delta_p, \delta_s)/(\omega_s - \omega_p) \tag{18.60a}$$

where

$$\begin{aligned}\Phi(\delta_p, \delta_s) = {} & 2\pi\left[0.005309(\log_{10}\delta_p)^2 + 0.07114\log_{10}\delta_p - 0.4761\right]\log_{10}\delta_s \\ & - 2\pi\left[0.00266(\log_{10}\delta_p)^2 + 0.5941\log_{10}\delta_p + 0.4278\right]\end{aligned} \tag{18.60b}$$

Here, ω_p and ω_s are the passband and stopband edge angles, whereas δ_p and δ_s are the passband and stopband ripple magnitudes. From the above estimate, it is seen that as the transition bandwidth $\omega_s - \omega_p$ is made smaller, the required filter order increases inversely proportionally to it. Since the direct-form implementation exploiting the coefficient symmetry requires approximately $N/2$ multipliers, this kind of implementation becomes very costly if the transition bandwidth is small.

The cost of implementation of a narrow transition-band FIR filter can be significantly reduced by using multiplier-efficient realizations, fast convolution algorithms, or multirate filtering. This section considers those multiplier-efficient realizations that use as basic building blocks the transfer functions obtained by replacing each unit delay in a conventional transfer function by multiple delays. We concentrate on the synthesis techniques described in Refs. [2–4,6,8–10].

18.4.1 Frequency-Response Masking Approach

A very elegant approach to significantly reducing the implementation cost of an FIR filter has been proposed by Lim [3]. In this approach, the overall transfer function is constructed as

$$H(z) = F(z^L)G_1(z) + \left[z^{-LN_F/2} - F(z^L)\right]G_2(z) \tag{18.61a}$$

where

$$F(z^L) = \sum_{n=0}^{N_F} f(n)z^{-nL}, \quad f(N_F - n) = f(n) \tag{18.61b}$$

$$G_1(z) = z^{-M_1}\sum_{n=0}^{N_1} g_1(n)z^{-n}, \quad g_1(N_1 - n) = g_1(n) \tag{18.61c}$$

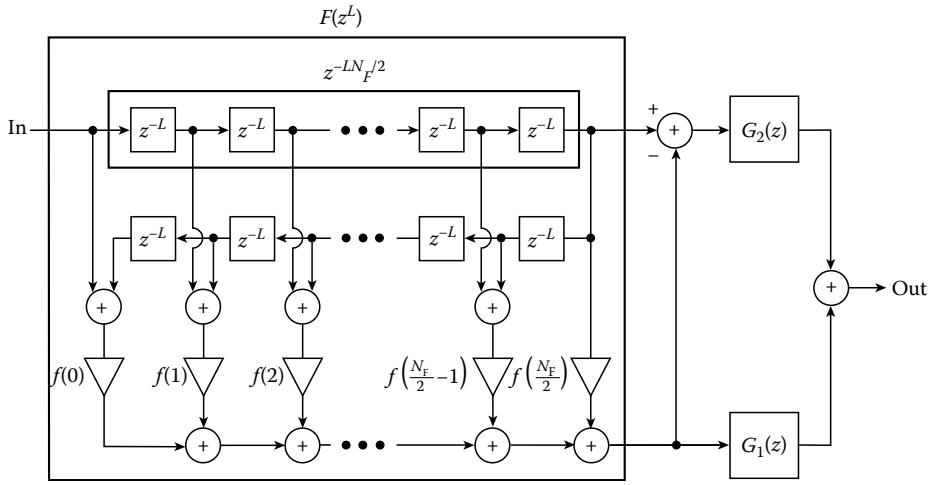


FIGURE 18.21 Efficient implementation for a filter synthesized using the frequency-response masking approach.

and

$$G_2(z) = z^{-M_2}\sum_{n=0}^{N_2} g_2(n)z^{-n}, \quad g_2(N_2 - n) = g_2(n) \tag{18.61d}$$

Here, N_F is even, whereas both N_1 and N_2 are either even or odd. For $N_1 \leq N_2$, $M_1 = 0$ and $M_2 = (N_1 - N_2)/2$, whereas for $N_1 < N_2$, $M_1 = (N_2 - N_1)/2$ and $M_2 = 0$. These selections guarantee that the delays of both of the terms of $H(z)$ are equal. An efficient implementation for the overall filter is depicted in Figure 18.21, where the delay term $z^{-LN_F/2}$ is shared with $F(z^L)$. Also, $G_1(z)$ and $G_2(z)$ can share their delays if a transposed direct-form implementation (exploiting the coefficient symmetry) is used.

The frequency response of the overall filter can be written as

$$H(e^{j\omega}) = H(\omega)e^{-j(LN_F + \max[N_1, N_2])\omega/2} \tag{18.62}$$

where $H(\omega)$ denotes the *zero-phase frequency response* of $H(z)$ and can be expressed as

$$H(\omega) = H_1(\omega) + H_2(\omega) \tag{18.63a}$$

where

$$H_1(\omega) = F(L\omega)G_1(\omega) \tag{18.63b}$$

and

$$H_2(\omega) = [1 - F(L\omega)]G_2(\omega) \tag{18.63c}$$

with

$$F(\omega) = f(N_F/2) + 2\sum_{n=1}^{N_F/2} f(N_F/2 - n)\cos n\omega \tag{18.63d}$$

and

$$G_k(\omega) = \begin{cases} g_k(N_k/2) + 2\sum_{n=1}^{N_k/2} g_k(N_k/2 - n)\cos n\omega & N_k \text{ even} \\ 2\sum_{n=0}^{(N_k-1)/2} g_k[(N_k-1)/2 - n]\cos[(n+1/2)\omega] & N_k \text{ odd} \end{cases} \tag{18.63e}$$

for $k = 1, 2$.

The efficiency as well as the synthesis of $H(z)$ are based on the properties of the pair of transfer functions $F(z^L)$ and $z^{-LN_F/2} - F(z^L)$, which can be generated from the pair of *prototype* transfer functions

$$F(z) = \sum_{n=0}^{N_F} f(n) z^{-n} \tag{18.64}$$

and $z^{-N_F/2} - F(z)$ by replacing z^{-1} by z^{-L}, that is, by substituting for each unit delay L unit delays. The order of the resulting filters is increased to LN_F, but since only every Lth impulse response value is nonzero, the filter complexity (number of adders and multipliers) remains the same. The above prototype pair forms a *complementary* filter pair since their zero-phase frequency responses, $F(\omega)$ and $1 - F(\omega)$ with $F(\omega)$ given by Equation 18.63d, add up to unity. Figure 18.22a illustrates the relations between these responses in the case where $F(z)$ and $z^{-N_F/2} - F(z)$ is a low-pass–high-pass filter pair with edges at θ and ϕ.

The substitution $z^{-L} \rightarrow z^{-1}$ preserves the complementary property resulting in the *periodic* responses $F(L\omega)$ and $1 - F(L\omega)$, which are frequency-axis compressed versions of the prototype responses such that the interval $[0, L\pi]$ is shrunk onto $[0, \pi]$ (see Figure 18.22b). Since the periodicity of the prototype responses is 2π, the periodicity of the resulting responses is $2\pi/L$ and they contain several passband and stopband regions in the interval $[0, \pi]$.

For a low-pass filter $H(z)$, one of the transition bands provided by $F(z^L)$ or $z^{-LN_F/2} - F(z^L)$ is used as that of the overall filter. In the first case, denoted by Case A, the edges are given by (Figure 18.23)

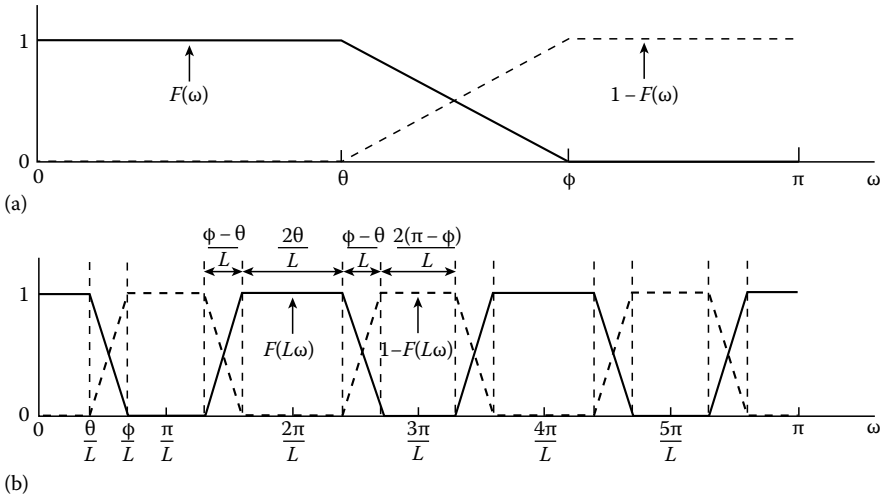


FIGURE 18.22 Generation of a complementary periodic filter pair by starting with a low-pass–high-pass complementary pair. (a) Prototype filter responses $F(\omega)$ and $1 - F(\omega)$. (b) Periodic responses $F(L\omega)$ and $1 - F(L\omega)$ for $L = 6$.

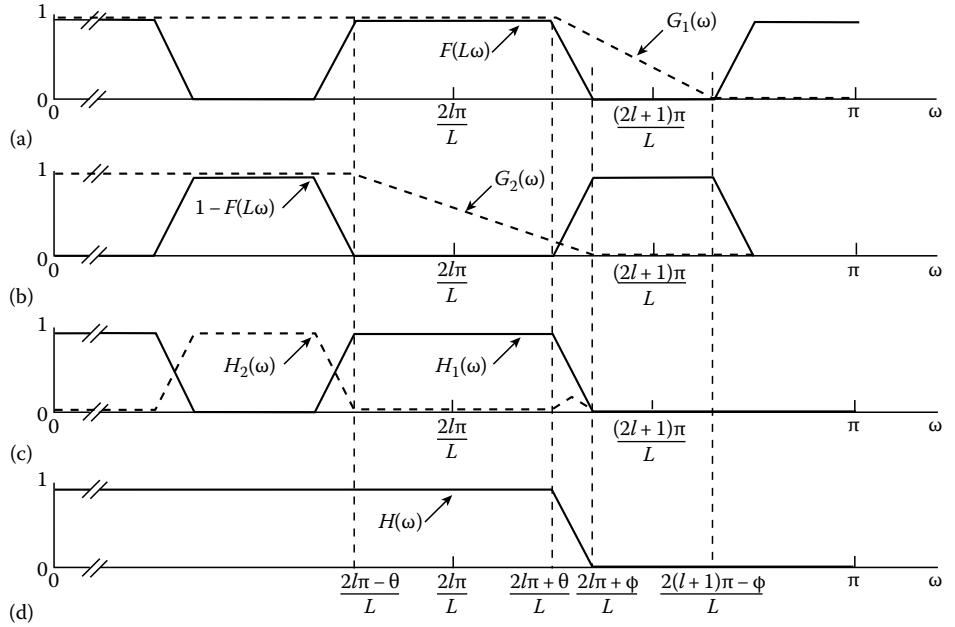


FIGURE 18.23 Case A design of a low-pass filter using the frequency-response masking technique.

$$\omega_p = (2l\pi + \theta)/L \quad \text{and} \quad \omega_s = (2l\pi + \phi)/L \tag{18.65}$$

where l is a fixed integer, and in the second case, referred to as Case B, by (Figure 18.24)

$$\omega_p = (2l\pi - \phi)/L \quad \text{and} \quad \omega_s = (2l\pi - \theta)/L \tag{18.66}$$

The widths of these transition bands are $(\phi - \theta)/L$, which is only $1/L$th of that of the prototype filters. Since the filter order is roughly inversely proportional to the transition bandwidth, this means that the arithmetic complexity of the periodic transfer functions to provide one of the transition bands is only $1/L$th of that of a conventional nonperiodic filter. Note that the orders of both the periodic filters and the corresponding nonperiodic filters are approximately the same, but the conventional filter does not contain zero-valued impulse response samples.

In order to exploit the attractive properties of the periodic transfer functions, the two low-order masking filters $G_1(z)$ and $G_2(z)$ are designed such that the subresponses $H_1(\omega)$ and $H_2(\omega)$ as given by Equations 18.63b and c approximate in the passband $F(L\omega)$ and $1 - F(L\omega)$, respectively, so that their sum approximates unity, as is desired. In the filter stopband, the role of the masking filters is to attenuate the extra unwanted passbands and transition bands of the periodic responses. In Case A, this is achieved by selecting the edges of $G_1(z)$ and $G_2(z)$ as (see Figure 18.23)

$$\omega_p^{(G_1)} = \omega_p = [2l\pi + \theta]/L \quad \text{and} \quad \omega_s^{(G_1)} = [2(l+1)\pi - \phi]/L \tag{18.67a}$$

$$\omega_p^{(G_2)} = [2l\pi - \theta]/L \quad \text{and} \quad \omega_s^{(G_2)} = \omega_s = [2l\pi + \phi]/L \tag{18.67b}$$

Since $F(L\omega) \approx 0$ on $[\omega_s, \omega_s^{(G1)}]$,the stopband region of $G_1(z)$ can start at $\omega = \omega_s{}^{(G1)}$, instead of $\omega = \omega_s$. Similarly, since $H_1(\omega) \approx F(L\omega) \approx 1$ and $[1 - F(L\omega)] \approx 0$ on $[\omega_p^{(G2)}, \omega_p]$, the passband region of $G_2(z)$ can start at $\omega = \omega_p{}^{(G2)}$, instead of $\omega = \omega_p$.

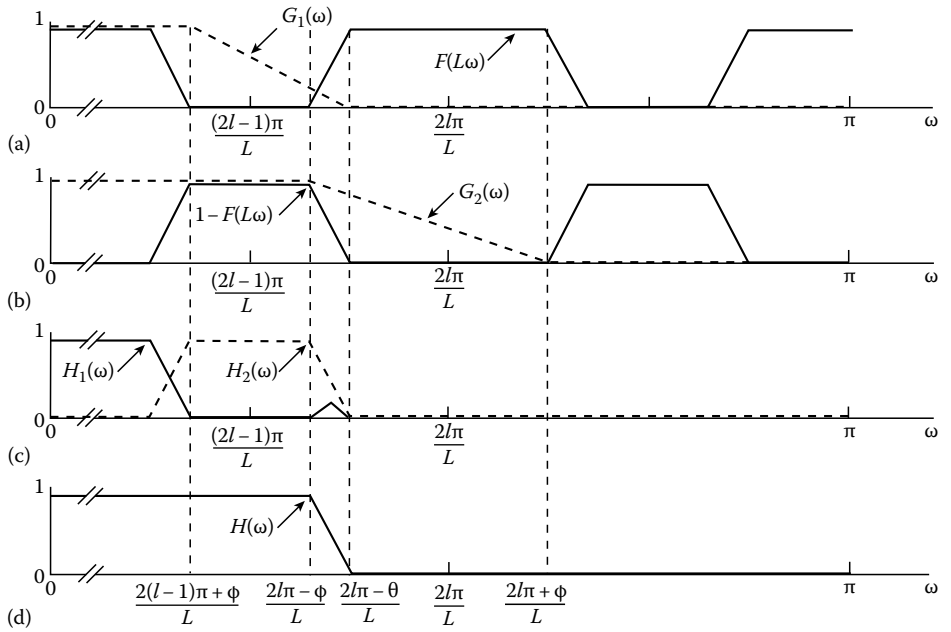


FIGURE 18.24 Case B design of a low-pass filter using the frequency-response masking technique.

For Case B designs, the required edges of the two masking filters, $G_1(z)$ and $G_2(z)$ are (see Figure 18.24)

$$\omega_p^{(G_1)} = [2(l-1)\pi + \phi]/L \quad \text{and} \quad \omega_s^{(G_1)} = \omega_s = [2l\pi - \theta]/L \tag{18.68a}$$

$$\omega_p^{(G_2)} = \omega_p = [2l\pi - \phi]/L \quad \text{and} \quad \omega_s^{(G_2)} = [2l\pi + \theta]/L \tag{18.68b}$$

The effects of the ripples of the subresponses on the ripples of the overall response $H(\omega)$ have been studied carefully in Ref. [3]. Based on these observations, the design of $H(z)$ with passband and stopband ripples of δ_p and δ_s can be accomplished for both Case A and Case B in the following two steps:

1. Design $G_k(z)$ for $k = 1, 2$ using either the Remez algorithm or linear programming such that $G_k(\omega)$ approximates unity on $[0, \omega_p^{(Gk)}]$ with tolerance $0.85\delta_p \cdots 0.9\delta_p$ and zero on $[\omega_s^{(Gk)}, \pi]$ with tolerance $0.85\delta_s \cdots 0.9\delta_s$.
2. Design $F(L\omega)$ such that the overall response $H(\omega)$ approximates unity on

$$\Omega_p^{(F)} = \begin{cases} \left[\omega_p^{(G_2)}, \omega_p\right] = [[2l\pi - \theta]/L, [2l\pi + \theta]/L] & \text{for Case A} \\ \left[\omega_p^{(G_1)}, \omega_p\right] = [[2(l-1)\pi + \phi]/L, [2l\pi - \phi]/L] & \text{for Case B} \end{cases} \tag{18.69a}$$

with tolerance δ_p and approximates zero on

$$\Omega_s^{(F)} = \begin{cases} \left[\omega_s, \omega_s^{(G_1)}\right] = [[2l\pi + \phi]/L, [2(l+1)\pi - \phi]/L] & \text{for Case A} \\ \left[\omega_s, \omega_s^{(G_2)}\right] = [[2l\pi - \theta]/L, [2l\pi + \theta]/L] & \text{for Case B} \end{cases} \tag{18.69b}$$

with tolerance δ_s.

TABLE 18.11 Error Function for Designing $F(\omega)$ Using the Remez Algorithm

$$E_F(\omega) = W_F(\omega)\,[F(\omega) - D_F(\omega)],$$

where

$$D_F(\omega) = [u(\omega) + l(\omega)]/2,\ W_F(\omega) = 2/[u(\omega) - l(\omega)]$$

with

$$u(\omega) = \min(\Psi_1(\omega) + \psi_1(\omega),\ \Psi_2(\omega) + \psi_2(\omega))$$

$$l(\omega) = \max(\Psi_1(\omega) - \psi_1(\omega),\ \Psi_2(\omega) - \psi_2(\omega))$$

$$\Psi_k(\omega) = \frac{D_H[h_k(\omega)] - G_2[h_k(\omega)]}{G_1[h_k(\omega)] - G_2[h_k(\omega)]},\quad k = 1,\ 2$$

$$\psi_k(\omega) = \frac{1/W_H[h_k(\omega)]}{|G_1[h_k(\omega)] - G_2[h_k(\omega)]|},\quad k = 1,\ 2$$

and

$$h_1(\omega) = (2l\pi + \omega)/L,\quad h_2(\omega) = \begin{cases} (2l\pi - \omega)/L & \text{for } \omega \in [0,\ \theta] \\ [2(l+1)\pi - \omega]/L & \text{for } \omega \in [\phi,\ \pi] \end{cases}$$

for Case A and

$$h_1(\omega) = (2l\pi - \omega)/L,\quad h_2(\omega) = \begin{cases} (2l\pi + \omega)/L & \text{for } \omega \in [0,\ \theta] \\ [2(l-1)\pi + \omega]/L & \text{for } \omega \in [\phi,\ \pi] \end{cases}$$

for Case B

The design of $F(L\omega)$ can be performed conveniently using linear programming [3]. Another, computationally more efficient, alternative is to use the Remez algorithm [10]. Its use is based on the fact that

$$|E_{\mathrm{H}}(\omega)| \le 1 \text{ for } \omega \in \Omega_{\mathrm{p}}^{(\mathrm{F})} \cup \Omega_{\mathrm{s}}^{(F)} \tag{18.70a}$$

where

$$E_{\mathrm{H}}(\omega) = W_{\mathrm{H}}(\omega)[H(\omega) - D_{\mathrm{H}}(\omega)] \tag{18.70b}$$

is satisfied when $F(\omega)$ is designed such that the maximum absolute value of the error function given in Table 18.11 becomes less than or equal to unity on $[0,\ \theta] \cup [\phi,\ \pi]$.

For step 2 of the above algorithm, $D_{\mathrm{H}}(\omega) = 1$ and $W_{\mathrm{H}}(\omega) = 1/\delta_{\mathrm{p}}$ on $\Omega_{\mathrm{p}}^{(F)}$, whereas $D_{\mathrm{H}}(\omega) = 0$ and $W_{\mathrm{H}}(\omega) = 1/\delta_s$ on $\Omega_{\mathrm{s}}^{(F)}$, giving for $k = 1,\ 2$

$$D_{\mathrm{H}}[h_k(\omega)] = \begin{cases} 1 & \text{for } \omega \in [0,\ \theta] \\ 0 & \text{for } \omega \in [\phi,\ \pi] \end{cases} \quad \text{and} \quad W_{\mathrm{H}}[h_k(\omega)] = \begin{cases} 1/\delta_{\mathrm{p}} & \text{for } \omega \in [0,\ \theta] \\ 1/\delta_{\mathrm{s}} & \text{for } \omega \in [\phi,\ \pi] \end{cases} \tag{18.71a}$$

for Case A and

$$D_{\mathrm{H}}[h_k(\omega)] = \begin{cases} 1 & \text{for } \omega \in [0,\ \theta] \\ 0 & \text{for } \omega \in [\phi,\ \pi] \end{cases} \quad \text{and} \quad W_{\mathrm{H}}[h_k(\omega)] = \begin{cases} 1/\delta_{\mathrm{s}} & \text{for } \omega \in [0,\ \theta] \\ 1/\delta_{\mathrm{p}} & \text{for } \omega \in [\phi,\ \pi] \end{cases} \tag{18.71b}$$

for Case B. Even though the resulting error function looks very complicated, it is straightforward to use the subroutines EFF and WATE in the Remez algorithm described in Ref. [5] for optimally designing $F(z)$.

The order of $G_1(z)$ can be considerably reduced by allowing larger ripples on those regions of $G_1(z)$ where $F(L\omega)$ has one of its stopbands. As a rule of thumb, the ripples on these regions can be selected to be 10 times larger [3]. Similarly, the order $G_2(z)$ can be decreased by allowing (ten times) larger ripples on those regions where $F(L\omega)$ has one of its passbands.

In practical filter synthesis problems, ω_p and ω_s are given and l, L, θ, and ϕ must be determined. To ensure that Equation 18.65 yields a desired solution with $0 \le \theta < \phi \le \pi$, it is required that (see Figure 18.23)

$$\frac{2l\pi}{L} \le \omega_p \quad \text{and} \quad \omega_s \le \frac{(2l+1)\pi}{L} \tag{18.72a}$$

for some positive integer l, giving

$$l = \lfloor L\omega_p/(2\pi) \rfloor, \quad \theta = L\omega_p - 2l\pi, \quad \text{and} \quad \phi = L\omega_s - 2l\pi \tag{18.72b}$$

where $\lfloor x \rfloor$ stands for the largest integer that is smaller than or equal to x. Similarly, to ensure that Equation 18.66 yields a desired solution with $0 \le \theta < \phi \le \pi$, it is required that (see Figure 18.23)

$$\frac{(2l-1)\pi}{L} \le \omega_p \quad \text{and} \quad \omega_s \le \frac{2l\pi}{L} \tag{18.73a}$$

for some positive integer l, giving

$$l = \lceil L\omega_s/(2\pi) \rceil, \quad \theta = 2l\pi - L\omega_s, \quad \text{and} \quad \phi = 2l\pi - L\omega_p \tag{18.73b}$$

where $\lfloor x \rfloor$ stands for the smallest integer that is larger than or equal to x. For any set of ω_p, ω_s, and L, either Equation 18.72b or Equation 18.73b (not both) will yield the desired θ and ϕ, provided that L is not too large. If $\theta = 0$ or $\phi = \pi$, then the resulting specifications for $F(\omega)$ are meaningless and the corresponding value of L cannot be used.

The remaining problem is to determine L to minimize the number of multipliers, which is $N_P/2 + 1 + 1\lfloor (N_1+2)/2 \rfloor + \lfloor (N_2+2)/2 \rfloor$ or $N_F + N_1 + N_2 + 3$ depending on whether the symmetries in the filter coefficients are exploited or not. Hence, in both cases, a good measure of the filter complexity is the sum of the orders of the subfilters. Instead of determining the actual minimum filter orders for various values of L, the computational workload can be significantly reduced based on the use of the estimation formula given by Equations 18.60a and b. Since the widths of transition bands of $F(z)$, $G_1(z)$, and $G_2(z)$ are $\phi - \theta$, $(2\pi - \phi - \theta)/L$ and $(\phi + \theta)/L$, respectively, good estimates for the corresponding filter orders are

$$N_F \approx \frac{\Phi(\delta_p, \delta_s)}{\phi - \theta}, \quad N_1 \approx \frac{L\Phi(\delta_p, \delta_s)}{2\pi - \phi - \theta}, \quad \text{and} \quad N_2 \approx \frac{L\Phi(\delta_p, \delta_s)}{\phi + \theta} \tag{18.74}$$

For the optimum nonperiodic direct-form design, the transition bandwidth is $\omega_s - \omega_p = (\phi - \theta)/L$, giving

$$N_{opt} \approx \frac{L\Phi(\delta_p, \delta_s)}{\phi - \theta} \tag{18.75}$$

The sum of the subfilter orders can be expressed in terms of N_{opt} as follows:

$$N_{ove} = N_{opt}\left[\frac{1}{L} + \frac{\phi - \theta}{2\pi - \phi - \theta} + \frac{\phi - \theta}{\phi + \theta}\right] \tag{18.76}$$

The smallest values of N_{ove} are typically obtained at those values of L for which $\theta + \phi \approx \pi$ and, correspondingly, $2\pi - \theta - \phi \approx \pi$. In this case, $N_1 \approx N_2$ and Equation 18.76 reduces, after substituting $\phi - \theta = L(\omega_s - \omega_p)$, to

$$N_{ove} = N_{opt}\left[\frac{1}{L} + 2L(\omega_s - \omega_p)/\pi\right] \tag{18.77}$$

At these values of L, N_F decreases and $N_1 \approx N_2$ increases inversely proportionally to L with the minimum of N_{ove} given by

$$N_{ove} = 2N_{opt}\sqrt{\frac{2(\omega_s - \omega_p)}{\pi}} \tag{18.78}$$

taking place at

$$L_{opt} = 1\Big/\sqrt{\frac{2(\omega_s - \omega_p)}{\pi}} \tag{18.79}$$

If for $L = L_{opt}$, $\theta + \phi$ is not approximately equal to π, then L minimizing the filter complexity can be found in the near vicinity of L_{opt}. The following example illustrates the use of the above estimation formulas.

Example 18.1

Consider the specifications: $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$, and $\delta_s = 0.001$. For the optimum conventional direct-form design, $N_{opt} = 2541$, requiring 1271 multipliers when the coefficient symmetry is exploited. Equation 18.79 gives $L_{opt} = 16$. Table 18.12 shows, for the admissible values of L in the vicinity of this value, l, θ, ϕ, the estimated orders for the subfilters, and the sum of the subfilter orders as well as whether the overall filter is a Case A or Case B design. For N_F, the minimum even order larger than or equal to the estimated order is used, whereas N_2 is forced to be even (odd) if N_1 is even (odd).

Also with the estimated filter orders of Table 18.12, $L = 16$ gives the best result. The actual filter orders are $N_F = 162$, $N_1 = 70$, $N_2 = 98$. The responses of the subfilters as well as that of the overall design are depicted in Figure 18.25. The overall number of multipliers and adders for this design are 168 and 330, respectively, which are 13 percent of those required by an equivalent conventional direct-form design (1271 and 2541). The overall filter order is 2690, which is only 6 percent higher than that of the direct-form design (2541).

TABLE 18.12 Estimated Filter Orders for the Admissible Values of L in the Vicinity of $L_{opt} = 16$

L	Case	l	θ	ϕ	N_F	N_1	N_2	$N_F + N_1 + N_2$
8	B	2	0.784π	0.8π	318	98	26	442
9	B	2	0.382π	0.4π	282	38	58	378
11	A	2	0.4π	0.422π	232	47	69	348
12	A	2	0.8π	0.824π	212	162	38	412
13	B	3	0.774π	0.8π	196	155	43	394
14	B	3	0.372π	0.4π	182	58	92	332
16	A	3	0.4π	0.432π	160	70	98	328
17	A	3	0.8π	0.834π	150	236	54	440
18	B	4	0.764π	0.8π	142	210	58	410
19	B	4	0.362π	0.4π	134	78	128	340
21	A	4	0.4π	0.442π	122	92	128	342
22	A	4	0.8π	0.844π	116	314	68	498

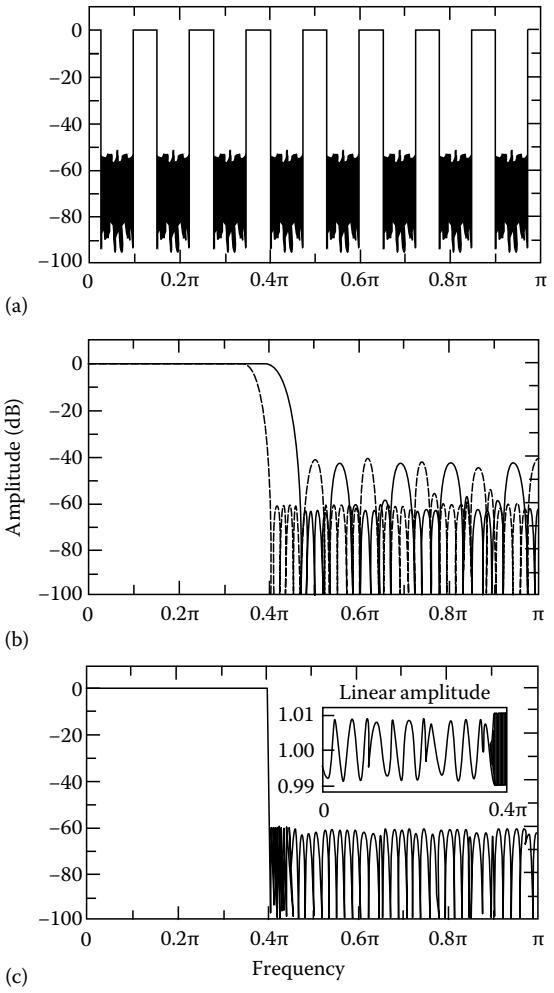


FIGURE 18.25 Amplitude responses for a filter synthesized using the frequency-response masking approach. (a) Periodic response $F(L\omega)$. (b) Responses $G_1(\omega)$ (solid line) and $G_2(\omega)$ (dashed line). (c) Overall response.

18.4.2 Multistage Frequency-Response Masking Approach

If the order of $F(z)$ is too high, its complexity can be reduced by implementing it using the frequency-response masking technique. Extending this to an arbitrary number of stages results in the multistage frequency-response masking approach [3,4], where $H(z)$ is generated iteratively as

$$H(z) \equiv F^{(0)}(z) = F^{(1)}(z^{L_1})G_1^{(1)}(z) + \left[z^{-L_1 N_F^{(1)}/2} - F^{(1)}(z^{L_1})\right] G_2^{(1)}(z) \tag{18.80a}$$

$$F^{(1)}(z) = F^{(2)}(z^{L_2})G_1^{(2)}(z) + \left[z^{-L_2 N_F^{(2)}/2} - F^{(2)}(z^{L_2})\right] G_2^{(2)}(z) \tag{18.80b}$$

$$\vdots$$

$$F^{(R-1)}(z) = F^{(R)}\left(z^{L_R}\right)G_1^{(R)}(z) + \left[z^{-L_R N_F^{(R)}/2} - F^{(R)}\left(z^{L_R}\right)\right] G_2^{(R)}(z) \tag{18.80c}$$

TABLE 18.13 Implementation Form for the Transfer Function in the Multistage Frequency-Response Masking Approach

$$H(z) \equiv F^{(0)}(z^{\hat{L}_0}) = F^{(1)}(z^{\hat{L}_1})G_1^{(1)}(z^{\hat{L}_0}) + [z^{-M_1} - F^{(1)}(z^{\hat{L}_1})]G_2^{(1)}(z^{\hat{L}_0})$$

$$F^{(1)}(z^{\hat{L}_1}) = F^{(2)}(z^{\hat{L}_2})G_1^{(2)}(z^{\hat{L}_1}) + [z^{-M_2} - F^{(2)}(z^{\hat{L}_2})]G_2^{(2)}(z^{\hat{L}_1})$$

$$\vdots$$

$$F^{(R-1)}(z^{\hat{L}_{R-1}}) = F^{(R)}(z^{\hat{L}_R})G_1^{(R)}(z^{\hat{L}_{R-1}}) + [z^{-M_R} - F^{(R)}(z^{\hat{L}_R})]G_2^{(R)}(z^{\hat{L}_{R-1}}),$$

where

$$\hat{L}_0 = 1,\ \hat{L}_r = \prod_{k=1}^{r} L_k, \quad r = 1, 2, \cdots, R$$

$$M_R = \hat{L}_R N_F^{(R)}/2, \quad M_{R-r} = M_{R-r+1} + m_{R-r}, \quad r = 1, 2, \cdots, R-1$$

$$m_{R-r} = \hat{L}_{R-r}\ \max\left\{N_1^{(R-r+1)}, N_2^{(R-r+1)}\right\}/2, \quad r = 1, 2, \cdots, R-1$$

$N_F^{(R)}$ is the order of $F^{(R)}(z)$

$N_1^{(r)}$ and $N_2^{(r)}$ are the orders of $G_1^{(r)}(z)$ and $G_2^{(r)}(z)$, respectively

Here, the $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s for $r = 1, 2, \ldots, R$ as well as $F^{(R)}(z)$ are the filters to be designed. For implementation purposes, $H(z)$ can be expressed in the form shown in Table 18.13. Figure 18.26 shows an efficient implementation for a three-stage filter, where the delay terms z^{-M3}, z^{-m2}, and z^{-m1}, can be shared with $F^{(3)}(z^{L_3})$. In order to obtain a desired overall solution, the orders of the $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s for $r = 2, 3, \ldots, R$, denoted by $N_1^{(r)}$ and $N_2^{(r)}$ in Table 18.13, have to be even.

Given the filter specifications and the L_r's for $r = 1, 2, \ldots, R$, the $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s as well as $F^{(R)}(z)$ can be synthesized in the following steps:

1. Set $r = 1$, $L = L_1$, and

$$D_H(\omega) = \begin{cases} 1 & \text{for } \omega \in [0, \omega_p] \\ 0 & \text{for } \omega \in [\omega_s, \pi] \end{cases}, \quad W_H(\omega) = \begin{cases} 1/\delta_p & \text{for } \omega \in [0, \omega_p] \\ 1/\delta_s & \text{for } \omega \in [\omega_s, \pi] \end{cases} \tag{18.81}$$

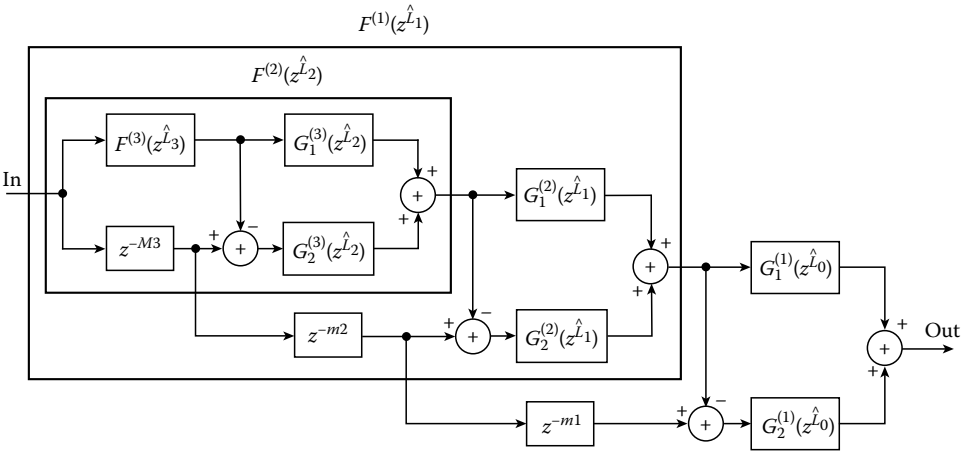


FIGURE 18.26 An implementation for a filter synthesized using the three-stage frequency-response masking approach.

2. Determine whether $F^{(r-1)}(z)$ is a Case A or Case B design as well as θ, ϕ, and l for $F^{(r)}(z)$ according to Equation 18.72b or Equation 18.73b. Also, determine $\omega_p^{(Gk)}$ and $\omega_s^{(Gk)}$ for $k = 1, 2$ from Equations 18.67a and b or Equations 18.68a and b.
3. Design $G_k^{(r)}(z)$ for $k = 1, 2$, using either the Remez algorithm or linear programming, in such a way that

$$\max_{\omega \in \left[0,\ \omega_p^{(G_k)}\right] \cup \left[\omega_s^{(G_k)},\ \pi\right]} \left| W_H(\omega)\left[G_k^{(r)}(\omega) - D_H(\omega)\right]\right| \leq 0.9 \tag{18.82}$$

4. Determine $W_F(\omega)$ and $D_F(\omega)$ from Table 18.11.
5. If $r = R$, then go to the next step. Otherwise, set $r = r + 1$, $L = L_r$, $W_H(\omega) = W_F(\omega)$, $D_H(\omega) = D_F(\omega)$, $\omega_p = \theta$, $\omega_s = \phi$, and go to step 2.
6. Design $F^{(R)}(z)$, using either the Remez algorithm or linear programming, in such a way that

$$\max_{\omega \in [0,\theta] \cup [\phi,\pi]} \left| W_F(\omega)\left[F^{(R)}(\omega) - D_F(\omega)\right]\right| \leq 1 \tag{18.83}$$

In the above algorithm, $G_1^{(1)}(z)$ and $G_2^{(1)}(z)$ are determined like in the one-stage frequency-response masking technique. The remaining filter part as given by Equation 18.80b has then to be designed such that the maximum absolute value of the error function given in Table 18.11 becomes less than or equal to unity on $[0, \theta] \cup [\phi, \pi]$. Using the substitutions $\omega_p = \theta$ and $\omega_s = \phi$, the synthesis problem for $F^{(1)}(z)$ becomes the same as for the overall filter with the only exception that the desired function $D_F(\omega)$ and the weighting function $W_F(\omega)$ are not constants in the passband and stopband regions. Therefore, the following $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s can be designed in the same manner. Finally, $F^{(R)}(z)$ is determined at step 6 like $F(z)$ in one-stage designs.

Given the filter specifications, the remaining problem is to select R as well as the L_r's to minimize the filter complexity. This problem has been considered in Ref. [4]. Assuming that for all the selected L_r's, $\theta + \phi \approx \pi$, the sum of the estimated orders of $F^{(R)}(z)$ and the $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s becomes

$$N_{ove}(R) = \left[1 \Big/ \prod_{r=1}^{R} L_r + [2(\omega_s - \omega_p)/\pi] \sum_{r=1}^{R} L_r\right] N_{opt} \tag{18.84}$$

The minimum of $N_{ove}(R)$ taking place at

$$L_1 = L_2 = \cdots = L_R = L_{opt}(R) = \left[\frac{2(\omega_s - \omega_p)}{\pi}\right]^{-1/(R+1)} \tag{18.85}$$

is

$$N_{ove}(R) = (R+1)\left[\frac{(2\omega_s - \omega_p)}{\pi}\right]^{R/(R+1)} N_{opt} \tag{18.86}$$

The derivation of the above formula is based on the assumption that the orders of all the $G_1^{(r)}(z)$'s and $G_2^{(r)}(z)$'s for $r = 1, 2, \ldots, R$ are equal, which is seldom true. Therefore, in order to minimize the overall filter complexity, the values of the L_r's should be varied in the vicinity of $L_{opt}(R)$. Given ω_p, ω_s, and R, good values for the L_r's can be obtained by the following procedure:

1. Set $r = 1$.
2. Determine $L = L_{opt}(R + 1 - r)$ from Equation 18.85.
3. For values of $\tilde{L}_r$ in the vicinity of L determine $\theta(\tilde{L}_r)$ and $\phi(\tilde{L}_r)$.

4. If $r = R$, then go to step 7. Otherwise, go to the next step.
5. Determine $L_r = \tilde{L}_r$ minimizing

$$(R+1-r)\left[\frac{2[\phi(\tilde{L}_r)-\theta(\tilde{L}_r)]}{\pi}\right]^{(R-r)/(R+1-r)}+\frac{\phi(\tilde{L}_r)-\theta(\tilde{L}_r)}{\theta(\tilde{L}_r)+\phi(\tilde{L}_r)}+\frac{\phi(\tilde{L}_r)-\theta(\tilde{L}_r)}{2\pi-\theta(\tilde{L}_r)-\phi(\tilde{L}_r)} \tag{18.87}$$

6. Set $r = r+1$, $\omega_p = \theta(L_r)$, $\omega_s = \phi(L_r)$, and go to step 2.
7. Determine $L_R = \tilde{L}_r$ minimizing

$$\frac{1}{\tilde{L}_R}+\frac{\phi(\tilde{L}_R)-\theta(\tilde{L}_R)}{\theta(\tilde{L}_R)+\phi(\tilde{L}_R)}+\frac{\phi(\tilde{L}_R)-\theta(\tilde{L}_R)}{2\pi-\theta(\tilde{L}_R)-\phi(\tilde{L}_R)} \tag{18.88}$$

At the first step in this procedure, L_1 is determined to minimize the estimated overall complexity of $G_1^{(1)}(z)$, $G_2^{(1)}(z)$, and the remaining $F_1^{(1)}(z)$, which is given by Equation 18.87 as a fraction of N_{opt}. Compared to Equation 18.76 for the one-stage design, $1/\tilde{L}_r$ is replaced in Equation 18.87 by the first term. This term estimates the complexity of $F_1^{(1)}(z)$ based on the use of Equation 18.86 with $\omega_p = \theta(\tilde{L}_r)$ and $\omega_s = \phi(\tilde{L}_r)$ and the fact that it is an $R-1$ stage design. Also, L_2 is redetermined based on the same assumptions and the process is continued in the same manner. Finally, L_R is determined to minimize the sum of the estimated orders of $G_1^{(R)}(z)$, $G_2^{(R)}(z)$, and $F^{(R)}(z)$ like in the one-stage design (cf. Equation 18.76).

Example 18.2

Consider the specifications of Example 18.1. For a two-stage design, the above procedure gives $L_1 = L_2 = 6$. For these values, $F^{(0)}(z) + H(z)$ and $F^{(1)}(z)$ are Case A designs ($l = 1$) with $\theta = 0.4\pi$ and $\phi = 0.412\pi$; and $\theta = 0.4\pi$ and $\phi = 0.472\pi$, respectively. The minimum orders of $G_1^{(1)}(z)$, $G_2^{(1)}(z)$, $G_1^{(2)}(z)$, $G_2^{(2)}(z)$, and $F^{(2)}(z)$ are 26, 40, 28, 36, and 74, respectively. Compared with the conventional direct-form FIR filter of order 2541, the number of multipliers and adders required by this design (107 and 204) are only 8 percent at the expense of a 15 percent increase in the overall filter order (to 2920). For a three-stage design, we get $L_1 = L_2 = L_3 = 4$. In this case, $F^{(0)}(z)$, $F^{(1)}(z)$, and $F^{(2)}(z)$ are Case B designs ($l = 1$) with $\theta = 0.392\pi$ and $\phi = 0.4\pi$; $\theta = 0.4\pi$ and $\phi = 0.432\pi$; and $\theta = 0.272\pi$ and $\phi = 0.4\pi$, respectively. The minimum orders of $G_1^{(1)}(z)$, $G_2^{(1)}(z)$, $G_1^{(2)}(z)$, $G_2^{(2)}(z)$, $G_1^{(3)}(z)$, $G_2^{(3)}(z)$, and $F^{(3)}(z)$ are 16, 28, 18, 24, 16, 32, and 40, respectively. The number of multipliers and adders (94 and 174) are only 7 percent of those required by the direct-form equivalent at the expense of a 26 percent increase in the overall filter order (to 3196). The amplitude responses of the resulting two-stage and three-stage designs are depicted in Figure 18.27.

18.4.3 Design of Narrowband Filters

Another general approach for designing multiplier-efficient FIR filters has been proposed by Jing and Fam [2]. This design technique is based on iteratively using the fact that there exist efficient implementation forms for filters with $\omega_s < \pi/2$ and for filters with $\omega_p > \pi/2$. A filter with $\omega_s < \pi/2$ is called a *narrowband* filter while that with $\omega_p > \pi/2$ is called a *wideband* filter. This section considers the design of narrowband filters, whereas Section 18.4.4 is devoted to the design of wideband filters. Finally, these techniques are combined, resulting in the Jing–Fam approach.

When the stopband edge of $H(z)$ is less than $\pi/2$, the first transition band of $F(z^L)$ can be used as that of $H(z)$ (Figure 18.28), that is,

$$\omega_p = \theta/L \quad \text{and} \quad \omega_s = \phi/L \tag{18.89}$$

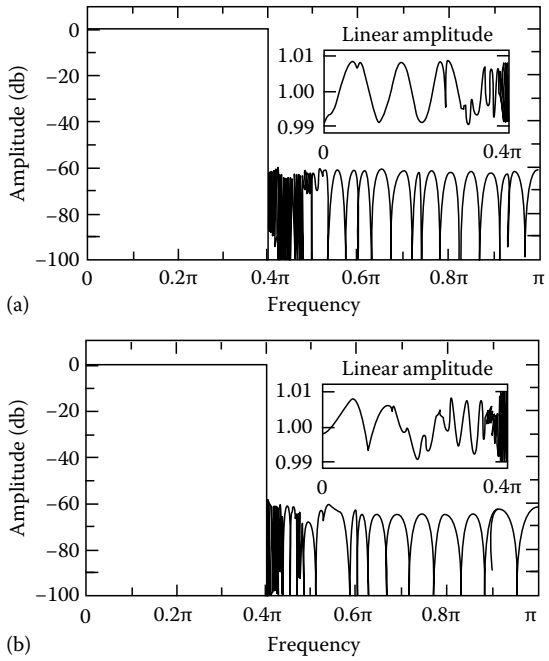


FIGURE 18.27 Amplitude responses for filters synthesized using the multistage frequency-response masking approach. (a) Two-stage filter. (b) Three-stage filter.

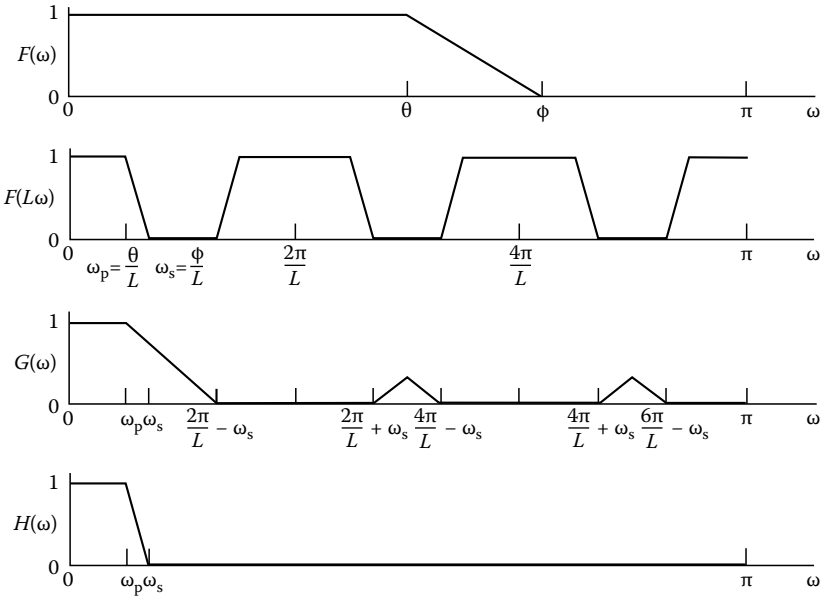


FIGURE 18.28 Synthesis of a narrowband filter as a cascade of a periodic and a nonperiodic filter.

In this case, the overall transfer function can be written in the following simplified form [6,8]:

$$H(z) = F(z^L)G(z) \tag{18.90}$$

where the orders of both $F(z)$ and $G(z)$ can be freely selected to be either even or odd. As shown in Figure 18.28, the role of $G(z)$ is to provide the desired attenuation on those regions where $F(z^L)$ has extra unwanted passband and transition band regions, that is, on

$$\Omega_s(L, \omega_s) = \bigcup_{k=1}^{\lfloor L/2 \rfloor} \left[k\frac{2\pi}{L} - \omega_s,\ \min\left(k\frac{2\pi}{L} + \omega_s, \pi \right) \right] \tag{18.91}$$

There exist two ways of designing the subfilters $F(z^L)$ and $G(z)$. In the first case, they are determined, by means of the Remez algorithm, to satisfy

$$1 - \delta_p^{(F)} \le F(\omega) \le 1 + \delta_p^{(F)} \quad \text{for } \omega \in [0,\ L\omega_p] \tag{18.92a}$$

$$-\delta_s \le F(\omega) \le \delta_s \quad \text{for } \omega \in [L\omega_s,\ \pi] \tag{18.92b}$$

$$1 - \delta_p^{(G)} \le G(\omega) \le 1 + \delta_p^{(G)} \quad \text{for } \omega \in [0,\ \omega_p] \tag{18.92c}$$

$$-\delta_s \le G(\omega) \le \delta_s \quad \text{for } \omega \in \Omega_s(L,\ \omega_s) \tag{18.92d}$$

where

$$\delta_p^{(G)} + \delta_p^{(F)} = \delta_p \tag{18.92e}$$

The ripples $\delta_p^{(F)}$ and $\delta_p^{(G)}$ can be selected, e.g., to be half the overall ripple δ_p. In the above specifications, both $F(z^L)$ and $G(z)$ have $[0,\ \omega_p]$ as a passband region.

Another approach, leading to a considerable reduction in the order of $G(z)$, is to design simultaneously $F(\omega)$ to satisfy

$$1 - \delta_p \le F(\omega)G(\omega/L) \le 1 + \delta_p \quad \text{for } \omega \in [0,\ L\omega_p] \tag{18.93a}$$

$$-\delta_s \le F(\omega)G(\omega/L) \le \delta_s \quad \text{for } \omega \in [L\omega_s,\ \pi] \tag{18.93b}$$

and $G(\omega)$ to satisfy

$$G(0) = 1 \tag{18.94a}$$

$$-\delta_s | le F(L\omega)G(\omega) \le \delta_s \quad \text{for } \omega \ \in\ \Omega_s(L,\ \omega_s) \tag{18.94b}$$

The desired overall solution can be obtained by iteratively determining, by means of the Remez algorithm, $F(z)$ to meet the criteria of Equations 18.93a and b and $G(z)$ to meet the criteria of Equations 18.94a and b. Typically, only three to five designs of both of the subfilters are required to arrive at a solution that does not change if further iterations are used. For more details, see Ref. [8] or [10]. Figure 18.29 shows typical responses for $G(z)$ and $F(z^L)$ and for the overall optimized design. As seen in this figure, $G(z)$ has all its zeros on the unit circle concentrating on providing the desired attenuation for the overall response on $\Omega_s(L, \omega_s)$, whereas $F(z^L)$ makes the overall response equiripple in the passband and in the stopband portion $[\omega_s,\ \pi/L]$.

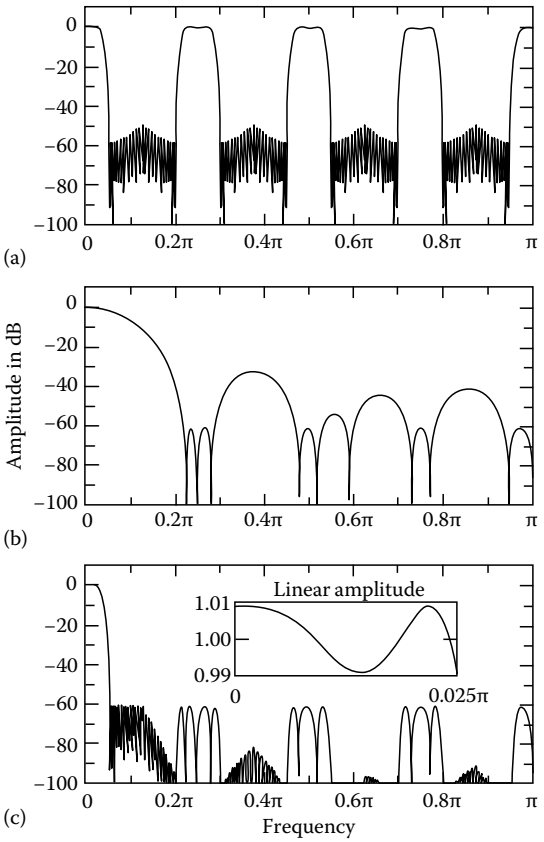


FIGURE 18.29 Typical amplitude responses for a filter of the form $H(z) = F(z^L)G(z)$. $L = 8$, $\omega_p = 0.025\pi$, $\omega_s = 0.05\pi$, $\delta_p = 0.01$, and $\delta_s = 0.001$. (a) $F(z^L)$ of order 26 in z^L. (b) $G(z)$ of order 19. (c) Overall filter.

For the order of $F(z)$, a good estimate is

$$N_F \approx \frac{\Phi(\delta_p, \delta_s)/L}{\omega_s - \omega_p} \tag{18.95}$$

so that it is $1/L$th of that of an optimum conventional nonperiodic filter meeting the given overall criteria. The order of $G(z)$, in turn, can be estimated accurately by [10]

$$N_G = \cosh^{-1}(1/\delta_s)\left[\frac{1}{X\left(\omega_p, \frac{2\pi}{L} - \frac{\omega_p + 2\omega_s}{3}\right)} + \frac{L/2}{X\left(\frac{L\omega_p}{2}, \pi - \frac{L(\omega_p + 2\omega_s)}{6}\right)}\right] \tag{18.96a}$$

where

$$X(\omega_1, \omega_2) = \cosh^{-1}[(2\cos\omega_1 - \cos\omega_2 + 1)/(1 + \cos\omega_2)] \tag{18.96b}$$

The minimization of the number of multipliers, $[(N_F + 2)/2] + [(N_G + 2)/2]$, with respect to L can be performed conveniently by evaluating the sum of the above estimated orders for the admissible values of L,

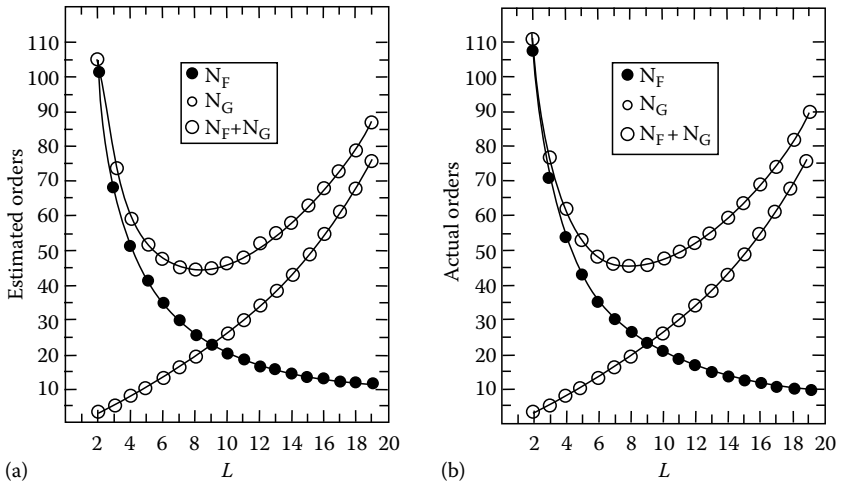


FIGURE 18.30 Estimated and actual subfilter orders as well as the sum of the subfilter orders vs. L in a typical narrowband case.

$2 \leq L < \pi/\omega_s$. The upper limit is a consequence of the fact that the stopband edge angle of $F(z)$, $\phi = L\omega_s$, must be less than π. The following example illustrates the minimization of the filter complexity.

Example 18.3

The narrowband specifications are $\omega_p = 0.025\pi$, $\omega_s = 0.05\pi$, $\delta_p = 0.01$, and $\delta_s = 0.001$. Figure 18.30a shows the estimated N_F, N_G, and $N_F + N_G$ as functions of L, whereas Figure 18.30b shows the corresponding actual minimum orders. It is seen that the estimated orders are so close to the actual ones that the minimization of the filter complexity can be accomplished based on the use of the above estimation formulas. It is also observed that $N_F + N_G$ is a unimodal function of L. With the estimates, $L = 8$ gives the best result. The estimated orders are $N_F = 25$ and $N_G = 19$, whereas the actual orders are $N_F = 26$ and $N_G = 19$. The amplitude responses for the subfilters and for the overall filter are depicted in Figure 18.29. This design requires 24 multipliers and 45 adders. The minimum order of a conventional direct-form design is 216, requiring 109 multipliers and 216 adders. The price paid for these 80% reductions in the filter complexity is a 5 percent increase in the overall filter order (from 216 to 227).

In the cases where L can be factored into the product

$$L = \prod_{r=1}^{R} L_r \tag{18.97}$$

where the L_r's are integers, further savings in the filter complexity can be achieved by designing $G(z)$ in the following multistage form [8]:

$$G(z) = G_1(z)G_2(z^{L_1})G_3(z^{L_1 L_2}) \cdots G_R(z^{L_1 L_2 \cdots L_{R-1}}) \tag{18.98}$$

Another alternative to reduce the number of adders and multipliers is to use special structures for implementing $G(z)$ [8–10].

18.4.4 Design of Wideband Filters

The synthesis of a wideband filter $H(z)$ can be converted into the design of a narrowband filter based on the following fact. If $\hat{H}(z)$ of even order $2M$ is a low-pass design with the following edges and ripples:

$$\hat{\omega}_p = \pi - \omega_s, \quad \hat{\omega}_s = \pi - \omega_p, \quad \hat{\delta}_p = \delta_s, \quad \hat{\delta}_s = \delta_p \tag{18.99}$$

then

$$H(z) = z^{-M} - (-1)^M \hat{H}(-z) \tag{18.100}$$

is a low-pass filter having the passband and stopband edge angles at ω_p and ω_s and the passband and stopband ripples of δ_p and δ_s. Hence, if ω_p and ω_s of $H(z)$ are larger than $\pi/2$, then $\hat{\omega}_p$ and $\hat{\omega}_s$ of $\hat{H}(z)$ are smaller than $\pi/2$ [10]. This enables us to design $\hat{H}(z)$ in the form

$$\hat{H}(z) = F(z^L)G(z) \tag{18.101}$$

using the techniques of Section 18.4.3, yielding

$$H(z) = z^{-M} - (-1)^M F\left[(-z)^L\right] G(-z) \tag{18.102a}$$

where

$$M = (LN_F + N_G)/2 \tag{18.102b}$$

is half the order of $F(z^L)G(z)$. For implementation purposes, $H(z)$ is expressed as

$$H(z) = z^{-M} - \hat{F}(z^L)\hat{G}(z), \quad \hat{F}(z^L) = (-1)^M F[(-z)^L], \quad \hat{G}(z) = G(-z) \tag{18.103}$$

An implementation of this transfer function is shown in Figure 18.31, where the delay term z^{-M} can be shared with $\hat{F}(z^L)$. To avoid half-sample delays, the order of $\hat{F}(z^L)\hat{G}(z)$ has to be even.

Example 18.4

The wideband specifications are $\omega_p = 0.95\pi$, $\omega_s = 0.975\pi$, $\delta_p = 0.001$, and $\delta_s = 0.01$. From Equation 18.99, the specifications of $\hat{H}(z)$ become $\omega_p = 0.025\pi$, $\hat{\omega}_5 = 0.05\pi$, $\hat{\delta}_p = 0.01$, and $\hat{\delta}_s = 0.001$. These are the narrowband specifications of Example 18.3. The desired wideband design is thus obtained by using the subfilters $F(z^L)$ and $G(z)$ of Figure 18.29 ($L = 8$, $N_F = 26$, and $N_G = 19$). However, the overall order is odd (227). A solution with even order is achieved by increasing the order of $G(z)$ by one ($N_G = 20$). Figure 18.32 shows the amplitude response of the resulting filter, requiring 25 multipliers, 46 adders, and 228 delay elements. The corresponding numbers for a conventional direct-form equivalent of order 216 are 109, 216, and 216, respectively.

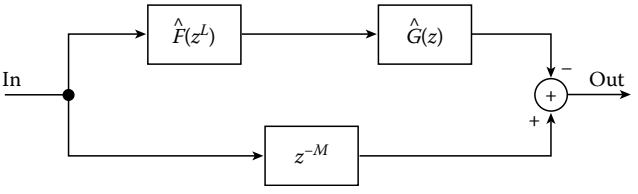


FIGURE 18.31 Implementation for a wideband filter in the form $H(z) = z^{-M} - (-1)^M F[(-z^L)G(-z)$. $\hat{F}(z^L) = (-1)^M \cdot F[(-z)^L]$ and $\hat{G}(z) = G(-z)$.

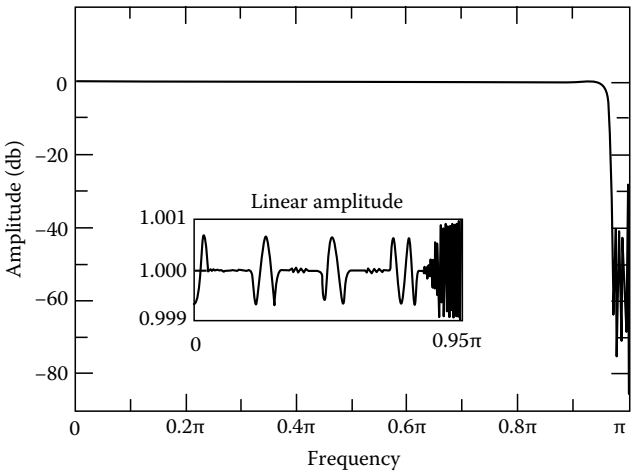


FIGURE 18.32 Amplitude response for a wideband filter implemented as shown in Figure 18.38.

18.4.5 Generalized Designs

The Jing–Fam approach [2] is based on iteratively using the facts that a narrowband filter can be implemented effectively as $H(z) = F(z^L)G(z)$ and a wideband filter in the form of Equations 18.102a and b. In this approach, a narrowband filter is generated [9] as

$$H(z) \equiv \hat{H}_1(z) = G_1(z)F_1(z^{L_1}) \tag{18.104a}$$

where

$$F_1(z) = z^{-M_1} - (-1)^{M_1}\hat{H}_2(-z) \quad \hat{H}_2(z) = G_2(z)F_2(z^{L_2}) \tag{18.104b}$$

$$F_2(z) = z^{-M_2} - (-1)^{M_2}\hat{H}_3(-z) \quad \hat{H}_3(z) = G_3(z)F_3(z^{L_3}) \tag{18.104c}$$

$$\vdots$$

$$F_{R-2}(z) = z^{-M_{R-2}} - (-1)^{M_{R-2}}\hat{H}_{R-1}(-z) \quad \hat{H}_{R-1}(z) = G_{R-1}(z)F_{R-1}(z^{L_{R-1}}) \tag{18.104d}$$

$$F_{R-1}(z) = z^{-M_{R-1}} - (-1)^{M_{R-1}}\hat{H}_R(-z) \quad \hat{H}_R(z) = G_R(z) \tag{18.104e}$$

with M_r for $r = 1, 2, \ldots, R-1$ being half the order of $\hat{H}_{r+1}(z)$. Here, the basic idea is to convert iteratively the design of the narrowband overall filter into the designs of narrowband transfer function $\hat{H}_r(z)$ for $r = 2, 3, \ldots, R$ until the transition bandwidth of the remaining $\tilde{H}_r(z) = G_R(z)$ becomes large enough and, correspondingly, its complexity (the number of multipliers) is low enough. The desired conversion is performed by properly selecting the L_r's and designing the lower-order filters $G_r(z)$ for $r = 1, 2, \ldots, R-1$.

In order to determine the conditions for the L_r's as well as the design criteria for the $G_r(z)$'s, we consider the rth iteration, where

$$\hat{H}_r(z) = G_r(z)F_r(z^{L_r}) \tag{18.105a}$$

with

$$F_r(z) = z^{-M_r} - (-1)^{M_r}\hat{H}_{r+1}(-z) \tag{18.105b}$$

Let the ripples of $\hat{H}_r(z)$ be $\hat{\delta}_p^{(r)}$ and $\hat{\delta}_s^{(r)}$ and the edges be located at $\omega_p^{(r)} < \pi/2$ and $\omega_s^{(r)} < \pi/2$. Since $F_r(z)$ is implemented in the form of Equation 18.105b, it cannot alone take care of shaping the passband response of $\hat{H}_r(z)$. Therefore, the simultaneous criteria for $G_r(z)$ and $F_r(z)$ are stated according to Equations 18.92a through e so that the passband and stopband regions of $G_r(z)$ are, respectively, $[0,\omega_p^{(r)}]$ and $\Omega_s L_r \omega_s^{(r)})$ with $\Omega_s(L, \omega_s)$ given by Equation 18.91. L_r has to be determined such that the edges of $F_r(z)$, $L_{r\omega p}^{(r)}$ and $L_{r\omega s}^{(r)}$, become larger than $\pi/2$ and, correspondingly, the edges of $\hat{H}_{r+1}(z)$, $\omega_p^{(r+1)} = \pi - L_{r\omega s}^{(r)}$ and $\omega_s^{(r+1)} = \pi - L_{r\omega p}^{(r)}$, become less than $\pi/2$.

In the case of the specifications of Equations 18.92a through e, the stopband ripple of $G_r(z)$, denoted for later use by $\delta_s^{(r)}$, and that of $F_r(z)$ are equal to $\hat{\delta}_s^{(r)}$, whereas the sum of the passband ripples is equal to $\hat{\delta}_p^{(r)}$. Denoting by $\hat{\delta}_p^{(r)}$ the passband ripple selected for $G_r(z)$, the corresponding ripple of $F_r(z)$ is $\hat{\delta}_p^{(r)} - \delta_p^{(r)}$. Since $F_r(z)$ and $\hat{H}_{(r+1)}(z)$ interchange the ripples, the ripple requirements for $\hat{H}_{r+1}(z)$ are $\hat{\delta}_p^{(r+1)} = \hat{\delta}_s^{(r)}$ and $\hat{\delta}_s^{(r+1)} = \hat{\delta}_p^{(r)} - \delta_p^{(r)}$.

The criteria for the $G_r(z)$'s for $r = 1, 2, \ldots, R$ can thus be stated as

$$1 - \delta_p^{(r)} \le G_r(\omega) \le 1 + \delta_p^{(r)} \quad \text{for } \omega \in [0, \omega_p^{(r)}] \tag{18.106a}$$

$$-\delta_s^{(r)} \le G_r(\omega) \le \delta_s^{(r)} \quad \text{for } \omega \in \Omega_s^{(r)} \tag{18.106b}$$

where

$$\Omega_s^{(r)} = \begin{cases} \bigcup_{k=1}^{\lfloor L_r/2 \rfloor} \left[k\frac{2\pi}{L_r} - \omega_s^{(r)},\ \min\left(k\frac{2\pi}{L_r} + \omega_s^{(r)},\ \pi \right) \right] & \text{for } r < R \\ \left[\omega_s^{(R)},\ \pi \right] & \text{for } r = R \end{cases} \tag{18.106c}$$

Here, the $\omega_p^{(r)}$'s and $\omega_s^{(r)}$'s for $R = 2, 3, \ldots, R$ are determined iteratively as

$$\omega_p^{(r)} = \pi - L_{r-1}\omega_s^{(r-1)}, \quad \omega_s^{(r)} = \pi - L_{r-1}\omega_p^{(r-1)} \tag{18.106d}$$

where $\omega_p^{(1)} = \omega_p$ and $\omega_s^{(1)} = \omega_s$ are the edges of the overall design, and the $\delta_s^{(r)}$'s as

$$\delta_s^{(r)} = \begin{cases} \delta_p - \sum_{\substack{k=1 \\ k \text{ odd}}}^{r-1} \delta_p^{(k)} & \text{for } r \text{ even} \\ \delta_s - \sum_{\substack{k=2 \\ k \text{ even}}}^{r-1} \delta_p^{(k)} & \text{for } r \text{ odd} \end{cases} \tag{18.106e}$$

where δ_p and δ_s are the ripple values of the overall filter and $\delta_p^{(r)}$ is the passband ripple selected for $G_r(z)$. In order for the overall filter to meet the given ripple requirements, $\delta_s^{(R)}$ and the $\delta_p^{(r)}$'s have to satisfy for R even

$$\sum_{\substack{k=2 \\ k \text{ even}}}^{R} \delta_p^{(k)} = \delta_s, \quad \delta_s^{(R)} + \sum_{\substack{k=1 \\ k \text{ odd}}}^{R-1} \delta_p^{(k)} = \delta_p \tag{18.107a}$$

or for R odd

$$\sum_{\substack{k=1 \\ k \text{ odd}}}^{R} \delta_p^{(k)} = \delta_p, \quad \delta_s^{(R)} + \sum_{\substack{k=2 \\ k \text{ even}}}^{R-1} \delta_p^{(k)} = \delta_s \tag{18.107b}$$

In the above, the L_r's have to be determined such that the $\omega_s^{(r)}$'s for $r < R$ become smaller than $\pi/2$. It is also desired that for the last filter stage $G_R(z)$, $\omega_s^{(R)}$ is smaller than $\pi/2$.

If $2\pi/L_r - \omega_s^{(r)} < \pi/2$ for $r < R$ or $\omega_s^{(R)} < \pi/2$, then the arithmetic complexity of $G_r(z)$ can be reduced by designing it, using the techniques of previous sections, in the form

$$G_r(z) = G_r^{(1)}(z^{K_r})G_r^{(2)}(z) \tag{18.108}$$

It is preferred to design the subfilters of $G_r(z)$ in such a way that the passband shaping is done entirely by $G_r^{(1)}(z^{K_r})$. The number of multipliers in the $G_r(z)$'s for $r = 1, 2, \ldots, R-1$ can be reduced by the experimentally observed fact that the overall filter still meets the given criteria when the stopband regions of these filters are decreased by using in Equation 18.106c the substitution

$$\left(2\omega_s^{(r)} + \omega_p^{(r)}\right)/3 \mapsto \omega_s^{(r)} \tag{18.109}$$

After some manipulations, $H(z)$ as given by Equation 18.104a through e and Equation 18.106 can be rewritten in the explicit form shown in Table 18.14. If $G_r(z)$ is a single-stage design, then $G_r^{(1)}(z^{K_r}) + 1$. In order to obtain the desired overall solution, the overall order of $G_r(z)$ for $r \geq 2$, denoted by N_r in Table 18.14, has to be even. Realizations for the overall transfer function are given in Figure 18.33, where

$$m_r = \hat{M}_r - \hat{M}_{r+1} = \frac{1}{2}\hat{L}_r N_r, \quad r = 2, 3, \ldots, R-1, \quad m_R = \hat{M}_R \tag{18.110}$$

The structure of Figure 18.33b is preferred since the delay terms z^{-m_r} can be shared with $H_R^{(1)}(z^{K_R\hat{L}_R})$ or, if this filter stage is not present, with $H_R^{(2)}(z^{K_R\hat{L}_R})$. This is because the overall order of this filter stage is usually larger than the sum of the m_r's.

If the edges ω_p and ω_s of the overall filter are larger than $\pi/2$, then we set $H(z) + F_1(z)$ in Equation 18.104a. In this case, $\delta_p^{(1)} + 0$, $L_1 + 1$, and $G_1(z)$,, $\omega_p^{(1)}$ and $\omega_s^{(1)}$ are absent. Furthermore, $\omega_p^{(2)} = \pi - \omega_s$, and $\omega_s^{(2)} = \pi - \omega_p$, and $H_1(z)$ is absent in Figure 18.33 and in Table 18.14.

TABLE 18.14 Explicit Form for the Transfer Function in the Jing–Fam Approach

$$H(z) = H_1(z^{\hat{L}_1})\Big\{I_2 z^{-\hat{M}_2} + H_2(z^{\hat{L}_2})\Big[I_3 z^{-\hat{M}_3} + H_3(z^{\hat{L}_3})(\cdots)$$
$$\Big\{I_{R-1}z^{-\hat{M}_{R-1}} + H_{R-1}(z^{\hat{L}_{R-1}})\Big[I_R z^{-\hat{M}_R} + H_R(z^{\hat{L}_R})\Big]\Big\}\ldots\Big)\Big]\},$$

where

$$H_r(z^{\hat{L}_r}) = H_r^{(1)}(z^{K_r\hat{L}_r})H_r^{(2)}(z^{\hat{L}_r})$$

$$H_r^{(1)}(z) = G_r^{(1)}\left(J_r^{(1)}z\right), \quad H_r^{(2)}(z) = S_r G_r^{(2)}\left(J_r^{(2)}z\right)$$

$$S_1 = 1, \quad S_r = -(-1)^{\hat{M}_r/\hat{L}_r}, \quad r = 2, 3, \ldots, R$$

$$J_1^{(2)} = 1, \quad J_2^{(2)} = -1, \quad J_r^{(2)} = -\left[J_{r-1}^{(2)}\right]^{\hat{L}_{r-1}}, \quad r = 3, 4, \ldots, R$$

$$J_r^{(1)} = \left[J_r^{(2)}\right]^{K_r}$$

$$\hat{L}_1 = 1, \quad \hat{L}_r = \prod_{k=1}^{r-1} L_k, \quad r = 2, 3, \ldots, R$$

$$\hat{M}_R = \tfrac{1}{2}\hat{L}_R N_R, \quad \hat{M}_{R-r} = \hat{M}_{R-r+1} + \tfrac{1}{2}\hat{L}_{R-r}N_{R-r}, \quad r = 1, 2, \ldots, R-2$$

$$I_2 = 1, \quad I_r = \left[J_{r-1}^{(2)}\right]^{\hat{M}_r/\hat{L}_{r-1}}, \quad r = 3, 4, \ldots, R$$

$$N_r = K_r N_r^{(1)} + N_r^{(2)}$$

$N_r^{(1)}$ and $N_r^{(2)}$ are the orders of $G_r^{(1)}(z)$ and $G_r^{(2)}(z)$, respectively

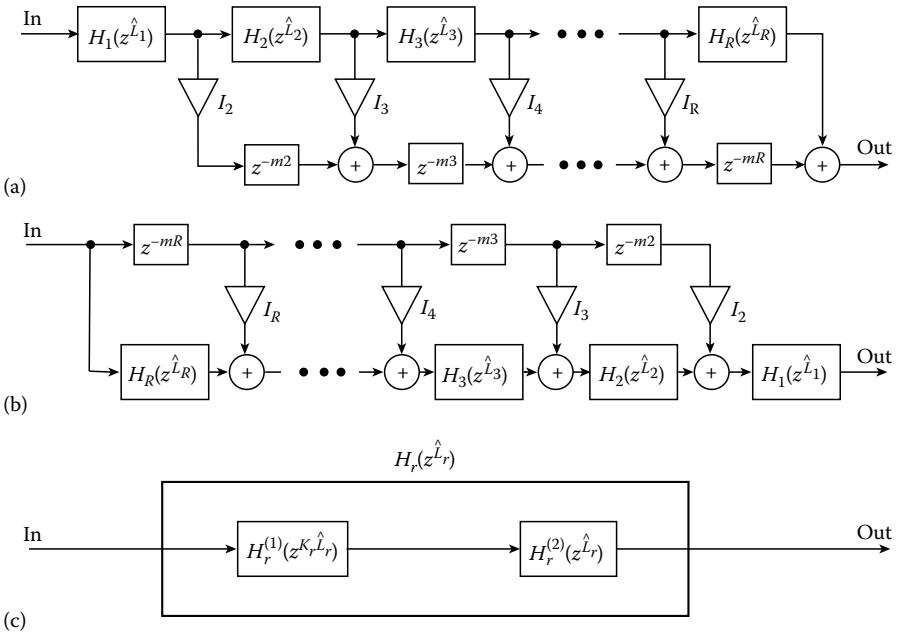


FIGURE 18.33 Implementations for a filter synthesized using the Jing–Fam approach. (a) Basic structure. (b) Transposed structure. (c) Structure for the subfilter $H_r(z^{O_{L_r}})$.

The remaining problem is to select R, the L_r's, the K_r's, and the ripple values such that the filter complexity is minimized. The following example illustrates this.

Example 18.5

Consider the specifications of Example 18.1, that is, $\boldsymbol{\omega}_p = 0.4\boldsymbol{\pi}$, $\boldsymbol{\omega}_s = 0.402\boldsymbol{\pi}$, $\boldsymbol{\delta}_p = 0.01$, and $\boldsymbol{\delta}_s = 0.001$. In this case, the only alternative is to select $L_1 = 2$. The resulting passband and stopband regions for $G_1(z)$ are (the substitution of Equation 18.109 is used)

$$\boldsymbol{\Omega}_p^{(1)} = [0,\ 0.4\boldsymbol{\pi}] \quad \text{and} \quad \boldsymbol{\Omega}_s^{(1)} = [0.5987\boldsymbol{\pi},\ \boldsymbol{\pi}]$$

For $\hat{H}_2(z)$, the edges become $\boldsymbol{\omega}_p^{(2)} = \boldsymbol{\pi} - L_{1\boldsymbol{\omega}s} = 0.196\boldsymbol{\pi}$ and $\boldsymbol{\omega}_s^{(2)} = \boldsymbol{\pi} - L_{1\boldsymbol{\omega}p} = 0.2\boldsymbol{\pi}$. For L_2, there are two alternatives to make the edges of $\hat{H}_3(z)$, $\boldsymbol{\omega}_p^{(3)} = \boldsymbol{\pi} - L_{2\boldsymbol{\omega}s}^{(2)}$ and $\boldsymbol{\omega}_s^{(3)} = \boldsymbol{\pi} - L_{2\boldsymbol{\omega}p}^{(2)}$, less than $\boldsymbol{\pi}/2$. These are $L_2 = 3$ and $L_2 = 4$. For $R = 5$ stages, there are the following four alternatives to make all the $\boldsymbol{\omega}_s^{(r)}$'s smaller than $\boldsymbol{\pi}/2$:

$$\begin{aligned}
&L_1 = 2, \quad L_2 = 4, \quad L_3 = 3, \quad L_4 = 2\\
&L_1 = 2, \quad L_2 = 4, \quad L_3 = 4, \quad L_4 = 4\\
&L_1 = 2, \quad L_2 = 3, \quad L_3 = 2, \quad L_4 = 4\\
&L_1 = 2, \quad L_2 = 3, \quad L_3 = 2, \quad L_4 = 3
\end{aligned}$$

Among these alternatives, the first one results in an overall filter with the minimum complexity. In this case, the edges of $\hat{H}_3(z)$, $\hat{H}_4(z)$, and $\hat{H}_5(z)+G_5(z)$ become as shown in Table 18.15. The corresponding passband and stopband regions for $G_2(z)$, $G_3(z)$, $G_4(z)$, and $G_5(z)$ are

TABLE 18.15 Data for a Filter Designed Using the Jing–Fam Approach

	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
$\omega p^{(r)}$	0.4π	0.196π	0.2π	0.352π	0.2π
$\omega s^{(r)}$	0.402π	0.2π	0.216π	0.4π	0.296π
$\delta p^{(r)}$	$\frac{1}{3}\times 10^{-2}$	$\frac{1}{3}\times 10^{-3}$	$\frac{1}{3}\times 10^{-2}$	$\frac{1}{3}\times 10^{-3}$	$\frac{1}{3}\times 10^{-2}$
$\delta s^{(r)}$	10^{-3}	$\frac{2}{3}\times 10^{-2}$	$\frac{2}{3}\times 10^{-3}$	$\frac{1}{3}\times 10^{-2}$	$\frac{1}{3}\times 10^{-3}$
L_r	2	4	3	2	—
K_r	—	3	2	—	3
$N_r^{(1)}$	—	20	11	—	22
$N_r^{(2)}$	31	10	8	26	14
N_r	31	70	30	26	80
$\hat{L}_r$	1	2	8	24	48
$J_r^{(1)}$	—	-1	1	—	-1
$J_r^{(2)}$	1	-1	-1	1	-1
$\hat{M}_r$	—	2422	2352	2232	1920
I_r	—	1	1	-1	1
S_r	1	1	-1	1	-1
m_r	—	70	120	312	1920

$$\Omega_p^{(2)} = [0,\ 0.196\pi], \quad \Omega_s^{(2)} = [0.3013\pi,\ 0.6987\pi] \cup [0.8013\pi,\ \pi]$$
$$\Omega_p^{(3)} = [0,\ 0.2\pi], \quad \Omega_s^{(3)} = [0.4560\pi,\ 0.8773\pi]$$
$$\Omega_p^{(4)} = [0,\ 0.352\pi], \quad \Omega_s^{(4)} = [0.616\pi,\ \pi]$$
$$\Omega_p^{(5)} = [0,\ 0.2\pi], \quad \Omega_s^{(5)} = [0.296\pi,\ \pi]$$

What remains is to determine the ripple requirements. From Equation 18.107b, it follows for $R=5$, $\delta_p^{(1)} + \delta_p^{(3)} + \delta_p^{(5)} = \delta_p$ and $\delta_p^{(2)} + \delta_p^{(4)} + \delta_p^{(6)} = \delta_s$. By simply selecting the ripple values in these summations to be equal, the required ripples for the $G_r(z)$'s become as shown in Table 18.15.

The first and fourth subfilter are single-stage filters since their stopband edges are larger than $\pi/2$, whereas the remaining three filters are two-stage designs. The parameters describing the overall filter are shown in Table 18.15, whereas Figure 18.34a depicts the response of this filter. The number of multipliers and the order of this design are 78 and 4875, whereas the corresponding numbers for the direct-form equivalent are 1271 and 2541. The number of multipliers required by the proposed design is thus only 6 percent of that of the direct-form filter. Since the complexity of $H_5(z^{\hat{L}_5})$ is similar to those of the earlier filter stages, $R=5$ is a good selection in this example.

The overall filter order as well as the number of multipliers can be decreased by selecting smaller ripple values for the first stages, thereby allowing larger ripples for the last stages. Proper selections for the ripple requirements and filter orders are shown in Table 18.16. The first four filters have been optimized such that their passband variations are minimized. The first criteria are met by a half-band filter of order 34, having the passband and stopband edges at 0.4013π and 0.5987π. Since every second impulse response coefficient of this filter is zero-valued except for the central coefficient with an easily

TABLE 18.16 Data for Another Filter Designed Using the Jing–Fam Approach

$\delta p^{(r)}$	7.3×10^{-4}	7.1×10^{-5}	3.5×10^{-4}	12.1×10^{-5}	89.2×10^{-4}
$\delta s^{(r)}$	10^{-3}	92.7×10^{-4}	92.9×10^{-5}	89.2×10^{-4}	80.8×10^{-5}
K_r	—	3	2	—	2
$N_r^{(1)}$	—	22	13	—	27
$N_r^{(2)}$	34	10	8	24	6

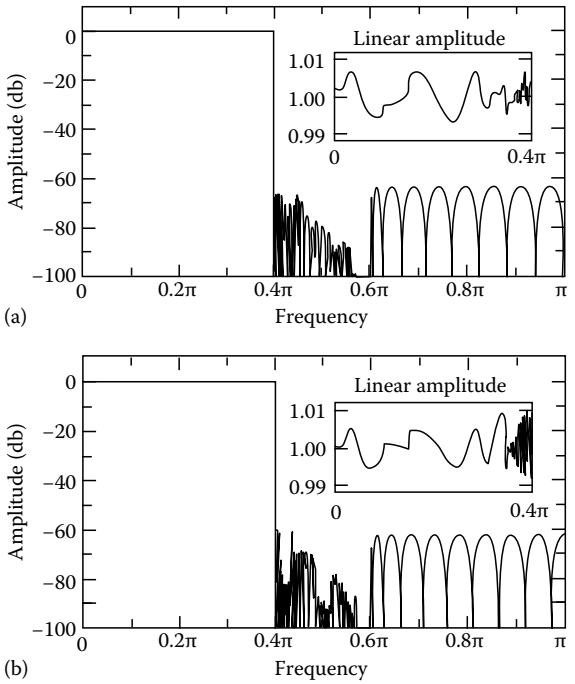


FIGURE 18.34 Amplitude responses for filters synthesized using the Jin–Fam approach.

implementable value of 1/2, this filter requires only nine multipliers. For the last stage, K_5 is reduced to 2 to decrease the overall filter order. The order of the resulting overall filter [see Figure 18.34b] is 3914, which is 54 percent higher than that of the direct-form equivalent. The number of multipliers is reduced to 70.

The Jing–Fam approach cannot be applied directly for synthesizing filters whose edges are very close to $\pi/2$. This problem can, however, be overcome by slightly changing the sampling rate or, if this is not possible, by shifting the edges by a factor 3/2 by using decimation by this factor at the filter input and interpolation by the same factor at the filter output [2]. One attractive feature of the Jing–Fam approach is that it can be combined with multirate filtering to reduce the filter complexity even further [7].

When comparing the above designs with the filters synthesized using the multistage frequency-response masking technique (Example 18.2), it is observed that the above designs require slightly fewer multipliers at the expense of an increased overall filter order. Both of these general approaches are applicable to those specifications that are not very narrowband or very wideband. For most very narrowband and wideband cases, filters synthesized in the simplified forms $H(z) = F(z^L)G(z)$ and $H(z) = z^{-M} - (-1)^M F[(-z)^L]G(-z)$, respectively, give the best results (see Examples 18.3 and 18.4).

References

1. O. Herrmann, L.R. Rabiner, and D.S.K. Chan, Practical design rules for optimum finite impulse response lowpass digital filters, *Bell System Tech. J.*, 52, 769–799, July–August 1973.
2. Z. Jing and A. T. Fam, A new structure for narrow transition band, lowpass digital filter design, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-32, 362–370, April 1984.
3. Y. C. Lim, Frequency-response masking approach for the synthesis of sharp linear phase digital filters, *IEEE Trans. Circuits Syst.*, CAS-33, 357–364, April 1986.

4. Y. C. Lim and Y. Lian The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique, *IEEE Trans. Circuits Syst.—II: Analog and Digital Signal Processing,* 40, 88–95, February 1993.
5. J. H. McClellan, T. W. Parks, and L. R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electroacoust.,* AU-21, 506–526, December 1973; also reprinted in *Selected Papers in Digital Signal Processing, II,* IEEE Digital Signal Processing Comm. and IEEE-ASSP, Eds., New York: IEEE Press, 1976, 97–117.
6. Y. Neuvo, C.-Y. Dong, and S. K. Mitra, Interpolated finite impulse response filters, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-32, 563–570, June 1984.
7. T. Ramstad and T. Saramäki, Multistage, multirate FIR filter structures for narrow transition-band filters, in *Proc.* 1990 *IEEE Int. Symp. Circuits Syst.,* New Orleans, LA, 1990, 2017–2021.
8. T. Saramäki, Y. Neuvo, and S. K. Mitra, Design of computationally efficient interpolated FIR filters, *IEEE Trans. Circuits Syst.,* CAS-35, 70–88, January 1988.
9. T. Saramäki and A. T. Fam, Subfilter approach for designing efficient FIR filters, in *Proc. 1988 IEEE Int. Symp. Circuits Syst.,* Espoo, Finland, 2903–2915, 1988.
10. T. Saramäki, Finite impulse response filter design, in *Handbook for Digital Signal Processing,* S. K. Mitra and J. F. Kaiser, Eds., New York: Wiley, 1993, Ch. 4, 155–277.

19

IIR Filters

19.1 Properties of IIR Filters **19**-1
System Function and Impulse Response • Causality (Physical Realizability) • Poles and Zeros • Stability • Frequency Response • Realizations • Minimum Phase • Allpass Filters
References **19**-10
19.2 Design of IIR Filters **19**-10
Introduction • Analog Filters • Design Using Analog Prototype Filters • Design Using Digital Frequency Transformations • Computer-Aided Design • Stability Issues
References **19**-41
19.3 Lattice Digital Filters **19**-41
Lattice Filters • Evaluation of the Reflection Coefficients k_n • Evaluation of the Tap Gains φ_n and λ_n • Evaluation of the Injector Gains θ_n and ϕ_n
References **19**-45

Sawasd Tantaratana
Thammasat University

Chalie Charoenlarpnopparut
Thammasat University

Phakphoom Boonyanant
National Electronics and Computer Technology Center

Yong Ching Lim
Nanyang Technological University

19.1 Properties of IIR Filters

Sawasd Tantaratana

19.1.1 System Function and Impulse Response

A digital filter with impulse response having infinite length (i.e., its values outside a finite interval cannot all be zero) is termed infinite impulse response (IIR) filter. The most important class of IIR filters can be described by the difference equation

$$\begin{aligned} y(n) = {} & b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) \\ & - a_1 y(n-1) - a_2 y(n-2) - \cdots - a_N y(n-N) \end{aligned} \tag{19.1}$$

where
$x(n)$ is the input
$y(n)$ is the output of the filter
$\{a_1, a_2, \ldots, a_N\}$ and $\{b_0, b_1, \ldots, b_M\}$ are constant coefficients

We assume that $a_N \neq 0$. The impulse response is the output of the system when it is driven by a unit impulse at $n = 0$, with the system being initially at rest, i.e., the output being zero prior to applying the input. We denote the impulse response by $h(n)$. With $x(0) = 1, x(n) = 0$ for $n \neq 0$, and $y(n) = 0$ for

$n < 0$, we can compute $h(n)$, $n \geq 0$, from Equation 19.1 in a recursive manner. Taking the z-transform of Equation 19.1, we obtain the system function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{19.2}$$

where N is the order of the filter. The system function and the impulse response are related through the z-transform and its inverse, i.e.,

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-1} \quad h(n) = \frac{1}{2\pi j} \oint_C H(z) z^{n-1} \mathrm{d}z \tag{19.3}$$

where C is a closed counterclockwise contour in the region of convergence. See Chapter 5 of *Fundamentals of Circuits and Filters* for a discussion of z-transform. We assume that $M \leq N$. Otherwise, the system function can be written as

$$H(z) = \left[c_0 + c_1 z^{-1} + \cdots + c_{M-N} z^{-(M-N)}\right] + \left[\frac{b_0' + b_1' z^{-1} + \cdots + b_M' z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}\right] \tag{19.4}$$

which is a finite impulse response (FIR) filter in parallel with an IIR filter, or as

$$H(z) = \left[c_0' + c_1' z^{-1} + \cdots + c_{M-N}' z^{-(M-N)}\right] \left[\frac{b_0'' + b_1'' z^{-1} + \cdots + b_M'' z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}\right] \tag{19.5}$$

which is an FIR filter in cascade with an IIR filter. FIR filters are covered in Chapter 18. This chapter covers IIR filters.

For ease of implementation, it is desirable that the coefficients $\{a_1, a_2, \ldots, a_N\}$ and $\{b_0, b_1, \ldots, b_M\}$ be real numbers (as opposed to complex numbers), which is another assumption that we make, unless specified otherwise.

19.1.2 Causality (Physical Realizability)

A causal (physically realizable) filter is one whose output value does not depend on the future input values. A noncausal filter cannot be realized in real time since some future inputs are needed in computing the current output value. The difference equation in Equation 19.1 can represent a causal system or a noncausal system. If the output $y(n)$ is calculated, for an increasing value of n, from $x(n), x(n-1), \ldots, x(n-M)$, $y(n-1), \ldots, y(n-N)$, according to the right-hand side of Equation 19.1 then the difference equation represents a causal system. On the other hand, we can rewrite Equation 19.1 as

$$\begin{aligned} y(n-N) = &\frac{1}{a_N}[b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) \\ &- y(n) - a_1 y(n-1) - \cdots - a_{N-1} y(n-N+1)] \end{aligned} \tag{19.6}$$

If the system calculates $y(n-N)$, for a decreasing due of n, using the right-hand side of Equation 19.6, then the system is noncausal since $y(n-N)$ depends on $x(n), \ldots, x(n-M)$, which are future input values. We shall assume that the IIR filter represented by Equation 19.1 is causal. It follows from this assumption that $h(n) = 0$ for $n < 0$ and that the convergence region of $H(z)$ is of the form: $|z| > r_0$, where r_0, is a nonnegative constant.

Noncausal filters are useful in practical application where the output need not be calculated in real time or where the variable n does not represent time, such as in image processing where n is a spatial variable. Generally, a noncausal filter can be modified to be causal by adding sufficient delay at the output.

19.1.3 Poles and Zeros

Rewriting Equation 19.2 we have

$$H(z) = z^{N-M} \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_{M-1} z + b_M}{z^N + a_1 z^{N-1} + \cdots + a_{N-1} z^{-N}} \tag{19.7}$$

Assuming $b_0, b_M \neq 0$, then there are N poles given by the roots of the denominator polynomial and M zeros given by the roots of the numerator polynomial. In addition, there are $N - M$ zeros at the origin on the complex plane. The locations of the poles and zeros can be plotted on the complex z plane. Denoting the poles by $p_1, p_2, \ldots, p_N$, and the nonzero zeros by $q_1, q_2, \ldots, q_M$, we can write

$$H(z) = b_0 z^{N-M} \frac{(z - q_1)(z - q_2) \cdots (z - q_M)}{(z - p_1)(z - p_2) \cdots (z - p_N)} \tag{19.8}$$

Since we assume that the coefficients $\{a_1, a_2, \ldots, a_N\}$ and $\{b_0, b_1, \ldots, b_M\}$ are each complex-valued pole (i.e., pole off the real axis on the z plane), there must be another pole that is the complex conjugate of the first. Similarly, complex-valued zeros must exist in complex–conjugate pairs. The combination of a complex–conjugate pole pair (or zero pair) yields a second-order polynomial with real coefficients. Real-valued pole (or zero) can appear single in Equation 19.8.

It is clear from Equation 19.8 that knowing all the pole and zero locations, we can write the system function to within a constant factor. Since the constant factor is only a gain, which can be adjusted as desired, specifying the locations of the poles and zeros essentially specifies the system function of the IIR filter.

19.1.4 Stability

A causal IIR filter is stable (in the sense that a bounded input gives rise to a bounded output) if all the poles lie inside the unit circle. If there are one or more simple poles on the unit circle (and all the others lie inside the unit circle), then the filter is marginally stable, giving a sustained oscillation. If there are multiple poles (more than one pole at the same location) on the unit circle or if there is at least one pole outside the unit circle, a slight input will give rise to an output with increasing magnitude. For most practical filters, all the poles are designed to lie inside the unit circle. In some special systems (such as oscillators), poles are placed on the unit circle to obtain the desired result.

Given the system function in the form of Equation 19.2 or Equation 19.7, the stability can be verified by finding all the poles of the filters and checking to see if all of them are inside the unit circle. Equivalently, stability can be verified directly from the coefficients $\{a_i\}$, using the Schür-Cohn algorithm [1]. For a second-order system, if the coefficient a_1, and a_2, lie inside the triangle in Figure 19.1, then the system is stable.

19.1.5 Frequency Response

The frequency response of the IIR filter is the due of the system function evaluated on the unit circle on the complex plane, i.e., with $z = e^{j2\pi f}$, where f varies from 0 to 1, or from $-1/2$ to $1/2$. The variable f represents the digital frequency. For simplicity, we write $H(f)$ for $H(z)|_{z=\exp(j2\pi f)}$. Therefore,

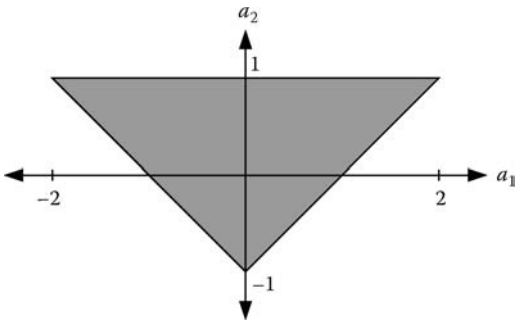


FIGURE 19.1 Region for the coefficients a_1 and a_2 that yield a stable second-order IIR filter.

$$H(f) = b_0 e^{j2\pi(M-N)f} \frac{(e^{j2\pi f} - q_1)(e^{j2\pi f} - q_2)\cdots(e^{j2\pi f} - q_M)}{(e^{j2\pi f} - p_1)(e^{j2\pi f} - p_2)\cdots(e^{j2\pi f} - p_N)} \tag{19.9}$$

$H(f)$ is generally a complex function of f, consisting of the real part $H_R(f)$ and the imaginary part $H_I(f)$. It can also be expressed in terms of the magnitude $|H(f)|$ and the phase $\theta(f)$

$$H(f) = H_R(f) + jH_I(f) = |H(f)|e^{j\theta(f)} \tag{19.10}$$

From Equation 19.9 we see that the magnitude response $|H(f)|$ equals the product of the magnitudes of the individual factors in the numerator, divided by the product of the magnitudes of the individual factors in the denominator. The magnitude square can be written as

$$|H(f)|^2 = H(f)H^*(f) = [H_R(f)]^2 + [H_I(f)]^2 \tag{19.11}$$

Since $H^*(f) = H^*(1/z^*)|_{z=\exp(j2\pi f)}$ and $H^*(1/z^*) = H(z^{-1})$ when all the coefficients of $H(z)$ are real, we have

$$|H(f)|^2 = H(z) \cdot H(z^{-1})|_{z=\exp(j2\pi f)} \tag{19.12}$$

Using Equation 19.12, the magnitude square can be put in the form

$$|H(f)|^2 = \frac{\sum_{k=0}^{M} \tilde{b}_k \cos(2\pi k f)}{\sum_{k=0}^{N} \tilde{a}_k \cos(2\pi k f)} \tag{19.13}$$

where the coefficients are given by

$$\begin{aligned} \tilde{b}_0 &= \sum_{j=0}^{M} b_j^2 \quad \tilde{b}_k = 2\sum_{j=k}^{M} b_j b_{j-k} \quad k = 1, \ldots, M \\ \tilde{a}_0 &= \sum_{j=0}^{N} a_j^2 \quad \tilde{a}_k = 2\sum_{j=k}^{N} a_j a_{j-k} \quad k = 1, \ldots, N \end{aligned} \tag{19.14}$$

with the understanding that $a_0 = 1$. Given $\{\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_M\}$ we can find $\{b_0, b_1, \ldots, b_M\}$ and vice versa. Similarly, $\{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_N\}$ and $\{a_1, a_2, \ldots, a_N\}$ can be computed from each other. The form in Equation 19.13 is useful in computer-aided design of IIR filters using linear programming [2].

We see from Equation 19.9 that the phase response $\theta(f)$ equals the sum of the phases of the individual factors in the numerator minus the sum of the phases of the individual factors in the denominator. The phase can be written in terms of the real and imaginary parts of $H(f)$ as

$$\theta(f) = \arctan\left[\frac{H_{\mathrm{I}}(f)}{H_{\mathrm{R}}(f)}\right] \tag{19.15}$$

A filter having linear phase in a frequency band (e.g., in the passband) means that there is no phase distortion in that band.

The group delay is defined as

$$\tau(f) = -\frac{1}{2\pi}\frac{\mathrm{d}}{\mathrm{d}f}\theta(f) \tag{19.16}$$

The group delay corresponds to the delay, from the input to the output, of the envelope of a narrowband signal [3]. A linear phase gives rise to a constant group delay. Nonlinearity in the phase appears as deviation of the group delay from a constant value.

The magnitude response of IIR filter does not change, except for a constant factor, if a zero is replaced by the reciprocal of its complex conjugate, i.e., if $(z - q)$ is replaced with $(z - 1/q^*)$. This can be seen as follows. Letting $\tilde{H}(z)$ be the system function without the factor $(z - q)$, we have

$$\begin{aligned}
|H(f)|^2 &= H(z)H^*(1/z^*)|_{z=\exp(j2\pi f)} \\
&= \hat{H}(z)\hat{H}^*(1/z^*)\frac{(z-q)(z^{-1}-q^*)}{(z-1/q^*)(z^{-1}-1/q)}\bigg|_{z=\exp(2j\pi f)} \\
&= |q|^2
\end{aligned}$$

Similarly, replacing the pole at p with a pole at $1/p^*$ will not alter the magnitude of the response except for a constant factor. This property is useful in changing an unstable IIR filter to a stable one without altering the magnitude response.

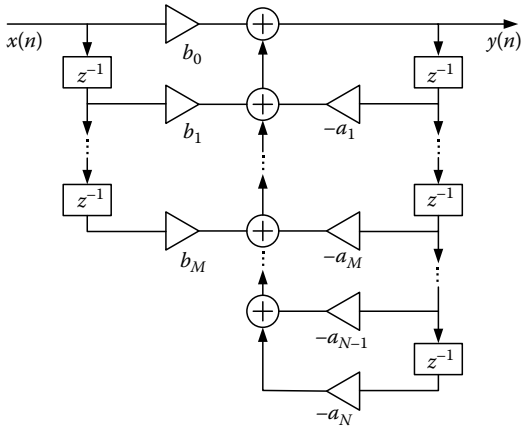
Compared to an FIR filter, an IIR filter requires a much lower order to achieve the same requirement of the magnitude response. However, the phase of a stable casual IIR filter cannot be made linear. This is the major reason not to use an IIR filter in applications where linear phase is essential. Nevertheless, using phase compensation such as allpass filters (see Section 19.1.8), the phase of an IIR filter can be adjusted close to linear. This process increases the order of the overall system, however. Note that if causality is not required, then a linear-phase IIR filter can be obtained using a time-reversal filter [2].

19.1.6 Realizations

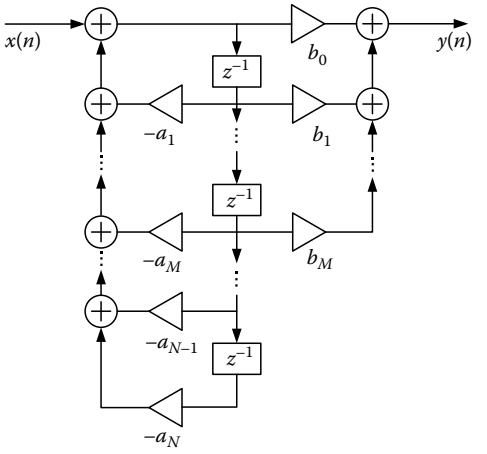
A realization of an IIR filter according to Equation 19.1 is shown in Figure 19.2a, which is called Direct Form I. By rearranging the structure, we can obtain Direct Form II, as shown in Figure 19.2b. Through transposition, we can obtain Transposed Direct Form I and Transposed Direct Form II as shown in Figure 19.2c and d.

The system function can be put in the form

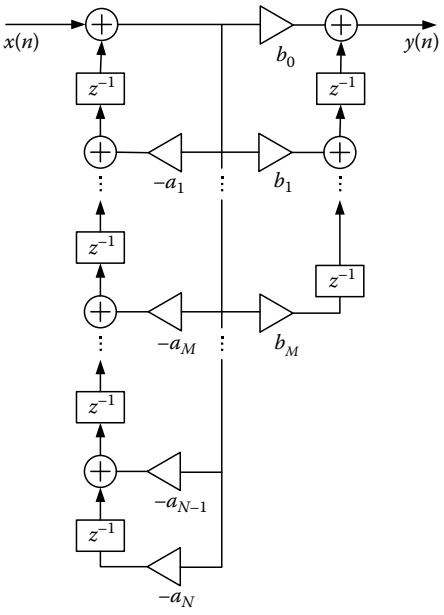
$$H(z) = \prod_{i=1}^{K}\frac{b_{i0} + b_{i1}z^{-1} + b_{i2}z^{-2}}{1 + a_{i1}z^{-1} + a_{i2}z^{-2}} \tag{19.17}$$



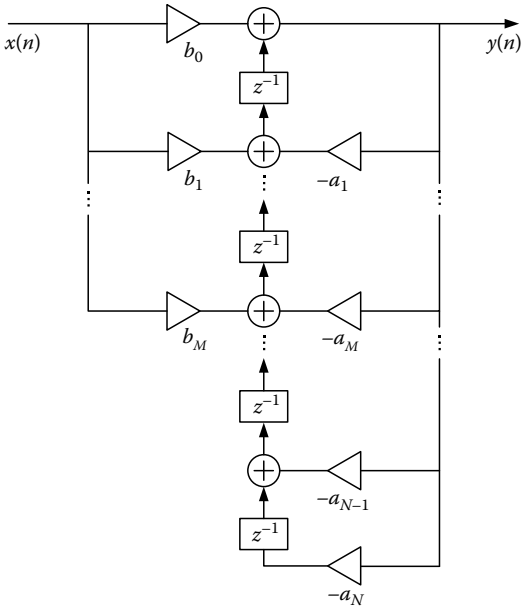
(a) Direct form I



(b) Direct form II



(c) Transposed direct form I



(d) Transposed direct form II

FIGURE 19.2 Direct form realizations of IIR filters.

by factoring the numerators and denominators into second-order factors, or in the form

$$H(z) = \frac{b_N}{a_N} + \sum_{i=1}^{K} \frac{b_{i0} + b_{i1} z^{-1}}{1 + a_{i1} z^{-1} + a_{i2} z^{-2}} \tag{19.18}$$

by partial fraction expansion. The value of K is $N/2$ when N is even and is $(N+1)/2$ when N is odd. When N is odd, one of a_{i2} must be zero, as well as one of b_{i2}, in Equation 19.17 and one of b_{i1} in Equation 19.18. All the coefficients in Equations 19.17 and 19.18 are real numbers. According to Equation 19.17, the IIR filter can be realized by K second-order IIR filters in cascade, as shown in Figure 19.3a. According

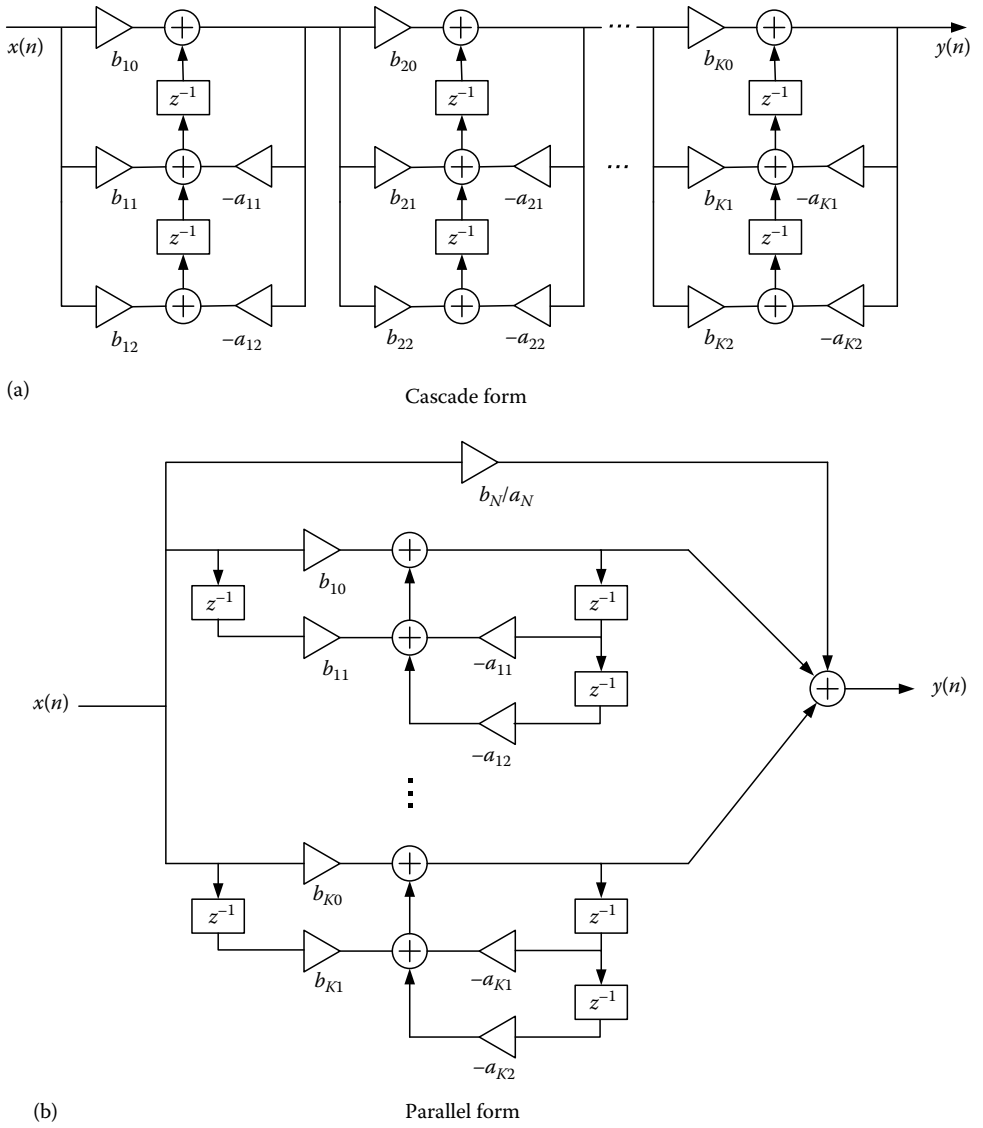


FIGURE 19.3 Cascade (a) and parallel (b) realizations of IIR filters.

to Equation 19.18, the IIR filter is realized by K second-order IIR filters and one scaler (i.e., b_N/a_N) in parallel, as depicted in Figure 19.3b. Each second-order subsystem can use any of the structures given in Figure 19.2.

There are many other realizations for IIR filters, such as state-space structures [4], wave structures, and lattice structures (Section 19.3).

Actual implementation of IIR filters requires that the signals and the coefficients be represented in a finite number of bits (or digits). Quantization of the coefficients to a finite number of bits essentially changes the filter coefficients, hence the frequency response changes. Coefficient quantization of a stable IIR filter may yield an unstable filter. For example, consider a second-order IIR filter with $a_1 = 1.26$ and $a_2 = 0.3$, which correspond to pole locations of -0.9413 and -0.3187, respectively. Suppose that we quantize these coefficients to two bits after the decimal point, yielding a quantized a_1 of 1.01 in binary or

1.25 and a quantized a_2 of 0.01 in binary or 0.25. This pair corresponds to pole locations at -1.0 and -0.25, respectively. Since one pole is on the unit circle, the IIR filter with quantized coefficients produces an oscillation. In this example, the quantization is equivalent to moving a point inside the triangle in Figure 19.1 to a point on the edge of the triangle. Different realizations are affected differently by coefficient quantization. Chapter 20 investigates coefficient quantization and roundoff noise in detail.

19.1.7 Minimum Phase

An IIR filter is a minimum-phase filter if all the zeros and poles are inside the unit circle. A minimum-phase filter introduces the smallest group delay among all filters that have the same magnitude response. A minimum-phase IIR filter can be constructed from a nonminimum-phase filter by replacing each zero (or pole) outside the unit circle with a zero (or pole) that is the reciprocal of its complex conjugate, as illustrated in Figure 19.4. This process moves all zeros and poles outside the unit circle to the inside. The magnitude response does not change, except for a constant factor, which is easily adjusted. Given an IIR filter $H(z)$ with input $x(n)$ and output $y(n)$, the inverse filter $1/H(z)$ can reconstruct $x(n)$ from $y(n)$ by feeding $y(n)$ to the input of $1/H(z)$. Assuming that both the filter and the inverse filter are causal, both of them can be stable only if $H(z)$ is a minimum-phase filter.

19.1.8 Allpass Filters

An allpass filter has a magnitude response of unity (or constant). An Nth-order IIR allpass filter with real coefficients has a system function given by

$$H(z) = z^{-N}\frac{D(z)}{D(z^{-1})} = z^{-N}\frac{a_N z^N + \cdots + a_2 z^2 + a_1 z + 1}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{19.19}$$

$$= z^{-N}\frac{(1 - p_1 z)(1 - p_2 z)\cdots(1 - p_N z)}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})\cdots(1 - p_N z^{-1})} \tag{19.20}$$

Since $H(z)(z^{-1}) = 1$, it follows that $|H(f)|^2 = 1$. The factor z^{-N} is included so that the filter is causal. Equation 19.20 implies that zeros and poles come in reciprocal pairs: if there is a pole at $z = p$, then there is a zero at $z = 1/p$, as illustrated in Figure 19.5.

Since the coefficients are real, poles and zeros off the real axis must exist in quadruplets: poles at p and p^* and zeros at $1/p$ and $1/p^*$, where $|p| < 1$ for stability. For poles and zeros on the real axis, they exist in reciprocal pairs: pole at p and zero at $1/p$, where p is real and $|p| < 1$ for stability. Since the numerator and the denominator in (19.19) share the same set of coefficients, we need only N multiplications in

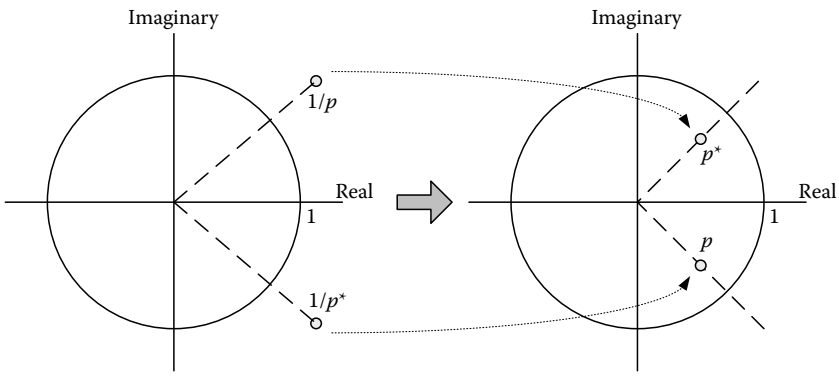


FIGURE 19.4 Changing a zero location to obtain a minimum-phase filter.

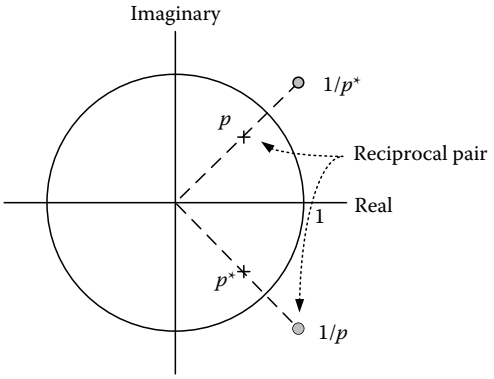


FIGURE 19.5 Pole-zero reciprocal pair in an allpass IIR filter.

realizing an *N*th-order allpass filter. The system function in Equation 19.19 can be written as the product (or sum) of first- and second-order allpass filters. The system function and the phase response of a first-order allpass filter is given by

$$H(z) = \frac{a_1 z + 1}{z + a_1} \tag{19.21}$$

$$\theta(f) = \arctan\left[\frac{(a_1^2 - 1)\sin(\omega)}{2a_1 + (a_1^2 + 1)\cos(\omega)}\right] \tag{19.22}$$

where $\omega = 2\pi f$. For a second-order allpass filter, these are

$$H(z) = \frac{a_2 z^2 + a_1 z + 1}{z^2 + a_1 z + a_2} \tag{19.23}$$

$$\theta(f) = \arctan\left[\frac{2a_1(a_2 - 1)\sin(\omega) + (a_2^2 - 1)\sin(2\omega)}{2a_2 + a_1^2 + 2a_1(a_2 + 1)\cos(\omega) + (a_2^2 + 1)\cos(2\omega)}\right] \tag{19.24}$$

The group delay $\tau(f)$ of an allpass filter is always ≥ 0. The output signal energy of an allpass filter is the same as the input signal energy, i.e., $\sum_{n=-\infty}^{\infty} |y(n)|^2 = \sum_{n=-\infty}^{\infty} |x(n)|^2$, which means that the allpass filter is a lossless system. Note that if we attempt to find a minimum-phase filter from a stable allpass filter, by moving all the zeros inside the unit circle, all poles and zeros would cancel out, yielding the trivial filter with a system function of unity.

A more general form of Equation 19.19, allowing the coefficients to be complex, is *N*th-order allpass filer with system function

$$H(z) = z^{-N}\frac{D^*(z^*)}{D(z^{-1})} = z^{-N}\frac{a_N^* + \cdots + a_2^* z^2 + a_1^* z + 1}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{19.25}$$

$$= z^{-N}\frac{(1 - p_1^* z)(1 - p_2^* z)\cdots(1 - p_N^* z)}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})\cdots(1 - p_N z^{-1})} \tag{19.26}$$

Therefore, for a pole at $z = p$ there is a zero at $z = 1/p^*$, i.e., poles and zeros exist in reciprocal–conjugate pairs.

Allpass filters have been used as building blocks for various applications [5]. Particularly, an allpass filter can be designed to approximate a desired phase response. Therefore, an allpass filter in cascade with an IIR filter can be used to compensate the nonlinear phase of the IIR filter. Such a cascade filter has

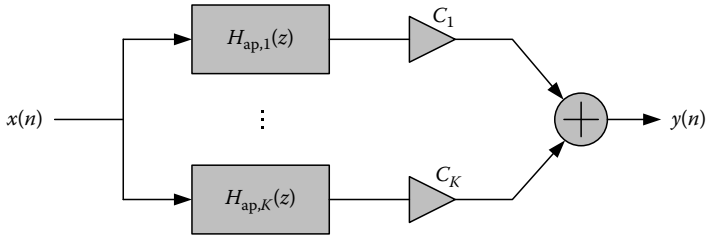


FIGURE 19.6 Block diagram of an IIR filter, using allpass filters.

system function of the form $H(z) = H_{IIR}(z)H_{ap}(z)$, where $H_{IIR}(z)$ is an IIR filter satisfying some magnitude response and $H_{ap}(z)$ is an allpass filter that compensates for the nonlinearity of the phase response of $H_{IIR}(z)$. Allpass filters in parallel connection can be used to approximate a desired magnitude response. For this, the system function is in the form $H(z) = \sum_{i=1}^{K} c_i H_{ap,i}(z)$, where $H_{ap,i}(z)$ is an allpass filter and c_i is a coefficient. A block diagram is shown in Figure 19.6.

References

1. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, 2nd ed. New York: Macmillan, 1992.
2. L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1975.
3. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NT: Prentice-Hall, 1989.
4. R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Reading, MA: Addison-Wesley, 1987.
5. P. A. Regalia, S. K. Mitra, and P. P. Vaidyanathan, The digital all-pass filter: A versatile signal processing building block, *Proc. IEEE*, 76, 19–37, Jan. 1988.

19.2 Design of IIR Filters

Sawasd Tantaratana, Chalie Charoenlarpnopparut, and Phakphoom Boonyanant

19.2.1 Introduction

A filter is generally designed to satisfy a frequency response specification. IIR filter design normally focuses on satisfying a magnitude response specification. If the phase response is essential, it is usually satisfied by a phase compensation filter, such as an allpass filter (see Section 19.1.8). We will adopt a magnitude specification that is normalized so that the maximum magnitude is 1. The magnitude square in the passband must be at least $1/(1+\varepsilon^2)$ and at most 1; while it must be no larger than δ^2 in the stopband, where ε and δ are normally small. The passband edge is denoted by f_p and the stopband edge by f_s. Figure 19.7a shows such a specification for a low-pass filter (LPF). The region between the passband and the stopband is the transition band. There is no constraint on the response in the transition band. Another specification that is often used is shown in Figure 19.7b using δ_1, and δ_2, to specify the acceptable magnitude. Given δ_1, and δ_2, they can be converted to ε and δ using $\varepsilon = 2\delta_1^{0.5}/(1-\delta_1)$ and $\delta = \delta_2/(1+\delta_1)$. The magnitude is often specified in decibels, which is $20\log_{10}|H(f)|$. Specifications for other types of filters (high-pass, bandpass, and bandstop) are similar.

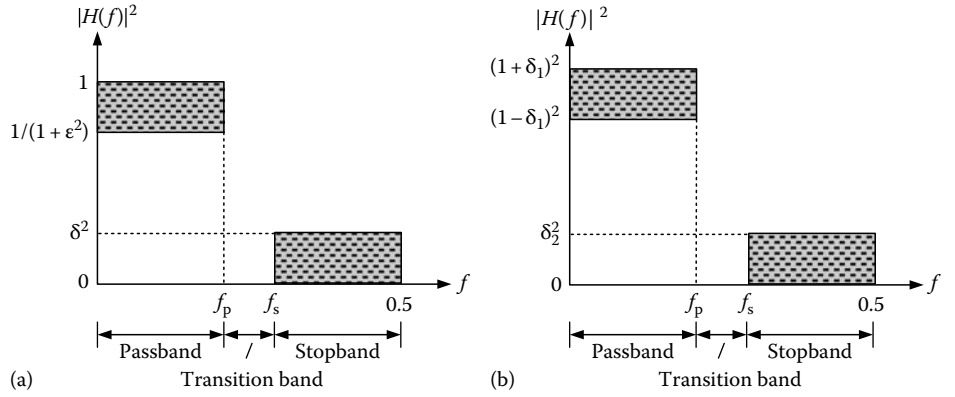


FIGURE 19.7 Specifications for a digital LPF. (a) Specification using ε and δ; (b) specification using δ_1 and δ_2.

We can classify various IIR filter design methods into three categories: the design using analog prototype filter, the design using digital frequency transformation, and computer-aided design. In the first category, an analog filter is designed to the (analog) specification and the analog filter transfer function is transformed to digital system function using some kind of transformation. The second category assumes that a digital LPF can be designed. The desired digital filter is obtained from the digital LPF by a digital frequency transformation. The last category uses some algorithm to choose the coefficients so that the response is as close (in some sense) as possible to the desired filter. Design methods in the first two categories are simple to do, requiring only a handheld calculator. Computer-aided design requires some computer programming, but it can be used to design nonstandard filters.

19.2.2 Analog Filters

Here, we describe four basic types of analog LPFs that can be used as prototype for designing IIR filters. For each type, we give the transfer function, its magnitude response, and the order N needed to satisfy the (analog) specification. We will use $H_a(s)$ for the transfer function of an analog filter, where s is the variable in the Laplace transform. Each of these filters have all its poles on the left-half s plane, so that it is stable. We will use the variable λ to represent the analog frequency in radians/second. The frequency response $H_a(\lambda)$ is the transfer function evaluated at $s = j\lambda$. The analog LPF specification is given by

$$\begin{aligned}(1+\varepsilon^2)^{-1} \le |H_a(\lambda)|^2 \le 1 \quad &\text{for } 0 \le (\lambda/2\pi) \le (\lambda_p/2\pi) \\ 0 \le |H_a(\lambda)|^2 \le \delta^2 \quad &\text{for } (\lambda_s/2\pi) \le (\lambda/2\pi) \le \infty\end{aligned} \tag{19.27}$$

where λ_p and λ_s are the passband edge and stopband edge, respectively. The specification is sketched in Figure 19.8.

19.2.2.1 Butterworth Filters

The transfer function of an Nth-order Butterworth filter is given by

$$H_a(s) = \begin{cases} \displaystyle\prod_{i=1}^{N/2} \frac{1}{(s/\lambda_c)^2 - 2\mathrm{Re}(s_i)(s/\lambda_c)+1} & N = \text{even} \\ \dfrac{1}{(s/\lambda_c)+1} \displaystyle\prod_{i=1}^{(N-1)/2} \frac{1}{(s/\lambda_c)^2 - \mathrm{Re}(s_i)(s/\lambda_c)+1} & N = \text{odd} \end{cases} \tag{19.28}$$

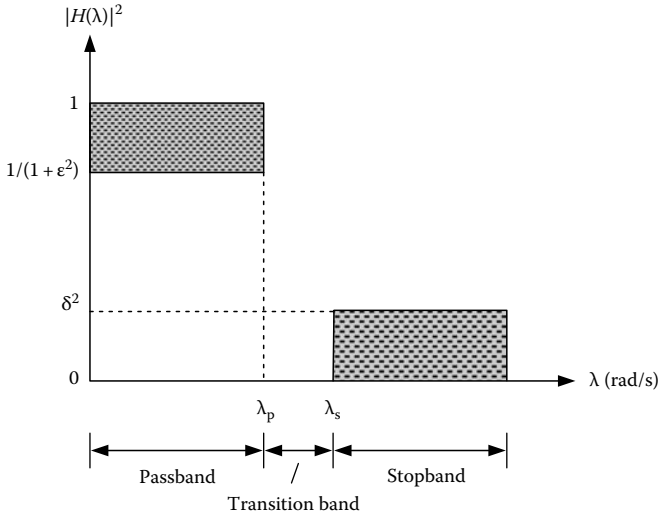


FIGURE 19.8 Specification for an analog LPF.

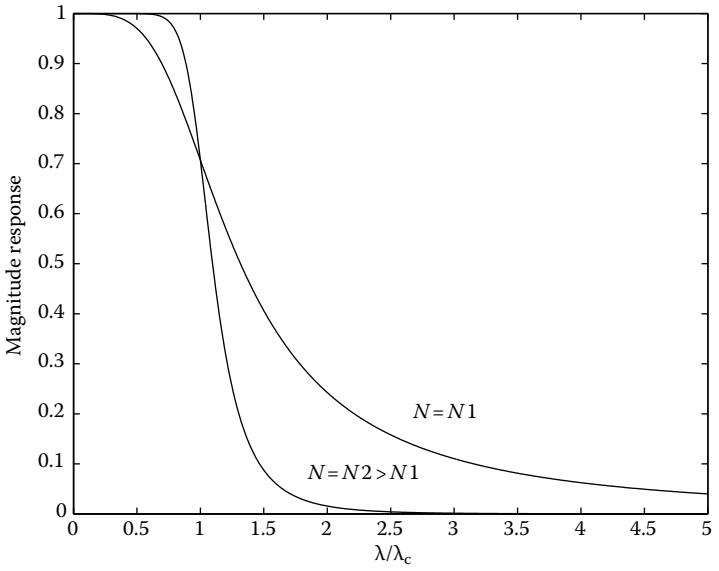


FIGURE 19.9 Magnitude responses of Butterworth filters.

where $\lambda_p \le \lambda_c \le \lambda_s$ and $s_i = \exp\{j[1 + (2i-1)/N]\pi/2\}$. The magnitude response square is

$$|H_a(\lambda)|^2 = \frac{1}{1 + (\lambda/\lambda_c)^{2N}} \tag{19.29}$$

Figure 19.9 shows the magnitude response $|H_a(\lambda)|$, with $\lambda_c = 1$. Note that a Butterworth filter is an all-pole (no zero) filter, with the poles being at $s = \lambda_c s_i$ and $s = \lambda_c s_i^*, i = 1, \ldots, N/2$ if N is even or $i = 1, \ldots, (N-1)/2$ if N is odd, where x^* denotes the complex conjugate of x. When N is odd, there

TABLE 19.1 Factorized, Normalized, Butterworth Polynomial $B_N(s)$

N	$B_N(s)$
1	$(s+1)$
2	$(s^2+\sqrt{2}s+1)$
3	$(s+1)(s^2+s+1)$
4	$(s^2+\sqrt{2-\sqrt{2}}s+1)(s^2+\sqrt{2+\sqrt{2}}s+1)$
5	$(s+1)\left(s^2+\left(\frac{\sqrt{5}-1}{2}\right)s+1\right)\left(s^2+\left(\frac{\sqrt{5}+1}{2}\right)s+1\right)$
6	$\left(s^2+\left(\frac{\sqrt{3}-1}{\sqrt{2}}\right)s+1\right)(s^2+\sqrt{2}s+1)\left(s^2+\left(\frac{\sqrt{3}+1}{\sqrt{2}}\right)s+1\right)$
7	$(s+1)(s^2+0.4450s+1)(s^2+1.2470s+1)(s^2+1.8019s+1)$
8	$(s^2+0.3902s+1)(s^2+1.1111s+1)(s^2+1.6629s+1)(s^2+1.9616s+1)$
9	$(s+1)^2(s^2+0.3473s+1)(s^2+1.5321s+1)(s^2+1.8794s+1)$
10	$(s^2+0.3219s+1)(s^2+0.9080s+1)(s^2+1.4142s+1)(s^2+1.7820s+1)(s^2+1.9754s+1)$

is another pole at $s=-\lambda_c$. All N poles are on the left-half s plane, located on the circle with radius λ_c. Therefore, the filter in Equation 19.28 is stable.

To satisfy the specification in Equation 19.27, the filter order can be calculated from

$$N = \text{integer} \geq \frac{\log\left[\varepsilon/(\delta^{-2}-1)^{1/2}\right]}{\log\left[\lambda_p/\lambda_s\right]} \tag{19.30}$$

The value of λ_c can be chosen as any value in the following range:

$$\lambda_p \varepsilon^{-1/N} \leq \lambda_c \leq \lambda_s(\delta^{-2}-1)^{-1/(2N)} \tag{19.31}$$

If we choose $\lambda_c = \lambda_p\varepsilon^{-1/N}$, then the magnitude response square passes through $1/(1+\varepsilon^2)$ at $\lambda=\lambda_p$. If we choose $\lambda_c = \lambda_s(\delta^{-2}-1)^{-1/(2N)}$, then the magnitude response square passes through δ^2 at $\lambda=\lambda_s$. If λ_c is between these two values, then the magnitude square will be $\geq 1/(1+\varepsilon^2)$ at $\lambda=\lambda_p$ and $\leq \delta^2$ at $\lambda=\lambda_s$.

For the sake of convenience, the transfer function of the Nth-order Butterworth filter with $\lambda_c=1$ can be found by

$$H_a(s) = \frac{1}{B_N(s)}$$

where $B_N(s)$ is the normalized Butterworth polynomial of order N shown in Table 19.1. For other values of λ_c', the transfer function can be computed similarly by analog–analog frequency transformation, i.e., replacing s in $H_a(s)$ with s/λ_c'.

19.2.2.2 Chebyshev Filters (Type-I Chebyshev Filters)

A Chebyshev filter is also an all-pole filter. The Nth-order Chebyshev filter has a transfer function given by

$$H_a(s) = C\prod_{i=1}^{N}\frac{1}{(s-p_i)} \tag{19.32}$$

where

$$C = \begin{cases} -\prod_{i=1}^{N} p_i & N \text{ is odd} \\ (1+\varepsilon^2)^{-1/2} \prod_{i=1}^{N} p_i & N \text{ is even} \end{cases} \tag{19.33}$$

$$p_i = -\lambda_p \sinh(\phi) \sin\left(\frac{2i-1}{2N}\pi\right) + j\lambda_p \cosh(\phi) \cos\left(\frac{2i-1}{2N}\pi\right) \tag{19.34}$$

$$\phi = \frac{1}{N} \ln\left[\frac{1+(1+\varepsilon^2)^{1/2}}{\varepsilon}\right] \tag{19.35}$$

The value of C normalizes the magnitude so that the maximum magnitude is 1. Note that C is always a positive constant. The poles are on the left-half s plane, lying on an ellipse centered at the origin with a minor radius of $\lambda_p \sinh(\phi)$ and major radius of $\lambda_p \cosh(\phi)$. Except for one pole when N is odd, all the poles have a complex–conjugate pair. Specifically, $p_i = p^*_{N-i+1}, i = 1, 2, \cdots, N/2$ or $(N-1)/2$. Combining each complex–conjugate pair in Equation 19.32 yields a second-order factor with real coefficients. The magnitude response can be computed from Equations 19.33 through 19.35 with $s = j\lambda$. Its square can also be written as

$$|H_a(\lambda)|^2 = \frac{1}{1+\varepsilon^2 T_N^2(\lambda/\lambda_p)} \tag{19.36}$$

where $T_N(x)$ is the Nth degree Chebyshev polynomial of the first kind, which is shown in Table 19.2 and also given recursively by

$$\begin{aligned} T_0(x) &= 1 \quad T_1(x) = x \\ T_{n+1}(x) &= 2xT^n(x) - T_{n-1}(x) \quad n \geq 1 \end{aligned} \tag{19.37}$$

Notice that $T_N^2(\pm 1) = 1$. Therefore, we have from Equation 19.36 that the magnitude square passes through $1/(1+\varepsilon^2)$ at $\lambda = \lambda_p$, i.e., $|H_a(\lambda_p)|^2 = 1/(1+\varepsilon^2)$. Note also that $T_N(0) = (-1)^{N/2}$ for even N and it is 0 for odd N. Therefore, $|H_a(0)|^2$ equals $1/(1+\varepsilon^2)$ for even N and it equals 1 for odd N. Figure 19.10 shows some examples of magnitude response square.

TABLE 19.2 Coefficients of Chebyshev Polynomials $T_n(x)$ of the First Kind, of Order n, in Ascending Powers of Variable x

n	Coefficients of $T_n(x)$
1	0, 1
2	−1, 0, 2
3	0, −3, 0, 4
4	1, 0, −8, 0, 8
5	0, 5, 0, −20, 0, 16
6	−1, 0, 18, 0, −48, 0, 32
7	0, −7, 0, 56, 0, −112, 0, 64
8	1, 0, −32, 0, 160, 0, −256, 0, 128
9	0, 9, 0, −120, 0, 432, 0, −576, 0, 256
10	−1, 0, 50, 0, −400, 0, 1120, 0, −1280, 0, 512

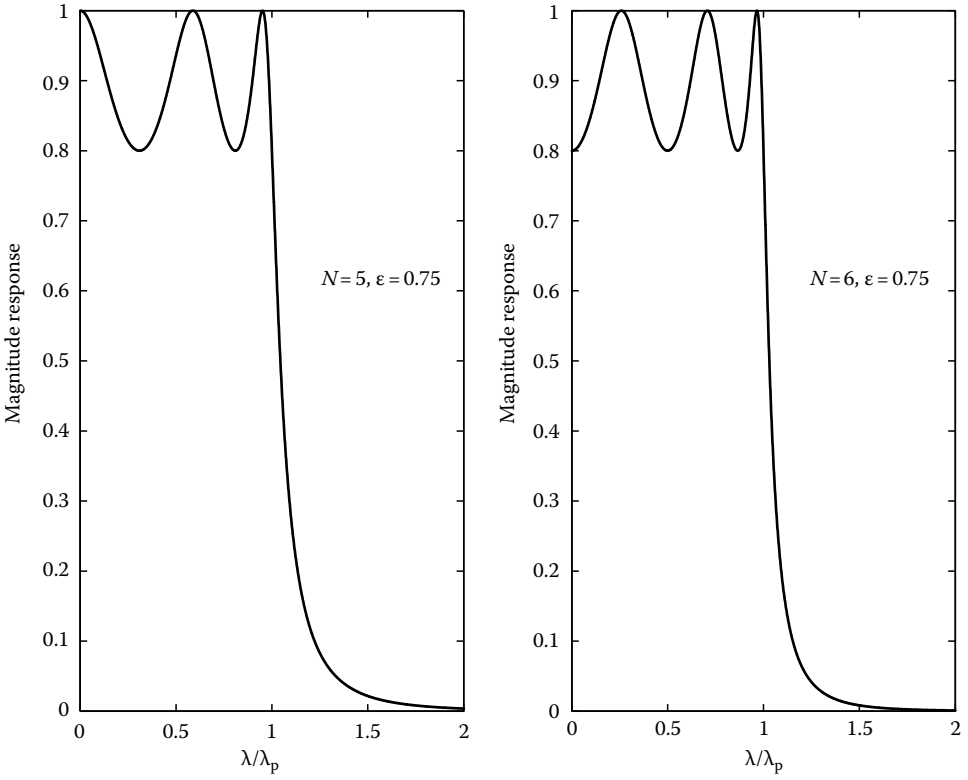


FIGURE 19.10 Magnitude responses of Chebyshev filters.

The filter order required to satisfy the specification in Equation 19.27 is

$$N \geq \frac{\cosh^{-1}\left[(\delta^{-2}-1)^{1/2}/\varepsilon\right]}{\cosh^{-1}(\lambda_s/\lambda_p)} = \frac{\log\{[(\delta^{-2}-1)^{1/2}/\varepsilon] + [(\delta^{-2}-1)/\varepsilon^2 - 1]^{1/2}\}}{\log\{(\lambda_s/\lambda_p) + [(\lambda_s/\lambda_p)^2 - 1]^{1/2}\}} \tag{19.38}$$

which can be computed knowing ε, δ, λ_p, and λ_s.

Table 19.3 shows the normalized Chebyshev polynomials $C_N(s)$ for different values of N and passband ripples. The transfer function of a unity dc gain, unity cutoff frequency Chebyshev Type-I filter is given by

$$H_a(s) = \frac{1}{C_N(s)}$$

19.2.2.3 Inverse Chebyshev Filters (Type-II Chebyshev Filters)

Notice from Figure 19.10 that the Chebyshev filter has magnitude response containing equiripples in the passband. The equiripples can be arranged to go inside the stopband, for which case we obtain inverse Chebyshev filters. The magnitude response square of the inverse Chebyshev filter is

TABLE 19.3 Normalized Chebyshev Polynomials $C_N(s)$ for Passband Ripple = 0.5, 1, and 3 dB

N	$C_N(s)$ for Passband Ripple 0.5 dB
1	$0.3493s + 1$
2	$0.6595s^2 + 0.9403s + 1$
3	$1.3972s^3 + 1.7506s^2 + 2.1446s + 1$
4	$2.6382s^4 + 3.1589s^3 + 4.5294s^2 + 2.7053s + 1$
5	$5.589s^5 + 6.5530s^4 + 10.8279s^3 + 7.3192s^2 + 4.2058s + 1$
6	$10.5527s^6 + 12.2324s^5 + 22.9188s^4 + 16.7763s^3 + 12.3663s^2 + 4.5626s + 1$
N	$C_N(s)$ for Passband Ripple 1 dB
1	$0.5088s + 1$
2	$0.9070s^2 + 0.9957s + 1$
3	$2.0354s^3 + 2.0117s^2 + 2.5206s + 1$
4	$3.6281s^4 + 3.4569s^3 + 5.2750s^2 + 2.6943s + 1$
5	$8.1416s^5 + 7.6272s^4 + 13.7496s^3 + 7.9331s^2 + 4.7265s + 1$
6	$14.5123s^6 + 13.4711s^5 + 28.0208s^4 + 17.4459s^3 + 13.6321s^2 + 4.4565s + 1$
N	$C_N(s)$ for Passband Ripple 3 dB
1	$0.9976s + 1$
2	$1.4125s^2 + 0.9109s + 1$
3	$3.9905s^3 + 2.3833s^2 + 3.7046s + 1$
4	$5.6501s^4 + 3.2860s^3 + 6.6057s^2 + 2.287s + 1$
5	$15.9621s^5 + 9.1702s^4 + 22.5867s^3 + 8.7622s^2 + 6.5120s + 1$
6	$22.6005s^6 + 12.8981s^5 + 37.5813s^4 + 15.6082s^3 + 15.8000s^2 + 3.6936s + 1$

$$|H_a(\lambda)|^2 = \frac{1}{1 + \frac{(\delta^{-2}-1)}{T_N^2(\lambda_s/\lambda)}} \tag{19.39}$$

Since $T_N^2(\pm 1) = 1$, Equation 19.39 gives $|H_a(\lambda_s)|^2 = \delta^2$. Figure 19.11 depicts some examples of Equation 19.38. Note that $|H_a(\infty)|$ equals 0 if N is odd and it equals δ if N is even.

The transfer function giving rise to Equation 19.39 is given by

$$H_a(s) = \begin{cases} C\prod_{i=1}^{N} \frac{(s-q_i)}{(s-p_i)} & N = \text{even} \\ \frac{C}{(s-p_{(N+1)/2})} \prod_{i=1, i\neq(N+1)/2}^{N} \frac{(s-q_i)}{(s-p_i)} & N = \text{odd} \end{cases} \tag{19.40}$$

where

$$C = \begin{cases} \prod_{i=1}^{N} \frac{p_i}{q_i} & N \text{ is even} \\ -p_{(N+1)/2} \prod_{i=1, i\neq(N+1)/2}^{N} \frac{p_i}{q_i} & N \text{ is odd} \end{cases} \tag{19.41}$$

$$p_i = \frac{\lambda_s}{\alpha_i^2 + \beta_i^2}(\alpha_i - j\beta_i) \quad q_i = j\frac{\lambda_s}{\cos\left(\frac{2i-1}{2N}\pi\right)} \tag{19.42}$$

$$\phi = \frac{1}{N}\cosh^{-1}(\delta^{-1}) = \frac{1}{N}\ln[\delta^{-1} + (\delta^{-2} - 1)^{1/2}] \tag{19.43}$$

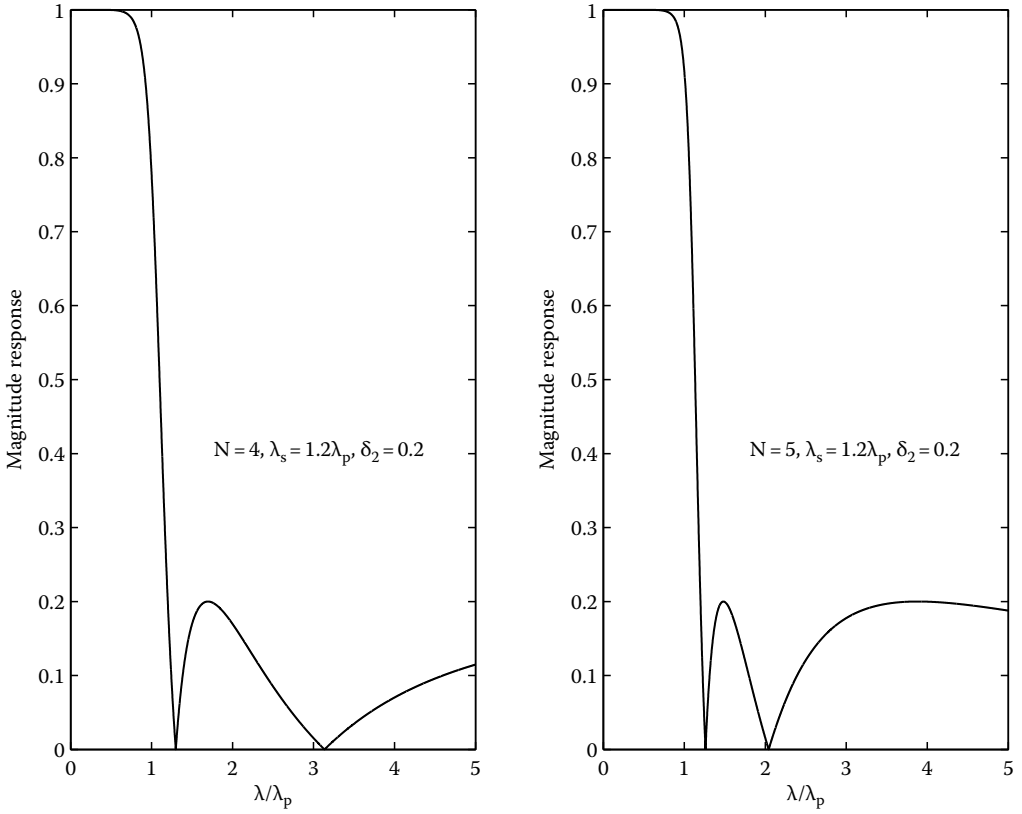


FIGURE 19.11 Magnitude responses of inverse Chebyshev filters.

Note that the zeros are on the imaginary axis on the *s* plane. The filter order N required to satisfy the specification in Equation 19.27 is the same as the order for the Chebyshev filter, given by Equation 19.38.

Another form for the inverse Chebyshev filter has magnitude response square given by

$$|H_a(\lambda)|^2 = \frac{1}{1 + \varepsilon^2 \frac{T_N^2(\lambda_s/\lambda_p)}{T_N^2(\lambda_s/\lambda)}} \tag{19.44}$$

which passes through $1/(1+\varepsilon^2)$ at $\lambda = \lambda_p$. For further details of this form see Ref. [1].

19.2.2.4 Elliptic Filters (Cauer Filters)

Elliptic filters have equiripples in both the passband and the stopband. We summarize the magnitude response and the transfer function of an elliptic filter as follows. Detail of derivation can be found in Refs. [2,3].

The magnitude response square is given by

$$|H_a(\lambda)|^2 = \frac{1}{1 + \varepsilon^2 R_N^2(\lambda)} \tag{19.45}$$

where $R_N(\lambda)$ is the Chebyshev rational function given by

$$R_N(\lambda) = \begin{cases} \frac{(\delta^{-2}-1)^{1/4}}{\varepsilon^{1/2}} \lambda \prod_{i=1}^{(N-1)/2} \frac{\lambda^2 - \lambda_r \mathrm{sn}^2\left[\frac{2iK(\lambda_r)}{N}, \lambda_r\right]}{\lambda^2 \lambda_r \mathrm{sn}^2\left[\frac{2iK(\lambda_r)}{N}, \lambda_r\right] - 1} & N = \text{odd} \\ \frac{(\delta^{-2}-1)^{1/4}}{\varepsilon^{1/2}} \lambda \prod_{i=1}^{(N-1)/2} \frac{\lambda^2 - \lambda_r \mathrm{sn}^2\left[\frac{(2i-1)K(\lambda_r)}{N}, \lambda_r\right]}{\lambda^2 \lambda_r \mathrm{sn}^2\left[\frac{(2i-1(\lambda_r)}{N}, \lambda_r\right] - 1} & N = \text{even} \end{cases} \tag{19.46}$$

Here, $\lambda_r = \lambda_p/\lambda_s, K(t)$ is the complete elliptic integral of the first kind given by

$$K(t) = \int_0^{\pi/2} \frac{d\theta}{(1 - t^2 \sin^2 \theta)^{1/2}} = \int_0^1 \frac{dx}{[(1-x^2)(1-t^2x^2)]^{1/2}} \tag{19.47}$$

The Jacobian elliptic sine function $\mathrm{sn}[u, t]$ is defined as

$$\mathrm{sn}[u, t] = \sin\phi \quad \text{if } u = \int_0^{\phi} \frac{d\theta}{(1 - t^2 \sin^2 \theta)^{1/2}} \tag{19.48}$$

The integral

$$F(\phi, t) = \int_0^{\phi} \frac{d\theta}{(1 - t^2 \sin^2 \theta)^{1/2}} = \int_0^{\sin\phi} \frac{dx}{[(1-x^2)(1-t^2x^2)]^{1/2}} \tag{19.49}$$

is called the elliptic integral of the first kind. Note that $K(t) = F(\pi/2, t)$.

The transfer function corresponding to the magnitude response in Equation 19.45 is

$$H_a(s) = \begin{cases} \frac{C}{(s+p_0)} \prod_{i=1}^{(N-1)/2} \frac{(s^2+B_i)}{(s^2+A_{i1}s+A_{i2})} & N \text{ odd} \\ C \prod_{i=1}^{N/2} \frac{(s^2+B_i)}{(s^2+A_{i1}s+A_{i2})} & N \text{ even} \end{cases} \tag{19.50}$$

$$C = \begin{cases} p_0 \prod_{i=1}^{(N-1)/2} \frac{A_{i2}}{B_i} & N \text{ odd} \\ \frac{1}{(1+\varepsilon^2)^{1/2}} \prod_{i=1}^{N/2} \frac{A_{i2}}{B_i} & N \text{ even} \end{cases} \tag{19.51}$$

The pole p_0 and the coefficients B_i, A_{i1} are calculated as follows:

$$\lambda_r = \frac{\lambda_p}{\lambda_s} \quad \lambda_c = \sqrt{\lambda_p \lambda_s} \quad \alpha = 0.5 \frac{1 - (1 - \lambda_r^2)^{1/4}}{1 + (1 - \lambda_r^2)^{1/4}} \tag{19.52}$$

$$\beta = e^{-\pi K[(1-\lambda_r^2)^{1/2}]/K(\lambda_r)} \approx \alpha + 2\alpha^5 + 15\alpha^9 + 150\alpha^{13} \tag{19.53}$$

$$\gamma = \frac{1}{2N} \ln\left[\frac{(1+\varepsilon^2)^{1/2} + 1}{(1+\varepsilon^2)^{1/2} - 1}\right] \tag{19.54}$$

$$\sigma = \left| \frac{2\beta^{1/4} \prod_{k=0}^{\infty} (-1)^k \beta^{k(k+1)} \sinh[(2k+1)\gamma]}{1 + 2\prod_{k=1}^{\infty} (-1)^k \beta^{k^2} \cosh[2k\gamma]} \right| \tag{19.55}$$

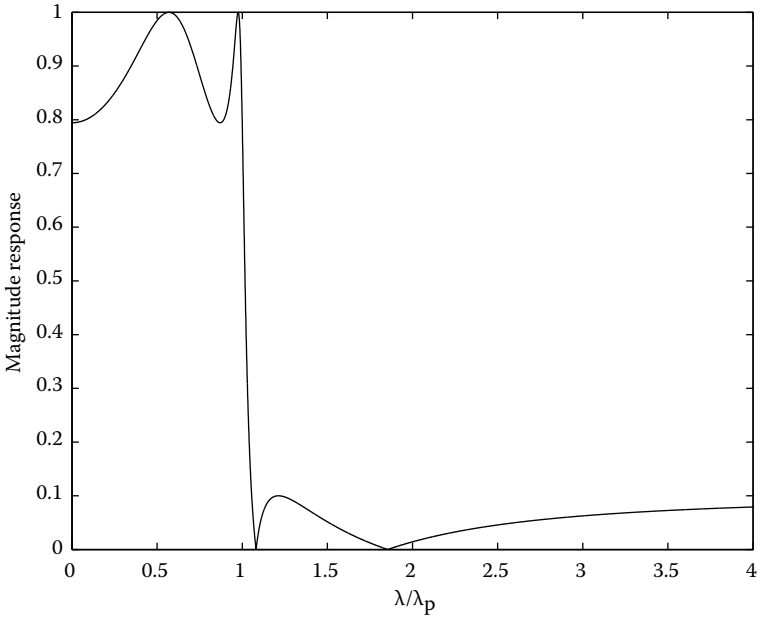


FIGURE 19.12 Magnitude response of elliptic filter.

$$\zeta = (1 + \lambda_r \sigma^2)\left(1 + \frac{\sigma^2}{\lambda_r}\right) \quad \eta = \begin{cases} i & N \text{ odd} \\ i - 0.5 & N \text{ even} \end{cases} \tag{19.56}$$

$$\psi_i = \frac{2\beta^{1/4} \sum_{k=0}^{\infty} (-1)^k \beta^{k(k+1)} \sin\left[(2k+1)\pi\eta/N\right]}{1 + 2\sum_{k=1}^{\infty} (-1)^k \beta^{k^2} \cos\left[2k\pi\eta/N\right]} \tag{19.57}$$

$$\mu_i = \left[(1 - \lambda_r \psi_i^2)\left(1 - \frac{\psi_i^2}{\lambda_r}\right)\right]^{1/2} \tag{19.58}$$

$$p_0 = \lambda_c \sigma \quad B_i = \frac{\lambda_c^2}{\psi_i^2} \quad A_{i1} = \frac{2\lambda_c \sigma \mu_i}{1 + \sigma^2 \psi_i^2} \quad A_{i2} = \lambda_c^2 \frac{\sigma^2 \mu_i^2 + \zeta \psi_i^2}{\left[1 + \sigma^2 \psi_i^2\right]^2} \tag{19.59}$$

The infinite summations above converge very quickly, so that only a few terms are needed in actual calculation. A simple program can be written to compute the values in Equations 19.52 through 19.59.

The filter order required to satisfy Equation 19.27 is calculated from

$$N \geq \frac{1}{\log(\beta)} \log\left[\frac{\varepsilon^2}{16(\sigma^{-2} - 1)}\right] \tag{19.60}$$

where β is given by Equation 19.53. An example of the magnitude response is plotted in Figure 19.12. We see that there are ripples in both the passband and the stopband.

19.2.2.5 Comparison

In comparing the filters given above, the Butterworth filter requires the highest order and the elliptic filter requires the smallest order to satisfy the same passband and stopband specifications. The Butterworth filter and the inverse Chebyshev filter have nicer (closer to linear) phase characteristics in the passband

than Chebyshev and elliptic filters. The magnitude responses of the Butterworth and Chebyshev filters decrease monotonically in the stopband to zero, which reduces the aliasing caused by some analog-to-digital transformation.

19.2.3 Design Using Analog Prototype Filters

In this subsection, we consider designing IIR filters using analog prototype filters. This method is suitable for designing the standard types of filters: low-pass filter (LPF), high-pass filter (HPF), bandpass filter (BPF), and bandstop filter (BSF). The basic idea is to transform the digital specification to analog specification, design an analog flter, and then transform the analog filter transfer function to digital filter system function. Several types of transformation have been studied.

The design steps are outlined in Figure 19.13. Given the desired magnitude response $|H^x(f)|$ of digital LPF, HPF, BPF, or BSF, it is transformed to analog magnitude specification (of the corresponding type: LPF, HPF, BPF, or BSF) $|H_a^x(\lambda)|$. The analog magnitude specification is then transformed to analog LPF magnitude specification $|H_a(\lambda)|$. We then design an analog prototype filter as discussed in Section 19.2.2, obtaining analog LPF transfer function $H_a(s)$. Next, the analog LPF transfer function is transformed to analog transfer function $H_a^x(s)$ of the desired type (LPF, HPF, BPF, or BSF), followed by a transformation to digital filter system function $H^x(z)$. By combining the appropriate steps, we can obtain transformations to go directly from $|H^x(f)|$ to $|H_a(\lambda)|$ and directly from $H_a(s)$ to $H(z)$, as indicated by the dotted lines in Figure 19.13. Note that for designing digital LPF, the middle steps involving $|H_a^x(\lambda)|$ and $H_a^x(s)$ are not applicable.

19.2.3.1 Transformations

There are several types of transformations. They arise from approximating continuous-time signals and systems by discrete-time signals and system. Table 19.4 shows several transformations, with their advantages and disadvantages. The constant T is the sampling interval. The resulting mapping is used for transforming $H_a(s)$ to $H(z)$. For example, in the backward difference approximation we obtain $H(z)$ by replacing the variable s with $(1 - z^{-1})/T$ in $H_a(s)$, i.e., $H(z) = H_a(s)|_{s=1-z^{-1}/T}$. The bilinear

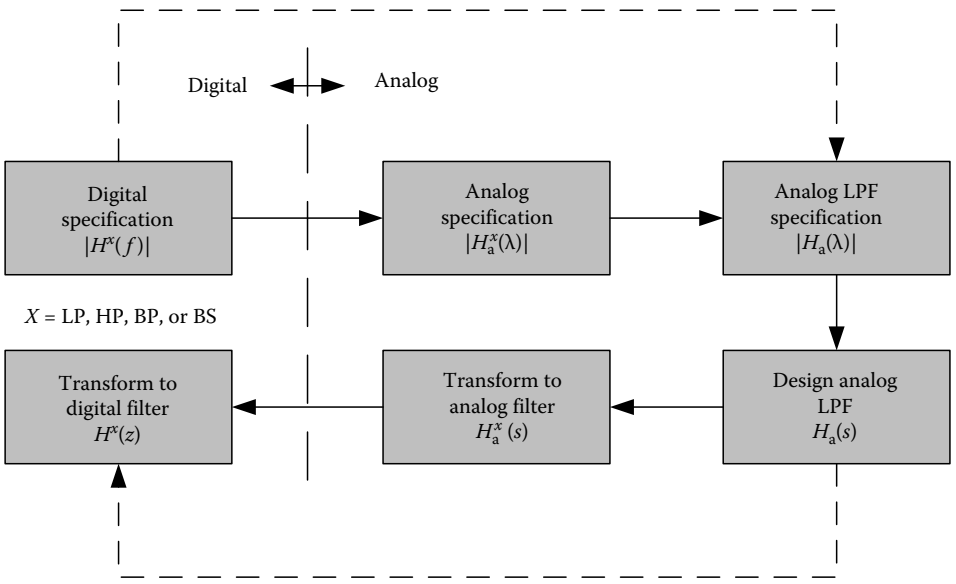
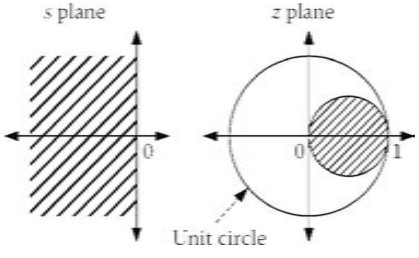
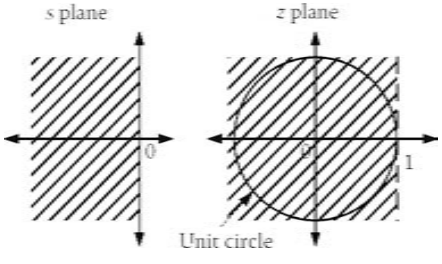


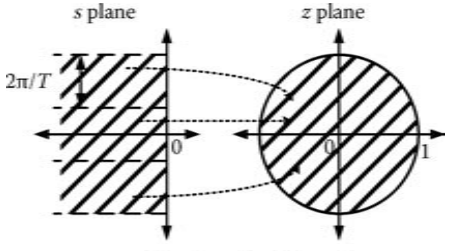
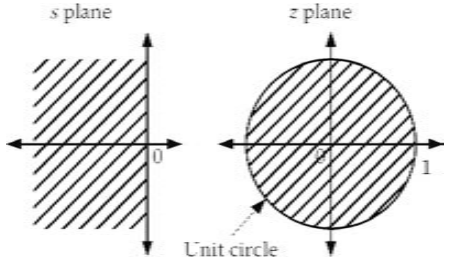
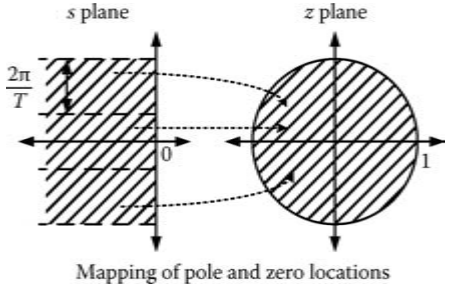
FIGURE 19.13 Diagram outlining the steps involved in designing IIR filter using analog prototype filter.

TABLE 19.4 Various Types of Analog-to-Digital Transformation

Type of Transformation	Principle	Resulting Mapping	Advantages	Disadvantages
Backward difference approximation	$\frac{dy}{dt} \approx \frac{y(n) - y(n-1)}{T}$	$s = \frac{1 - z^{-1}}{T}$, $z = \frac{1}{1 - sT}$ 	Stable analog filter yields stable digital filter	Left-half s plane is not mapped onto unit circle on z plane. Pole locations will be in the circle centered at 0.5 with radius 0.5 on the z plane
Forward difference approximation	$\frac{dy}{dt} \approx \frac{y(n+1) - y(n)}{T}$	$s = \frac{z - 1}{T}$, $z = 1 + sT$ 		Stable analog filler does not yield stable digital filter Left-half s plane is not mapped onto unit circle on z plane
Impulse invariant method	Sample the analog impulse response: $h(n) = h_a(nT)$	Transform $H_a(s) = \sum_{i=1}^{N} \frac{b_i}{s - p_i}$ to $H(z) = \sum_{i=1}^{N} \frac{b_i}{[1 - z^{-1} \exp(p_i T)]}$	Preserve shape of impulse response Stable analog filter yields stable digital filter The analog frequency and digital frequency are linearly related, $\lambda T = 2\pi f, -0.5 \le f \le 0.5$	Aliasing in the frequency domain (due to many-to-one mapping from s to z plane) $T \cdot H(f) = \sum_{k=-\infty}^{\infty} H_a\left(2\pi \frac{f + k}{T}\right)$

(*continued*)

TABLE 19.4 (continued) Various Types of Analog-to-Digital Transformation

Type of Transformation	Principle	Resulting Mapping	Advantages	Disadvantages
		 Mapping of pole locations		
Bilinear transformation	Approximation $y(t) = \int_{t-T}^{t} y'(\tau)d\tau + y(t-T)$ by $y(n) = \frac{T}{2}[y'(n) + y'(n-1) + y(n-1)]$	$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right), \quad z = \frac{(2/T)+s}{(2/T)-s}$ 	Stable analog filter yields stable digital filter Left-half s plane is mapped onto unit circle z plane, on a one-to-one mapping	Frequency warping nonlinear relation between analog frequency and digital frequency
Matched z transformation	Map zero on s plane directly to pole and zero on z plane	Transform $H_a(s) = \sum_{i=1}^{N} \frac{s-q_i}{s-p_i}$ to $H(z) = \sum_{i=1}^{N} \frac{[1-z^{-1}\exp(q_i T)]}{[1-z^{-1}\exp(p_i T)]}$  Mapping of pole and zero locations	Stable analog filter yields stable digital filter	Aliasing in the frequency domain (due to many-to-one mapping from s to z plane)

transformation is the best all-around method, followed by the impulse invariant method. Therefore, we describe these two transformations in more detail.

19.2.3.1.1 Bilinear Transformations

Using this transformation, the analog filter is converted to digital filter by replacing s in the analog filter transfer function with $(2/T)(1 - z^{-1})/(1 + z^{-1})$, i.e.,

$$H(z) = H_a(s)|_{s=(2/T)(1-z^{-1})/(1+z^{-1})} \tag{19.61}$$

From the mapping, we can show as follows that the imaginary axis on the s plane is mapped to the unit circle on the z plane. Letting $s = j\lambda$, we have

$$\begin{aligned} z &= \frac{(2/T) + s}{(2/T) - s} = \frac{(2/T) + j\lambda}{(2/T) - j\lambda} \\ &= \frac{\sqrt{(2/T)^2 + (\lambda)^2}\, e^{j \arctan(\lambda/(2/T))}}{\sqrt{(2/T)^2 + (\lambda)^2}\, e^{j \arctan(-\lambda/(2/T))}} \end{aligned} \tag{19.62}$$

which is the unit circle on the z plane as λ goes from $-\infty$ to ∞. Writing $z = e^{j2\pi f}$ in Equation 19.62, we obtain the relation between the analog frequency λ and the digital frequency f:

$$f = \frac{1}{\pi} \arctan\left(\frac{\lambda T}{2}\right) \quad \lambda = \frac{2}{T} \tan(\pi f) \tag{19.63}$$

which is plotted in Figure 19.14. Equation 19.63 is used for converting digital specification to analog specification, i.e., $\lambda_s = (2/T)\tan(\pi f_s)$ and $\lambda_p = (2/T)\tan(\pi f_p)$. In a complete design process, starting from the digital specification and ending at the digital filter system function, as outlined in Figure 19.13, the sampling interval T is canceled out in the process. Hence, it has no effect and any convenient value (such as 1 or 2) can be used.

19.2.3.1.2 Impulse Invariance Method

This method approximates the analog filter impulse response $h_a(t)$ by its samples separated by T seconds. The result is the impulse response $h(n)$ of the digital filter, i.e., $h(n) = h_a(nT)$. From this relation, it can be shown that

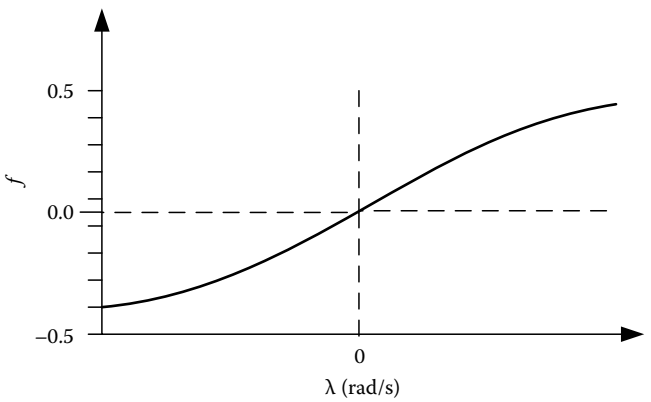


FIGURE 19.14 Relation between λ and f for bilinear transformation.

$$H(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(\lambda)\Big|_{\lambda=2\pi(f+k)/T} = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(2\pi\frac{f+k}{T}\right) \tag{19.64}$$

The analog and digital frequencies are related by

$$f = \frac{\lambda T}{\pi}, \quad |f| \le 0.5 \tag{19.65}$$

From Equation 19.64, the digital filter frequency response is the sum of shifted versions of the analog filter frequency response. There is aliasing if $H_a(\lambda)$ is not zero for $|\lambda/2\pi| > 1/(2T)$. Therefore, the analog filter used in this method should have a frequency response that goes to zero quickly as λ goes to ∞. Because of the aliasing, this method cannot be used for designing a HPF. Writing the analog filter transfer function in the form

$$H_a(s) = \sum_{i=1}^{N} \frac{b_i}{(s-p_i)} \tag{19.66}$$

it follows that the analog impulse response is given by $h_a(t) = \sum_{i=1}^{N} b_i e^{p_i t}$ and the digital filter can be obtained as

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n)z^{-n} = \sum_{n=0}^{\infty} h(nT)z^{-n} \\ &= \sum_{i=1}^{N} b_i \sum_{n=0}^{\infty} (e^{p_i T}z^{-1})^n = \sum_{i=1}^{N} \frac{b_i}{1-e^{p_i T}z^{-1}} \end{aligned} \tag{19.67}$$

Therefore, an analog filter transfer function $H_a(s) = \sum_{i=1}^{N} b_i/(s-p_i)$ gets transformed to a digital filter system function $H(z) = \sum_{i=1}^{N} b_i/(1-e^{p_i T}z^{-1})$, as shown in Table 19.4. Similar to the bilinear transformation, in a complete design process the choice of T has no effect (except for the final magnitude scaling factor).

19.2.3.2 Low-Pass Filters

We give one example in designing an LPF using the impulse invariant method and one example using the bilinear transformation. In this example, suppose that we wish to design a digital filter using an analog Butterworth prototype filter. The digital filter specification is

$$\begin{aligned} &20\log|H(f)| \ge -2\,\text{dB} \quad \text{for } 0 \le f \le 0.11 \\ &20\log|H(f)| \le -10\,\text{dB} \quad \text{for } 0.2 \le f \le 0.5 \end{aligned}$$

where the log is of base 10. Therefore, we have $\varepsilon = 0.7648$, $\delta = 0.3162$, $f_p = 0.11$, and $f_s = 0.2$. Let us use the impulse invariant method. Therefore, the analog passband edge and stopband edge are $\lambda_p = 0.22\pi/T$ and $\lambda_s = 0.4\pi/T$, respectively. We use the same ripple requirements: $\varepsilon = 0.7648$ and $\delta = 0.3162$. Using these values, a Butterworth filter order is calculated from Equation 19.30, yielding $N \ge 2.3$. So, we choose $N = 3$. With $\lambda_c = \lambda_p \varepsilon^{-1/N} = 0.2406\pi/T$, we find the analog filter transfer function to be

$$\begin{aligned} H_a(s) &= \frac{\lambda_c^3}{(s+\lambda_c)(s^2+\lambda_c s+\lambda_c^2)} \\ &= \lambda_c\left[\frac{1}{s+\lambda_c} + \frac{-0.5-j0.5/\sqrt{3}}{s+0.5(1-j\sqrt{3})\lambda_c} + \frac{-0.5+j0.5/\sqrt{3}}{s+0.5(1+j\sqrt{3})\lambda_c}\right] \\ &= \frac{0.7559}{T}\left[\frac{1}{s+0.7559/T} + \frac{-0.5-j0.5/\sqrt{3}}{s+0.3779(1-j\sqrt{3})/T} + \frac{-0.5+j0.5/\sqrt{3}}{s+0.3779(1+j\sqrt{3})/T}\right] \end{aligned}$$

Using Equations 19.66 and 19.67 we obtain the digital filter system function:

$$H(z) = \frac{0.7559}{T}\left[\frac{1}{1-e^{-0.7559}z^{-1}} + \frac{-0.5-j0.5/\sqrt{3}}{1-e^{-0.3779(1-j\sqrt{3})}z^{-1}} + \frac{-0.5+j0.5/\sqrt{3}}{1-e^{-0.3779(1-j\sqrt{3})}z^{-1}}\right]$$
$$= \frac{0.7559}{T}\left[\frac{1}{1-0.4696z^{-1}} - \frac{1-0.7846z^{-1}}{1-1.0873z^{-1}+0.4696z^{-2}}\right]$$

Due to aliasing, the maximum value of the resulting magnitude response (which is at $f = 0$ or $z = 1$) is no longer equal to 1, although the analog filter has maximum magnitude (at $\lambda = 0$ or $s = 0$) of 1. Note that the choice of T affects only the scaling factor, which is only a constant gain factor. If we adjust the system function so that the maximum magnitude is 1, that is, $|H(f)| = 1$, we have

$$H(z) = \frac{0.7565}{T}\left[\frac{1}{1-0.4696z^{-1}} - \frac{1-0.7846z^{-1}}{1-1.0873z^{-1}+0.4696z^{-2}}\right]$$

The magnitude response in decibels and the phase response in degrees are plotted in Figure 19.15. From the result, $|H(f)| = -1.97$ dB at $f = 0.11$ and $|H(f)| = -13.42$ dB at $f = 0.2$; both satisfy the desired specification. The aliasing in this example is small enough that the resulting response still meets the specification. It is possible that the aliasing is large enough that the designed filter does not meet the specification. To compensate for the unknown aliasing, we may want to use smaller ε and δ in designing the analog prototype filter.

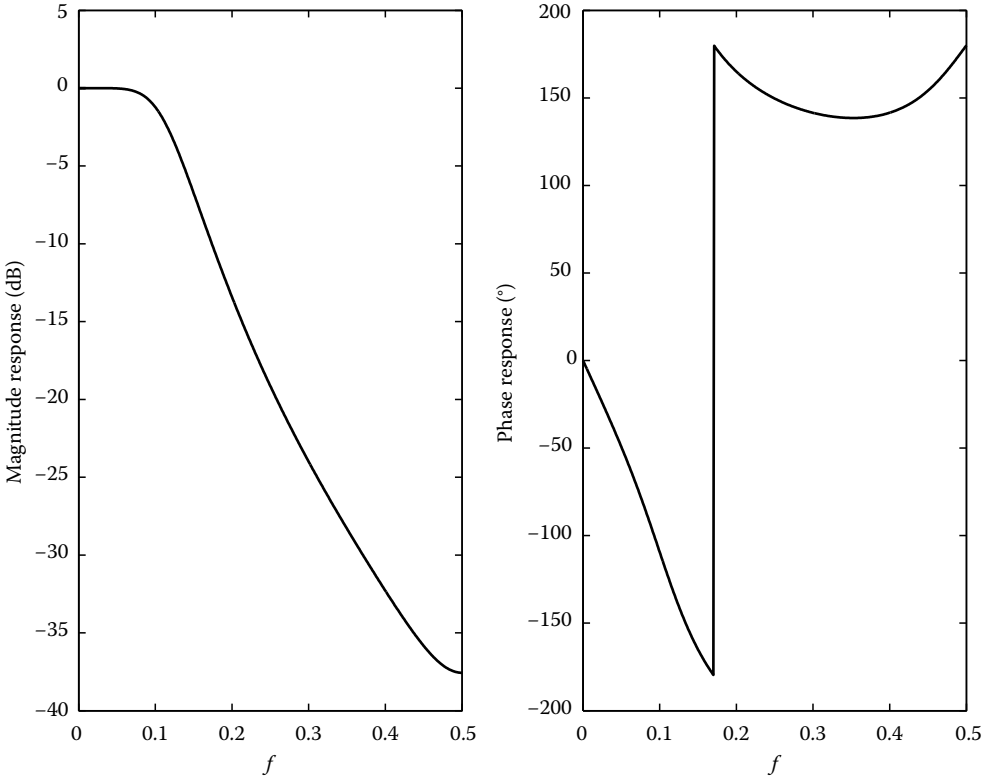


FIGURE 19.15 Frequency response of the LPF designed using impulse invariant method.

In this next example, we demonstrate the design method using bilinear transformation, with an analog elliptic prototype filter. Let the desired filter specification be

$$|H(f)|^2 \geq 0.8 \text{ (or } -0.97\,\text{dB)} \quad \text{for } 0 \leq f \leq 0.1125$$
$$|H(f)|^2 \leq 2.5 \times 10^{-5} \text{ (or } -46.02\,\text{dB)} \quad \text{for } 0.15 \leq f \leq 0.5$$

which means $\varepsilon = 0.5$, $\delta = 0.005$, $f_p = 0.1125$, and $f_s = 0.15$. For bilinear transformation, we calculate the analog passband and stopband edges as $\lambda_p = (2/T)\tan(\pi f_p) = 0.7378/T$ and $\lambda_s = (2/T)\tan(\pi f_s) = 1.0190/T$, respectively. Therefore, $\lambda_p/\lambda_s = 0.7240$. From Equation 19.60, we obtain the order $N \geq 4.8$. So, we use $N = 5$. The analog elliptic filter transfer function is calculated from Equations 19.50 through 19.59 to be

$$H_a(s) = \frac{7.8726 \times 10^{-3}\left[\left(\frac{sT}{2}\right)^2 + 0.6006\right]\left[\left(\frac{sT}{2}\right)^2 + 0.2782\right]}{\left[\left(\frac{sT}{2}\right) + 0.1311\right]\left[\left(\frac{sT}{2}\right)^2 + 0.1689\left(\frac{sT}{2}\right) + 0.0739\right]\left[\left(\frac{sT}{2}\right)^2 - 0.0457\left(\frac{sT}{2}\right) + 0.1358\right]}$$

To convert to digital filter system function, we replace s with $(2/T)(1 - z^{-1})/(1 + z^{-1})$. Equivalently, we replace $sT/2$ with $(1 - z^{-1})/(1 + z^{-1})$, yielding

$$H(z) = \frac{1.0511 \times 10^{-2}(1 + z^{-1})(1 - 0.4991Z^{-1} + z^{-2})(1 - 1.1294z^{-1} + z^{-2})}{(1 - 0.7682z^{-1})(1 - 1.4903z^{-1} + 0.7282z^{-2})(1 - 1.5855z^{-1} + 1.0838z^{-2})}$$

Note that the choice of T has no effect on the resulting system function. The magnitude response in decibels and the phase response are plotted in Figure 19.16, which satisfies the desired magnitude specification. Note the equiripples in both the passband and the stopband.

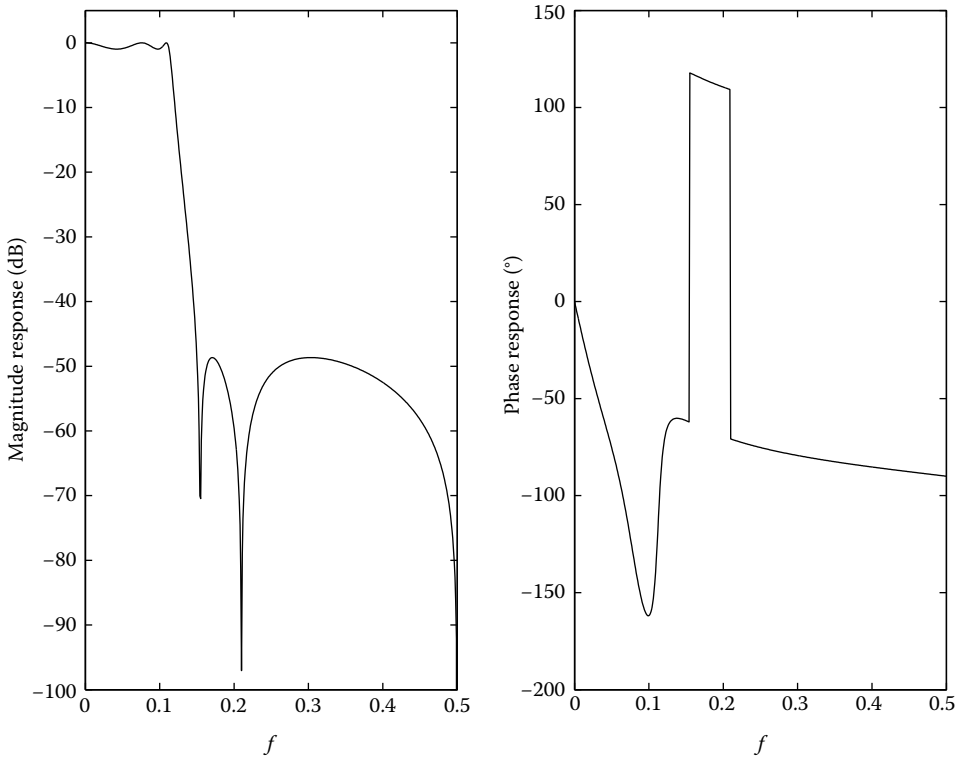


FIGURE 19.16 Frequency response of the LPF designed using bilinear transformation.

19.2.3.3 High-Pass Filters

As mentioned above, the impulse invariant method is not suitable for HPFs due to aliasing. Therefore, we only discuss the bilinear transformation. In addition to the procedure used with designing an LPF, we need to transform the analog high-pass specification to analog low-pass specification and transform the resulting analog LPF to analog HPF. There is a simple transformation for this job: replacing s in the analog LPF transfer function with $1/s$. In terms of the frequency, $j\lambda$ becomes $1/j\lambda = j(-1/\lambda)$, i.e., a low frequency is changed to a (negative) high frequency. Therefore, an analog LPF becomes an analog HPF. When combined with the bilinear transformation, this process gives the transformation

$$s = \frac{T}{2}\frac{(1+z^{-1})}{(1-z^{-1})} \quad \text{or} \quad z = \frac{s+(T/2)}{s-(T/2)} \tag{19.68}$$

Writing $s = j\lambda$, we can show that $z = \exp\{j[2\arctan(2\lambda/T) - \pi]\}$. With $z = \exp(j2\pi f)$, we have

$$\lambda = \frac{T}{2}\tan[\pi(f+0.5)] \tag{19.69}$$

To write f in terms of λ, we can show that, after adjusting the range of f to $[-1/2, 1/2]$,

$$f = \begin{cases} -\frac{1}{2} + \frac{1}{\pi}\arctan(2\lambda/T) & \lambda > 0 \\ +\frac{1}{2} + \frac{1}{\pi}\arctan(2\lambda/T) & \lambda < 0 \end{cases} \tag{19.70}$$

Equations 19.69 and 19.70 give the relation between the digital frequency and the analog frequency, corresponding to the transformation in Equation 19.68. This relation is plotted in Figure 19.17, from which we see that a low digital frequency corresponds to a high analog frequency and vice versa.

We can summarize the design steps as follows. Given a digital HPF specification as in Figure 19.18, it is converted to an analog LPF specification using Equation 19.70 to obtain the passband and stopband edges λ_p and λ_s, from f_p and f_s, respectively. With λ_p, λ_s, ε, and δ, we design the low-pass analog prototype filter. Let the transfer function be $H_a(s)$. This transfer function is then converted to digital HPF system function by replacing s with $(T/2)(1+z^{-1})/(1-z^{-1})$.

Note that this corresponds to the procedure in Figure 19.13, with the bypass of the "analog specification" block and the "transform to analog filter" block, as indicated by the dotted lines in Figure 19.13.

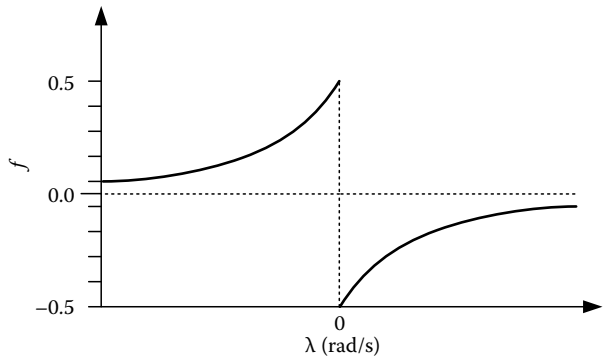


FIGURE 19.17 The relation for designing an HPF.

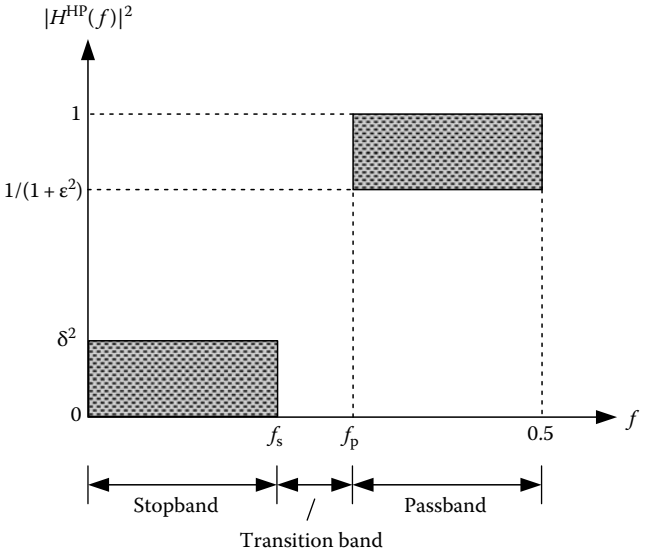


FIGURE 19.18 Digital HPF specification.

As an example, consider designing a digital HPF with the following specification:

$$|H^{HP}(f)|^2 \geq 0.8 \text{ (or } -0.97\,\text{dB)} \quad \text{for } 0.4 \leq f \leq 0.5$$
$$|H^{HP}(f)|^2 \leq 2.5 \times 10^{-5} \text{ (or } -46.02\,\text{dB)} \quad \text{for } 0 \leq f \leq 0.3$$

Since T does not affect the result, we let $T = 2$ for convenience. We calculate the analog LPF passband and stopband edges as $\lambda_p = \tan[\pi(0.5 + f_p)] = -0.3249$ rad/s and $\lambda_s = \tan[\pi(0.5 + f_s)] = -0.7265$ rad/s. Since the magnitude response is in symmetry with respect to $\lambda = 0$, we use $\lambda_p = 0.3249$ rad/s and $\lambda_s = 0.7265$ rad/s. Therefore, $\lambda_s/\lambda_p = 2.2361$. Suppose that we choose the inverse Chebyshev filter as the analog prototype filter. From Equation 19.37, we obtain the order $N \geq 4.6$. So, we use $N = 5$. From Equation 19.40, the low-pass analog inverse Chebyshev filter transfer function is

$$H_a(s) = \frac{1.8160 \times 10^{-2}(s^2 + 0.5835)(s^2 + 1.5276)}{(s + 0.4822)(s^2 + 0.6772s + 0.2018)(s^2 - 0.2131s + 0.1663)}$$

To convert to digital filter system function, we replace s with $(1 + z^{-1})/(1 - z^{-1})$, yielding

$$H^{HP}(s) = \frac{1.8160 \times 10^{-2}(1 - z^{-1})(1 + 0.5261z^{-1} + z^{-2})(1 - 0.4175z^{-1} + z^{-2})}{(1 + 0.3493z^{-1})(1 + 0.8498z^{-1} + 0.2792z^{-2})(1 + 1.2088z^{-1} + 0.6910z^{-2})}$$

The magnitude response and the phase response are plotted in Figure 19.19.

19.2.3.4 Bandpass Filters

A magnitude response specification for a digital BPF is depicted in Figure 19.20a. Note that there are two passband edges (f_{p1} and f_{p2}) and two stopband edges (f_{s1} and f_{s2}). For the bilinear transformation $s = (2/T)(1 - z^{-1})/(1 - z^{-1})$ we can transform the digital BPF specification to an analog BPF specification by letting

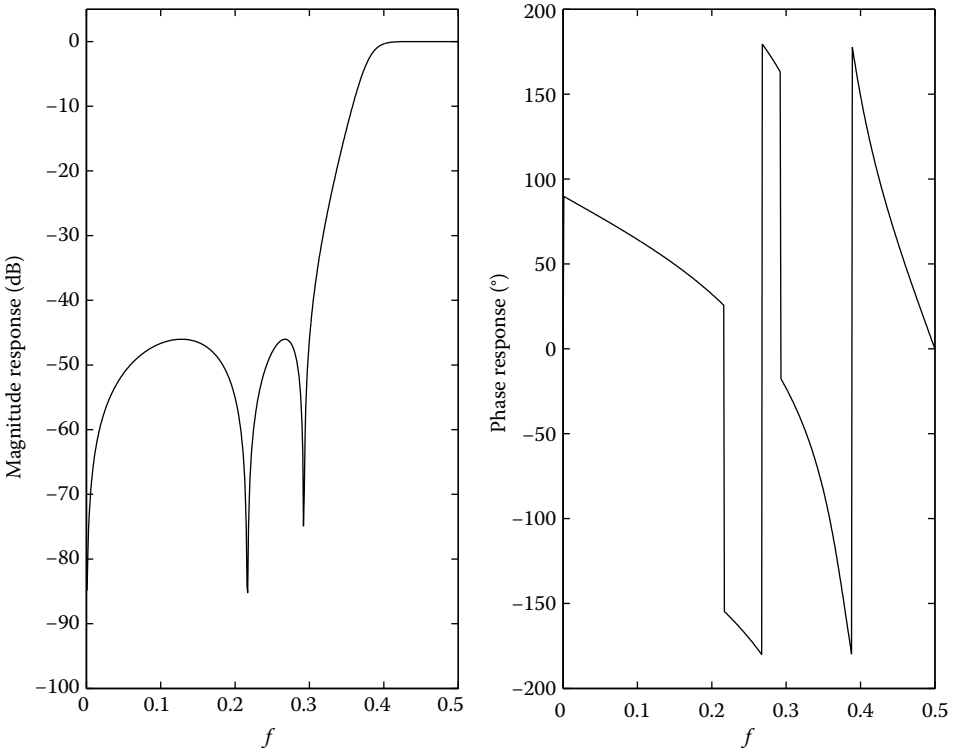


FIGURE 19.19 Frequency response of the HPF designed using bilinear transformation.

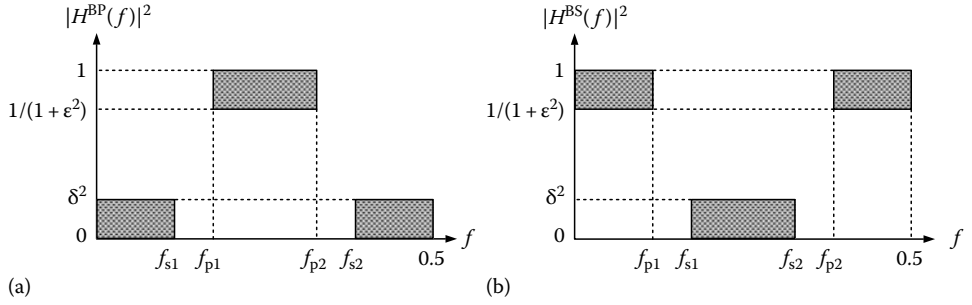


FIGURE 19.20 Magnitude specifications for digital BPF and BSF: (a) digital BPF specification and (b) digital BSF specification.

$$\begin{aligned} \lambda_{p1} &= 2T\tan(\pi f_{p1}) \quad \lambda_{p2} = 2T\tan(\pi f_{p2}) \\ \lambda_{s1} &= 2T\tan(\pi f_{s1}) \quad \lambda_{s2} = 2T\tan(\pi f_{s2}) \end{aligned} \tag{19.71}$$

and keeping the same ε and δ.

Now, we need a transformation between an analog BPF and an analog LPF. To distinguish between the variable s and λ for the two filters, let us use s' and λ' for the analog LPF and s and λ for the analog BPF, respectively. A transformation for converting an analog LPF to an analog BPF is given by

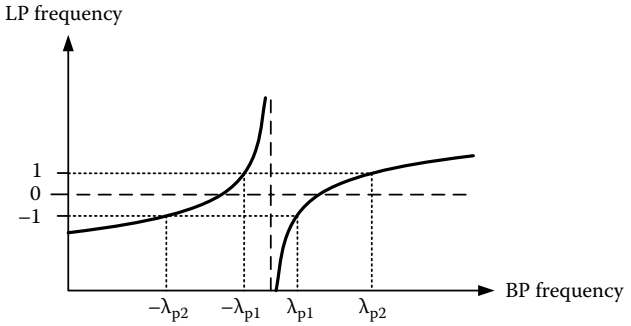


FIGURE 19.21 Relation between λ and λ' for bandpass-to-low-pass conversion.

$$s' = \frac{s^2 + \lambda_0^2}{Ws} \quad \text{or} \quad \lambda' = \frac{\lambda^2 - \lambda_0^2}{W\lambda} \tag{19.72}$$

where

$$W = \lambda_{p2} - \lambda_{p1} \quad \text{and} \quad \lambda_0^2 = \lambda_{p1}\lambda_{p2} \tag{19.73}$$

Figure 19.21 depicts an example of the relation between λ and λ'. Note that λ_{p1} and λ_{p2} get mapped to $\lambda' = -1$ and $+1$, respectively. Therefore, the analog LPF has a passband edge of 1. The values of λ_{s1} and λ_{s2} get mapped to $\lambda'_{s1} = -|(\lambda_{s1}^2 - \lambda_0^2)/(W\lambda_{s1})|$ and $\lambda'_{s2} = |(\lambda_{s2}^2 - \lambda_0^2)/(W\lambda_{s2})|$. However, these two values may not be negative of each other. Since the analog LPF must have a symmetric magnitude response, we must use the more stringent of the two stopband edges, i.e., the smaller of $|\lambda'_{s1}|$ and $|\lambda'_{s2}|$. Letting

$$\lambda'_s = \min\{|\lambda'_{s1}|, |\lambda'_{s2}|\} = \min\left\{\left|\frac{\lambda_{s1}^2 - \lambda_0^2}{W\lambda_{s1}}\right|, \left|\frac{\lambda_{s2}^2 - \lambda_0^2}{W\lambda_{s2}}\right|\right\} \tag{19.74}$$

we now have the analog LPF specification. Therefore, a prototype analog LPF can be designed.

The design process can be summarized as follows. First, the desired digital BPF magnitude specification is converted to an analog BPF magnitude specification using Equation 19.71. Then the analog BPF specification is converted to an analog LPF specification using λ'_s calculated from Equation 19.74 and $\lambda'_p = 1$. Next, a prototype analog LPF is designed with the values of $\varepsilon, \delta, \lambda'_p = 1$, yielding an analog LPF transfer function $H_a(s')$. The LPF transfer function is converted to an analog BPF transfer function $H_a^{HP}(s)$, using the transformation (from s' to s) given in Equation 19.72. Finally, the analog BPF transfer function is converted to a digital BPF transfer function $H^{BP}(z)$ using the bilinear transformation $s = (2/T)(1 - z^{-1})/(1 + z^{-1})$. As before, the value of T does not affect the result.

For example, let the desired digital BPF have the following specification:

$$|H^{BP}(f)|^2 \begin{cases} \geq 0.8\ (-0.97\ \text{dB}) & \text{for } 0.25 \leq f \leq 0.3 \\ \leq 2.5 \times 10^{-5}\ (-46.02\ \text{dB}) & \text{for } 0 \leq f \leq 0.2 \text{ and } 0.35 \leq f \leq 0.5 \end{cases}$$

which means $\varepsilon = 0.5, \delta = 0.005, f_{p1} = 0.25, f_{p2} = 0.3, f_{s1} = 0.2$, and $f_{s2} = 0.35$. Let $T = 2$ for convenience. Using $\lambda = \tan(\pi f)$, we obtain the analog BPF passband and stopband edges as $\lambda_{p1} = \tan(\pi f_{p1}) = 1.0$ rad/s, $\lambda_{p2} = \tan(\pi f_{p2}) = 1.3764$ rad/s, $\lambda_{s1} = \tan(\pi f_{s1}) = 0.7265$ rad/s, and $\lambda_{s2} = \tan(\pi f_{s2}) = 1.9626$ rad/s. Therefore, $\lambda_0^2 = 1.3764$ and $W = 0.3764$. So, we have $\lambda'_s = \min\{3.1030, 3, 3509\} = 3.1030$ rad/s. Suppose

that we use the elliptic LPF as an analog prototype filter. With $\varepsilon = 0.5$, $\delta = 0.005$, $\lambda'_p = 1$, and $\lambda'_s = 3.1030$ rad/s, we need an elliptic filter of order $N = 3$. The low-pass analog elliptic filter transfer function is

$$H_a(s') = \frac{4.1129 \times 10^{-2}(s'^2 + 12.6640)}{(s' + 0.5174)(s'^2 + 0.763s' + 1.0067)}$$

Replacing s' with $(s^2 + 1.3764)/(0.3764s)$ yields the analog BPF transfer function

$$H_a^{BP}(s) = \frac{1.5480 \times 10^{-2}s(s^4 + 4.5467s^2 + 1.8944)}{(s^2 + 0.1947s + 1.3764)(s^4 + 0.1793s^3 + 2.8953s^2 + 0.2467s + 1.8944)}$$

Note that an Nth-order LPF becomes a $2N$th-order BPF. To convert to digital filter system function, we replace s with $(1 - z^{-1})/(1 + z^{-1})$, yielding

$$H^{BP}(z) = \frac{7.2077 \times 10^{-3}(1 - z^{-2}) \times (1 + 0.4807z^{-1} + 1.1117z^{-2} + 0.4807z^{-3} + z^{-4})}{(1 + 0.2928z^{-1} + 0.8485z^{-2}) \times (1 + 0.5973z^{-1} + 1.8623z^{-2} + 0.5539z^{-3} + 0.8629z^{-4})}$$

The magnitude and phase responses are plotted in Figure 19.22.

Note that for the transformation in Equation 19.72, we can also let $W = \lambda_{s2} - \lambda_{s1}$ and $\lambda_0^2 = \lambda_{s1}\lambda_{s2}$, instead of Equation 19.73. Such a choice will give $\lambda'_s = 1$. The passband edge for the prototype LPF is now calculated from $\lambda'_p = \min\{|(\lambda_{p1}^2 - \lambda_0^2)/(W\lambda_{p1})|\ |\lambda_{p2}^2 - \lambda_0^2)/(W\lambda_{p2})|\}$.

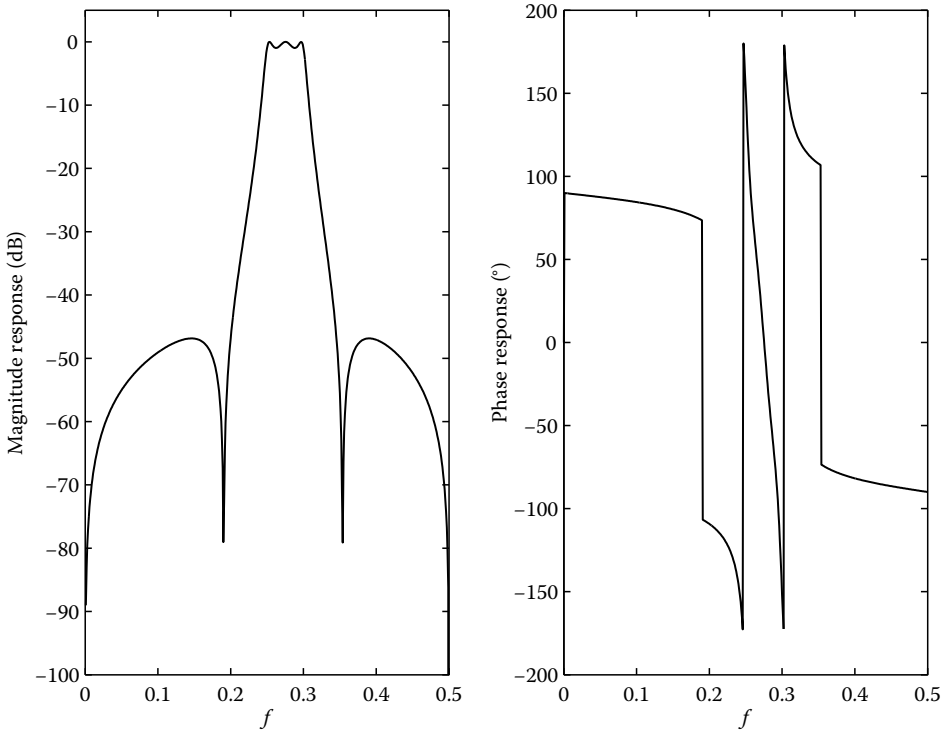


FIGURE 19.22 Frequency response of the designed digital BPF.

19.2.3.5 Bandstop Filters

A digital BSF specification is depicted in Figure 19.20b. As in the case of the BPF there are two passband edges (f_{p1} and f_{p2}) and two stopband edges (f_{s1} and f_{s2}). A transformation from analog BSF to analog LPF is given by

$$s' = \frac{Ws}{s^2 + \lambda_0^2} \quad \text{or} \quad \lambda' = \frac{W\lambda}{\lambda_0^2 - \lambda^2} \tag{19.75}$$

where W and λ_0^2 are given by Equation 19.73. Note that the expression for s in Equation 19.75 is the reciprocal of that in Equation 19.72. The relation between the LPF frequency λ' and the BSF frequency λ is depicted in Figure 19.23. The passband edges λ_{p1} and λ_{p2} get mapped to $\lambda' = 1$ and -1, respectively. The values of λ_{s1} and λ_{s2} get mapped to $\lambda'_{s1} = W\lambda_{s1}/(\lambda_0^2 - \lambda_{s1}^2)$ and $\lambda'_{s2} = W\lambda_{s2}/(\lambda_0^2 - \lambda_{s2}^2)$. Therefore, the passband edge and stopband edge of the prototype analog LPF are 1 and λ'_s, respectively, where

$$\lambda'_s = \min\{|\lambda'_{s1}|, |\lambda'_{s2}|\} = \min\left\{\left|\frac{W_{s1}\lambda}{\lambda_0^2 - \lambda_{s1}^2}\right|, \left|\frac{W\lambda_{s2}}{\lambda_0^2 - \lambda_{s2}^2}\right|\right\} \tag{19.76}$$

The design process for the BSF can follow the same process as the design for the BPF, except that we use Equations 19.75 and 19.76 instead of Equations 19.62 and 19.64.

Similar to the case of the BPF, we can also let $W = \lambda_{s2} - \lambda_{s1}$ and $\lambda_0^2 = \lambda_{s1}\lambda_{s2}$ instead of Equation 19.73, for the transformation in Equation 19.75. The stopband edge and the passband edge for the prototype LPF are now $\lambda'_s = 1$ and $\lambda'_p = \min\{|W\lambda_{p1}/(\lambda_0^2 - \lambda_{p1}^2)|, |W\lambda_{p2}/(\lambda_0^2 - \lambda_{p2}^2)|\}$.

19.2.4 Design Using Digital Frequency Transformations

This method assumes that we can design a digital LPF. The desired filter is then obtained from the digital LPF by transforming the digital LPF in the z domain. Let us denote the z variable for the digital LPF by z' and that for the desired digital filter by z. Similarly, we use f' for the digital frequency of the digital LPF and f for the frequency of the desired digital filter. Suppose that the digital LPF has system function $H(z')$ and the desired digital filter has system function $H^x(z)$, where x stands for LP, HP, BP, or BS. The system function $H^x(z)$ is obtained from $H(z')$ by replacing z' with an appropriate function of z. The LPF $H(z')$ can be designed using the method discussed in the Section 19.2.3, or by some other means. The specification for the digital LPF is obtained from the specification of the desired digital filter through the relation between f' and f. The relation depends on the specific transformation. Note that the

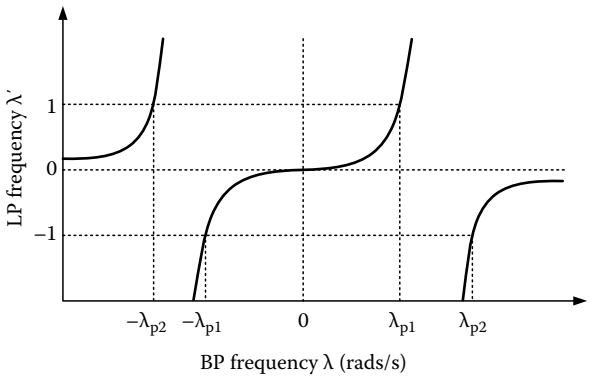


FIGURE 19.23 Relation between λ and λ' for bandstop-to-lowpass conversion.

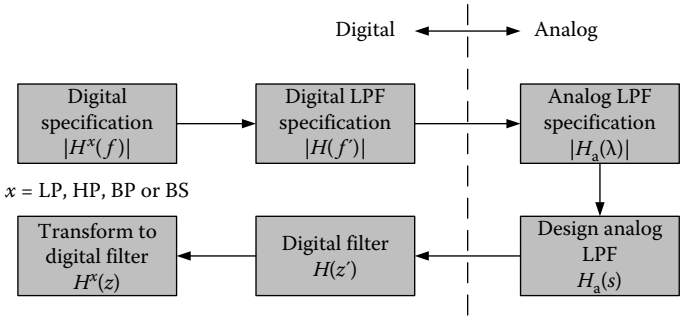


FIGURE 19.24 Design process using digital frequency transformation.

difference between the method in this section and the method described above is that the transformation between the desired type of filter and the LPF is in the digital domain (the z domain) for the current method whereas it is in the analog domain (the s domain) in the previous method. Figure 19.24 shows the design process using digital frequency transformation. The advantage of the current method is that in designing a desired digital HPF, BPF, or BSF, we design a digital LPF, which can make use of the impulse invariant method, in addition to the bilinear transformation. This is not the case for the method discussed previously, due to excessive aliasing.

19.2.4.1 Low-Pass Filters

We can transform a digital LPF to a digital LPF using the transformation

$$z' = \frac{z + \alpha}{1 + \alpha z} \quad |\alpha| < 1 \tag{19.77}$$

With $z = \exp(j2\pi f)$ and $z' = \exp(j2\pi f')$, we can show that the digital frequencies are related by

$$f' = \frac{1}{2\pi} \arctan\left[\frac{(1 - \alpha^2)\sin 2\pi f}{2\alpha + (1 + \alpha^2)\cos 2\pi f}\right] \tag{19.78}$$

The relation given by Equation 19.78 is plotted in Figure 19.25a. If $\alpha = 0$, then $z' = z$ and $f' = f$, which is the trivial case. When $\sigma \neq 0$, there is frequency warping introduced by the transformation. After choosing a, we can transform the desired digital LPF specification to another digital LPF specification, i.e., calculate f'_p and f'_s from f_p and f_s. With f_p, f_s, ε, and δ, a digital LPF can then be designed to satisfy the specification. The resulting system function is then transformed to the desired LPF using Equation 19.77. This method may yield a filter of lower or higher order (due to the frequency warping) compared to the case that there is no digital frequency transformation ($\alpha = 0$).

As an alternative to specifying α, we can specify f'_p, which, together with f_p, specifies the value of α, according to Equation 19.68. With the value of α, we can calculate f'_s from f_s. We can also exchange the role of the passband edge with the stopband edge, i.e., we specify f'_s and compute α from the values of f'_s and f_s. With α, f'_p can be determined from f_p.

19.2.4.2 High-Pass Filters

We can transform a digital LPF to a digital HPF using the transformation

$$z' = -\frac{z + \alpha}{1 + \alpha z} \quad |\alpha| < 1 \tag{19.79}$$

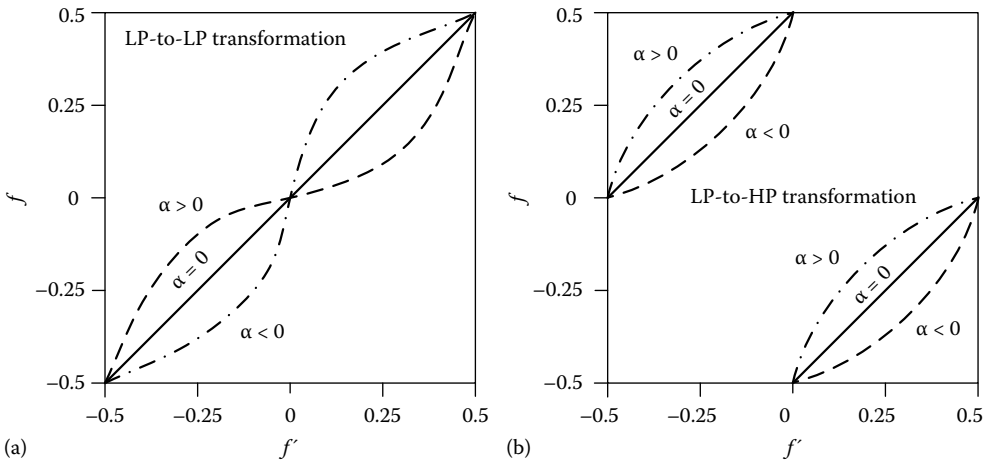


FIGURE 19.25 Frequency relation for digital frequency transformations: (a) lowpass to lowpass and (b) lowpass to highpass.

With $z = \exp(j2\pi f)$ and $z' = \exp(j2\pi f')$, we obtain the relation between the two digital frequencies:

$$f' = \frac{1}{2\pi}\arctan\left[\frac{-(1-\alpha^2)\sin 2\pi f}{-2\alpha - (1+\alpha^2)\cos 2\pi f}\right] \tag{19.80}$$

The relation is plotted in Figure 19.25b. If $\alpha = 0$, then $z' = -z$ and $f' = f + 0.5$. The design process proceeds as follows. After choosing the value of α, the desired digital HPF specification is transformed to the digital LPF specification, using the relation in Equation 19.80. Using the resulting values of f'_p and f'_s, together with the ripple specifications (ε and δ), a digital LPF is designed to satisfy the specification. The resulting LPF system function $H(z')$ is then transformed to the desired HPF by substituting z' with $-(z+\alpha)/(1+\alpha z)$, given by Equation 19.79. Similar to the case of the LPF, we can specify f'_p (instead of α) and calculate the required value of α from f'_p and f_p.

19.2.4.3 Bandpass Filters

To transform a digital LPF to a digital BPF using the transformation

$$z' = -\frac{1 + \frac{2\alpha k}{k+1}z + \frac{k-1}{k+1}z^2}{\frac{k-1}{k+1} + \frac{2\alpha k}{k+1}z + z^2} \quad |\alpha| < 1,\ k > 0 \tag{19.81}$$

This implies the following relation between the two digital frequencies:

$$f' = \frac{1}{2\pi}\arctan\left[\frac{(1-b)\{2a\sin 2\pi f + (1+b)\sin 4\pi f\}}{-a^2 - 2b - 2a(1+b)\cos 2\pi f - (b^2+1)\cos 4\pi f}\right] \tag{19.82}$$

where $a = 2\alpha k/(k+1)$ and $b = (k-1)/(k+1)$. An example of Equation 19.82 is plotted in Figure 19.26. If $\alpha = 0$, the curve would be odd symmetric with respect to $f = 0.25$. The design process is similar to the case of the HPF, except that there are now two passband edges, f'_{pi} and $|f'_{p2}|$, and two stopband

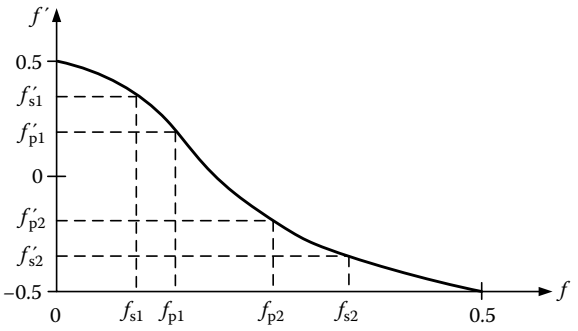


FIGURE 19.26 Frequency relation for BP to LP digital frequency transformation.

edges, f'_{s1} and $|f'_{p2}|$, for the digital LPF to satisfy (see Figure 19.25). To satisfy both sets, we let $f'_p = \max\{f'_{p1}, |f'_{p2}|\}$ and $f'_s = \min\{f_{s1}, |f'_{s2}|\}$ be the passband and stopband edges for the digital LPF filter.

If we specify f'_p, then together with f_{p1} and f_{p2}, they determine the values of α and k:

$$\alpha = \frac{\cos[\pi(f_{p2} + f_{p1})]}{\cos[\pi(f_{p2} - f_{p1})]} \quad k = \cot[\pi(f_{p2} - f_{p1})]\tan(\pi f'_p) \tag{19.83}$$

With the values of α and k, we calculate the values of f'_{s1} and f'_{s2}, from Equation 19.82 and let $f'_s = \min\{f'_{s1}, |f'_{s2}|\}$. Thus we have f'_p, f'_s, ε, and δ as the digital LPF specification. After a digital LPF is designed to satisfy this specification, it is converted to digital BPF by the transformation in Equation 19.81.

19.2.4.4 Bandstop Filters

To transform a digital LPF to a digital BSF, we can use

$$z' = \frac{1 + \frac{2\alpha k}{k+1}z + \frac{k-1}{k+1}z^2}{\frac{k-1}{k+1} + \frac{2\alpha k}{k+1}z + z^2} \quad |\alpha| < 1,\ k > 0 \tag{19.84}$$

The corresponding relation between the two digital frequencies is

$$f' = \frac{1}{2\pi}\arctan\left[-\frac{(1-b)\{2a\sin 2\pi f + (1+b)\sin 4\pi f\}}{a^2 + 2b + 2a(1+b)\cos 2\pi f + (b^2+1)\cos 4\pi f}\right] \tag{19.85}$$

where $a = 2\alpha k/(k+1)$ and $b = (k-1)/(k+1)$. An example is plotted in Figure 19.27. The design process is the same as described in Section 19.2.4.3.

When f'_{pi} is specified, together with f_{p1} and f_{p2}, the values of α and k can be calculated from

$$\alpha = \frac{\cos[\pi(f_{p2} + f_{p1})]}{\cos[\pi(f_{p2} - f_{p1})]} \quad k = \tan[\pi(f_{p2} - f_{p1})]\tan(\pi f'_p) \tag{19.86}$$

With these values, we calculate the values of f'_{s1} and f'_{s2} from Equation 19.85. Letting $f'_s = \min\{f'_{s1}, |f'_{s2}|\}$, we have f'_p, f'_s, ε, and δ, which constitute the digital LPF specification. A digital LPF is then designed and converted to digital BSF by the transformation in Equation 19.84.

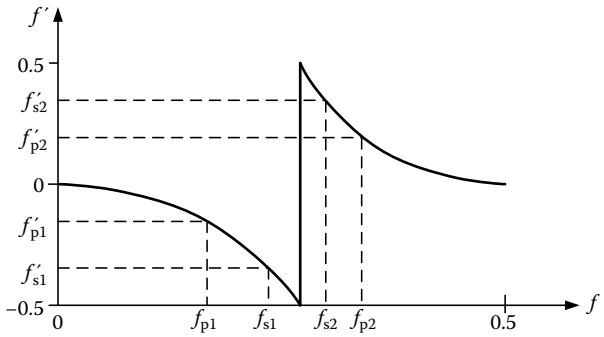


FIGURE 19.27 Frequency relation for BS to LP digital frequency transformation.

19.2.5 Computer-Aided Design

The general idea is to use an algorithm to search for coefficients such that the resulting response (magnitude and/or phase) is "close" to the desired response. The "closeness" is in some well-defined sense. The advantage of such method is that it can be used to design nonstandard filters such as multiband filters, phase equalizers, differentiators, etc. However, it requires a computer program to execute the algorithm. In addition, it usually cannot directly determine the filter order such that the passband and stopband ripples are within the desired ranges. The order is usually determined through several trials.

A large number of procedures have been studied for designing IIR filter in literatures. They can be classified into two categories. One is the indirect approach [2] which can be divided into two steps. First, an FIR filter that meets the required specifications is designed. Then, a lower order IIR filter is obtained, which maintains the original magnitude and phase specifications by applying model reduction techniques. The other is the direct approach in which IIR filters are directly designed to meet the frequency response specification using the least square criterion or Chebyshev (minimax) criterion. Several techniques are employed to achieve this propose. We review some of them in this section.

19.2.5.1 Least Squares ($\mathcal{L}_2$-Norm) Criterion

Consider the transfer function of an IIR filter given as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{19.87}$$

where $X(z)$ and $Y(z)$ are the denominator and numerator polynomials, respectively. We wish to design the frequency response $H(e^{j\omega})$ such that it optimally approximates the desired frequency response $D(e^{j\omega})$. The filter coefficients can be found by minimizing the weighted square error

$$E(\omega) = \int W^2(\omega)|D(e^{j\omega}) - H(e^{j\omega})|^2 \mathrm{d}\omega \tag{19.88}$$

where $W(\omega)$ denotes a nonnegative weighting function, which is zero in the transition band. The weighted least squares (WLS) method has been successfully applied to design an FIR filter [1] where its coefficients are obtained from a well-known least squares solution. However, for IIR filter design, the WLS criterion in Equation 19.88 is no longer a quadratic form of the filter coefficients. Therefore, nonlinear optimization techniques with some stability constraints are required to solve the cost function (Equation 19.88). However, the global minimum of $E(\omega)$ may not be guaranteed. We will leave the stability issue for the last part of this section.

To avoid the complexity of using nonlinear optimization techniques, the iterative procedure replaces the cost function as

$$\begin{aligned} E_k(\omega) &= \int |\varepsilon_k(\omega)|^2 d\omega \\ &= \int \frac{W^2(\omega)}{|X_{k-1}(e^{j\omega})|^2} |D(e^{j\omega})X_k(e^{j\omega}) - Y_k(e^{j\omega})|^2 d\omega \end{aligned} \tag{19.89}$$

where $X_k(e^{j\omega})$ and $Y_k(e^{j\omega})$ are polynomials to be determined in the kth iteration. Several methods have been proposed to solve this problem in the literature and we present a few of them here.

19.2.5.1.1 Adaptive Weighted Least Squares

The adaptive WLS method for IIR filter design can be formulated from the error function $E_k(\omega)$ in Equation 19.89

$$E_k(\omega) = \frac{W(\omega)}{X_{k-1}(e^{j\omega})} [D(e^{j\omega})X_k(e^{j\omega}) - Y_k(e^{j\omega})] \tag{19.90}$$

By defining the following vectors

$$\begin{aligned} \mathbf{c}_0 &= \left[e^{-j\omega} \cdots e^{-j\omega N}\right]^{\mathrm{T}} \\ \mathbf{c}_1 &= \left[1\ e^{-j\omega} \cdots e^{-j\omega M}\right]^{\mathrm{T}} \\ \mathbf{a}_k &= \left[a_1^{(k)} \cdots a_N^{(k)}\right]^{\mathrm{T}} \\ \mathbf{b}_k &= \left[b_0^{(k)} \cdots b_M^{(k)}\right]^{\mathrm{T}} \end{aligned}$$

the error function can be represented in matrix form as

$$E_k(\omega) = \frac{W(\omega)}{1 + \mathbf{c}_0^{\mathrm{T}}(\omega)\mathbf{a}_{k-1}} [D(e^{j\omega}) + D(e^{j\omega})\mathbf{c}_0^{\mathrm{T}}(\omega)\mathbf{a}_k - \mathbf{c}_1^{\mathrm{T}}(\omega)\mathbf{b}_k] \tag{19.91}$$

where $X_k(e^{j\omega}) = 1 + \mathbf{c}_0^{\mathrm{T}}(\omega)\mathbf{a}_k$ and $Y_k(e^{j\omega}) = \mathbf{c}_1^{\mathrm{T}}(\omega)\mathbf{b}_k$. Note that the superscript T denotes the transposition operation. Furthermore, if the above error function is evaluated on a dense frequency grid, the following vector equation can be formed

$$\mathbf{e}_k = \mathbf{W}_k(\mathbf{d} - \mathbf{C}\mathbf{x}_k) \tag{19.92}$$

where

$$\begin{aligned} \mathbf{e}_k &= \left[\varepsilon_k(e^{j\omega_1})\ \varepsilon_k(e^{j\omega_2}) \cdots\right]^{\mathrm{T}} \\ \mathbf{W}_k &= \mathrm{diag}[W_k(\omega_1)W_k(\omega_2) \cdots] \\ \mathbf{d} &= \left[D(e^{j\omega_1})\ D(e^{j\omega_2}) \cdots\right] \\ \mathbf{C} &= \begin{bmatrix} -e^{-j\omega_1}D(e^{j\omega_1}) & \cdots & -e^{-jN\omega_1}D(e^{j\omega_1}) & 1 & e^{-j\omega_1} & \cdots & e^{-jM\omega_1} \\ -e^{-j\omega_2}D(e^{j\omega_2}) & \cdots & -e^{-jN\omega_2}D(e^{j\omega_2}) & 1 & e^{-j\omega_2} & \cdots & e^{-jM\omega_2} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ \mathbf{x}_k &= \begin{bmatrix} \mathbf{a}_k \\ \mathbf{b}_k \end{bmatrix} \end{aligned}$$

Here, the weighting function $\hat{W}_k(\omega)$ is defined as

$$\hat{W}_k(\omega) = \frac{W(\omega)}{|X_{k-1}(e^{j\omega})|} \tag{19.93}$$

The cost function in Equation 19.89 can be approximated as

$$E_k(\omega) = \|\varepsilon(\omega)\|_2^2 \approx \mathbf{e}_k^H \mathbf{e}_k \tag{19.94}$$

where superscript $(\cdot)^H$ denotes the conjugate transpose operation. Minimizing the cost function $E_k(\omega)$ leads to the well-known least squares solution,

$$\mathbf{x}_k = [\mathrm{Re}(\mathbf{C}^H)\mathbf{W}_k\mathrm{Re}(\mathbf{C}) + \mathrm{Im}(\mathbf{C}^H)\mathbf{W}_k\mathrm{Im}(\mathbf{C})]^{-1}[\mathrm{Re}(\mathbf{C}^H)\mathbf{W}_k\mathrm{Re}(\mathbf{d}) + \mathrm{Im}(\mathbf{C}^H)\mathbf{W}_k\mathrm{Im}(\mathbf{d})] \tag{19.95}$$

Once the filter coefficients are obtained, some weight updating procedures [1] are then applied to achieve the equiripple filter response.

19.2.5.1.2 Quadratic Programming

Using the same cost function in Equation 19.89, the problem can be formulated in a standard quadratic programming form as

$$E = \mathbf{x}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{x}_k + \mathbf{x}_k^{\mathrm{T}}\mathbf{p}_k + \text{constant} \tag{19.96}$$

where

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{a}_k \\ \mathbf{b}_k \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{12}^T & \mathbf{H}_{22} \end{bmatrix}, \quad \text{and} \quad \mathbf{p}_k = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

with

$$\mathbf{H}_{11} = \int \hat{W}_k(\omega)|D(e^{j\omega})|^2(\mathbf{c}_0\mathbf{c}_0^H)\mathrm{d}\omega$$

$$\mathbf{H}_{12} = \int \hat{W}_k(\omega)\mathrm{Re}(D(e^{j\omega})(\mathbf{c}_0\mathbf{c}_1^H))\mathrm{d}\omega$$

$$\mathbf{H}_{22} = \int \hat{W}_k(\omega)(\mathbf{c}_1\mathbf{c}_1^H)\mathrm{d}\omega$$

$$\mathbf{p}_1 = \int \hat{W}_k(\omega)|D(e^{j\omega})|^2\ \mathrm{Re}(\mathbf{c}_0)\mathrm{d}\omega$$

$$\mathbf{p}_2 = -\int \hat{W}_k(\omega)\mathrm{Re}(D(e^{j\omega})\mathbf{c}_1)\mathrm{d}\omega$$

$$\hat{W}_k(\omega) = \frac{W^2(\omega)}{|X_{k-1}(e^{j\omega})|^2}$$

The stability and amplitude requirements are taken care of by imposing linear inequality constraints [4]. Filter coefficients can be found by minimizing the cost function (Equation 19.96) using standard quadratic programming tools.

19.2.5.1.3 Eigenfilter

The eigenfilter approach was proposed in Ref. [5] for designing linear-phase FIR filters. The goal is to express the cost function (Equation 19.89) in form of

$$E = \mathbf{h}^{\mathrm{T}}\mathbf{P}\mathbf{h} \tag{19.97}$$

where

h is a vector containing the unknown filter coefficients matrix
P is symmetric, real, and positive-definite

If **h** has unit norm, i.e., $\mathbf{h}^{\mathrm{T}}\mathbf{h} = 1$, to avoid trivial solution, then the optimal **h** which minimizes the cost function E is simply the eigenvector corresponding to the minimum eigenvalue of **P**. The eigenfilter method can also be applied to design IIR filter. The problem may be formulated in two ways.

1. Time-domain approach
   In this method, the objective error function is formulated in time domain as

$$e(n) = \sum_{k=0}^{N} a(k)h_1(n-k) - \sum_{k=0}^{M} b(k)\delta(n-k) \tag{19.98}$$

   where $h_1(n)$ is the impulse response of a target transfer function $H(z) = H_1(z) + H_1(z^{-1})$. The filter $H_1(z)$ is stable and causal so that a noncausal implementation of the system is necessary. With some mathematical manipulation, the error function (Equation 19.98) can be expressed in the standard form of Equation 19.97. Therefore, the IIR filter coefficients $a(n)$ and $b(n)$ can be found. This method can also be extended to design 2-D IIR filters. However, to achieve good approximation in filter design, computation must be performed on matrices of very large size.
2. Frequency-domain approach
   In this approach, the error function in Equation 19.90 is used to formulate the problem in another way as

$$\varepsilon(\omega) = W(\omega)\mathbf{h}^{\mathrm{T}}\mathbf{c} \tag{19.99}$$

   where

$$\begin{aligned}\mathbf{c} &= [D(e^{j\omega})\ D(e^{j\omega})e^{-j\omega}\cdots D(e^{j\omega})e^{-j\omega N} - 1 - e^{-j\omega}\cdots e^{-j\omega M}]^{\mathrm{T}}\\ \mathbf{h} &= [a_0\ a_1\cdots a_N\ b_0\ b_1\cdots b_M]^{\mathrm{T}}\end{aligned} \tag{19.100}$$

   Consequently, the cost function in Equation 19.89 can be expressed as

$$E = \int \mathbf{h}^{\mathrm{T}}\mathbf{c}^{*}\mathbf{c}^{\mathrm{T}}\mathbf{h}W(\omega)\mathrm{d}\omega = \mathbf{h}^{\mathrm{T}}\mathbf{P}\mathbf{h}, \quad \omega \in \Omega \tag{19.101}$$

   where
   superscript * denotes the conjugation operation
   Ω is the frequency region of interest

   the matrix

$$\mathbf{P} = \int \mathbf{c}^{*}\mathbf{c}^{\mathrm{T}}W(\omega)\mathrm{d}\omega \tag{19.102}$$

   is a symmetric, real, positive-definite matrix.

By subjecting $\mathbf{h}$ to the usual unit norm condition $\mathbf{h}^{\mathrm{T}}\mathbf{h} = 1$, the optimum filter coefficients that minimize the cost function are the elements of the eigenvector of the matrix $\mathbf{P}$ corresponding to the minimum eigenvalue. The eigenfilter method can solve constrained filter design problem. However, to obtain equiripple filters, a weight adaptive procedure is needed.

19.2.5.2 Weighted Chebyshev ($\mathcal{L}_\infty$-Norm) Criterion

An IIR filter can also be formulated on weighted Chebyshev criterion. Filter coefficients are chosen such that its weighted Chebyshev (minimax) error between desired and actual frequency response is minimized. The iterative cost function in Equation 19.89 is now evaluated on Chebyshev criterion as

$$\begin{aligned} E &= \max_{\omega\in\Omega} \frac{W(\omega)}{|X_{k-1}(e^{j\omega})|} |D(e^{j\omega})X_k(e^{j\omega}) - Y_k(e^{j\omega})| \\ &= \max_{\omega\in\Omega} W_k(\omega)|D(e^{j\omega}) + \mathbf{x}_k^T\mathbf{s}| \end{aligned} \tag{19.103}$$

where

$$\mathbf{s} = [D(e^{j\omega})\mathbf{c}_0 - \mathbf{c}_1]^{\mathrm{T}}$$

$\mathbf{x}_k$ is the coefficient vector previously defined in Equation 19.92.

The solution of the minimax problem in Equation 19.103 can be found by solving the following equivalent linear programming problem

$$\min_{\mathbf{x}_k} \varepsilon_k \tag{19.104}$$

subject to

$$W_k(\omega)|D(e^{j\omega}) + \mathbf{x}_k^{\mathrm{T}}\mathbf{s}| \le \varepsilon_k \tag{19.105}$$

With a stability constraint, the above linear programming problem can be arranged in standard form of linear programming technique and solved using off-the-shelf linear programming software.

19.2.6 Stability Issues

An IIR filter can be unstable if there are some poles outside the unit circle. However, in case that the phase response is not important in the design, an unstable IIR filter obtained from a design algorithm can be stabilized by conjugate reciprocal substitution of the unstable factor without changing its amplitude. If, however, phase response is a part of design specification, some other techniques must be used. There are four major approaches that can be applied for filter stabilization when optimization techniques previously described are used. The following is a summary. More details and references can be found in Ref. [6].

1. First approach is proposed by Deczky [3]. In this method, a standard gradient-based optimization is modified so that the searching trajectory is only inside the border of stability. However, the algorithm has high computation complexity. A standard optimization tool cannot be used.
2. In the second approach, the target frequency response is chosen such that the desired filter is stable as in Ref. [7]. The target filter is restricted and may be too difficult to obtain filter stabilization.

3. Third approach is to impose constraints on the denominator X(z) of the filter transfer function. A stable IIR filter is achieved by imposing the real part of X(z) to be nonnegative. However, using these methods, some optimal solution that is excluded from the constraints may not be found.
4. Fourth approach is to transform a transfer function such that its optimum solution always lies in the stable area [6]. However, this method leads to a very complicated cost function that requires nonlinear optimization methods.

References

1. Y. C. Lim, J. H. Lee, C. K. Chen, and R. H. Yang, A weighted least squares algorithm for quasiequiripple FIR and IIR digital filter design, *IEEE Trans. Signal Process.*, 40, 551–558, Mar. 1992.
2. B. Beliczynki, J. Kale, and G. D. Cain, Approximation of FIR by IIR digital filter: An algorithm based on balanced model reduction, *IEEE Trans. Signal Process.*, 40, 532–542, Mar. 1992.
3. A. Deczky, Synthesis of recursive filters using the minimum p-error criterion, *IEEE Trans. Audio Electroacoust*, AU-20, 257–263, 1972.
4. W. S. Lu, Design of stable IIR digital filter with equirripple passbands and peak-constrained least-squares stopbands, *IEEE Trans. Circuits Syst. II*, 46, 1421–1426, Nov. 1999.
5. P. P. Vaidyanathan and T. Q. Nguyen, Eigenfilter: A new approach to least-squares FIR filter design and applications including Nyquist filters, *IEEE Trans. Circuits Syst.*, CAS-34, 11–23, Jan. 1987.
6. A. Tarczynski, G. D. Cain, E. Hermanowicz, and M. Rojewski, A WISE method for designing IIR filters, *IEEE Trans. Signal Process.*, 49, 1421–1432, July 2001.
7. T. Kobayashi and S. Imai, Design of IIR digital filters with arbitrary log magnitude function by WLS techniques, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-38, 247–252, Feb. 1990.

19.3 Lattice Digital Filters

Yong Ching Lim

19.3.1 Lattice Filters

There are several families of lattice structures for the implementation of IIR filters. Two of the most commonly encountered families are the tapped numerator structure shown in Figure 19.28 [3] and the injected numerator structure shown in Figure 19.29 [3]. It should be noted that not all the taps and injectors of the filters are nontrivial. For example, if $\lambda_i = 0$ for all i, the structure of Figure 19.28 simplifies to that of Figure 19.30 [1]. If $\phi_i = 0$ for $i > 0$, the structure of Figure 19.29 reduces to that of Figure 19.31. For both families, the denominator of the filter's transfer function is synthesized using a lattice network. The transfer function's numerator of the tapped numerator structure is realized by a

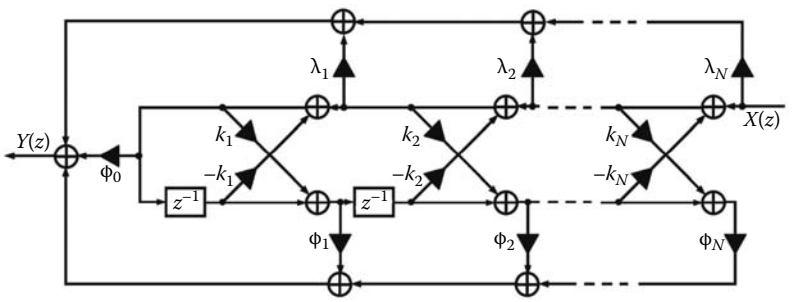


FIGURE 19.28 General structure of a tapped numerator lattice filter.

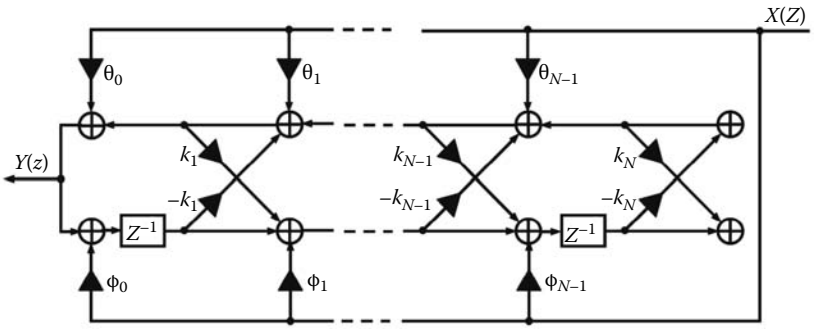


FIGURE 19.29 General structure of an injected numerator lattice filter.

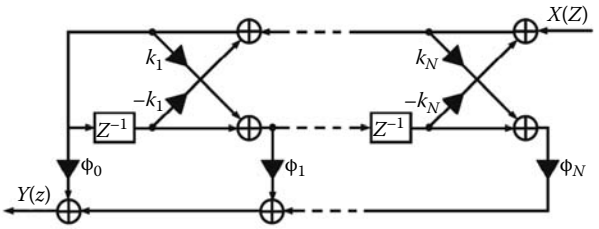


FIGURE 19.30 Structure of a tapped numerator lattice filter with $\lambda_i = 0$ for all i.

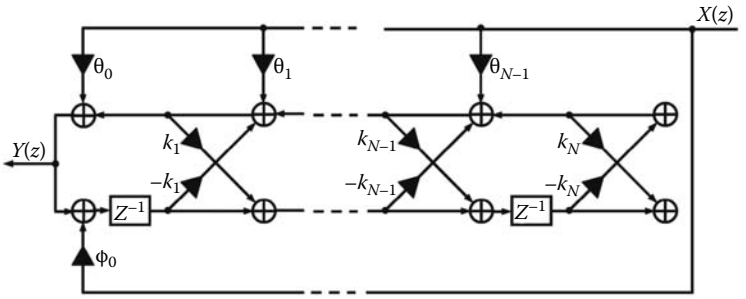


FIGURE 19.31 Structure of an injected lattice filter with $\phi_i = 0$ for $i > 0$.

weighted sum of the signals tapped from $N + 1$ appropriate points of the lattice. For the injected numerator structure, the transfer function's numerator is realized by weighting and injecting the input into $N + 1$ appropriate points on the lattice. The lattice itself may appear in several forms as shown in Figure 19.32 [1]. Figure 19.33 shows the structure of a third-order injected numerator filter synthesized using the one-multiplier lattice of Figure 19.32b.

19.3.2 Evaluation of the Reflection Coefficients k_n [2]

The nth reflection coefficient k_n for both families of filters may be evaluated as follows: Let the transfer function of the filter $H(z)$ be given by

$$H(z) = \frac{B(z)}{A(z)} \tag{19.106}$$

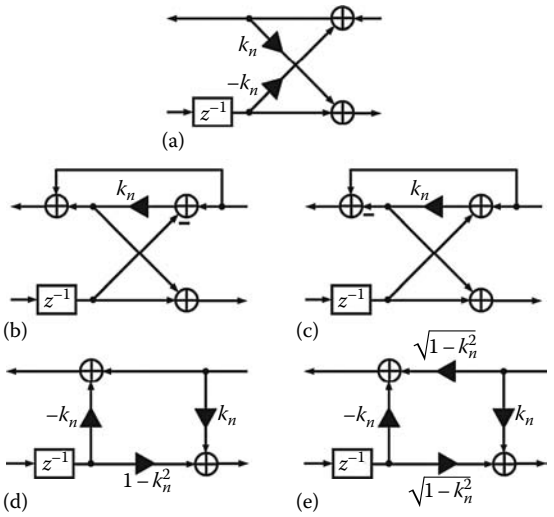


FIGURE 19.32 (a) Two-multiplier lattice; (b) and (c) one-multiplier lattice; (d) three-multiplier lattice; and (e) four-multiplier lattice.

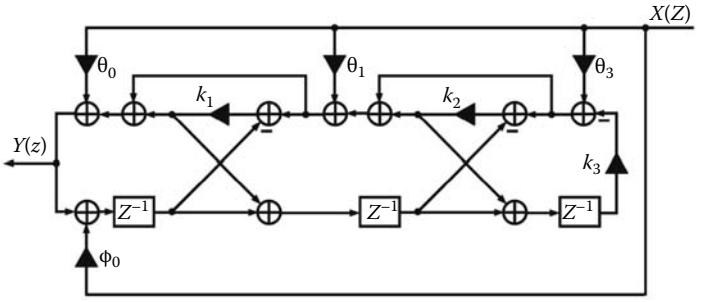


FIGURE 19.33 Third-order one-multiplier injected numerator lattice filter.

where

$$B(z) = \sum_{n=0}^{N} b_n z^{-n} \tag{19.107}$$

$$A(z) = 1 + \sum_{n=1}^{N} a_n z^{-n} \tag{19.108}$$

Define

$$D_N(z) = A(z) \tag{19.109}$$

$$D_{n-1}(z) = \frac{D_n(z) - k_n z^{-n} D_n(z^{-1})}{1 - k_n^2} \tag{19.110}$$

$$= 1 + \sum_{r=1}^{n-1} d_{n-1}(r) z^{-r} \tag{19.111}$$

The algorithm for computing k_n for all n runs as follows:

1. Set $n = N$.
2. Compute $D_n(z)$.
3. $k_n = d_n(n)$.
4. Decrement n.
5. If $n = 0$, stop; otherwise, go to 2.

19.3.3 Evaluation of the Tap Gains φ_n and λ_n [3]

For the tapped numerator filters, φ_n and λ_n may be computed as follows:
Define

$$\Gamma_N(z) = B(z) \tag{19.112}$$

$$\Gamma_{n-1}(z) = \Gamma_n(z) - \lambda_n D_n(z) - \psi_n z^{-n} D_n\left(z^{-1}\right) \tag{19.113}$$

$$= \sum_{r=0}^{n-1} \gamma_{n-1}(r) z^{-r} \tag{19.114}$$

The algorithm for computing φ_n and λ_n for all n runs as follows:

1. Set $n = N$.
2. Compute $\Gamma_n(z)$.
   Set either $\psi_n = 0$, $\lambda_n = 0$, or $\lambda_n = 0$.
   If $\psi_n = 0$, $\lambda_n = \gamma_n(n)/k_n$.
   If $\lambda_n = 0$, $\psi_n = \gamma_n(n)$.
3. Decrement n.
4. If $n = -1$, stop; otherwise, go to 2.

19.3.4 Evaluation of the Injector Gains θ_n and ϕ_n [3]

For the injected numerator filters, θ_n and ϕ_n may be computed as follows:
Define

$$L_0^0(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{19.115}$$

$$L_m^n(z) = \begin{bmatrix} 1 & k_n z^{-1} \\ k_n & z^{-1} \end{bmatrix} L_m^{n-1}(z), \quad n > m \tag{19.116}$$

$$= \begin{bmatrix} P_m^n(z) & Q_m^n(z) \\ R_m^n(z) & S_m^n(z) \end{bmatrix} \tag{19.117}$$

$$P_m^n(z) = 1 + \sum_{r=1}^{n-m-1} p_m^n(r) z^{-r} \tag{19.118}$$

$$Q_m^n(z) = \sum_{r=1}^{n-m} q_m^n(r) z^{-r} \tag{19.119}$$

$$\Xi_{N-1} = B(z) + \phi_0 Q_0^N(z) \tag{19.120}$$

$$\Xi_{n-1}(z) = \Xi_n(z) + \phi_{N-n} Q_{N-n}^N(z) - \theta_{N-n-1} P_{N-n-1}^N(z) \tag{19.121}$$

$$= \sum_{r=0}^{n-1} \xi_{n-1}(r) z^{-r} \tag{19.122}$$

The algorithm for computing θ_n and ϕ_n for all n runs as follows:

1. $\phi_0 = -b_N / q_0^N(N)$.
   Set $n = 0$.
2. Increment n.
   Compute $\Xi_{N-n}(z)$.
   Set either $\phi_n = 0$ or $\theta_{n-1} = 0$.
   If $\phi_n = 0$, $\theta_{n-1} = \xi_{N-n}(N-n) / p_{n-1}^N(N-n)$
   If $\theta_{n-1} = 0$, $\phi_n = -\xi_{N-n}(N-n) / q_n^N(N-n)$
3. If $n = N - 1$ go to 4; otherwise, go to 2.
4. $\theta_{N-1} = \xi_0(0)$. Stop.

References

1. A. H. Gray, Jr. and J. D. Markel, Digital lattice and ladder filter synthesis, *IEEE Trans. Audio Electroacoust.*, AU-21, 491–500, Dec. 1973.
2. A. H. Gray, Jr. and J. D. Markel, A normalized digital filter structure, *IEEE Trans. Acoustics, Speech, Signal Process.*, ASSP-23, 268–277, June 1975.
3. Y. C. Lim, On the synthesis of the IIR digital filters derived from single channel AR lattice network, *IEEE Trans. Acoustics, Speech, Signal Process.*, ASSP-32, 741–749, Aug. 1984.

20

Finite Wordlength Effects

20.1 Number Representation **20**-1
20.2 Fixed-Point Quantization Errors **20**-2
20.3 Floating-Point Quantization Errors **20**-4
20.4 Roundoff Noise **20**-5
Roundoff Noise in FIR Filters • Roundoff Noise in Fixed-Point IIR Filters • Roundoff Noise in Floating-Point IIR Filters
20.5 Limit Cycles **20**-12
20.6 Overflow Oscillations **20**-14
20.7 Coefficient Quantization Error **20**-15
20.8 Realization Considerations **20**-17
References **20**-18

Bruce W. Bomar
University of Tennessee Space Institute

Practical digital filters must be implemented with finite precision numbers and arithmetic. As a result, both the filter coefficients and the filter input and output signals are in discrete form. This leads to four types of finite wordlength effects.

Discretization (quantization) of the filter coefficients has the effect of perturbing the location of the filter poles and zeros. As a result, the actual filter response differs slightly from the ideal response. This *deterministic* frequency response error is referred to as coefficient quantization error.

The use of finite precision arithmetic makes it necessary to quantize filter calculations by rounding or truncation. Roundoff noise is that error in the filter output that results from rounding or truncation calculations within the filter. As the name implies, this error looks like low-level noise at the filter output.

Quantization of the filter calculations also renders the filter slightly nonlinear. For large signals this nonlinearity is negligible and roundoff noise is the major concern. However, for recursive filters with a zero or constant input, this nonlinearity can cause spurious oscillations called limit cycles.

With fixed-point arithmetic it is possible for filter calculations to overflow. The term overflow oscillation, sometimes also called adder overflow limit cycle, refers to a high-level oscillation that can exist in an otherwise stable filter due to the nonlinearity associated with the overflow of internal filter calculations.

In this chapter, we examine each of these finite wordlength effects. Both fixed-point and floating-point number representations are considered.

20.1 Number Representation

In digital signal processing, $(B+1)$-bit fixed-point numbers are usually represented as two's-complement signed fractions in the format

$$b_0 \cdot b_{-1} b_{-2} \cdots b_{-B}$$

The number represented is then

$$X = -b_0 + b_{-1}2^{-1} + b_{-2}2^{-2} + \cdots + b_{-B}2^{-B} \tag{20.1}$$

where b_0 is the sign bit and the number range is $-1 \le X < 1$. The advantage of this representation is that the product of two numbers in the range from -1 to 1 is another number in the same range.

Floating-point numbers are represented as

$$X = (-1)^s m 2^c \tag{20.2}$$

where

s is the sign bit
m is the mantissa
c is the characteristic *or* exponent

To make the representation of a number unique, the mantissa is *normalized* so that $0.5 \le m < 1$.

Although floating-point numbers are always represented in the form of Equation 20.2, the way in which this representation is actually *stored* in a machine may differ. Since $m \ge 0.5$, it is not necessary to store the 2^{-1}-weight bit of m, which is always set. Therefore, in practice numbers are usually stored as

$$X = (-1)^s (0.5 + f) 2^c \tag{20.3}$$

where f is an unsigned fraction, $0 \le f < 0.5$.

Most floating-point processors now use the IEEE Standard 754 32-bit floating-point format for storing numbers. According to this standard, the exponent is stored as an unsigned integer p where

$$p = c + 126 \tag{20.4}$$

Therefore, a number is stored as

$$X = (-1)^s (0.5 + f) 2^{p-126} \tag{20.5}$$

where s is the sign bit, f is a 23-b unsigned fraction in the range $0 \le f < 0.5$, and p is an 8-b unsigned integer in the range $0 \le p \le 255$. The total number of bits is $1 + 23 + 8 = 32$. For example, in IEEE format 3/4 is written $(-1)^0\ (0.5 + 0.25)2^0$ so $s = 0$, $p = 126$, and $f = 0.25$. The value $X = 0$ is a unique case and is represented by all bits zero (i.e., $s = 0$, $f = 0$, and $p = 0$). Although the 2^{-1}-weight mantissa bit is not actually stored, it does exist so the mantissa has 24 b plus a sign bit.

20.2 Fixed-Point Quantization Errors

In fixed-point arithmetic, a multiply doubles the number of significant bits. For example, the product of the two 5-b numbers 0.0011 and 0.1001 is the 10-b number 00.00011011. The extra bit to the left of the decimal point can be discarded without introducing any error. However, the least significant four of the remaining bits must ultimately be discarded by some form of quantization so that the result can be stored to 5 b for use in other calculations. In the example above this results in 0.0010 (quantization by rounding) or 0.0001 (quantization by truncating). When a sum of products calculation is performed,

the quantization can be performed either after each multiply or after all products have been summed with double-length precision.

We will examine three types of fixed-point quantization—rounding, truncation, and magnitude truncation. If X is an exact value then the rounded value will be denoted $Q_r(X)$, the truncated value $Q_t(X)$, and the magnitude truncated value $Q_{mt}(X)$. If the quantized value has B bits to the right of the decimal point, the quantization step size is

$$\Delta = 2^{-B} \tag{20.6}$$

Since rounding selects the quantized value nearest the unquantized value, it gives a value which is never more than $\pm\Delta/2$ away from the exact value. If we denote the rounding error by

$$\varepsilon_r = Q_r(X) - X \tag{20.7}$$

then

$$-\frac{\Delta}{2} \le \varepsilon_r \le \frac{\Delta}{2} \tag{20.8}$$

Truncation simply discards the low-order bits giving a quantized value that is always less than or equal to the exact value so

$$-\Delta < \varepsilon_t \le 0 \tag{20.9}$$

Magnitude truncation chooses the nearest quantized value that has a magnitude less than or equal to the exact value so

$$-\Delta < \varepsilon_{mt} < \Delta \tag{20.10}$$

The error resulting from quantization can be modeled as a random variable uniformly distributed over the appropriate error range. Therefore, calculations with roundoff error can be considered error-free calculations that have been corrupted by additive white noise. The mean of this noise for rounding is

$$m_{\varepsilon_r} = E\{\varepsilon_r\} = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} \varepsilon_r \, d\varepsilon_r = 0 \tag{20.11}$$

where $E\{\ \}$ represents the operation of taking the expected value of a random variable. Similarly, the variance of the noise for rounding is

$$\sigma_{\varepsilon_r}^2 = E\{(\varepsilon_r - m_{\varepsilon_r})^2\} = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} (\varepsilon_r - m_{\varepsilon_r})^2 \, d\varepsilon_r = \frac{\Delta^2}{12} \tag{20.12}$$

Likewise, for truncation,

$$\begin{aligned} m_{\varepsilon_t} &= E\{\varepsilon_t\} = -\frac{\Delta}{2} \\ \sigma_{\varepsilon_t}^2 &= E\{(\varepsilon_t - m_{\varepsilon_t})^2\} = \frac{\Delta^2}{12} \end{aligned} \tag{20.13}$$

and, for magnitude truncation

$$\begin{aligned} m_{\varepsilon_{\text{mt}}} &= E\{\varepsilon_{\text{mt}}\} = 0 \\ \sigma^2_{\varepsilon_{\text{mt}}} &= E\{(\varepsilon_{\text{mt}} - m_{\varepsilon_{\text{mt}}})^2\} = \frac{\Delta^2}{3} \end{aligned} \tag{20.14}$$

20.3 Floating-Point Quantization Errors

With floating-point arithmetic it is necessary to quantize after both multiplications and additions. The addition quantization arises because, prior to addition, the mantissa of the smaller number in the sum is shifted right until the exponent of both numbers is the same. In general, this gives a sum mantissa that is too long and so must be quantized.

We will assume that quantization in floating-point arithmetic is performed by rounding. Because of the exponent in floating-point arithmetic, it is the relative error that is important. The relative error is defined as

$$\varepsilon_{\text{r}} = \frac{Q_{\text{r}}(X) - X}{X} = \frac{\varepsilon_{\text{r}}}{X} \tag{20.15}$$

Since $X = (-1)^s m 2^c$, $Q_{\text{r}}(X) = (-1)^s\, Q_{\text{r}}(m) 2^c$ and

$$\varepsilon_{\text{r}} = \frac{Q_{\text{r}}(m) - m}{m} = \frac{\varepsilon}{m} \tag{20.16}$$

If the quantized mantissa has B bits to the right of the decimal point, $|\varepsilon_{\text{r}}| < \Delta/2$ where, as before, $\Delta = 2^{-B}$. Therefore, since $0.5 \le m < 1$,

$$|\varepsilon_{\text{r}}| < \Delta \tag{20.17}$$

If we assume that ε is uniformly distributed over the range from $-\Delta/2$ to $\Delta/2$ and m is uniformly distributed over 0.5 to 1,

$$\begin{aligned} m_{\varepsilon_{\text{r}}} &= E\left\{\frac{\varepsilon}{m}\right\} = 0 \\ \sigma^2_{\varepsilon_{\text{r}}} &= E\left\{\left(\frac{\varepsilon}{m}\right)^2\right\} = \frac{2}{\Delta}\int_{1/2}^{1}\int_{-\Delta/2}^{\Delta/2} \frac{\varepsilon^2}{m^2}\, d\varepsilon\, dm \\ &= \frac{\Delta^2}{6} = (0.167)2^{-2B} \end{aligned} \tag{20.18}$$

In practice, the distribution of m is not exactly uniform. Actual measurements of roundoff noise in Ref. [1] suggested that

$$\sigma^2_{\varepsilon_{\text{r}}} \approx 0.23\Delta^2 \tag{20.19}$$

while a detailed theoretical and experimental analysis in Ref. [2] determined

$$\sigma^2_{\varepsilon_{\text{r}}} \approx 0.18\Delta^2 \tag{20.20}$$

From Equation 20.15 we can represent a quantized floating-point value in terms of the unquantized value and the random variable ε_r using

$$Q_r(X) = X(1 + \varepsilon_r) \tag{20.21}$$

Therefore, the finite-precision product X_1X_2 and the sum $X_1 + X_2$ can be written

$$fl(X_1X_2) = X_1X_2(1 + \varepsilon_r) \tag{20.22}$$

and

$$fl(X_1 + X_2) = (X_1 + X_2)(1 + \varepsilon_r) \tag{20.23}$$

where ε_r is zero-mean with the variance of Equation 20.20.

20.4 Roundoff Noise

To determine the roundoff noise at the output of a digital filter we will assume that the noise due to a quantization is stationary, white, and uncorrelated with the filter input, output, and internal variables. This assumption is good if the filter input changes from sample to sample in a sufficiently complex manner. It is not valid for zero or constant inputs for which the effects of rounding are analyzed from a limit cycle perspective.

To satisfy the assumption of a sufficiently complex input, roundoff noise in digital filters is often calculated for the case of a zero-mean white noise filter input signal $x(n)$ of variance σ_x^2. This simplifies calculation of the output roundoff noise because expected values of the form $E\{x(n)x(n-k)\}$ are zero for $k \neq 0$ and give σ_x^2 when $k = 0$. This approach to analysis has been found to give estimates of the output roundoff noise that are close to the noise actually observed for other input signals.

Another assumption that will be made in calculating roundoff noise is that the product of two quantization errors is zero. To justify this assumption, consider the case of a 16-b fixed-point processor. In this case a quantization error is of the order 2^{-15}, while the product of two quantization errors is of the order 2^{-30}, which is negligible by comparison.

If a linear system with impulse response $g(n)$ is excited by white noise with mean m_x and variance σ_x^2, the output is noise of mean [3, pp. 788–790]

$$m_y = m_x \sum_{n=-\infty}^{\infty} g(n) \tag{20.24}$$

and variance

$$\sigma_y^2 = \sigma_x^2 \sum_{n=-\infty}^{\infty} g^2(n) \tag{20.25}$$

Therefore, if $g(n)$ is the impulse response from the point where a roundoff takes place to the filter output, the contribution of that roundoff to the variance (mean-square value) of the output roundoff noise is given by Equation 20.25 with σ_x^2 replaced with the variance of the roundoff. If there is more than one source of roundoff error in the filter, it is assumed that the errors are uncorrelated so the output noise variance is simply the sum of the contributions from each source.

20.4.1 Roundoff Noise in FIR Filters

The simplest case to analyze is a finite impulse response (FIR) filter realized via the convolution summation

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{20.26}$$

When fixed-point arithmetic is used and quantization is performed after each multiply, the result of the N multiplies is N-times the quantization noise of a single multiply. For example, rounding after each multiply gives, from Equations 20.6 and 20.12, an output noise variance of

$$\sigma_o^2 = N\frac{2^{-2B}}{12} \tag{20.27}$$

Virtually all digital signal processor integrated circuits contain one or more double-length accumulator registers which permit the sum-of-products in Equation 20.26 to be accumulated without quantization. In this case only a single quantization is necessary following the summation and

$$\sigma_o^2 = \frac{2^{-2B}}{12} \tag{20.28}$$

For the floating-point roundoff noise case we will consider Equation 20.26 for $N = 4$ and then generalize the result to other values of N. The finite-precision output can be written as the exact output plus an error term $e(n)$. Thus,

$$\begin{aligned} y(n) + e(n) = (\{[h(0)x(n)[1+\varepsilon_1(n)]] \\ &+ h(1)x(n-1)[1+\varepsilon_2(n)]\,[1+\varepsilon_3(n)] \\ &+ h(2)x(n-2)[1+\varepsilon_4(n)]\}\{1+\varepsilon_5(n)\} \\ &+ h(3)x(n-3)[1+\varepsilon_6(n)])[1+\varepsilon_7(n)] \end{aligned} \tag{20.29}$$

In Equation 20.29, $\varepsilon_1(n)$ represents the error in the first product, $\varepsilon_2(n)$ the error in the second product, $\varepsilon_3(n)$ the error in the first addition, etc. Notice that it has been assumed that the products are summed in the order implied by the summation of Equation 20.26.

Expanding Equation 20.29, ignoring products of error terms, and recognizing $y(n)$ gives

$$\begin{aligned} e(n) = h(0)x(n)[\varepsilon_1(n) + \varepsilon_3(n) + \varepsilon_5(n) + \varepsilon_7(n)] \\ + h(1)x(n-1)[\varepsilon_2(n) + \varepsilon_3(n) + \varepsilon_5(n) + \varepsilon_7(n)] \\ + h(2)x(n-2)[\varepsilon_4(n) + \varepsilon_5(n) + \varepsilon_7(n)] \\ + h(3)x(n-3)[\varepsilon_6(n) + \varepsilon_7(n)] \end{aligned} \tag{20.30}$$

Assuming that the input is white noise of variance σ_x^2 so that $E\{x(n)x(n-k)\}$ is zero for $k \neq 0$, and assuming that the errors are uncorrelated,

$$E\{e^2(n)\} = [4h^2(0) + 4h^2(1) + 3h^2(2) + 2h^2(3)]\sigma_x^2\sigma_{\varepsilon_r}^2 \tag{20.31}$$

In general, for any N,

$$\sigma_o^2 = E\{e^2(n)\} = \left[Nh^2(0) + \sum_{k=1}^{N-1}(N+1-k)h^2(k)\right]\sigma_x^2\sigma_{\varepsilon_r}^2 \tag{20.32}$$

Notice that if the order of summation of the product terms in the convolution summation is changed, then the order in which the $h(k)$'s appear in Equation 20.32 changes. If the order is changed so that the $h(k)$ with smallest magnitude is first, followed by the next smallest, etc., then the roundoff noise variance is minimized. However, performing the convolution summation in nonsequential order greatly complicates data indexing and so may not be worth the reduction obtained in roundoff noise.

20.4.2 Roundoff Noise in Fixed-Point IIR Filters

To determine the roundoff noise of a fixed-point infinite impulse response (IIR) filter realization, consider a causal first-order filter with impulse response

$$h(n) = a^n u(n) \tag{20.33}$$

realized by the difference equation

$$y(n) = ay(n-1) + x(n) \tag{20.34}$$

Due to roundoff error, the output actually obtained is

$$\hat{y}(n) = Q\{ay(n-1) + x(n)\} = ay(n-1) + x(n) + e(n) \tag{20.35}$$

where $e(n)$ is a random roundoff noise sequence. Since $e(n)$ is injected at the same point as the input, it propagates through a system with impulse response $h(n)$. Therefore, for fixed-point arithmetic with rounding, the output roundoff noise variance from Equations 20.6, 20.12, 20.25, and 20.33 is

$$\sigma_o^2 = \frac{\Delta^2}{12}\sum_{n=-\infty}^{\infty} h^2(n) = \frac{\Delta^2}{12}\sum_{n=0}^{\infty} a^{2n} = \frac{2^{-2B}}{12}\frac{1}{1-a^2} \tag{20.36}$$

With fixed-point arithmetic there is the possibility of overflow following addition. To avoid overflow it is necessary to restrict the input signal amplitude. This can be accomplished by either placing a *scaling* multiplier at the filter input or by simply limiting the maximum input signal amplitude. Consider the case of the first-order filter of Equation 20.34. The transfer function of this filter is

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{1}{e^{j\omega} - a} \tag{20.37}$$

so

$$\left|H(e^{j\omega})\right|^2 = \frac{1}{1 + a^2 - 2a\cos(\omega)} \tag{20.38}$$

and

$$\left|H(e^{j\omega})\right|_{\max} = \frac{1}{1-|a|} \tag{20.39}$$

The peak gain of the filter is $1/(1-|a|)$ so limiting input signal amplitudes to $|x(n)| \leq 1-|a|$ will make overflows unlikely.

An expression for the output roundoff noise-to-signal ratio can easily be obtained for the case where the filter input is white noise, uniformly distributed over the interval from $-(1-|a|)$ to $(1-|a|)$ [4,5]. In this case

$$\sigma_x^2 = \frac{1}{2(1-|a|)} \int_{-(1-|a|)}^{1-|a|} x^2 \mathrm{d}x = \frac{1}{3}(1-|a|)^2 \tag{20.40}$$

so, from Equation 20.25,

$$\sigma_y^2 = \frac{1}{3}\frac{(1-|a|)^2}{1-a^2} \tag{20.41}$$

Combining Equations 20.36 and 20.41 then gives

$$\frac{\sigma_o^2}{\sigma_y^2} = \left(\frac{2^{-2B}}{12}\frac{1}{1-a^2}\right)\left(3\frac{1-a^2}{(1-|a|)^2}\right) = \frac{2^{-2B}}{12}\frac{3}{(1-|a|)^2} \tag{20.42}$$

Notice that the noise-to-signal ratio increases without bound as $|a| \to 1$.

Similar results can be obtained for the case of the causal second-order filter realized by the difference equation

$$y(n) = 2r\cos(\theta)y(n-1) - r^2 y(n-2) + x(n) \tag{20.43}$$

This filter has complex–conjugate poles at $re^{\pm j\theta}$ and impulse response

$$h(n) = \frac{1}{\sin(\theta)} r^n \sin[(n+1)\theta]\ u(n) \tag{20.44}$$

Due to roundoff error, the output actually obtained is

$$\hat{y}(n) = 2r\cos(\theta)y(n-1) - r^2 y(n-2) + x(n) + e(n) \tag{20.45}$$

There are two noise sources contributing to $e(n)$ if quantization is performed after each multiply, and there is one noise source if quantization is performed after summation. Since

$$\sum_{n=-\infty}^{\infty} h^2(n) = \frac{1+r^2}{1-r^2}\frac{1}{(1+r^2)^2 - 4r^2\cos^2(\theta)} \tag{20.46}$$

the output roundoff noise is

$$\sigma_o^2 = \nu\frac{2^{-2B}}{12}\frac{1+r^2}{1-r^2}\frac{1}{(1+r^2)^2 - 4r^2\cos^2(\theta)} \tag{20.47}$$

where $\nu = 1$ for quantization after summation, and $\nu = 2$ for quantization after each multiply.

To obtain an output noise-to-signal ratio we note that

$$H(e^{j\omega}) = \frac{1}{1 - 2r\cos(\theta)e^{-j\omega} + r^2 e^{-j2\omega}} \tag{20.48}$$

and, using the approach of [6],

$$\left|H(e^{j\omega})\right|^2_{\max} = \frac{1}{4r^2\left\{\left[\text{sat}\left(\frac{1+r^2}{2r}\cos(\theta)\right) - \frac{1+r^2}{2r}\cos(\theta)\right]^2 + \left[\frac{1-r^2}{2r}\sin(\theta)\right]^2\right\}} \tag{20.49}$$

where

$$\text{sat}(\mu) = \begin{cases} 1 & \mu > 1 \\ \mu & -1 \le \mu \le 1 \\ -1 & \mu < -1 \end{cases} \tag{20.50}$$

Following the same approach as for the first-order case then gives

$$\frac{\sigma_o^2}{\sigma_y^2} = \nu\frac{2^{-2B}}{12}\frac{1+r^2}{1-r^2}\frac{3}{(1+r^2)^2 - 4r^2\cos^2(\theta)} \times \frac{1}{4r^2\left\{\left[\text{sat}\left(\frac{1+r^2}{2r}\cos(\theta)\right) - \frac{1+r^2}{2r}\cos(\theta)\right]^2 + \left[\frac{1-r^2}{2r}\sin(\theta)\right]^2\right\}} \tag{20.51}$$

Figure 20.1 is a contour plot showing the noise-to-signal ratio of Equation 20.51 for $\nu = 1$ in units of the noise variance of a single quantization $2^{-2B}/12$. The plot is symmetrical about $\theta = 90°$, so only the range from 0° to 90° is shown. Notice that as $r \to 1$, the roundoff noise increases without bound. Also notice that the noise increases as $\theta \to 0°$.

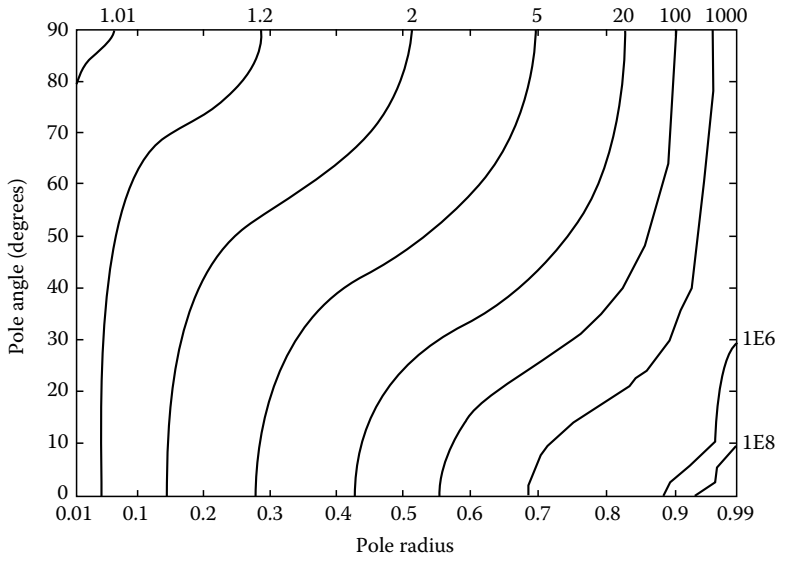


FIGURE 20.1 Normalized fixed-point roundoff noise variance.

It is possible to design state-space filter realizations that minimize fixed-point roundoff noise [7–10]. Depending on the transfer function being realized, these structures may provide a roundoff noise level that is orders-of-magnitude lower than for a nonoptimal realization. The price paid for this reduction in roundoff noise is an increase in the number of computations required to implement the filter. For an Nth-order filter the increase is from roughly $2N$ multiplies for a direct form realization to roughly $(N+1)^2$ for an optimal realization. However, if the filter is realized by the parallel or cascade connection of first- and second-order optimal subfilters, the increase is only to about $4N$ multiplies. Furthermore, near-optimal realizations exist that increase the number of multiplies to only about $3N$ [10].

20.4.3 Roundoff Noise in Floating-Point IIR Filters

For floating-point arithmetic it is first necessary to determine the injected noise variance of each quantization. For the first-order filter this is done by writing the computed output as

$$y(n) + e(n) = [ay(n-1)(1+\varepsilon_1(n)) + x(n)](1+\varepsilon_2(n)) \tag{20.52}$$

where

$\varepsilon_1(n)$ represents the error due to the multiplication
$\varepsilon_2(n)$ represents the error due to the addition

Neglecting the product of errors, Equation 20.52 becomes

$$\begin{aligned} y(n) + e(n) \approx\ & ay(n-1) + x(n) + ay(n-1)\varepsilon_1(n) \\ & + ay(n-1)\varepsilon_2(n) + x(n)\varepsilon_2(n) \end{aligned} \tag{20.53}$$

Comparing Equations 20.34 and 20.53, it is clear that

$$e(n) = ay(n-1)\varepsilon_1(n) + ay(n-1)\varepsilon_2(n) + x(n)\varepsilon_2(n) \tag{20.54}$$

Taking the expected value of $e^2(n)$ to obtain the injected noise variance then gives

$$\begin{aligned} E\{e^2(n)\} =\ & a^2E\{y^2(n-1)\}E\left\{\varepsilon_1^2(n)\right\} + a^2E\{y^2(n-1)\}E\left\{\varepsilon_2^2(n)\right\} \\ & + E\{x^2(n)\}E\left\{\varepsilon_2^2(n)\right\} + E\{x(n)y(n-1)\}E\left\{\varepsilon_2^2(n)\right\} \end{aligned} \tag{20.55}$$

To carry this further it is necessary to know something about the input. If we assume the input is zero-mean white noise with variance σ_x^2, then $E\{x^2(n)\} = \sigma_x^2$ and the input is uncorrelated with past values of the output so $E\{x(n)y(n-1)\} = 0$ giving

$$E\{e^2(n)\} = 2a^2\sigma_y^2\sigma_{\varepsilon_r}^2 + \sigma_x^2\sigma_{\varepsilon_r}^2 \tag{20.56}$$

and

$$\begin{aligned} \sigma_o^2 &= \left(2a^2\sigma_y^2\sigma_{\varepsilon_r}^2 + \sigma_x^2\sigma_{\varepsilon_r}^2\right)\sum_{n=-\infty}^{\infty} h^2(n) \\ &= \frac{2a^2\sigma_y^2 + \sigma_x^2}{1-a^2}\sigma_{\varepsilon_r}^2 \end{aligned} \tag{20.57}$$

However,

$$\sigma_y^2 = \sigma_x^2 \sum_{n=-\infty}^{\infty} h^2(n) = \frac{\sigma_x^2}{1-a^2} \tag{20.58}$$

$$\sigma_o^2 = \frac{1+a^2}{(1-a^2)^2}\sigma_{\varepsilon_r}^2\sigma_x^2 = \frac{1+a^2}{1-a^2}\sigma_{\varepsilon_r}^2\sigma_y^2 \tag{20.59}$$

and the output roundoff noise-to-signal ratio is

$$\frac{\sigma_o^2}{\sigma_y^2} = \frac{1+a^2}{1-a^2}\sigma_{\varepsilon_r}^2 \tag{20.60}$$

Similar results can be obtained for the second-order filter of Equation 20.43 by writing

$$\begin{aligned} y(n) + e(n) = \big([2r\cos(\theta)y(n-1)(1+\varepsilon_1(n)) - r^2y(n-2)(1+\varepsilon_2(n))] \\ \times [1+\varepsilon_3(n)] + x(n))(1+\varepsilon_4(n)) \end{aligned} \tag{20.61}$$

Expanding with the same assumptions as before gives

$$\begin{aligned} e(n) \approx\ & 2r\cos(\theta)y(n-1)[\varepsilon_1(n)+\varepsilon_3(n)+\varepsilon_4(n)] \\ & - r^2y(n-2)[\varepsilon_2(n)+\varepsilon_3(n)+\varepsilon_4(n)] + x(n)\varepsilon_4(n) \end{aligned} \tag{20.62}$$

and

$$\begin{aligned} E\{e^2(n)\} = {} & 4r^2\cos^2(\theta)\sigma_y^2\sigma_{\varepsilon_r}^2 + r^2\sigma_y^2 3\sigma_{\varepsilon_r}^2 \\ & + \sigma_x^2\sigma_{\varepsilon_r}^2 - 8r^3\cos(\theta)\sigma_{\varepsilon_r}^2 E\{y(n-1)y(n-2)\} \end{aligned} \tag{20.63}$$

However,

$$\begin{aligned} & E\{y(n-1)y(n-2)\} \\ & \quad = E\{2r\cos(\theta)y(n-2) - r^2y(n-3) + x(n-1)]y(n-2)\} \\ & \quad = 2r\cos(\theta)E\{y^2(n-2)\} - r^2E\{y(n-2)y(n-3)\} \\ & \quad = 2r\cos(\theta)E\{y^2(n-2)\} - r^2E\{y(n-1)y(n-2)\} \\ & \quad = \frac{2r\cos(\theta)}{1+r^2}\sigma_y^2 \end{aligned} \tag{20.64}$$

so

$$E\{e^2(n)\} = \sigma_{\varepsilon_r}^2\sigma_x^2 + \left[3r^4 + 12r^2\cos^2(\theta) - \frac{16r^4\cos^2(\theta)}{1+r^2}\right]\sigma_{\varepsilon_r}^2\sigma_y^2 \tag{20.65}$$

and

$$\begin{aligned} \sigma_o^2 &= E(n)\sum_{n=-\infty}^{\infty} h^2(n) \\ &= \xi\left[\sigma_{\varepsilon_r}^2\sigma_x^2 + \left[3r^4 + 12r^2\cos^2(\theta) - \frac{16r^4\cos^2(\theta)}{1+r^2}\right]\sigma_{\varepsilon_r}^2\sigma_y^2\right] \end{aligned} \tag{20.66}$$

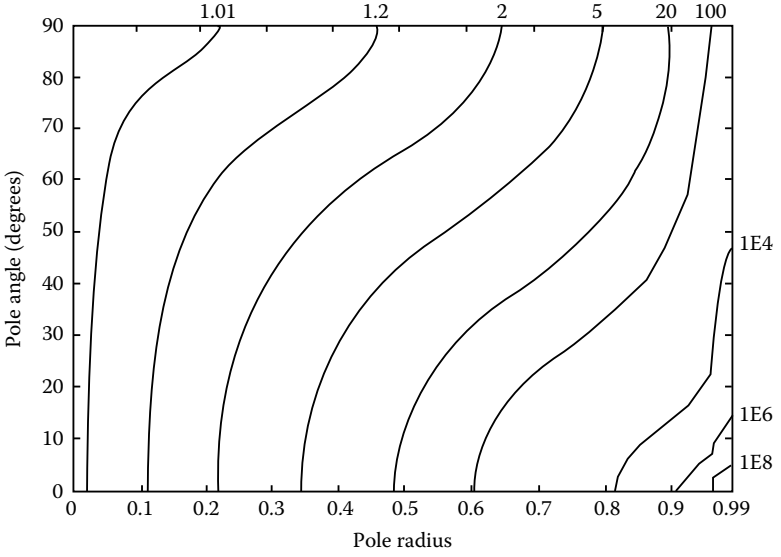


FIGURE 20.2 Normalized floating-point roundoff noise variance.

where from Equation 20.46,

$$\xi = \sum_{n=-\infty}^{\infty} h^2(n) = \frac{1+r^2}{1-r^2}\,\frac{1}{(1+r^2)^2 - 4r^2\cos^2(\theta)} \tag{20.67}$$

Since $\sigma_y^2 = \xi\sigma_x^2$, the output roundoff noise-to-signal ratio is then

$$\frac{\sigma_o^2}{\sigma_y^2} = \xi\left[1 + \xi\left[3r^4 + 12r^2\cos^2(\theta) - \frac{16r^4\cos^2(\theta)}{1+r^2}\right]\right]\sigma_{\varepsilon_r}^2 \tag{20.68}$$

Figure 20.2 is a contour plot showing the noise-to-signal ratio of Equation 20.68 in units of the noise variance of a single quantization $\sigma_{\varepsilon_r}^2$. The plot is symmetrical about $\theta = 90°$, so only the range from 0° to 90° is shown. Notice the similarity of this plot to that of Figure 20.1 for the fixed-point case. It has been observed that filter structures generally have very similar fixed-point and floating-point roundoff characteristics [2]. Therefore, the techniques of [7–10], which were developed for the fixed-point case, can also be used to design low-noise floating-point filter realizations. Furthermore, since it is not necessary to scale the floating-point realization, the low-noise realizations need not require significantly more computation than the direct form realization.

20.5 Limit Cycles

A limit cycle, sometimes referred to as a multiplier roundoff limit cycle, is a low-level oscillation that can exist in an otherwise stable filter as a result of the nonlinearity associated with rounding (or truncating) internal filter calculations [11]. Limit cycles require recursion to exist and do not occur in nonrecursive FIR filters.

As an example of a limit cycle, consider the second-order filter realized by

$$y(n) = Q_r\left\{\frac{7}{8}y(n-1) - \frac{5}{8}y(n-2) + x(n)\right\} \tag{20.69}$$

where $Q_r\{\ \}$ represents quantization by rounding. This is a stable filter with poles at $0.4375 \pm j0.6585$. Consider the implementation of this filter with 4-b (3 b and a sign bit) two's complement fixed-point arithmetic, zero initial conditions ($y(-1) = y(-2) = 0$), and an input sequence $x(n) = \frac{3}{8}\delta(n)$, where $\delta(n)$ is the unit impulse or unit sample. The following sequence is obtained:

$$
\begin{aligned}
y(0) &= Q_r\left\{\frac{3}{8}\right\} = \frac{3}{8} \\
y(1) &= Q_r\left\{\frac{21}{64}\right\} = \frac{3}{8} \\
y(2) &= Q_r\left\{\frac{3}{32}\right\} = \frac{1}{8} \\
y(3) &= Q_r\left\{-\frac{1}{8}\right\} = -\frac{1}{8} \\
y(4) &= Q_r\left\{-\frac{3}{16}\right\} = -\frac{1}{8} \\
y(5) &= Q_r\left\{-\frac{1}{32}\right\} = 0 \\
y(6) &= Q_r\left\{\frac{5}{64}\right\} = \frac{1}{8} \\
y(7) &= Q_r\left\{\frac{7}{64}\right\} = \frac{1}{8} \\
y(8) &= Q_r\left\{\frac{1}{32}\right\} = 0 \\
y(9) &= Q_r\left\{-\frac{5}{64}\right\} = -\frac{1}{8} \\
y(10) &= Q_r\left\{-\frac{7}{64}\right\} = -\frac{1}{8} \\
y(10) &= Q_r\left\{-\frac{1}{32}\right\} = 0 \\
y(12) &= Q_r\left\{\frac{5}{64}\right\} = \frac{1}{8} \\
&\vdots
\end{aligned}
\tag{20.70}
$$

Notice that while the input is zero except for the first sample, the output oscillates with amplitude 1/8 and period 6.

Limit cycles are primarily of concern in fixed-point recursive filters. As long as floating-point filters are realized as the parallel or cascade connection of first- and second-order subfilters, limit cycles will generally not be a problem since limit cycles are practically not observable in first- and second-order systems implemented with 32-b floating-point arithmetic [12]. It has been shown that such systems must have an extremely small margin of stability for limit cycles to exist at anything other than underflow levels, which are at an amplitude of less than 10^{-38} [12].

There are at least three ways of dealing with limit cycles when fixed-point arithmetic is used. One is to determine a bound on the maximum limit cycle amplitude, expressed as an integral number of quantization steps [13]. It is then possible to choose a word length that makes the limit cycle amplitude acceptably low. Alternately, limit cycles can be prevented by randomly rounding calculations up or down [14]. However, this approach is complicated to implement. The third approach is to properly choose the

filter realization structure and then quantize the filter calculations using magnitude truncation [15,16]. This approach has the disadvantage of producing more roundoff noise than truncation or rounding [see Equations 20.12 through 20.14].

20.6 Overflow Oscillations

With fixed-point arithmetic it is possible for filter calculations to overflow. This happens when two numbers of the same sign add to give a value having magnitude greater than one. Since numbers with magnitude greater than one are not representable, the result overflows. For example, the two's complement numbers 0.101 (5/8) and 0.100 (4/8) add to give 1.001 which is the two's complement representation of $-7/8$.

The overflow characteristic of two's complement arithmetic can be represented as $R\{\}$ where

$$R\{X\} = \begin{cases} X - 2 & X \geq 1 \\ X & -1 \leq X < -1 \\ X + 2 & X < -1 \end{cases} \tag{20.71}$$

For the example just considered, $R\{9/8\} = -7/8$.

An overflow oscillation, sometimes also referred to as an *adder overflow limit cycle,* is a high-level oscillation that can exist in an otherwise stable fixed-point filter due to the gross nonlinearity associated with the overflow of internal filter calculations [17]. Like limit cycles, overflow oscillations require recursion to exist and do not occur in nonrecursive FIR filters. Overflow oscillations also do not occur with floating-point arithmetic due to the virtual impossibility of overflow.

As an example of an overflow oscillation, once again consider the filter of Equation 20.69 with 4-b fixed-point two's complement arithmetic and with the two's complement overflow characteristic of Equation 20.71:

$$y(n) = Q_r\left\{R\left[\frac{7}{8}y(n-1) - \frac{5}{8}y(n-2) + x(n)\right]\right\} \tag{20.72}$$

In this case we apply the input

$$\begin{aligned} x(n) &= -\frac{3}{4}\delta(n) - \frac{5}{8}\delta(n-1) \\ &= \left\{-\frac{3}{4}, -\frac{5}{8}, 0, 0, \ldots\right\} \end{aligned} \tag{20.73}$$

giving the output sequence

$$\begin{aligned} y(0) &= Q_r\left\{R\left[-\frac{3}{4}\right]\right\} = Q_r\left\{-\frac{3}{4}\right\} = -\frac{3}{4} \\ y(1) &= Q_r\left\{R\left[-\frac{41}{32}\right]\right\} = Q_r\left\{\frac{23}{32}\right\} = \frac{3}{4} \\ y(2) &= Q_r\left\{R\left[\frac{9}{8}\right]\right\} = Q_r\left\{-\frac{7}{8}\right\} = -\frac{7}{8} \\ y(3) &= Q_r\left\{R\left[-\frac{79}{64}\right]\right\} = Q_r\left\{\frac{49}{64}\right\} = \frac{3}{4} \end{aligned}$$

$$
\begin{aligned}
y(4) &= Q_r\left\{R\left[\frac{77}{64}\right]\right\} = Q_r\left\{-\frac{51}{64}\right\} = -\frac{3}{4} \\
y(5) &= Q_r\left\{R\left[-\frac{9}{8}\right]\right\} = Q_r\left\{\frac{7}{8}\right\} = \frac{7}{8} \\
y(6) &= Q_r\left\{R\left[\frac{79}{64}\right]\right\} = Q_r\left\{\frac{-49}{64}\right\} = -\frac{3}{4} \\
y(7) &= Q_r\left\{R\left[-\frac{77}{64}\right]\right\} = Q_r\left\{\frac{51}{64}\right\} = \frac{3}{4} \\
y(8) &= Q_r\left\{R\left[\frac{9}{8}\right]\right\} = Q_r\left\{-\frac{7}{8}\right\} = -\frac{7}{8} \\
&\vdots
\end{aligned}
\tag{20.74}
$$

This is a large-scale oscillation with nearly full-scale amplitude.

There are several ways to prevent overflow oscillations in fixed-point filter realizations. The most obvious is to scale the filter calculations so as to render overflow impossible. However, this may unacceptably restrict the filter dynamic range. Another method is to force completed sums-of-products to saturate at ± 1, rather than overflowing [18,19]. It is important to saturate only the completed sum, since intermediate overflows in two's complement arithmetic do not affect the accuracy of the final result. Most fixed-point digital signal processors provide for automatic saturation of completed sums if their *saturation arithmetic* feature is enabled. Yet another way to avoid overflow oscillations is to use a filter structure for which any internal filter transient is guaranteed to decay to zero [20]. Such structures are desirable anyway, since they tend to have low roundoff noise and be insensitive to coefficient quantization [21].

20.7 Coefficient Quantization Error

Each filter structure has its own finite, generally nonuniform grids of realizable pole and zero locations when the filter coefficients are quantized to a finite word length. In general the pole and zero locations desired in a filter do not correspond exactly to the realizable locations. The error in filter performance (usually measured in terms of a frequency response error) resulting from the placement of the poles and zeros at the nonideal but realizable locations is referred to as coefficient quantization error.

Consider the second-order filter with complex–conjugate poles

$$
\begin{aligned}
\lambda &= re^{\pm j\theta} \\
&= \lambda_r \pm j\lambda_i \\
&= r\cos(\theta) \pm jr\sin(\theta)
\end{aligned}
\tag{20.75}
$$

and transfer function

$$
H(z) = \frac{1}{1 - 2r\cos(\theta)z^{-1} + r^2z^{-2}} \tag{20.76}
$$

realized by the difference equation

$$
y(n) = 2r\cos(\theta)y(n-1) - r^2y(n-2) + x(n) \tag{20.77}
$$

Figure 20.3 from Ref. [5] shows that quantizing the difference equation coefficients results in a nonuniform grid of realizable pole locations in the z plane. The grid is defined by the intersection of vertical lines

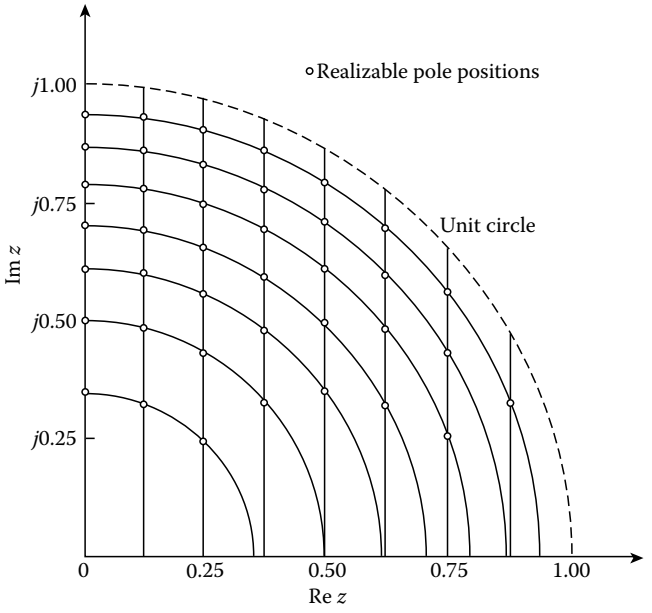


FIGURE 20.3 Realizable pole locations for the difference Equation of 20.76.

corresponding to quantization of $2\lambda_r$ and concentric circles corresponding to quantization of $-r^2$. The sparseness of realizable pole locations near $z = \pm 1$ will result in a large coefficient quantization error for poles in this region.

Figure 20.4 gives an alternative structure to Equation 20.77 for realizing the transfer function of Equation 20.76. Notice that quantizing the coefficients of this structure corresponds to quantizing λ_r and λ_i. As shown in Figure 20.5 from Ref. [5], this results in a uniform grid of realizable pole locations. Therefore, large coefficient quantization errors are avoided for all pole locations.

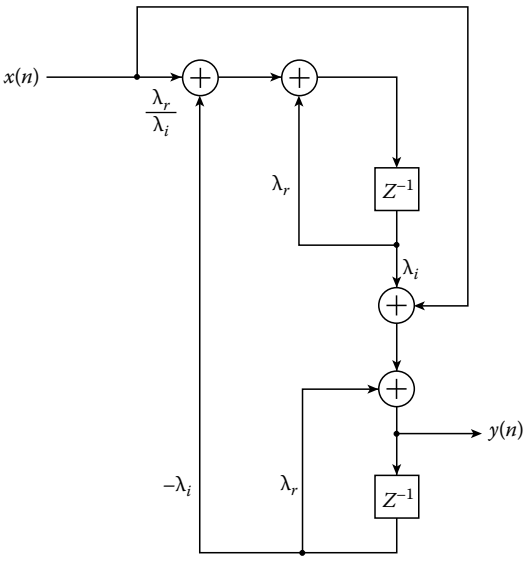


FIGURE 20.4 Alternate realization structure.

It is well established that filter structures with low roundoff noise tend to be robust to coefficient quantization, and vice versa [22–24]. For this reason, the uniform grid structure of Figure 20.4 is also popular because of its low roundoff noise. Likewise, the low-noise realizations of Ref. [7–10] can be expected to be relatively insensitive to coefficient quantization, and digital wave filters and lattice filters that are derived from low-sensitivity analog structures tend to have not only low coefficient sensitivity, but also low roundoff noise [25,26].

It is well known that in a high-order polynomial with clustered roots, the root location is a very sensitive function of the polynomial coefficients. Therefore, filter poles and zeros can be much more accurately controlled if higher order filters are realized by breaking them up into the parallel or cascade connection of first- and second-order subfilters. One exception to this rule is the case of linear-phase

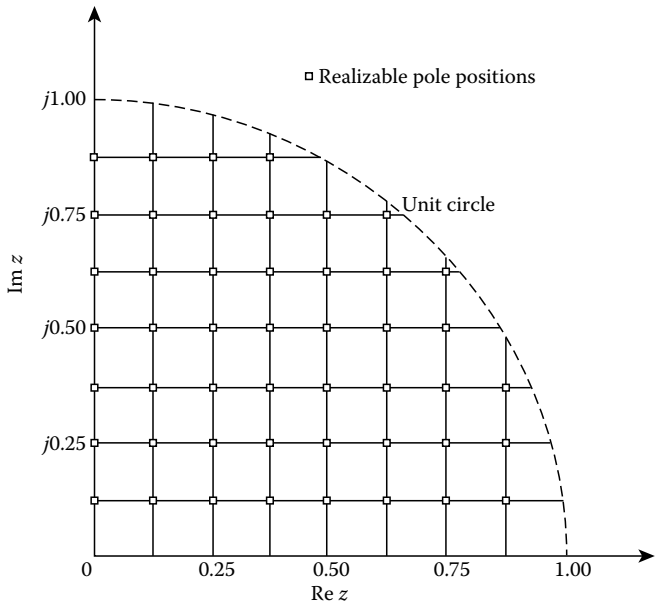


FIGURE 20.5 Realizable pole locations for the alternate realization structure.

FIR filters in which the symmetry of the polynomial coefficients and the spacing of the filter zeros around the unit circle usually permits an acceptable direct realization using the convolution summation.

Given a filter structure it is necessary to assign the ideal pole and zero locations to the realizable locations. This is generally done by simply rounding or truncating the filter coefficients to the available number of bits, or by assigning the ideal pole and zero locations to the nearest realizable locations. A more complicated alternative is to consider the original filter design problem as a problem in discrete optimization, and choose the realizable pole and zero locations that give the best approximation to the desired filter response [27–30].

20.8 Realization Considerations

Linear-phase FIR digital filters can generally be implemented with acceptable coefficient quantization sensitivity using the direct convolution sum method. When implemented in this way on a digital signal processor, fixed-point arithmetic is not only acceptable but may actually be preferable to floating-point arithmetic. Virtually all fixed-point digital signal processors accumulate a sum of products in a double-length accumulator. This means that only a single quantization is necessary to compute an output. Floating-point arithmetic, on the other hand, requires a quantization after every multiply and after every add in the convolution summation. With 32-b floating-point arithmetic these quantizations introduce a small enough error to be insignificant for many applications.

When realizing IIR filters, either a parallel or cascade connection of first- and second-order subfilters is almost always preferable to a high-order direct-form realization. With the availability of very low-cost floating-point digital signal processors, like the Texas Instruments TMS320C32, it is highly recommended that floating-point arithmetic be used for IIR filters. Floating-point arithmetic simultaneously eliminates most concerns regarding scaling, limit cycles, and overflow oscillations. Regardless of the arithmetic employed, a low roundoff noise structure should be used for the second-order sections. Good choices are given in Refs. [2,10]. Recall that realizations with low fixed-point roundoff noise also have low floating-point roundoff noise. The use of a low roundoff noise structure for the second-order

sections also tends to give a realization with low coefficient quantization sensitivity. First-order sections are not as critical in determining the roundoff noise and coefficient sensitivity of a realization, and so can generally be implemented with a simple direct form structure.

References

1. C. Weinstein and A. V. Oppenheim, A comparison of roundoff noise in floating-point and fixed-point digital filter realizations, *Proc. IEEE,* 57, 1181–1183, June 1969.
2. L. M. Smith, B. W. Bomar, R. D. Joseph, and G. C. Yang, Floating-point roundoff noise analysis of second-order state-space digital filter structures, *IEEE Trans. Circuits Syst. II*, 39, 90–98, February 1992.
3. J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*, 1st edn., New York: Macmillan, 1988.
4. A. V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.
5. A. V. Oppenheim and C. J. Weinstein, Effects of finite register length in digital filtering and the fast Fourier transform, *Proc. IEEE*, 60, 957–976, August 1972.
6. B. W. Bomar and R. D. Joseph, Calculation of L_∞ norms for scaling second-order state-space digital filter sections, *IEEE Trans. Circuits Syst*., CAS-34, 983–984, August 1987.
7. C. T. Mullis and R. A. Roberts, Synthesis of minimum roundoff noise fixed-point digital filters, *IEEE Trans. Circuits Syst*., CAS-23, 551–562, September 1976.
8. L. B. Jackson, A. G. Lindgren, and Y. Kim, Optimal synthesis of second-order state-space structures for digital filters, *IEEE Trans. Circuits Syst*., CAS-26, 149–153, March 1979.
9. C. W. Barnes, On the design of optimal state-space realizations of second-order digital filters, *IEEE Trans. Circuits Syst*., CAS-31, 602–608, July 1984.
10. B. W. Bomar, New second-order state-space structures for realizing low roundoff noise digital filters, *IEEE Trans. Acoust*., *Speech, Signal Process*., ASSP-33, 106–110, February 1985.
11. S. R. Parker and S. F. Hess, Limit-cycle oscillations in digital filters, *IEEE Trans. Circuit Theory*, CT-18, 687–697, November 1971.
12. P. H. Bauer, Limit cycle bounds for floating-point implementations of second-order recursive digital filters, *IEEE Trans. Circuits Syst. II*, 40, 493–501, August 1993.
13. B. D. Green and L. E. Turner, New limit cycle bounds for digital filters, *IEEE Trans. Circuits Syst*., 35, 365–374, April 1988.
14. M. Buttner, A novel approach to eliminate limit cycles in digital filters with a minimum increase in the quantization noise, in *Proc. 1976 IEEE Int. Symp. Circuits Syst*., April 1976, 291–294.
15. P. S. R. Diniz and A. Antoniou, More economical state-space digital filter structures which are free of constant-input limit cycles, *IEEE Trans. Acoust., Speech, Signal Process*., ASSP-34, 807–815, August 1986.
16. B. W. Bomar, Low-rounoff-noise limit-cycle-free implementation of recursive transfer functions on a fixed-point digital signal processor, *IEEE Trans. Ind. Electron*., 41, 70–78, February 1994.
17. P. M. Ebert, J. E. Mazo, and M. G. Taylor, Overflow oscillations in digital filters, *Bell Syst., Tech. J*., 48. 2999–3020, November 1969.
18. A. N. Willson, Jr., Limit cycles due to adder overflow in digital filters, *IEEE Trans. Circuit Theory*, CT-19, 342–346, July 1972.
19. J. H. F. Ritzerfield, A condition for the overflow stability of second-order digital filters that is satisfied by all scaled state-space structures using saturation, *IEEE Trans. Circuits Syst*., 36, 1049–1057, August 1989.
20. W. T. Mills, C. T. Mullis, and R. A. Roberts, Digital filter realizations without overflow oscillations, *IEEE Trans. Acoust., Speech, Signal Process*., ASSP-26, 334–338, August 1978.

21. B. W. Bomar, On the design of second-order state-space digital filter sections, *IEEE Trans. Circuits Syst.*, 36, 542–552, April 1989.
22. L. B. Jackson, Roundoff noise bounds derived from coefficient sensitivities for digital filters, *IEEE Trans. Circuits Syst.*, CAS-23, 481–485, August 1976.
23. D. B. V. Rao, Analysis of coefficient quantization errors in state-space digital filters, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-34, 131–139, February 1986.
24. L. Thiele, On the sensitivity of linear state-space systems, *IEEE Trans. Circuits Syst.*, CAS-33, 502–510, May 1986.
25. A. Antoniou, *Digital Filters: Analysis and Design*, New York: McGraw-Hill, 1979.
26. Y. C. Lim, On the synthesis of IIR digital filters derived from single channel AR lattice network, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-32, 741–749, August 1984.
27. E. Avenhaus, On the design of digital filters with coefficients of limited wordlength, *IEEE Trans. Audio Electroacoust.*, AU-20, 206–212, August 1972.
28. M. Suk and S. K. Mitra, Computer-aided design of digital filters with finite wordlengths, *IEEE Trans. Audio Electroacoust.*, AU-20, 356–363, December 1972.
29. C. Charalambous and M. J. Best, Optimization of recursive digital filters with finite wordlengths, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-22, 424–431, December 1979.
30. Y. C. Lim, Design of discrete-coefficient-value linear-phase FIR filters with optimum normalized peak ripple magnitude, *IEEE Trans. Circuits Syst.*, 37, 1480–1486, December 1990.

21

VLSI Implementation of Digital Filters

Joseph B. Evans
University of Kansas

Timothy R. Newman
University of Kansas

21.1 Introduction **21**-1
21.2 General-Purpose Processors **21**-2
Historical Perspective • Current Processors • Future Directions
21.3 Special-Purpose Implementations **21**-8
FIR Filter Examples • IIR Filter Examples • Future Trends
21.4 Programmable Logic Implementations **21**-11
FIR Filter Implementations • IIR Filter Implementations • Future Trends
References **21**-14
Further Information **21**-15

21.1 Introduction

Digital implementations of filters are preferred over analog realizations for many reasons. Improvements in VLSI technology have enabled digital filters to be used in an increasing number of application domains.

There are a variety of methods that can be used to implement digital filters. In this chapter we focus on the use of traditional VLSI digital logic families such as CMOS, rather than more exotic approaches. The vast majority of implementations encountered in practice make use of traditional technologies because the performance and cost characteristics of these approaches are so favorable.

Digital filter implementations can be classified into several categories based on the architectural approach used: general purpose, special purpose, and programmable logic implementations. The choice of a particular approach should be based upon the flexibility and performance required by a particular application. General-purpose architectures possess a great deal of flexibility, but are somewhat limited in performance, being best suited for relatively low sampling frequencies, usually under 10 MHz. Special-purpose architectures are capable of much higher performance, with sampling frequencies as high as 1 GHz, but are often only configurable for one application domain. Programmable logic implementations lie somewhere between these extremes, providing both flexibility and reasonably high performance, with sampling rates as high as 200 MHz.

Digital filtering implementations have been strongly influenced by evolution of VLSI technology. The regular computational structures encountered in filters are well suited for VLSI implementation. This regularity often translates into efficient parallelism and pipelining. Further, the small set of computational structures required in digital filtering makes automatic synthesis of special-purpose and programmable logic designs feasible. The design automation of digital filter implementation is relatively simple

compared to the general design synthesis problem. For this reason, digital filters are often the test case for evaluating new device and computer-aided design technologies.

21.2 General-Purpose Processors

General-purpose digital signal processors are by far the most commonly used method for digital filter implementation, particularly at audio bandwidths. These systems possess architectures well suited to digital filtering, as well as other digital signal processing (DSP) algorithms.

21.2.1 Historical Perspective

General-purpose digital signal processors trace their lineage back to the microprocessors of the early 1980s. The generic microprocessors of that period were ill suited for the implementation of DSP algorithms, due to the lack of hardware support for numerical algorithms of significant complexity in those architectures. The primary requirement for DSP implementation was identified to be hardware support for multiplication, due to the large number of multiply-accumulate (MAC) operations in DSP algorithms and their large contribution to computational delays. The earliest widely available single chip general-purpose DSP implementation was from AT&T, which evolved into the AT&T DSP20 family. Products such as the Texas Instruments TMS32010 and NEC 7720 soon followed this. The early DSP chips exhibited several shortcomings, such as difficult programming paradigms, awkward architectures for many applications, and limited numerical precision. Many of these difficulties were imposed by the limits of the VLSI technology of the time, and some by inexperience in this particular application area. Despite these shortcomings, however, the early processors were well suited to the implementation of digital filter algorithms, because digital filtering was identified as one of the target areas for these architectures. This match between architecture and algorithms continues to be exhibited in current general-purpose DSP chips.

21.2.2 Current Processors

There are a variety of general-purpose digital signal processors currently commercially available. We will look at several of the most common architectural families in detail, although this discussion will not be comprehensive by any means. The processors are best classified into two categories, fixed-point processors and floating-point processors. In both cases, these architectures are commonly (although not exclusively) based on a single arithmetic unit shared among all computations, which leads to constraints on the sampling rates that may be attained.

Fixed-point processors exhibit extremely high performance in terms of maximum throughput as compared to their floating-point counterparts. In addition, fixed-point processors are typically inexpensive as compared to floating-point options, due to the smaller integrated circuit die area occupied by fixed-point processing blocks. A major difficulty encountered in implementing filters on fixed-point processors is that overflow and underflow need to be prevented by careful attention to scaling, and round-off effects may be significant.

Floating-point processors, on the other hand, are significantly easier to program, particularly in the case of complex algorithms, at the cost of lower performance and larger die area. Given the regular structure of most digital filtering algorithms and computer-aided design support for filters based on limited precision arithmetic, fixed-point implementations may be the more cost effective option for this type of algorithm. Because of the prevalence of both types of general-purpose processor, examples of each will be examined in detail.

Two widely used floating-point processor families will be studied, although there are many contenders in this field. These families are the Texas Instruments family of floating-point DSPs, in particular the TI TMS320C3x (TI, 1992) family, and the Analog Devices ADSP-21020 family (Schweber, 1993). More

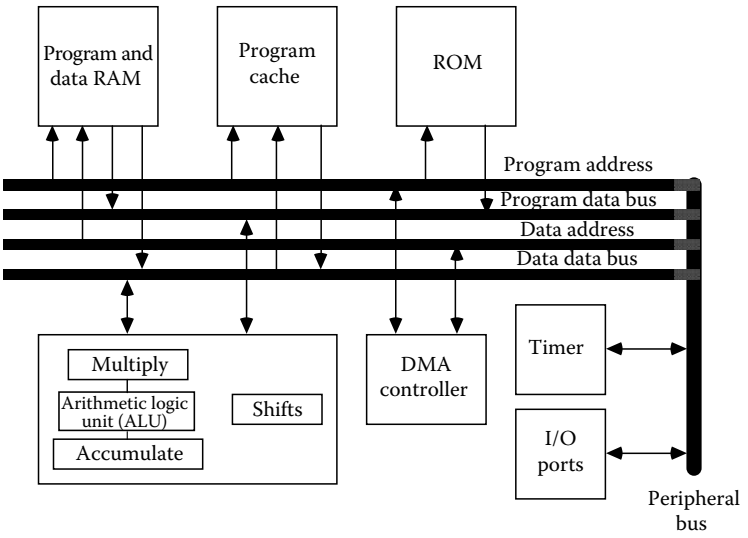


FIGURE 21.1 Texas Instruments TMS320C30 architecture.

recent examples of the TI family are the TMS320C67x DSP chips (TI, 2006a), and more recent Analog Devices parts are the ADSP-2116x SHARC chips (Analog Devices, 2006).

The architecture of the TI TMS320C30 is illustrated in Figure 21.1. The floating-point word size used by this processor was 32 bits. The most prominent feature of this chip was the floating-point arithmetic unit, which contains a floating-point multiplier and adder. This unit was highly pipelined to support high throughput, at the cost of latency; when data is input to the multiplier, for example, the results will not appear on the output from that unit until several clock cycles later. Other features included a separate integer unit for control calculations, and significant amounts (2k words) of SRAM for data and on-chip instruction memory. On-chip ROM (4k words) was also optionally provided in order to eliminate the need for an external boot ROM in some applications. This chip also included a 64-word instruction cache to allow its use with lower speed memories. The modified Harvard architecture, that is, the separate data and instruction buses, provided for concurrent instruction and data word transfers within one cycle time. The TMS320C30 offered instruction cycle times as low as 60 ns. A code segment which implements portions of an finite impulse response (FIR) filter on this device was

```
    RPTS   RC
    MPYF3  *AR0++(1),*AR1++(1)%,R0
||  ADDF3  R0,R2,R2
    ADDF   R0,R2,R0
```

where the `MPYF3` instruction performs a pipelined multiply operation in parallel with data and coefficient pointer increments. The `ADDF3` instruction is performed in parallel with the `MPYF3` instruction, as denoted by the "||" symbol. Because these operations are in parallel, only one instruction cycle per tap is required. An FIR filter tap was benchmarked at 60 ns on this chip. Similarly, a typical biquad infinite impulse response (IIR) filter code segment was

```
MPYF3  *AR0,*AR1,R0
MPYF3  *++AR0(1),*AR1--(1)%,R1
MPYF3  *++AR0(1),*AR1,R0
```

```
||  ADDF3  R0,R2,R2
    MPYF3  *++AR0(1),*AR1--(1)%,R0
||  ADDF3  R0,R2,R2
    MPYF3  *++AR0(1),R2,R2
||  STF    R2,*AR1++(1)%
    ADDF   R0,R2
    ADDF   R1,R2,R0
```

where the MPYF3 and ADDF3 instructions implement the primary filter arithmetic and memory pointer modification operations in parallel, as in the previous example. The biquad IIR benchmark on this processor was 300 ns. More recent members of the TI floating-point family such as the TMS320C6727 support two parallel FIR filter taps at 2.86 ns and two IIR biquad sections at 106 ns.

Another floating-point chip worthy of note is the Analog Devices ADSP-21020 series. The architecture of the ADSP-21020 chip is shown in Figure 21.2. This chip can be seen to share a number of features with the TMS320C3x family, that is, a 32 bit by 32 bit floating-point MAC unit (not pipelined), modified Harvard architecture, and 16 words of on-chip memory. In this case, the scratchpad memory was organized into register files, much like a general-purpose RISC architecture register set. The memory capacity of this device was significantly smaller than that of its competitors. As in the case of the TMS320C3x, on the other hand, an instruction cache (32 words) was also provided. The cycle time for the ADSP-21020 was 40 ns. An N tap FIR filter code segment illustrates the operation of this device,

```
        i0=coef;
        f9=0.0;
        f1=0; f4=dm(i0,m0); f5=pm(i8,m8);
        lcntr=N, DO bottom until lce;
bottom:     f1=f1+f9; f9=f4*f5; f4=dm(i0,m0); f5=pm(i8,m8);
        f1=f1+f9;
```

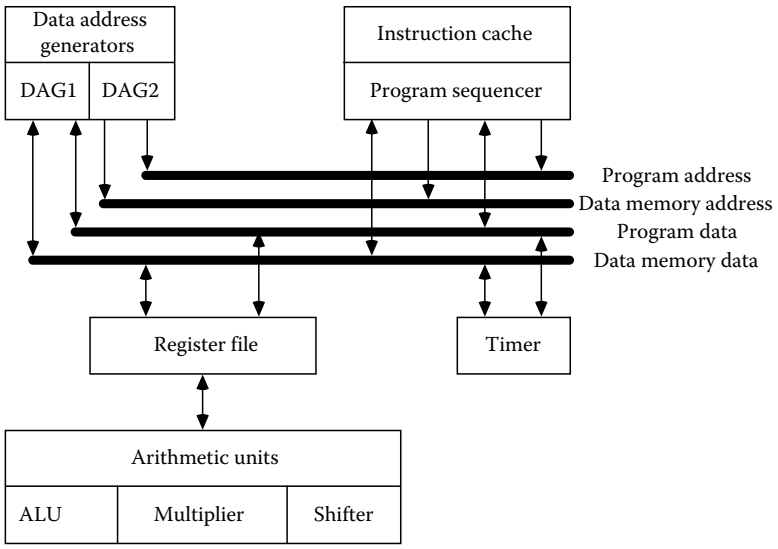


FIGURE 21.2 Analog devices ADSP-21020 architecture.

where the "*" and "+" instructions perform the MAC operations, and the dm() and pm() instructions perform the memory address update operations in parallel. An FIR filter tap thus executed in one instruction per tap on the ADSP-21020, or in 40 ns. An IIR filter biquad section required 200 ns on this chip. More recent members of this family such as the ADSP-21367 support FIR filter taps at 3 ns and IIR biquad sections at 60 ns. Note that while the assembly language for the Analog Devices chip was significantly different from that of the Texas Instruments chip, the architectural similarities are striking.

Two families of fixed-point digital signal processors will also be examined and compared. These are the Texas Instruments TMS320C5x family (TI, 1993) and the Motorola DSP56000 series of devices (Motorola, 1989). More recent examples of the TI family are the TMS320C62x DSP chips (TI, 2006b) and more recent Motorola parts are the DSP56300 series of chips (Motorola, 2007).

The Texas Instruments TMS320C5x series devices were high performance digital signal processors derived from the original TI DSP chip, the TMS32010, and its successor, the TMS320C2x. The architecture of the TMS320C50 is shown in Figure 21.3. This chip was based on the Harvard architecture, that is, separate data and instruction buses. This additional bandwidth between processing elements supported rapid concurrent transfers of data and instructions. This chip used a 16 bit by 16 bit fixed-point multiplier and a 32-bit accumulator, and up to 10k words on-chip scratchpad RAM. This architecture supported instruction rates of 50 ns. An FIR filter code segment is shown below, where the primary filter tap operations were performed by the `MACD` instruction,

```
RPTK  N
MACD  *-,COEFFP
```

This exhibits a general similarity with that for the TI floating-point chips, in particular a single instruction cycle per tap, although in this case a single instruction was executed as opposed to two parallel instructions on the TMS320C3x. The memory addressing scheme was also significantly different. An FIR filter on the TMS320C5x could thus be implemented in 25 ns per tap. An *N*th-order IIR filter code segment is show below, where the `MACD` and `AC` instructions performed the primary multiplication operations,

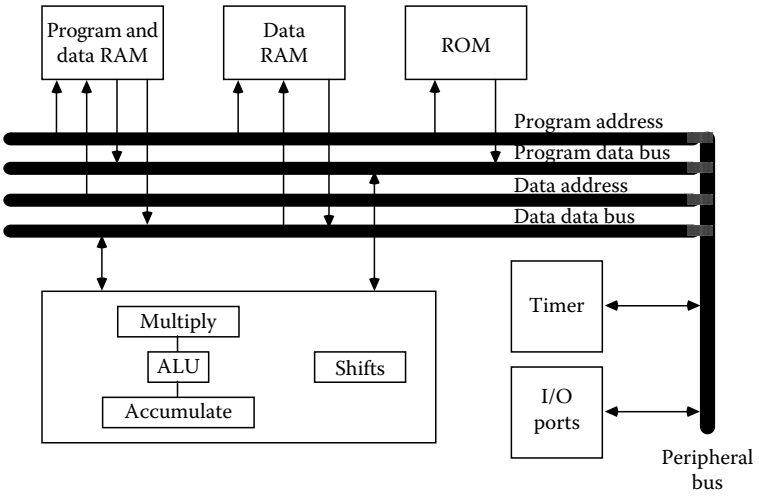


FIGURE 21.3 Texas Instruments TMS320C50 architecture.

```
ZPR
LACC  *,15,AR1
RPT   #(N−2)
AC    COEFFB,*−
APAC
SACH  *,1
ADRK  N−1
RPTZ  #(N-1)
MACD  COEFFA,*−
LTA   *,AR2
SACH  *,1
```

A single IIR biquad section could be performed in 250 ns on this chip. More recent members of the TI family such as the TMS320C6203C support two parallel FIR filter taps at 3.33 ns speeds and IIR biquad sections at 66.6 ns.

The Motorola 56001 series was a fixed-point architecture with 24-bit word size, as opposed to the smaller word sizes in most fixed-point DSP chips. The architecture of the 56001 is depicted in Figure 21.4. This chip shared many of the same features as other DSP chips, that is, Harvard architecture, on-chip scratchpad memory (512 words), and hardware MAC support, in this case 24 bit by 24 bit operators which form a 56 bit result. The instruction cycle time of the Motorola 56001 was 97.5 ns. An FIR filter implemented on the 56001 might use the code segment shown below,

```
MOVE  #AADDR, R0
MOVE  #BADDR+n, R4
NOP
```

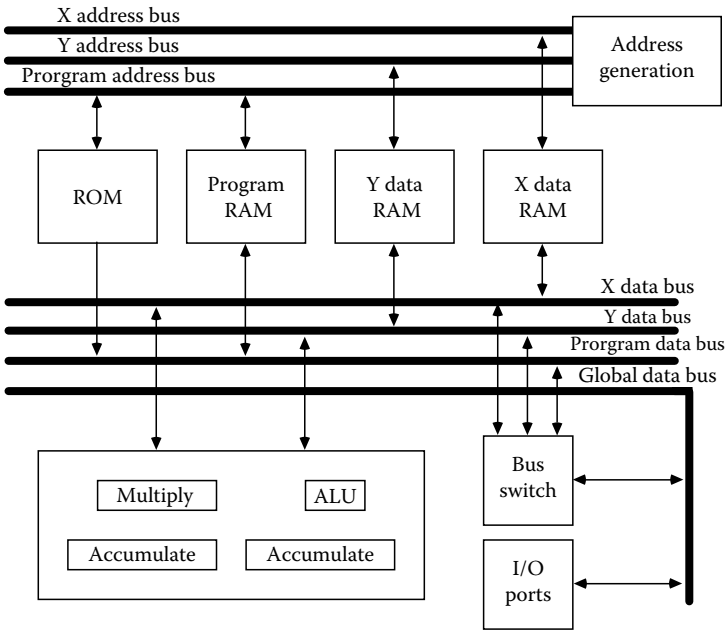


FIGURE 21.4 Motorola 56001 architecture.

```
CLR  A          X:(R0)+,X0  Y:(R4)-,Y0
REP  #N
MAC  X0,Y0,A    X:(R0)+,X0  Y:(R4)-,Y0
RND  A
```

where the MAC instruction retrieves data from the appropriate registers, loads it into the multiplier, and leaves the result in the accumulator. The 56001 could perform FIR filtering at a rate of one instruction per tap, or 97.5 ns per tap. An IIR filter code segment used the MAC instruction, as well as several others to set up the registers for the arithmetic unit, as shown below.

```
OR    #$08,MR
RND   A           X:(R0)-,X0    Y:(R4)+,Y0
MAC   -Y0,X0,A    X:(R0)-,X1    Y:(R4)+,Y0
MAC   -Y0,X1,A    X1,X:(R0)+    Y:(R4)+,Y0
MAC   Y0,X0,A     A,X:(R0)      Y:(R4),Y0
MAC   Y0,X1,A                   1
MOVE        A,X:OUTPUT
```

The 56001 could compute a second-order IIR biquad in seven instruction cycles, or 682.5 ns. More recent members of this family such as the DSP56L307 support FIR filter taps at an asymptotic speed of 6.25 ns and IIR biquad sections at 56.25 ns.

From these examples, it can be seen that general-purpose DSP processors possess many common features which make them well suited for digital filtering. The hardware MAC unit, Harvard architecture, and on-chip memory are consistent characteristics of these devices. The major shortcoming of such architectures for digital filtering is the necessity to multiplex a single arithmetic unit (or very small number of ALUs), which implies that sampling rates above $1/NT$ are not possible, where N is the number of atomic operations (e.g., FIR filter taps) and T is the time to complete those operations.

21.2.3 Future Directions

Several trends have become apparent as VLSI technology has improved. One trend of note is the increasing use of parallelism, both on-chip and between chips. The support for multiprocessor communications in the TI TMS320C80 provides an avenue for direct parallel implementation of algorithms. Architectures based upon multiple fixed-point DSP processors on a single chip have also been fielded.

Another trend has been the development of better programming interfaces for the general-purpose chips. In particular, high level language compilers have improved to the point where they provide for reasonably good performance for complex algorithms, although still not superior to that obtained by manual assembly language programming.

This trend is being accelerated by an initiative called GNURadio (GNU Radio, 2007), which is making it easier to implement all types of digital filters purely in software in order to enable creation of software radios. GNURadio provides signal processing blocks implemented in C++ that are connected together to form a complete communications system. The advantage of this software implementation is that it is generally hardware independent and can run on any general-purpose processor, opening up the world of digital implementations to anyone with a PC. The added flexibility comes at a loss of efficiency. Due to the generalized nature of the hardware independent system, the computations will not be implemented as efficiently as they would be if written in assembly for a specific processor. An IIR filter code segment from the GNURadio project is shown below:

```
acc=d_fftaps[0] * input;
for (i=1; i<n; i++)
  acc += (d_fftaps[i] * d_prev_input[latest+i]
        + d_fbtaps[i] * d_prev_output[latest+i]);
```

This C++ code segment from the GNURadio library implement a direct form 1 IIR filter where fftaps are the feed-forward taps and fbtaps are the feed-back taps. To use the GNURadio software for practical signal analysis, some sort of analog to digital apparatus is required. This can be as simple as a standard PC sound card or as complex as the universal software radio peripheral (USRP), which is sold by the maintainers of the GNURadio project as an RF front end that contains an onboard field programmable gate arrays (FPGA) and several optional daughter boards for the frequency band of interest (USRP, 2007).

Another trend that is worthy of note is the development of low-power DSP implementations. These devices are targeted at the wireless personal communications system (PCS) marketplace, where minimum power usage is critical. The developments in this area have been particularly striking, given the strong dependence of power consumption on clock frequency, which is usually high in DSP implementations. Through a combination of careful circuit design, power supply voltage reductions, and architectural innovations, extremely low power implementations have been realized.

A final trend is related to the progress of general-purpose processors relative to DSP chips. The evolution of general-purpose DSP implementations may have come full circle, as general-purpose processors such as the Intel Pentium family and digital equipment company (DEC) Alpha family possess on-chip floating-point multiplication units, as well as memory bandwidths equaling or exceeding that of the DSP chips. These features are reflected in the performance of these chips on standard benchmarks (Stewart et al., 1992), in which the DEC Alpha outperforms the fastest DSP engines. Similar results were obtained from the Pentium upon the implementation of the MMX capabilities; even older Pentium chipsets outperform most floating-point and fixed-point DSP chips (BTDI, 2000).

These trends, particularly multiple processor parallelism and general-purpose ease of programming, are illustrated by the Cell Broadband Engine Architecture developed via a joint effort between Sony, Toshiba, and IBM (IBM, 2007). The Cell combines a standard Power architecture core with streamlined coprocessing components, with the first major use of the Cell being in Sony's Playstation 3 console. It was designed to bridge the gap between general-purpose processors and more specialized high-performance processors. The architecture consists of a main processor called the power processing element (PPE), eight coprocessors called the synergistic processing elements (SPEs), and a high-bandwidth bus connecting all of these elements. The PPE is able to run a conventional operating system because of its Power Architecture core, and has control over the eight SPEs allowing high performance and mathematically intensive tasks to be achieved.

21.3 Special-Purpose Implementations

The tremendous growth in the capabilities of VLSI technology and the corresponding decrease in the fabrication costs have lead to the wide availability advent of application-specific integrated circuits (ASICs). These devices are tailored to a particular application or domain of applications in order to provide the highest possible performance at low per-unit costs.

Although it is difficult to generalize, special-purpose implementations share some common features. The first is the high degree of parallelism in these designs. For example, a typical special-purpose FIR filter implementation will contain tens or hundreds of MAC units, each of which executes a filter tap operation at the same time. This is in contrast to most general-purpose architectures, in which a single MAC unit is shared. Another common feature is extensive pipelining between arithmetic operators; this leads to high sampling rates and high throughput, at some cost in latency. Finally, these implementations are often lacking in flexibility, being designed for specific application domains. The number of filter taps

may be fixed, or the filter coefficients themselves may be fixed. In almost all instances, these implementations are based on fixed-point arithmetic.

Because the implementation cost of multiplication operations is so large compared to other operations, significant research effort has been expended on developing fast and efficient multiplier architectures, as well as digital filter design techniques that can be used to reduce the number of multiplications. A large number of multiplier architectures have been developed, ranging from bit-serial structures to bit and word level pipelined array designs (Ma and Taylor, 1990). The most appropriate architecture is a function of the application requirements, as various area versus speed options are available. The other major research direction is the minimization of multiplication operations. In this case, multiplications are eliminated by conscientious structuring of the realization, as in linear phase filters, circumvented by use of alternate number systems such as the residue number system (RNS), or simplified to a limited number of shift-and-add operations. The later option has been used successfully in a large number of both FIR and IIR realizations, some of which will be discussed below.

Historically, bit-serial implementations of digital filters have been of some interest to researchers and practitioners in the early days of VLSI because of the relatively high cost of silicon area devoted to both devices and routing (Denyer and Renshaw, 1985). Even in primitive technologies, bit-serial implementations could exploit the natural parallelism in digital filtering algorithms.

As clock rates have risen, and silicon area has become more economical, parallel implementations have become the most effective way of implementing high-performance digital filters. The concept of the systolic array has strongly influenced the implementation of both FIR and IIR filters (Kung, 1988). Systolic arrays are characterized by spatial and temporal locality, that is, algorithms and processing elements should be structured to minimize interconnection distances between nodes and to provide at least a single delay element between nodes. Interconnection distances need to be kept to a minimum to reduce delays associated with signal routing, which is becoming the dominant limiting factor in VLSI systems. Imposing pipeline delays between nodes minimizes computational delay paths and leads to high throughput.

These characteristic features of special-purpose digital filter designs will be illustrated by examples of FIR and IIR filter implementations. It should be noted that it is increasingly difficult to identify ASICs that only perform digital filtering; as VLSI capabilities increase, this functionality is more typically embedded with other functions in very application-focused devices.

21.3.1 FIR Filter Examples

FIR filters may be implemented in a number of ways, depending on application requirements. The primary factors that must be considered are the filter length, sampling rate, and area, which determine the amount of parallelism that can be applied. Once the degree of parallelism and pipelining are determined, the appropriate general filter structure can be determined.

A typical high-performance FIR filter implementation (Khoo et al., 1993) provided sampling rates of 180 MHz for 32 linear phase taps. This chip used canonical signed digit (CSD) coefficients. This representation is based on a number system in which the digits take the values $(-1, 0, 1)$. A filter tap can be implemented with a small number of these digits, and hence that tap requires a small number of shift-and-add operations. Each coefficient was implemented based on two bit shift-and-add units, as depicted in Figure 21.5. Delay elements are bypassed during configuration to allow realization of coefficients with additional bits. This chip also made use of extensive pipelining, carry-save addition, and advanced single-phase clocking techniques to provide high throughput.

In part due to the highly structured nature of FIR filtering algorithms, automatic design tools have been used to successfully implement high-performance FIR filters similar to that just presented. These methods often integrate the filter and architectural design into a unified process which can effectively utilize silicon area to provide the desired performance.

At the other extreme of performance was the Motorola 56200 FIR filter chip (Motorola, 1988) (Figure 21.6). This chip, although quite old, represents an approach to the custom implementation of

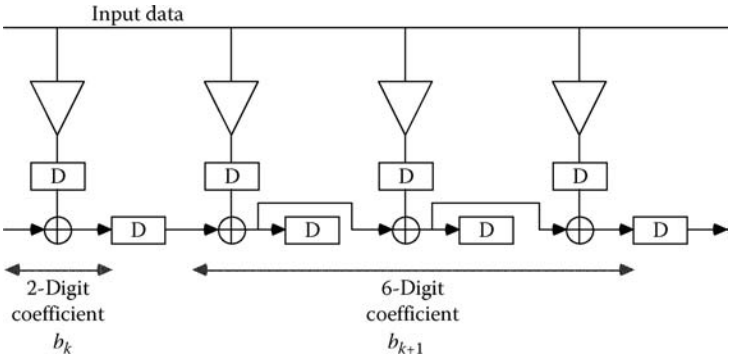


FIGURE 21.5 Custom FIR filter architecture for 180 MHz sampling rates.

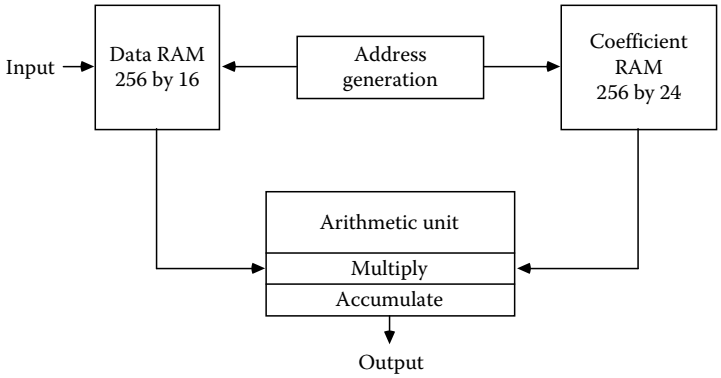


FIGURE 21.6 Motorola 56200 architecture.

long (several hundred taps) FIR filters. In this case, a single processing element was multiplexed among all of the filter taps, similar in concept to the approach used in general-purpose DSP processors. Due to the regularity of the filter structure, extensive pipelining in the arithmetic unit could be used to support a large number of taps at audio rates. This chip could be used to realize a 256 tap FIR filter at sampling rates up to 19 kHz, with higher performance for shorter filters. Longer filters could be implemented using cascaded processors.

A comparison of implementations (Hartley et al., 1989; Hatamian and Rao, 1990; Khoo et al., 1993; Laskowski and Samueli 1992; Ruetz, 1989; Yassa et al., 1987; Yoshino et al., 1990) illustrates the range of design and performance options. This is illustrated in Table 21.1, where the ''score'' is calculated

TABLE 21.1 FIR Filter ASIC Comparison

Design	Taps	Area (mm^2)	Rate (MHz)	Technology (μm)	Score
Khoo 93	32	20.1	180.0	1.2	495.2
Laskowski 92	43	40.95	150.0	1.2	139.3
Yoshino 90	64	48.65	100.0	0.8	33.68
Ruetz 89	64	225.0	22.0	1.5	21.12
Yassa 87	16	17.65	30.0 (est.)	1.25	53.12
Hartley 89	4	25.8 (est.)	37.0	1.25	11.20
Hatamian 90	40	22.0	100.0	0.9	132.5

according to the sampling rate multiplied by the number of taps per unit area, with normalization for the particular technology used. This simplistic comparison does not consider differences in word length or coefficient codings, but it does provide some insight into the results of the various design approaches. A significant number of other digital FIR filtering chips exist, both research prototypes and commercial products; this exposition only outlines some of the architectural options.

21.3.2 IIR Filter Examples

Custom IIR filter implementations are also most commonly based on parallel architectures, although there are somewhat fewer custom realizations of IIR filters than FIR filters. A significant difficulty in the implementation of high performance IIR filters in the need for feedback in the computation of an IIR filter section. This limits the throughput that can be attained to at least one MAC cycle in a straightforward realization. Another difficulty is the numerical stability of IIR filters with short coefficients, which makes aggressive quantization of coefficients less promising.

In order to address the difficulties with throughput limitation due to feedback, structures based on systolic concepts have been developed. Although the feedback problem imposes a severe constraint on the implementation, use of bit and word level systolic structures which pipeline data most significant digit first can minimize the impact of this restriction (Woods et al., 1990). Using these techniques, and a Signed Binary Number Representation (SBNR) similar to a CSD code, first-order sections with sampling rates of 15 MHz were demonstrated in a 1.5 μm standard cell process in an area of 21.8 mm^2. This particular design used fairly large coefficient and data words, however, at 12 bits and 11 bits, respectively.

The numerical stability problem has been addressed through a variety of techniques. One of these is based on minimizing limited precision effects by manipulation of traditional canonical filter structures and clever partitioning of arithmetic operations. A more recent and general approach is based on modeling the digital implementations of filters after their analog counterparts; these classes of filters are known as wave digital filters (WDFs) (Fettweis, 1986). WDFs exhibit good passband ripple and stopband attenuation, with high tolerance to limited wordlength effects. Because of the latter property, efficient implementations based on short word sizes are feasible. A WDF design for a second-order section in custom 1.5 μm CMOS based on a restricted coefficient set akin to CSD supported 10 MHz sampling rates in an area of 12.9 mm^2 (Wicks and Summerfield, 1993).

21.3.3 Future Trends

The future trends in digital filter implementation appear to be a fairly straightforward function of the increasing capability of VLSI devices. In particular, more taps and filter sections per chip and higher sampling rates are becoming achievable. Related to these trends are higher degrees of on-chip parallelism. Further, programmability is more reasonable as density and speed margins increase, although there is still a high cost in area and performance. Finally, special-purpose implementations show extraordinary promise in the area of low power systems, where custom circuit design techniques and application-specific architectural features can be combined to best advantage.

21.4 Programmable Logic Implementations

The rapid evolution of VLSI technology has enabled the development of several high-density programmable logic architectures. There are several novel features that make these devices of interest beyond their traditional field-of-state machine implementation. In particular, the density of the largest of these devices is over 12,000,000 gates (Xilinx, 2007), which encompasses the level of complexity found in the majority of ASICs (although some ASICs are significantly more complex). This level of complexity is sufficient to support many designs that would traditionally need to be implemented as ASICs. The speed of

programmable logic devices (PLDs) and FPGAs is quite reasonable, with toggle rates on the order of 550 MHz (Xilinx, 2007). While this is not as great as custom implementations, it does allow many applications to be realized in this new technology.

One of the most significant features of FPGA implementations is the capability for in-system reprogrammability in many FPGA families. Unlike traditional field programmable parts based on anti-fuse technology and which can only be programmed once, many of the new architectures are based on memory technology. This means that entirely new computational architectures can be implemented simply by reprogramming the logic functions and interconnection routing on the chip. Ongoing research efforts have been directed toward using FPGAs as generalized coprocessors for supercomputing and signal processing applications.

The implications of programmable device technology for filter implementation are significant. These devices provide an enormous amount of flexibility, which can be used in the implementation of a variety of novel architectures on a single chip. This is particularly useful for rapid prototyping of digital filtering algorithms, where several high-performance designs can be evaluated in a target environment on the same hardware platform. Further, complex adaptive systems based on this technology and which use a variety of signal processing and digital filter techniques are becoming increasingly popular in a variety of applications.

Because many of the programmable logic architectures are based on SRAM technology, the density of these devices can be expected to grow in parallel with the RAM growth curve, that is, at approximately 60% per year. Further, since these devices may be used for a large variety of applications, they have become high-volume commodity parts, and hence prices are relatively low compared to more specialized and low-volume DSP chips. This implies that new DSP systems that were not previously technically and economically feasible to implement in this technology are now feasible.

One of the extra costs of this approach, as opposed to the full custom strategy, is the need for support chips. Several chips are typically needed, including memory to store the programmable device configuration, as well as logic to control the downloading of the program. These issues are generally outweighed by the flexibility provided by programmable solutions.

We will next examine the implementation of several FIR and IIR digital filtering architectures based on FPGAs.

21.4.1 FIR Filter Implementations

Several approaches to the FPGA implementation of FIR filters can be taken. Due to the flexibility of these parts, switching from one architecture to the next only requires reprogramming the device, subject to constraints on I/O pin locations. Two fundamental strategies for realizing FIR filters will be illustrated here, one which is suited to relatively short filters (or longer filters cascaded across several chips) operating at high rates, and another which is suited for longer filters at lower rates.

A high-performance FIR filter example (Evans, 1993), illustrated in Figure 21.7, was based on the observation that since the entire device is reprogrammable, architectures in which filter coefficient multiplications are implemented as ''hardwired'' shifts can be easily reconfigured depending on the desired filter response. In this example, each of the coefficients was represented in a CSD code with a limited number of nontrivial (e.g., nonzero) bits, which allowed each tap to be implemented as a small number of shift-and-add operations. A filter tap could be implemented in two columns of logic blocks on a Xilinx 3100-series FPGA (Xilinx, 1993), where the two columns of full adders and associated delays implement a tap based on CSD coefficients with two nontrivial bits. With this architecture, up to 11 taps could be implemented on a single Xilinx XC3195 FPGA at sampling rates of above 40 MHz. Longer filters could be implemented by a cascade of FPGA devices.

An FIR filter architecture for longer filters was based upon implementation of several traditional MAC units on one chip, as shown in Figure 21.8. Each of these MAC units could then be shared among a large number of filter tap computations, much as the single MAC unit in the Motorola 56200 was multiplexed.

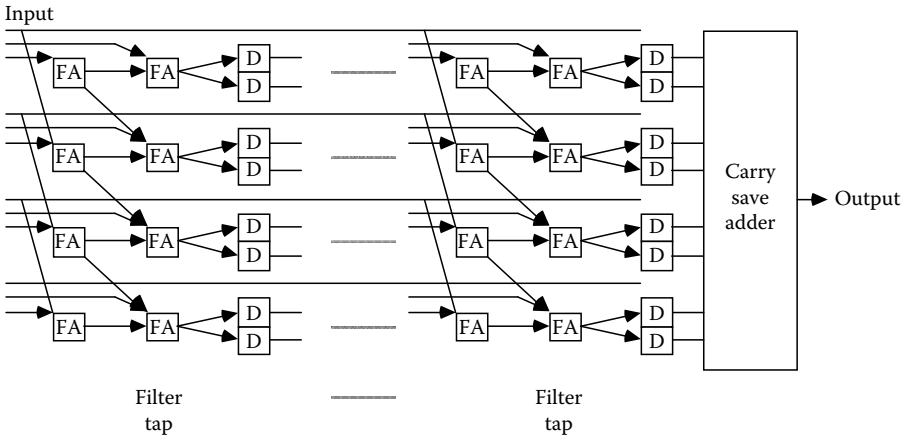


FIGURE 21.7 High-performance FIR architecture on FPGAs.

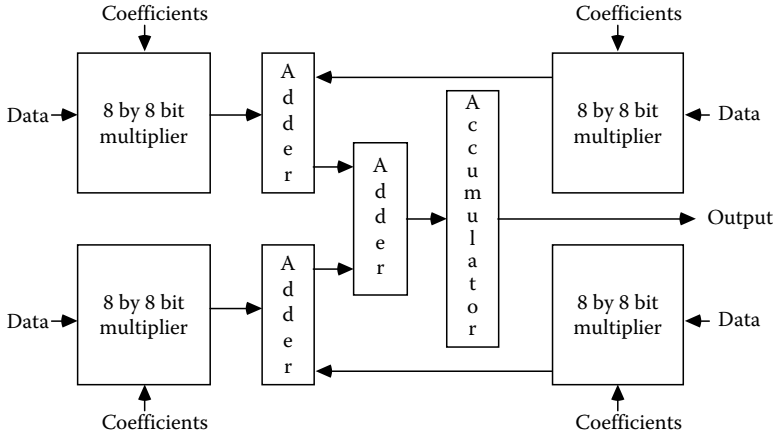


FIGURE 21.8 FIR architecture on FPGAs for large number of taps.

Since four multipliers could be implemented in the Xilinx 4000-series, the inherent parallelism of FIR filters can be exploited to support sampling rates of up to 1.25 MHz for 32 taps in that technology.

21.4.2 IIR Filter Implementations

As in the case of FIR filters, IIR filters can be implemented using a ''hardwired'' architecture suited to high performance, or a more traditional approach based on general MAC units. In the case of IIR filters, however, the hardwired implementation is significantly more desirable than the alternate approach due to the difficulty in rescheduling multiplexed processing elements in a system with feedback.

An architecture which is reconfigured to implement different filters will generally provide both high-performance and good area efficiency. An example of such a system is shown in Figure 21.9, in which two IIR biquad sections were implemented on a single FPGA using a traditional canonical filter structure (Chou et al., 1993). Each of the columns realized a shift-and-add for one nontrivial bit of a coefficient, where the shaded blocks also contained delay registers. This implementation yielded sampling rates of better than 10 MHz for typical coefficients.

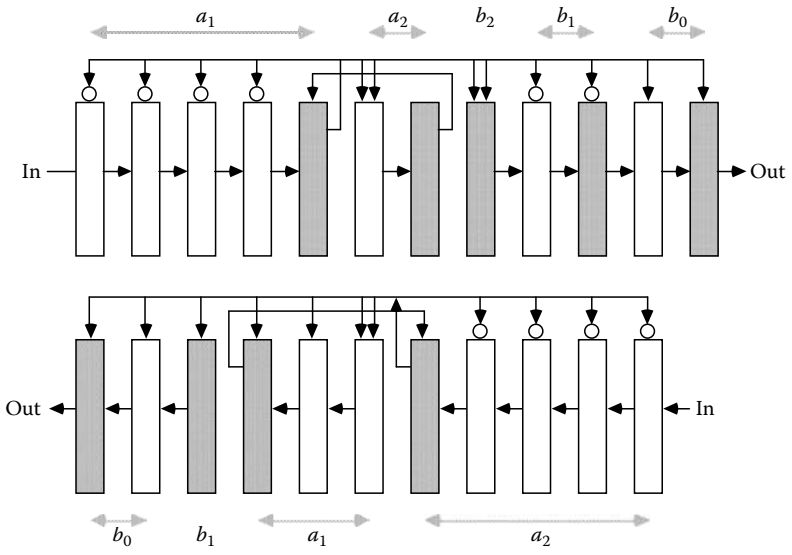


FIGURE 21.9 High-performance IIR architecture on FPGAs.

A more traditional approach to the realization of IIR filters using MAC units is also possible, but may be less efficient. The general architecture is similar to that of the FIR filter in Figure 21.8, with slight modifications to the routing between arithmetic units and support for scaling necessary in an IIR biquad section.

21.4.3 Future Trends

Because of the rapid advances in FPGA technology, higher performance digital filtering may in fact be possible with programmable logic than with typical custom ASIC approaches (Moeller, 1999). In addition, there are a wide variety of DSP core functions being offered by FPGA manufacturers (Xilinx, 2000) which will further accelerate this revolution in DSP implementation.

References

Analog Devices, 2006. *ADSP-21367 SHARC DSP Hardware Reference*, Rev. A., Analog Devices, Norwood, MA.

BTDI, 2000. *BDTImark Scores*, Berkeley Design Technologies, Inc., Berkeley, CA.

Chou, C. -J., S. Mohanakrishnan, and J. Evans, 1993. FPGA implementation of digital filters, *International Conference on Signal Processing Applications and Technology*, pp. 80–88, Santa Clara, CA.

Denyer, P. and D. Renshaw, 1985. *VLSI Signal Processing: A Bit-Serial Approach*, Addison-Wesley, Reading, MA.

Evans, J., 1993. An efficient FIR filter architecture, *IEEE International Symposium on Circuits and Systems*, pp. 627–630, Chicago, IL.

Fettweis, A., 1986. Wave digital filters: Theory and practice, *Proceedings of the IEEE*, 74(2), 270–327.

GNU Radio, 2007. GNU Radio 3.0.3 Release, http://www.gnu.org/software/gnuradio/index.html

Hartley, R., P. Corbett, et al., 1989. A high speed FIR filter designed by Compiler, *IEEE Cust. IC Conf.*, pp. 20.2.1–20.2.4, San Diego, CA.

Hatamian, M. and S. Rao, 1990. A 100 MHz 40-tap programmable FIR filter chip, *IEEE International Symposium on Circuits and Systems*, pp. 3053–3056, New Orleans, LA.

IBM, 2007, *The CELL project at IBM Research*, http://www.research.ibm.com/cell/

Khoo, K.-Y., A. Kwentus, and A. Willson, Jr., 1993. An efficient 175 MHz programmable FIR digital filter, *IEEE International Symposium on Circuits and Systems*, pp. 72–75, Chicago, IL.

Kung, S. Y., 1988. *VLSI Array Processors*, Prentice-Hall, Englewood Cliffs, NJ.

Laskowski, J. and Samueli, H., 1992. A 150-mhz 43-tap half-band fir digital filter in 1.2-μm CMOS generated by silicon compiler, *IEEE Custom Integrated Circuits Conference*, pp. 11.4.1–11.4.4.

Ma, G.-K. and F. Taylor, 1990. Multiplier policies for digital signal processing, *IEEE ASSP Mag.*, January, pp. 6–20.

Moeller, T. J., 1999. Field programmable gate arrays for radar front-end digital signal processing, MS thesis, MIT, Cambridge, MA.

Motorola, 1988. *DSP56200 Cascadable Adaptive Finite Impulse Response Digital Filter Chip*, Motorola, Phoenix, AZ.

Motorola, 1989. *DSP56001 Digital Signal Processor User's Manual*, Motorola, Phoenix, AZ.

Motorola, 2007. *DSP56374 24-Bit Digital Signal Processor User's Manual*, Motorola, Austin, TX.

Ruetz, P., 1989. The architectures and design of a 20-MHz real-time DSP chip set, *IEEE J. Solid State Circuits*, 24(2), 338–348.

Schweber, W., 1993. Floating-point DSP for high-speed signal processing, *Analog Dialogue*, 25(4), 3–5.

Stewart, L., A. Payne, and T. Levergood, 1992. Are DSP chips obsolete?, *International Conference on Signal Processing Applications and Technology*, Nov. 1992, pp. 178–187, Cambridge, MA.

Texas Instruments (TI), 1992. *TMS320C3x User's Manual*, Texas Instruments, Dallas, TX.

Texas Instruments (TI), 1993. *TMS320C5x User's Manual*, Texas Instruments, Dallas, TX.

Texas Instruments (TI), 2006a. *TMS320C6727 Floating-Point Digital Signal Processor*, Texas Instruments, Dallas, TX.

Texas Instruments (TI), 2006b. *TMS320C6205 Fixed-Point Digital Signal Processors*, Texas Instruments, Dallas, TX.

USRP, 2007, *Universal Software Radio Peripheral User's Manual*, Ettus Research LLC, Mountain View, CA.

Wicks, A. and S. Summerfield, 1993. VLSI implementation of high speed wave digital filters based on a restricted coefficient set, *IEEE International Symposium on Circuits and Systems*, pp. 603–606, Chicago, IL.

Woods, R., J. McCanny, S. Knowles, and O. McNally, 1990. A high performance IIR digital filter chip, *IEEE International Symposium on Circuits and Systems*, pp. 1410–1413, New Orleans, LA.

Xilinx, 1993. *The Programmable Logic Data Book*, Xilinx, San Jose, CA.

Xilinx, 2000. *XtremeDSP Technical Backgrounder*, Xilinx, San Jose, CA.

Xilinx, 2007. *Virtex-5 Platform FPGA Handbook*, Xilinx, San Jose, CA.

Yassa, F., J. Jasica, et al., 1987. A silicon compiler for digital signal processing: methodology, implementation, and applications, *Proc. IEEE*, 75(9), 1272–1282.

Yoshino, T., R. Jain, et al., 1990. A 100-MHz 64-Tap FIR digital filter in 0.8 μm BiCMOS gate array, *IEEE J. Solid State Circuits*, 25(6), 1494–1501.

Further Information

The publication IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing frequently contains articles on the VLSI implementation of digital filters as well as design methods for efficient implementation. The *IEEE Transactions on Signal Processing* often includes articles in these areas as well. Articles in the *IEEE Journal on Solid State Circuits*, the *IEEE Transactions on VLSI Systems*, and the *IEE Electronics Letters* regularly cover particular implementations of digital filters.

The conference proceedings for the IEEE International Symposium on Circuits and Systems and the IEEE International Conference on Acoustics, Speech, and Signal Processing also contain a wealth of information on digital filter implementation.

The textbook *VLSI Array Processors* by S. Y. Kung discusses the concept of systolic arrays at length. The textbook *Software Radio: A Modern Approach to Radio Engineering* by Jeffrey Reed covers issues engineers must understand in order to utilize DSP in software radio subsystems. The textbook *Digital Signal Processing with Field Programmable Gate Arrays* by Uwe Meyer-Baese discusses how FPGA implementations are revolutionizing digital signal processing and covers implementations of FIR and IIR filters along with several other DSP processing systems on FPGAs.

22

Two-Dimensional FIR Filters

22.1 Introduction **22**-1
22.2 Preliminary Design Considerations **22**-2
Filter Specifications and Approximation Criteria • Zero-Phase FIR Filters and Symmetry Considerations • Guidelines on the Use of the Design Techniques
22.3 General Design Methods for Arbitrary Specifications **22**-6
Design of 2-D FIR Filters by Windowing • Frequency Sampling and Linear Programming-Based Method • FIR Filters Optimal in L_p Norm • Iterative Method for Approximate Minimax Design
22.4 Special Design Procedure for Restricted Classes **22**-16
Separable 2-D FIR Filter Design • Frequency Transformation Method • Design Using Nonrectangular Transformations and Sampling Rate Conversions
22.5 2-D FIR Filter Implementation **22**-24
22.6 Two-Dimensional Filter Banks **22**-25
Acknowledgments **22**-26
References **22**-27
Further Information **22**-29

Rashid Ansari
University of Illinois at Chicago

A. Enis Cetin
Bilkent University

22.1 Introduction

In this chapter, methods of designing two-dimensional (2-D) finite-extent impulse response (FIR) discrete-time filters are described. 2-D FIR filters offer the advantages of phase linearity and guaranteed stability, which makes them attractive in applications. Over the years an extensive array of techniques for designing 2-D FIR filters has been accumulated [14,23,30]. These techniques can be conveniently classified into the two categories of general and specialized designs. Techniques in the category of general design are intended for approximation of *arbitrary* desired frequency responses usually with no structural constraints on the filter. These techniques include approaches such as windowing of the ideal impulse response (IIR) [22] or the use of suitable optimality criteria possibly implemented with iterative algorithms. On the other hand, techniques in the category of special design are applicable to restricted classes of filters, either due to the nature of the response being approximated or due to imposition of structural constraints on the filter used in the design. The specialized designs are a consequence of the observation that commonly used filters have characteristic underlying features that can be exploited to simplify the problem of design and implementation. The stopbands and passbands of filters encountered

in practice are often defined by straight-line, circular, or elliptical boundaries. Specialized design methodologies have been developed for handling these cases and they are typically based on techniques such as the transformation of one-dimensional (1-D) filters or the rotation and translation of separable filter responses. If the desired response possesses symmetries, then the symmetries imply relationships among the filter coefficients, which are exploited in both the design and the implementation of the filters. In some design problems it may be advantageous to impose structural constraints in the form of parallel and cascade connections.

The material in this chapter is organized as follows. A preliminary discussion of characteristics of 2-D FIR filters and issues relevant to the design methods appears in Section 22.2. Following this, methods of general and special FIR filter design are described in Sections 22.3 and 22.4, respectively. Several examples of design illustrating the procedure are also presented. Issues in 2-D FIR filter implementation are briefly discussed in Section 22.5. Finally, two-dimensional filter banks are briefly discussed in Section 22.6, and a list of sources for further information is provided.

22.2 Preliminary Design Considerations

In any 2-D filter design there is a choice between FIR and IIR filters, and their relative merits are briefly examined next. 2-D FIR filters offer certain advantages over 2-D IIR filters as a result of which FIR filters have found widespread use in applications such as image and video processing. One key attribute of an FIR filter is that it can be designed with strictly linear passband phase, and it can be implemented with small delays without the need to reverse the signal array during processing. A 2-D FIR filter impulse response has only a finite number of nonzero samples which guarantee stability. On the other hand, stability is difficult to test in the case of 2-D IIR filters due to the absence of a 2-D counterpart of the fundamental theorem of algebra, and a 2-D polynomial is almost never factorizable. If a 2-D FIR filter is implemented nonrecursively with finite precision, then it does not exhibit limit cycle oscillations. Arithmetic quantization noise and coefficient quantization effects in FIR filter implementation are usually very low. A key disadvantage of FIR filters is that they typically have higher computational complexity than IIR filters for meeting the same magnitude specifications, especially in cases where the specifications are stringent.

The term 2-D FIR filter refers to a linear shift-invariant system whose input–output relation is represented by a convolution [14]

$$y(n_1, n_2) = \sum_{(k_1,k_2)\in} \sum_{I} h(k_1, k_2)x(n_1 - k_1, n_2 - k_2), \tag{22.1}$$

where

$x(n_1, n_2)$ and $y(n_1, n_2)$ are the input and the output sequences, respectively
$h(n_1, n_2)$ is the impulse response sequence
I is the support of the impulse response sequence

FIR filters have compact support, meaning that only a finite number of coefficients are nonzero. This makes the impulse response sequence of FIR filters absolutely summable, thereby ensuring filter stability. Usually the filter support, I, is chosen to be a rectangular region centered at the origin, e.g., $I = \{(n_1, n_2)$: $-N_1 \le n_1 \le N_1$, $-N_2 \le n_2 \le N_2\}$. However, there are some important cases where it is more advantageous to select a nonrectangular region as the filter support [32].

Once the extent of the impulse response support is determined, the sequence $h(n_1, n_2)$ should be chosen in order to meet given filter specifications under suitable approximation criteria. These aspects are elaborated in Section 22.2.1. This is followed by a discussion of phase linearity and filter response symmetry considerations and then some guidelines on using the design methods are provided (Sections 22.2.2 and 22.2.3).

22.2.1 Filter Specifications and Approximation Criteria

The problem of designing a 2-D FIR filter consists of determining the impulse response sequence, $h(n_1, n_2)$, or its system function, $H(z_1, z_2)$, in order to satisfy given requirements on the filter response. The filter requirements are usually specified in the frequency domain, and only this case is considered here. The frequency response,* $H(\omega_1, \omega_2)$, corresponding to the impulse response $h(n_1, n_2)$, with a support, I, is expressed as

$$H(\omega_1, \omega_2) = \sum_{(n_1, n_2) \in I} h(n_1, n_2) e^{-j(n_1\omega_1 + n_2\omega_2)}. \tag{22.2}$$

Note that $H(\omega_1, \omega_2) = H(\omega_1 + 2\pi, \omega_2) = H(\omega_1, \omega_2 + 2\pi)$ for all (ω_1, ω_2). In other words, $H(\omega_1, \omega_2)$ is a periodic function with a period 2π in both ω_1 and ω_2. This implies that by defining $H(\omega_1, \omega_2)$ in the region $\{-\pi < \omega_1 \leq \pi, -\pi < \omega_2 \leq \pi\}$, the frequency response of the filter for all (ω_1, ω_2) is determined.

For 2-D FIR filters the specifications are usually given in terms of the magnitude response, $|H(\omega_1, \omega_2)|$. Attention in this chapter is confined to the case of a two-level magnitude design, where the desired magnitude levels are either 1.0 (in the passband) or 0.0 (in the stopband). Some of the procedures can be easily modified to accommodate multilevel magnitude specifications, as, for instance, in a case that requires the magnitude to increase linearly with distance from the origin in the frequency domain.

Consider the design of a 2-D FIR low-pass filter whose specifications are shown in Figure 22.1. The magnitude of the low-pass filter ideally takes the value 1.0 in the passband region, F_p, which is centered around the origin, $(\omega_1, \omega_2) = (0, 0)$, and 0.0 in the stopband region, F_s. As a magnitude discontinuity is not possible with a finite filter support, I, it is necessary to interpose a transition region, F_t, between F_p and F_s. Also, magnitude bounds $|H(\omega_1, \omega_2) - 1| \leq \delta_p$ in the passband and $|H(\omega_1, \omega_2)| \leq \delta_s$ in the stopband are specified, where the parameters δ_p and δ_s are positive real numbers, typically much less than 1.0. The frequency response $H(\omega_1, \omega_2)$ is assumed to be real. Consequently, the low-pass filter is specified in the frequency domain by the regions, F_p, F_s, and the tolerance parameters, δ_p and δ_s. A variety of stopband and passband shapes can be specified in a similar manner.

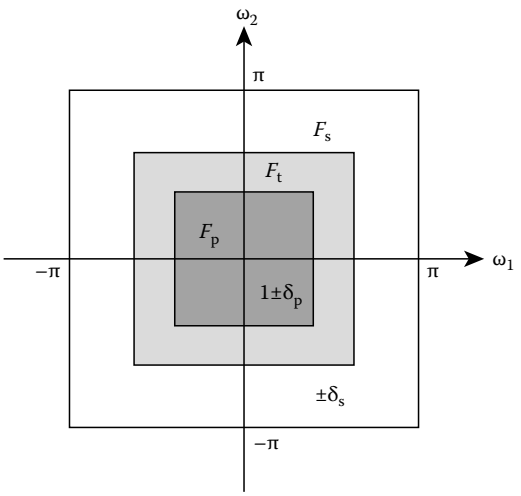


FIGURE 22.1 Frequency response specifications for a 2D low-pass filter ($|H(\omega_1, \omega_2) - 1| \leq \delta_p$ for $(\omega_1, \omega_2) \in F_p$ and $|H(\omega_1, \omega_2)| \leq \delta_s$ for $(\omega_1, \omega_2) \in F_s$).

In order to meet given specifications, an adequate filter order (defined here to be the number of nonzero impulse response samples) needs to be determined. If the specifications are stringent, with tight tolerance parameters and small transition regions, then the filter support region, I, must be large. In other words, there is a trade-off between the filter support region, I, and the frequency domain specifications. In the general case the filter order is not known *a priori*, and may be determined either through an iterative process or using estimation rules if available. If the filter order is given, then in order to determine an optimum solution to the design problem, an appropriate optimality criterion is needed. Commonly used criteria in 2-D filter design are minimization of the L_p norm, p finite, of the approximation error, or the L_∞ norm. If desired, a maximal flatness requirement at desired frequencies can be imposed

* Here $\omega_1 = 2\pi f_1$ and $\omega_2 = 2\pi f_2$ are the horizontal and vertical angular frequencies, respectively.

[24]. It should be noted that if the specifications are given in terms of the tolerance bounds on magnitude, as described above, then the use of L_∞ criterion is appropriate. However, the use of other criteria such as a weighted L_2 norm can serve to arrive at an almost minimax solution [2,10].

22.2.2 Zero-Phase FIR Filters and Symmetry Considerations

Phase linearity is important in many filtering applications. As in the 1-D case, a number of conditions for phase linearity can be obtained depending on the nature of symmetry. But the discussion here is limited to the case of ''zero phase'' design, with a purely real frequency response. A salient feature of 2-D FIR filters is that realizable FIR filters, which have purely real frequency responses, are easily designed. The term ''zero phase'' is somewhat misleading in the sense that the frequency response may be negative at some frequencies. The term should be understood in the sense of ''zero phase in passband'' because the passband frequency response is within a small deviation of the value 1.0. The frequency response may assume negative values in the stopband region where phase linearity is immaterial. In frequency domain, the zero-phase or real frequency response condition corresponds to

$$H(\omega_1, \omega_2) = H^*(\omega_1, \omega_2), \tag{22.3}$$

where $H^*(\omega_1, \omega_2)$ denotes the complex conjugate of $H(\omega_1, \omega_2)$. The condition Equation 22.3 is equivalent to

$$h(n_1, n_2) = h^*(-n_1, -n_2) \tag{22.4}$$

in the spatial-domain. Making a common practical assumption that $h(n_1, n_2)$ is real, the above condition reduces to

$$h(n_1, n_2) = h(-n_1, -n_2), \tag{22.5}$$

implying a region of support with the above symmetry about the origin.

Henceforth, only the design of zero-phase FIR filters is considered. With $h(n_1, n_2)$ real, and satisfying Equation 22.5, the frequency response, $H(\omega_1, \omega_2)$, is expressed as

$$\begin{aligned} H(\omega_1, \omega_2) &= h(0,0) + \sum_{(n_1, n_2) \in I_1} h(n_1, n_2) e^{-j(\omega_1 n_1 + \omega_2 n_2)} + \sum_{(n_1, n_2) \in I_2} h(n_1, n_2) e^{-j(\omega_1 n_1 + \omega_2 n_2)} \\ &= h(0,0) + \sum_{(n_1, n_2) \in I_1} 2h(n_1, n_2) \cos(\omega_1 n_1 + \omega_2 n_2), \end{aligned} \tag{22.6}$$

where I_1 and I_2 are disjoint regions such that $I_1 \cup I_2 \cup \{(0,0)\} = I$, and if $(n_1, n_2) \in I_1$, then $(-n_1, -n_2) \in I_2$.

In order to understand the importance of phase linearity in image processing, consider an example that illustrates the effect of nonlinear-phase filters on images. In Figure 22.2a, an image that is corrupted by white Gaussian noise is shown. This image is filtered with a nonlinear-phase low-pass filter and the resultant image is shown in Figure 22.2b. It is observed that edges and textured regions are severely distorted in Figure 22.2b. This is due to the fact that the spatial alignment of frequency components that define an edge in the original is altered by the phase nonlinearity. The same image is also filtered with a zero-phase low-pass filter, $H(\omega_1, \omega_2)$, which has the same magnitude characteristics as the nonlinear-phase filter. The resulting image is shown in Figure 22.2c. It is seen that the edges are perceptually preserved in Figure 22.2c, although blurred due to the low-pass nature of the filter. In this example, a separable zero-phase low-pass filter, $H(\omega_1, \omega_2) = H_1(\omega_1)\, H_1(\omega_2)$, is used, where $H_1(\omega)$ is a 1-D Lagrange filter with a cutoff $\pi/2$. In spatial domain $h(n_1, n_2) = h_1(n_1)\, h_1(n_2)$ where $h_1(n) = \{\ldots, 0, -1/32, 0, 9/32,$

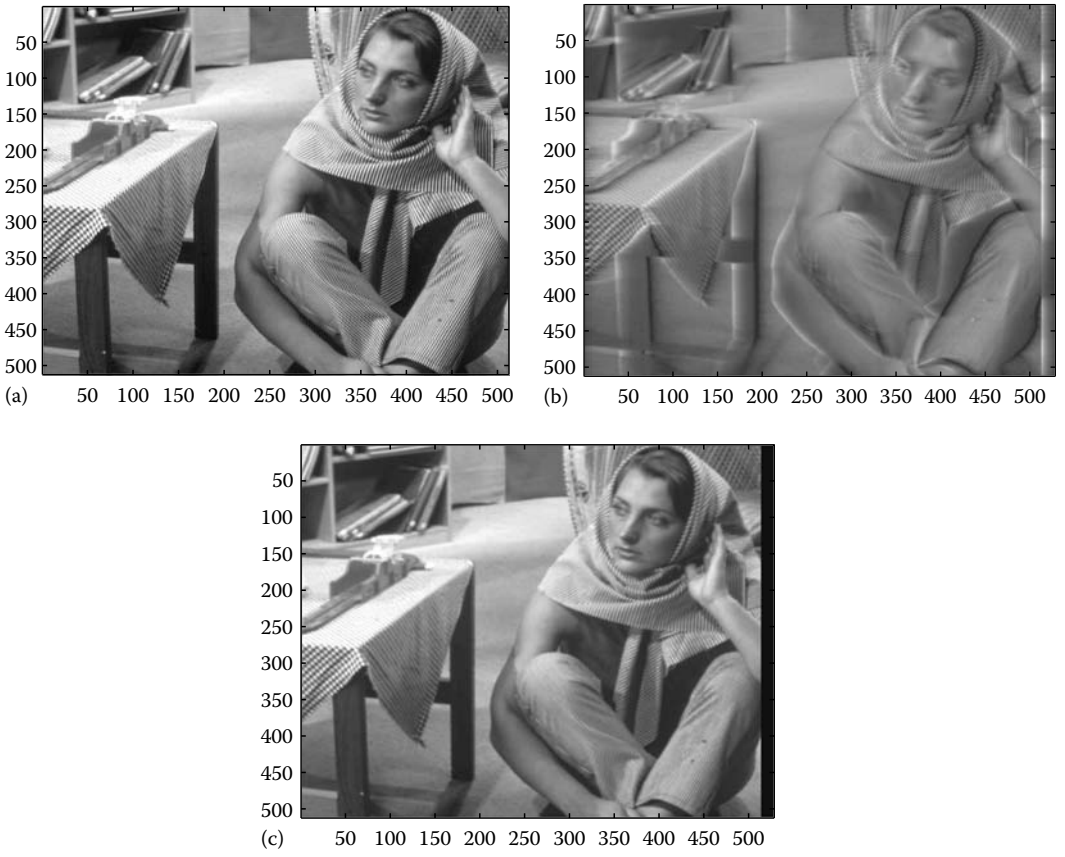


FIGURE 22.2 (a) Original image of 696×576 pixels; (b) nonlinear-phase low-pass filtered image; and (c) zero-phase low-pass filtered image.

$1/2, 9/32, 0, -1/32, 0, \ldots\}$ is the impulse response of the seventh-order symmetric (zero-phase) 1-D Lagrange filter. The nonlinear-phase filter is a cascade of the above zero-phase filter with an allpass filter.

In some filter design problems, symmetries in frequency domain specifications can be exploited by imposing restrictions on the filter coefficients and the shape of the support region for the impulse response. A variety of symmetries that can be exploited is extensively studied in Refs. [3,32,44,45]. For example, a condition often encountered in practice is the symmetry with respect to each of the two frequency axes. In this case, the frequency response of a zero-phase filter satisfies

$$H(\omega_1, \omega_2) = H(-\omega_1, \omega_2) = H(\omega_1, -\omega_2). \tag{22.7}$$

This yields an impulse response that is symmetric with respect to the n_1 and n_2 axes, i.e.,

$$h(n_1, n_2) = h(-n_1, n_2) = h(n_1, -n_2). \tag{22.8}$$

By imposing symmetry conditions, one reduces the number of independently varying filter coefficients that must be determined in the design. This can be exploited in reducing both the computational complexity of the filter design and the number of arithmetic operations required in the implementation.

22.2.3 Guidelines on the Use of the Design Techniques

The design techniques described in this chapter are classified into the two categories of general and specialized designs. The user should use the techniques of general design in cases requiring approximation of arbitrary desired frequency responses, usually with no structural constraints on the filter. The specialized designs are recommended in cases where filters exhibit certain underlying features that can be exploited to simplify the problem of design and implementation.

In the category of general design, four methods are described. Of these, the windowing procedure is quick and simple. It is useful in situations where implementation efficiency is not critical, especially in single-use applications. The second procedure is based on linear programming, and is suitable for design problems where equiripple solutions are desired to meet frequency domain specifications. The remaining two procedures may also be used for meeting frequency domain specifications, and lead to nearly equiripple solution. The third procedure provides solutions for L_p approximations. The fourth procedure is an iterative procedure that is easy to implement, and is convenient in situations where additional constraints are to be placed on the filter.

In the category of specialized design described here, the solutions are derived from 1-D filters. These often lead to computationally efficient implementation, and are recommended in situations where low implementation complexity is critical, and the filter characteristics possess features that can be exploited in the design. An important practical class of filters is one where specifications can be decomposed into a set of separable filter designs requiring essentially the design of suitable 1-D filters. Here the separable design procedure should be used. Another class of filters is one where the passbands and stopbands are characterized by circular, elliptical, or special straight-line boundaries. In this case a frequency transformation method called the McClellan transformation procedure proves effective. The desired 2-D filter constant-magnitude contours are defined by a proper choice of parameters in a transformation of variables applied to a 1-D zero-phase filter. Finally, in some cases filter specifications are characterized by ideal frequency responses in which passbands and stopbands are separated by straight-line boundaries that are not suitable for applying the McClellan transformation procedure. In this case the design may be carried out by nonrectangular transformations and sampling grid conversions. The importance of this design method stems from the implementation efficiency that results from a generalized notion of separable processing.

22.3 General Design Methods for Arbitrary Specifications

Some general methods of meeting arbitrary specifications are now described. These are typically based on extending techniques of 1-D design. However, there are important differences. The Parks–McClellan procedure for minimax approximation based on the alternation theorem does not find a direct extension. This is because the set of cosine functions used in the 2-D approximation does not satisfy the Haar condition on the domain of interest [25], and the Chebyshev approximation does not have a unique solution. However, techniques that employ exchange algorithms have been developed for the 2-D case [20,25,36].

Here we consider four procedures in some detail. The first technique is based on windowing. It is simple, but not optimum for Chebyshev approximation. The second technique is based on frequency sampling, and this can be used to arrive at equiripple solutions using linear programming. Finally, two techniques for arriving iteratively at a nearly equiripple solution are described. The first of these is based on L_p approximations using nonlinear optimization. The second is based on the use of alternating projections in the sample and the frequency domains.

22.3.1 Design of 2-D FIR Filters by Windowing

This design method is basically an extension of the window-based 1-D FIR filter design to the case of 2-D filters. An IIR sequence, which is usually an infinite-extent sequence, is suitably windowed to make the

support finite. 1-D FIR filter design by windowing and classes of 1-D windows are described in detail in Section 22.2.

Let $h_{\mathrm{id}}(n_1, n_2)$ and $H_{\mathrm{id}}(\omega_1, \omega_2)$ be the impulse and frequency responses of the ideal filter, respectively. The impulse response of the required 2-D filter, $h(n_1, n_2)$, is obtained as a product of the IIR sequence and a suitable 2-D window sequence which has a finite-extent support, I, that is,

$$h(n_1, n_2) = \begin{cases} h_{id}(n_1, n_2)w(n_1, n_2), & (n_1, n_2) \in I, \\ 0, & \text{otherwise}, \end{cases} \tag{22.9}$$

where $w(n_1, n_2)$ is the window sequence. The resultant frequency response, $H(\omega_1, \omega_2)$, is a smoothed version of the ideal frequency response as $H(\omega_1, \omega_2)$ is related to the $H_{\mathrm{id}}(\omega_1, \omega_2)$ via the periodic convolution, that is,

$$H(\omega_1, \omega_2) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H_{\mathrm{id}}(\Omega_1, \Omega_2) W(\omega_1 - \Omega_1, \omega_2 - \Omega_2) \mathrm{d}\Omega_1 \mathrm{d}\Omega_2, \tag{22.10}$$

where $W(\omega_1, \omega_2)$ is the frequency response of the window sequence, $w(n_1, n_2)$.

As in the 1-D case, a 2-D window sequence, $w(n_1, n_2)$, should satisfy three requirements:

1. It must have a finite-extent support, I.
2. Its discrete-space Fourier transform should in some sense approximate the 2-D impulse function, $\delta(\omega_1, \omega_2)$.
3. It should be real, with a zero-phase discrete-space Fourier transform.

Usually 2-D windows are derived from 1-D windows. Three methods of constructing windows are briefly examined. One method consists of obtaining a separable window from two 1-D windows, that is,

$$w_{\mathrm{r}}(n_1, n_2) = w_1(n_1)w_2(n_2), \tag{22.11}$$

where $w_1(n)$ and $w_2(n)$ are the 1-D windows. Thus, the support of the resultant 2-D window, $w_{\mathrm{r}}(n_1, n_2)$, is a rectangular region. The frequency response of the 2-D window is also separable, i.e., $W_{\mathrm{r}}(\omega_1, \omega_2) = W_1(\omega_1)\ W_2(\omega_2)$.

The second method of constructing a window, due to Huang [22], consists of sampling the surface generated by rotating a 1-D continuous-time window, $w(t)$, as follows:

$$w_{\mathrm{c}}(n_1, n_2) = w\left(\sqrt{n_1^2 + n_2^2}\right), \tag{22.12}$$

where $w(t) = 0$, $t \geq N$. The impulse response support is $I = \left\{n_1, n_2 \colon \sqrt{n_1^2 + n_2^2} < N\right\}$. Note that the 2-D Fourier transform of the $w_{\mathrm{c}}(n_1, n_2)$ is not equal to the circularly rotated version of the Fourier transform of $w(t)$.

Finally, in the third method, proposed by Yu and Mitra [53], the window is constructed by using a 1-D to 2-D transformation belonging to a class called the McClellan transformations [33]. These transformations are discussed in greater detail in Section 22.4. Here we consider a special case of the transform that produces approximately circular contours in the 2-D frequency domain. Briefly, the discrete-space frequency transform of the 2-D window sequence obtained with a McClellan transformation applied to a 1-D window is given by

$$T(\omega_1, \omega_2) = \sum_{n=-N}^{N} w(n) e^{-j\omega n} \Big|_{\cos(\omega)=0.5\cos(\omega_1)+0.5\cos(\omega_2)+0.5\cos(\omega_1)\cos(\omega_2)-0.5}$$
$$= w(0) + 2\sum_{n=1}^{N} w(n)\cos(n\omega) \Big|_{\cos(\omega)=0.5\cos(\omega_1)+0.5\cos(\omega_2)+0.5\cos(\omega_1)\cos(\omega_2)-0.5}$$
$$= \sum_{n=0}^{N} b(n)\cos^n(\omega) \Big|_{\cos(\omega)=0.5\cos(\omega_1)+0.5\cos(\omega_2)+0.5\cos(\omega_1)\cos(\omega_2)-0.5} \tag{22.13}$$

where $w(n)$ is an arbitrary symmetric 1-D window of duration $2N+1$ centered at the origin, and the coefficients, $b(n)$, are obtained from $w(n)$ via Chebyshev polynomials [33]. After some algebraic manipulations it can be shown that

$$T(\omega_1, \omega_2) = \sum_{n_1=-N}^{N} \sum_{n_2=-N}^{N} w_t(n_1, n_2) e^{-j(n_1\omega_1 + n_2\omega_2)}, \tag{22.14}$$

where $w_t(n_1, n_2)$ is a zero-phase 2-D window of size $(2N+1) \times (2N+1)$ obtained by using the McClellan transformation.

The construction of 2-D windows using the above three methods is now examined. In the case of windows obtained by the separable and the McClellan transformation approaches, the 1-D prototype is a Hamming window,

$$w_h(n) = \begin{cases} 0.54 + 0.46\cos(\pi n/N), & |n| < N, \\ 0, & \text{otherwise.} \end{cases} \tag{22.15}$$

In the second case $w_c(n_1, n_2) = 0.54 + 0.46\cos(\pi\sqrt{n_1^2 + n_2^2}/N)$. By selecting $w_1(n) = w_2(n) = w_h(n)$ in Equation 22.11 we get a 2-D window, $w_r(n_1, n_2)$, of support $I = \{|n_1| < N,\ |n_2| < N\}$ which is a square-shaped symmetric region centered at the origin. For $N = 6$ the region of support, I, contains $11 \times 11 = 121$ points. Figure 22.3a shows the frequency response of this window. A second window is designed by using Equation 22.12, i.e., $w_c(n_1, n_2) = w_h(\sqrt{n_1^2 + n_2^2})$. For $N = 6$ the frequency response of this filter is shown in Figure 22.3b. The region of support is almost circular and it contains 113 points. From these examples, it is seen that the 2-D windows may not behave as well as 1-D windows. Speake and Mersereau [46] compared these two methods and observed that the main-lobe width and the highest attenuation level of the side-lobes of the 2-D windows differ from their 1-D prototypes.

Let us construct a 2-D window by the McClellan transformation with a 1-D Hamming window of order 13 ($N = 6$) as the prototype. The frequency response of the 2-D window, $w_t(n_1, n_2)$, is shown in Figure 22.3c. The frequency response of this window is almost circularly symmetric and it preserves the features of its 1-D prototype.

Consider the design of a circularly symmetric low-pass filter. The ideal frequency response for $(\omega_1, \omega_2) \in [-\pi, \pi] \times [-\pi, \pi]$ is given by

$$H_{id}(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{\omega_1^2 + \omega_2^2} \le \omega_c, \\ 0, & \text{otherwise,} \end{cases} \tag{22.16}$$

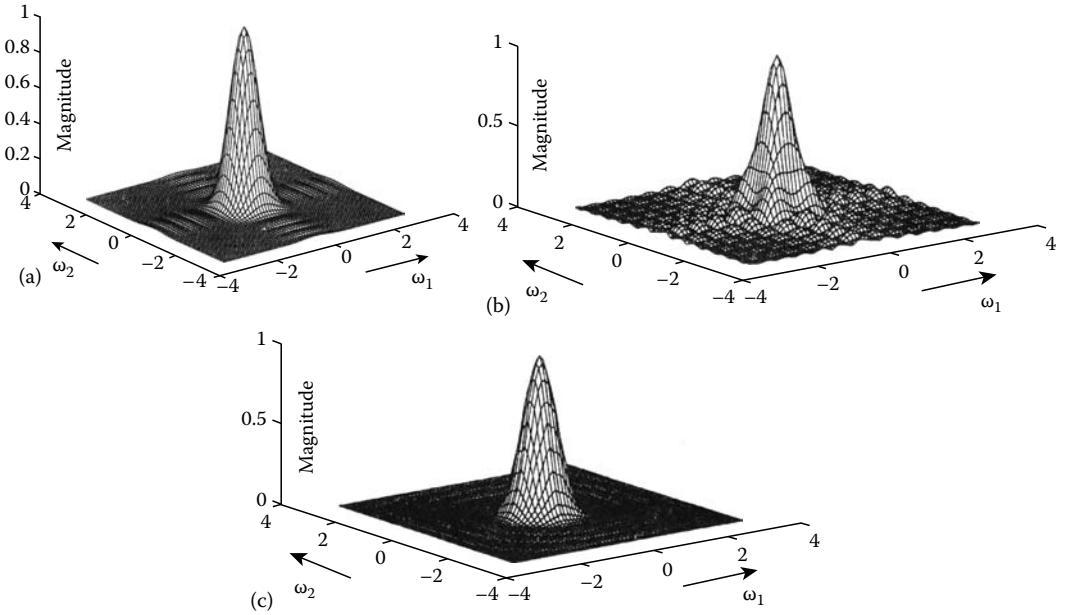


FIGURE 22.3 Frequency responses of the (a) separable, (b) Huang, and (c) McClellan 2D windows generated from a Hamming window of order 13 ($N = 6$).

whose impulse response is given by

$$h_{\text{id}}(n_1, n_2) = \frac{\omega_c J_1\left(\omega_c \sqrt{n_1^2 + n_2^2}\right)}{2\pi\sqrt{n_1^2 + n_2^2}}, \tag{22.17}$$

where

$J_1(\cdot)$ is the first-order Bessel function of the first kind
ω_c is the cutoff frequency

The frequency response of the 2-D FIR filter obtained with a rectangular window of size $2 \times 5 + 1$ by $2 \times 5 + 1$ is shown in Figure 22.4a. Note the Gibbs-phenomenon type ripples at the passband edges.

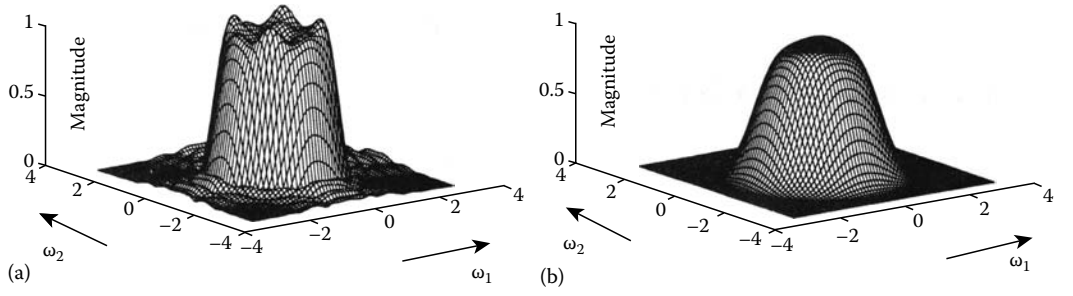


FIGURE 22.4 Frequency responses of the 2D filters designed with (a) a rectangular window and (b) a separable window of Figure 22.3a.

In Figure 22.4b the separable window of Figure 22.3a, derived from a Hamming window, is used to design the 2-D filter. Note that this 2-D filter has smaller ripples at the passband edges.

In windowing methods, it is often assumed that $H_{\mathrm{id}}(\omega_1, \omega_2)$ is given. However, if the specifications are given as described in Section 22.2, then a proper $H_{\mathrm{id}}(\omega_1, \omega_2)$ should be constructed. The ideal magnitudes are either 1.0 (in passband) or 0.0 (in stopband). However, there is a need to define a *cutoff* boundary, which lies within the transition band. This can be accomplished by using a suitable notion of "midway" cutoff between the transition boundaries. In practical cases where transition boundaries are given in terms of straight-line segments or smooth curves such as circles and ellipses, the construction of "midway" cutoff boundary is relatively straightforward. The IIR, $h_{\mathrm{id}}(n_1, n_2)$, is computed from the desired frequency response, $H_{\mathrm{id}}(\omega_1, \omega_2)$, either analytically (if possible), or by using the discrete Fourier transform (DFT). In the latter case the desired response, $H_{\mathrm{id}}(\omega_1, \omega_2)$, is first sampled on a rectangular grid in the Fourier domain, then an inverse DFT computation is carried out via a 2-D fast Fourier transform (FFT) algorithm to obtain an approximation to the sequence $h_{\mathrm{id}}(n_1, n_2)$. The resulting sequence is an aliased version of the IIR. Therefore, a sufficiently dense grid should be used in order to reduce the effects of aliasing.

In practice, several trials may be needed to design the final filter satisfying bounds both in the passbands and stopbands. The filter support is adjusted to obtain the smallest order to meet given requirements.

Filter design with windowing is a simple approach that is suitable for applications where a quick and nonoptimal design is needed. Additional information on windowing can be found in Refs. [26,46].

22.3.2 Frequency Sampling and Linear Programming-Based Method

This method is based on the application of the sampling theorem in the frequency domain. Consider the design of a 2-D filter with impulse response support of $N_1 \times N_2$ samples. The frequency response of the filter can be obtained from a conveniently chosen set of its samples on an $N_1 \times N_2$ grid. For example, the DFT of the impulse response can be used to interpolate the response for the entire region $[0, 2\pi] \times [0, 2\pi]$. The filter design then becomes a problem of choosing an appropriate set of DFT coefficients [21].

One choice of DFT coefficients consists of the ideal frequency response values, assuming a suitable cutoff. However, the resultant filters usually exhibit large magnitude deviations away from the DFT sample locations in the filter passbands and stopbands. The approximation error can be reduced by allowing the DFT values in the transition band to vary, and choosing them to minimize the deviation of the magnitude from the desired values. Another option is to allow all the DFT values to vary, and pick the optimal set of values for minimum error. The use of DFT-based interpolation allows for a computationally efficient implementation. The implementation cost of the method basically consists of a 2-D array product and inverse discrete Fourier transform (IDFT) computation, with appropriate addition.

Let us consider the set $S \subset Z^2$ that defines the equispaced frequency locations $\left(\frac{2k_1\pi}{N_1}, \frac{2k_2\pi}{N_2}\right)$:

$$S = \{(k_1, k_2): k_1 = 0, 1, \ldots, N_1, k_2 = 0, 1, \ldots, N_2\}. \tag{22.18}$$

The DFT values can be expressed as

$$H_{\mathrm{DFT}}[k_1, k_2] = |H(\omega_1, \omega_2)|_{(\omega_1, \omega_2) = \left(\frac{2k_1\pi}{N_1}, \frac{2k_2\pi}{N_2}\right)}, \ (k_1, k_2) \in S. \tag{22.19}$$

The filter coefficients, $h(n_1, n_2)$, are found by using an IDFT computation

$$h(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} H_{\mathrm{DFT}}[k_1, k_2] e^{j\left(\frac{2\pi}{N_1}k_1 n_1 + \frac{2\pi}{N_2}k_2 n_2\right)}, \ (n_1, n_2) \in S. \tag{22.20}$$

If Equation 22.20 is substituted in the expression for frequency response

$$H(\omega_1, \omega_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} h(n_1, n_2) e^{-j(\omega_1 n_1 + \omega_2 n_2)}, \tag{22.21}$$

we arrive at the interpolation formula

$$H(\omega_1, \omega_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} H_{\mathrm{DFT}}[k_1, k_2] A_{k_1 k_2}(\omega_1, \omega_2), \tag{22.22}$$

where

$$A_{k_1 k_2}(\omega_1, \omega_2) = \frac{1}{N_1 N_2} \left(\frac{1 - e^{-jN_1\omega_1}}{1 - e^{-j(\omega_1 - 2\pi k_1/N_1)}} \right) \left(\frac{1 - e^{-jN_2\omega_2}}{1 - e^{-j(\omega_2 - 2\pi k_2/N_2)}} \right). \tag{22.23}$$

Equation 22.22 serves as the basis of the frequency sampling design. As mentioned before, if the H_{DFT} are chosen directly according to the ideal response, then the magnitude deviations are usually large. To reduce the ripples, one option is to express the set S as the disjoint union of two sets S_t and S_c, where S_t contains indices corresponding to the transition band F_t, and S_c contains indices corresponding to the care-bands, i.e., the union of the passbands and stopbands, $F_p \cup F_s$. The expression for frequency response in Equation 22.22 can be split into two summations, one over S_t and the other over S_c

$$H(\omega_1, \omega_2) = \sum_{S_t} H_{\mathrm{DFT}}[k_1, k_2] A_{k_1 k_2}(\omega_1, \omega_2) + \sum_{S_c} H_{\mathrm{DFT}}[k_1, k_2] A_{k_1 k_2}(\omega_1, \omega_2), \tag{22.24}$$

where the first term on the right-hand side is optimized. The design equations can be put in the form

$$1 - \alpha\delta \le H(\omega_1, \omega_2) \le 1 + \alpha\delta, \quad (\omega_1, \omega_2) \in F_p \tag{22.25}$$

and

$$-\delta \le H(\omega_1, \omega_2) \le \delta, \quad (\omega_1, \omega_2) \in F_s, \tag{22.26}$$

where

δ is the peak approximation error in the stopband

$\alpha\delta$ is the peak approximation error in the passband, where α is any positive constant defining the relative weights of the deviations

The problem is readily cast as a linear programming problem with a sufficiently dense grid of points.

For equiripple design, all the DFT values H_{DFT} over S_t and S_c are allowed to vary. Following is an example of this design.

Example 22.1

The magnitude response for the approximation of a circularly symmetric response is shown in Figure 22.5. Here the passband is the interior of the circle $R_1 = \pi/3$ and the stopband is the exterior of the circle $R_2 = 2\pi/3$. With $N_1 = N_2 = 9$, the passband ripple is 0.08 dB and the minimum stopband attenuation is 32.5 dB.

22.3.3 FIR Filters Optimal in L_p Norm

A criterion different from the minimax criterion is briefly examined. Let us define the error at the frequency pair (ω_1, ω_2) as follows:

$$E(\omega_1, \omega_2) = H(\omega_1, \omega_2) - H_{\mathrm{id}}(\omega_1, \omega_2). \tag{22.27}$$



$R_1 = 1.5\pi/4.5$
$R_2 = 3\pi/4.5$
$N_1 = N_2 = 9$
Amplitude response



Log magnitude response
In-band ripple = 0.08
Peak attenuation = 32.5 dB

FIGURE 22.5 Frequency response of the circularly symmetric filter obtained by using the frequency sampling method. (From Hu, J.V. and Rabiner, L.R., *IEEE Trans. Audio Electroacoust.*, 20, 249, 1972. With permission. © 1972 IEEE.)

One design approach is to minimize the L_p norm of the error

$$\varepsilon_p = \left(\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} |E(\omega_1, \omega_2)|^p \, d\omega_1 d\omega_2 \right)^{\frac{1}{p}}. \tag{22.28}$$

Filter coefficients are selected by a suitable algorithm. For $p = 2$ Parseval's relation implies that

$$\varepsilon_2^2 = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} [h(n_1, n_2) - h_{\text{id}}(n_1, n_2)]^2. \tag{22.29}$$

By minimizing Equation 22.29 with respect to the filter coefficients, $h(n_1, n_2)$, which are nonzero only in a finite-extent region, I, one gets

$$h(n_1, n_2) = \begin{cases} h_{\text{id}}(n_1, n_2), & (n_1, n_2) \in I, \\ 0, & \text{otherwise}, \end{cases} \tag{22.30}$$

which is the filter designed by using a straightforward rectangular window. Due to the Gibbs phenomenon it may have large variations at the edges of passband and stopband regions. A suitable weighting function can be used to reduce the ripple [2], and an approximately equiripple solution can be obtained.

For the general case of $p \neq 2$ [32], the minimization of Equation 22.28 is a nonlinear optimization problem. The integral in Equation 22.28 is discretized and minimized by using an iterative nonlinear optimization technique. The solution for $p = 2$ is easy to obtain using linear equations. This serves as an excellent initial estimate for the coefficients in the case of larger values of p. As p increases, the solution becomes approximately equiripple. The error term, $E(\omega_1, \omega_2)$, in Equation 22.28 is nonuniformly weighted in passbands and stopbands, with larger weight given close to band-edges where deviations are typically larger.

22.3.4 Iterative Method for Approximate Minimax Design

We now consider a simple procedure based on alternating projections in the sample and frequency domains, which leads to an approximately equiripple response. In this method, the zero-phase FIR filter design problem is formulated to alternately satisfy the frequency domain constraints on the magnitude response bounds and spatial domain constraints on the impulse response support [1,11,12]. The algorithm is iterative and each iteration requires two 2-D FFT computations.

As pointed out in Section 22.2, 2-D FIR filter specifications are given as requirements on the magnitude response of the filter. It is desirable that the frequency response, $H(\omega_1, \omega_2)$, of the zero-phase FIR filter be within prescribed upper and lower bounds in its passbands and stopbands. Let us specify bounds on the frequency response $H(\omega_1, \omega_2)$ of the minimax FIR filter, $h(n_1, n_2)$, as follows:

$$H_{\mathrm{id}}(\omega_1, \omega_2) - E_{\mathrm{d}}(\omega_1, \omega_2) \leq H(\omega_1, \omega_2) \leq H_{\mathrm{id}}(\omega_1, \omega_2) + E_{\mathrm{d}}(\omega_1, \omega_2) \quad \omega_1, \omega_2 \in R, \tag{22.31}$$

where

$H_{\mathrm{id}}(\omega_1, \omega_2)$ is the ideal filter response

$E_{\mathrm{d}}(\omega_1, \omega_2)$ is a positive function of (ω_1, ω_2) which may take different values in different passbands and stopbands

R is a region defined in Equation 22.28 consisting of passbands and stopbands of the filter (note that $H(\omega_1, \omega_2)$ is real for a zero-phase filter)

Usually, $E_{\mathrm{d}}(\omega_1, \omega_2)$ is chosen constant in a passband or a stopband. Inequality equation (Equation 22.31) is the frequency domain constraint of the iterative filter design method.

In spatial domain the filter must have a finite-extent support, I, which is symmetric region around the origin. The spatial domain constraint requires that the filter coefficients must be equal to zero outside the region, I.

The iterative method begins with an arbitrary finite-extent, real sequence $h_0(n_1, n_2)$ that is symmetric ($h_0(n_1, n_2) = h_0(-n_1, n_2)$). Each iteration consists of making successive imposition of spatial and frequency domain constraints onto the current iterate. The kth iteration consists of the following steps:

- Compute the Fourier transform of the kth iterate $h_k(n_1, n_2)$ on a suitable grid of frequencies by using a 2-D FFT algorithm.
- Impose the frequency domain constraint as follows:

$$G_k(\omega_1, \omega_2) = \begin{cases} H_{\mathrm{id}}(\omega_1, \omega_2) + E_{\mathrm{d}}(\omega_1, \omega_2) & \text{if } H_k(\omega_1, \omega_2) > H_{\mathrm{id}}(\omega_1, \omega_2) + E_{\mathrm{d}}(\omega_1, \omega_2), \\ H_{\mathrm{id}}(\omega_1, \omega_2) - E_{\mathrm{d}}(\omega_1, \omega_2) & \text{if } H_k(\omega_1, \omega_2) < H_{\mathrm{id}}(\omega_1, \omega_2) - E_{\mathrm{d}}(\omega_1, \omega_2), \\ H_k(\omega_1, \omega_2) & \text{otherwise.} \end{cases} \tag{22.32}$$

- Compute the inverse Fourier transform of $G_k(\omega_1, \omega_2)$.
- Zero out $g_k(n_1, n_2)$ outside the region I to obtain h_{k+1}.

The flow diagram of this method is shown in Figure 22.6. It can be proven that the algorithm converges for all symmetric input sequences. This method requires the specification of the bounds or equivalently, $E_{\mathrm{d}}(\omega_1, \omega_2)$, and the filter support, I. In 2-D filter design, filter order estimates for prescribed frequency domain specifications are not available. Therefore, successive reduction of bounds is used. If the specifications are too tight, then the algorithm does not converge. In such cases one can either progressively enlarge the filter support region, or relax the bounds on the ideal frequency response.

The size of the 2-D FFT must be chosen sufficiently large. The passband and stopband edges are very important for the convergence of the algorithm. These edges must be represented accurately on the frequency grid of the FFT algorithm.

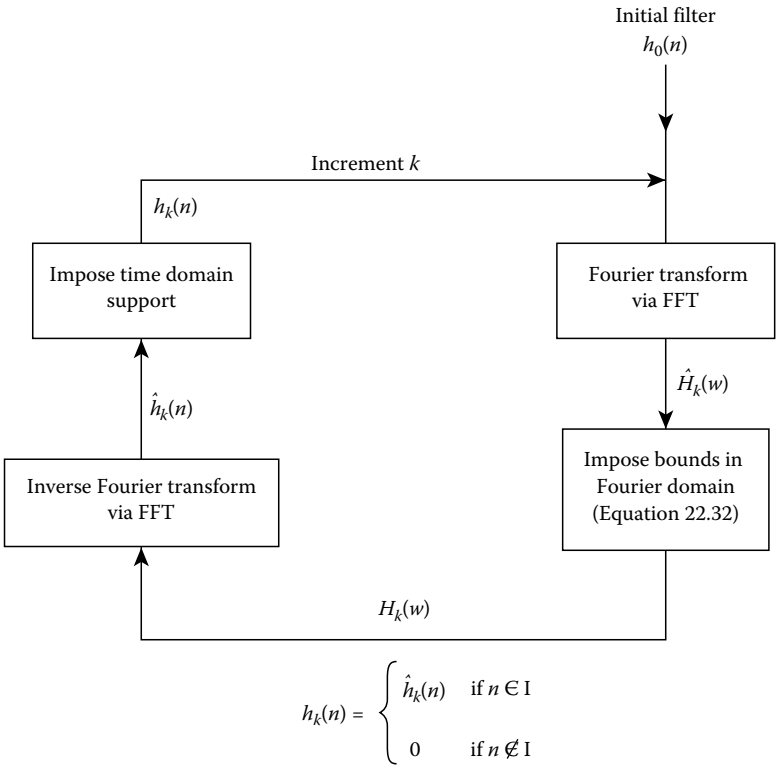


FIGURE 22.6 Flow diagram of the iterative filter design algorithm.

The shape of the filter support is very important in any 2-D filter design method. The support should be chosen to exploit the symmetries in the desired frequency response. For example, diamond-shaped supports show a clear advantage over the commonly assumed rectangular regions in designing diamond filters or 90° fan filters [4,17].

Since there are efficient FFT routines, 2-D FIR filters with large orders can be designed by using this procedure.

Example 22.2

Let us consider the design of a circularly symmetric low-pass filter. Maximum allowable deviation is $\delta_p = \delta_s = 0.05$ in both passband and the stopband. The passband and stopband cutoff boundaries have radii of 0.43 π and 0.63 π, respectively. This means that the functions $E_d(\omega_1, \omega_2) = 0.05$ in the passband and the stopband. In the transition band the frequency response is conveniently bounded by the lower bound of the stopband and the upper bound of the passband. The filter support is a square-shaped 17×17 region. The frequency response of this filter is shown in Figure 22.7.

Example 22.3

Let us now consider an example in which we observe the importance of filter support. We design a fan filter whose specifications are shown in Figure 22.8. Maximum allowable deviation is $\delta_p = \delta_s = 0.1$ in both passband and the stopband. If one uses a 7×7 square-shaped support which has 49 points, then it cannot meet the design specifications. However, a diamond-shaped support,

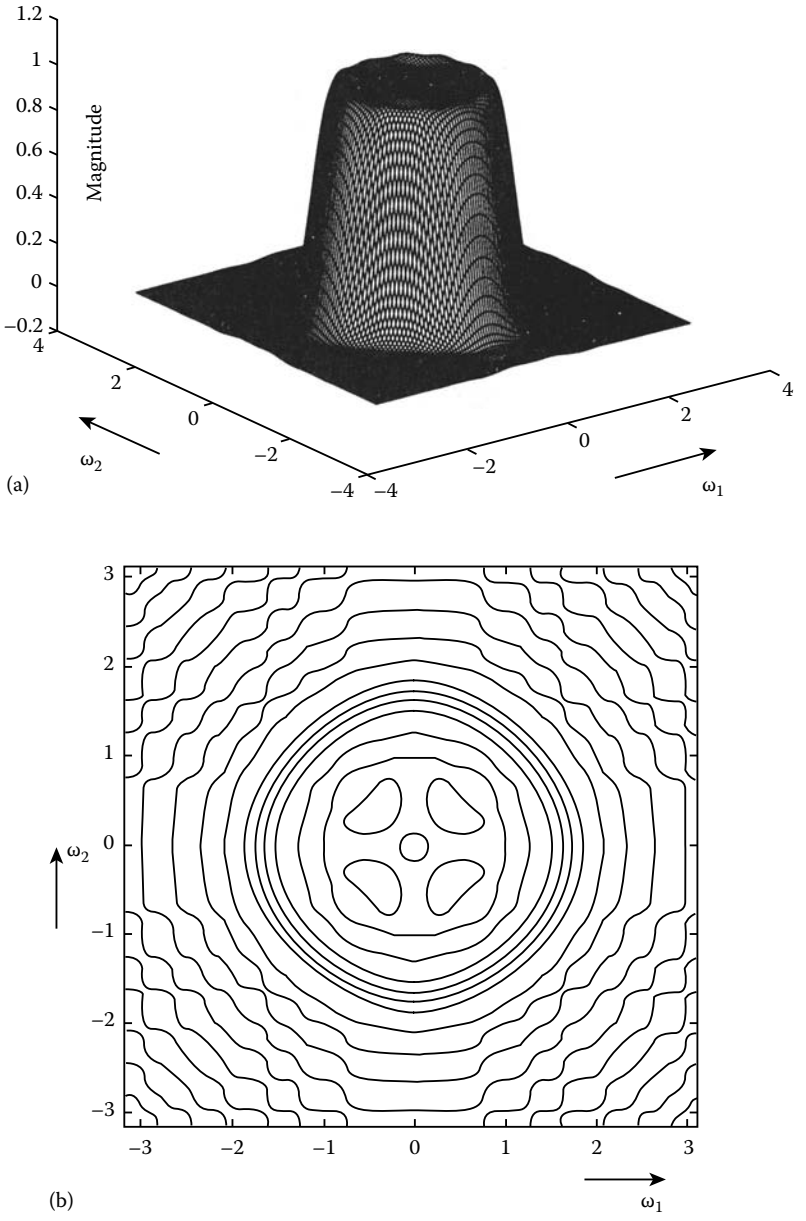


FIGURE 22.7 (a) Frequency response and (b) contour plot of the low-pass filter of Example 22.1.

$$I_d = \{-5 \le n_1 + n_2 \le 5\} \cap \{-5 \le n_1 - n_2 \le 5\}, \tag{22.33}$$

together with the restriction that ("de" for diamond-support with some eliminated samples)

$$I_{de} = I_d \cap \{n_1 + n_2 = \text{odd or } n_1 = n_2 = 0\} \tag{22.34}$$

produces a filter satisfying the bounds. The filter support region, I_{de}, contains 37 points. The resultant frequency response is shown in Figure 22.8.

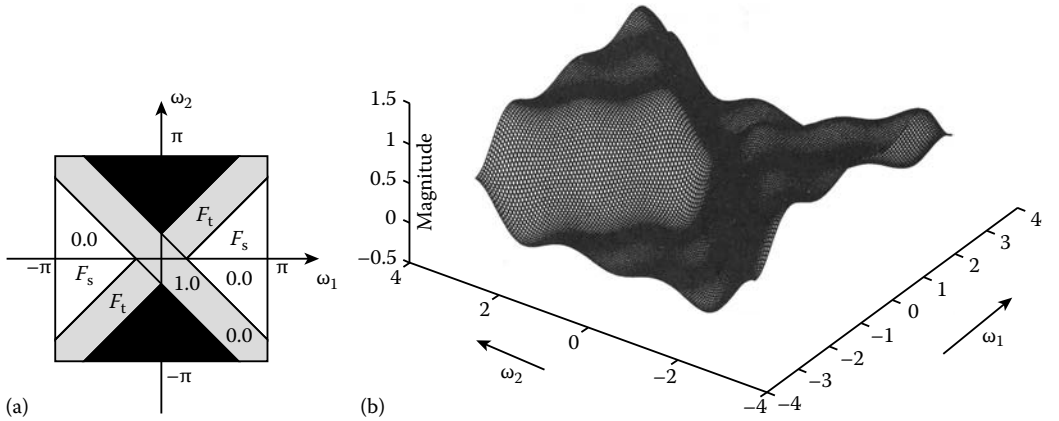


FIGURE 22.8 (a) Specifications and (b) perspective frequency response of the fan filter designed in Example 22.2.

22.4 Special Design Procedure for Restricted Classes

Many cases of practical importance typically require filters belonging to restricted classes. The stopbands and passbands of these filters are often defined by straight-line, circular, or elliptical boundaries. In these cases, specialized procedures lead to efficient design and low-cost implementation. The filters in these cases are derived from 1-D prototypes.

22.4.1 Separable 2-D FIR Filter Design

The design of 2-D FIR filters composed of 1-D building blocks is briefly discussed. In cases where the specifications are given in terms of multiple passbands in the shapes of rectangles with sides parallel to the frequency axes, the design problem can be decomposed into multiple designs. The resulting filter is a parallel connection of component filters that are themselves separable filters. The separable structure was encountered earlier in the construction of 2-D windows from 1-D windows in Section 22.3. The design approach is essentially the same. We will confine the discussion to cascade structures, which is a simple and very important practical case.

The frequency response of the 2-D separable FIR filter is expressed as

$$H(\omega_1, \omega_2) = H_1(\omega_1)H_2(\omega_2), \tag{22.35}$$

where $H_1(\omega)$ and $H_2(\omega)$ are frequency responses of two 1-D zero-phase FIR filters of durations N_1 and N_2. The corresponding 2-D filter is also a zero-phase FIR filter with $N \times M$ coefficients, and its impulse response is given by

$$h(n_1, n_2) = h_1(n_1)h_2(n_2), \tag{22.36}$$

where $h_1(n)$ and the $h_2(n)$ are the impulse responses of the 1-D FIR filters.

If the ideal frequency response can be expressed in a separable cascade form as in Equation 22.35, then the design problem is reduced to the case of appropriate 1-D filter designs. A simple but important example is the design of a 2-D low-pass filter with a symmetric square-shaped passband, PB = {(ω_1, ω_2): $|\omega_1| < \omega_c$, $|\omega_2| < \omega_c$}. Such a low-pass filter can be designed from a single 1-D FIR filter with a cutoff frequency of ω_c by using Equation 22.36. A low-pass filter constructed this way is used in Figure 22.2c.

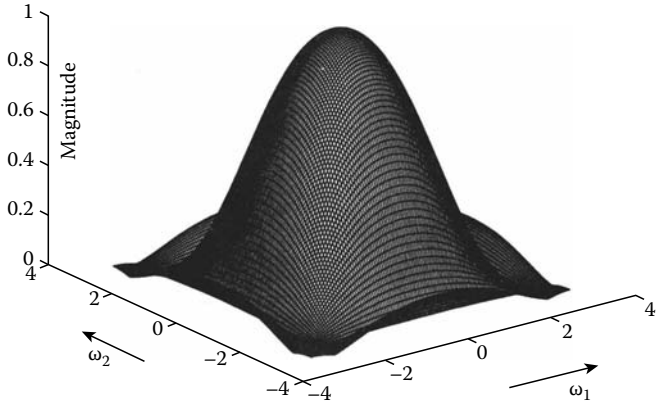


FIGURE 22.9 Frequency response of the separable low-pass filter $H(\omega_1, \omega_2) = H_1(\omega_1)H_1(\omega_2)$ where $H_1(\omega)$ is a seventh-order Lagrange filter.

The frequency response of this 2-D filter whose 1-D prototypes are seventh-order Lagrange filters is shown in Figure 22.9.

This design method is also used in designing 2-D filter banks which are utilized in subband coding of images and video signals [49,51,52]. The design of 2-D filter banks is discussed in Section 22.6.

22.4.2 Frequency Transformation Method

In this method a 2-D zero-phase FIR filter is designed from a 1-D zero-phase filter by a clever substitution of variables. The design procedure was first proposed by McClellan [33] and the frequency transformation is usually called the McClellan transformation [14,35,37,38].

Let $H_1(\omega)$ be the frequency response of a 1-D zero-phase filter with $2N + 1$ coefficients. The key idea of this method is to find a suitable transformation $\omega = G(\omega_1, \omega_2)$ such that the 2-D frequency response, $H(\omega_1, \omega_2)$, which is given by

$$H(\omega_1, \omega_2) = |H_1(\omega)|_{\omega = G(\omega_1, \omega_2)} \tag{22.37}$$

approximates the desired frequency response, $H_{\text{id}}(\omega_1, \omega_2)$.

Since the 1-D filter is a zero-phase filter, its frequency response is real, and it can be written as follows:

$$H_1(\omega) = h_1(0) + \sum_{n=1}^{N} 2h_1(n) \cos(\omega n), \tag{22.38}$$

where the term $\cos(\omega n)$ can be expressed as a function of $\cos(\omega)$ by using the nth-order Chebyshev polynomial, T_n,* i.e.,

$$\cos(\omega n) = T_n[\cos(\omega)]. \tag{22.39}$$

* Chebyshev polynomials are recursively defined as follows: $T_0(x) = 1$, $T_1(x) = x$, and $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$.

Using Equation 22.39, the 1-D frequency response can be written as

$$H_1(\omega) = \sum_{n=0}^{N} 2b(n)[\cos(\omega)]^n, \tag{22.40}$$

where the coefficients, $b(n)$, are related to the filter coefficients, $h(n)$.

In this design method the key step is to substitute a transformation function, $F(\omega_1, \omega_2)$, for $\cos(\omega)$ in Equation 22.40. In other words, the 2-D frequency response, $H(\omega_1, \omega_2)$, is obtained as follows:

$$\begin{aligned} H(\omega_1, \omega_2) &= H_1(\omega)\big|_{\cos(\omega)=F(\omega_1,\omega_2)} \\ &= \sum_{n=0}^{N} 2b(n)[F(\omega_1, \omega_2)]^n. \end{aligned} \tag{22.41}$$

The function, $F(\omega_1, \omega_2)$, is called the McClellan transformation.

The frequency response, $H(\omega_1, \omega_2)$, of the 2-D FIR filter is determined by two free functions, the 1-D prototype frequency response, $H_1(\omega)$, and the transformation, $F(\omega_1, \omega_2)$. In order to have $H(\omega_1, \omega_2)$ be the frequency response of an FIR filter, the transformation, $F(\omega_1, \omega_2)$, must itself be the frequency response of a 2-D FIR filter. McClellan proposed $F(\omega_1, \omega_2)$ to be the frequency response of a 3×3 zero-phase filter in Ref. [33]. In this case the transformation, $F(\omega_1, \omega_2)$, can be written as follows:

$$F(\omega_1, \omega_2) = A + B\ \cos(\omega_1) + C\ \cos(\omega_2) + D\ \cos(\omega_1 - \omega_2) + E\ \cos(\omega_1 + \omega_2), \tag{22.42}$$

where the real parameters, A, B, C, D, and E, are related to the coefficients of the 3×3 zero-phase FIR filter. For $A = -(1/2)$, $B = C = (1/2)$, $D = E = (1/4)$, the contour plot of the transformation, $F(\omega_1, \omega_2)$, is shown in Figure 22.10. Note that in this case the contours are approximately circularly symmetric around

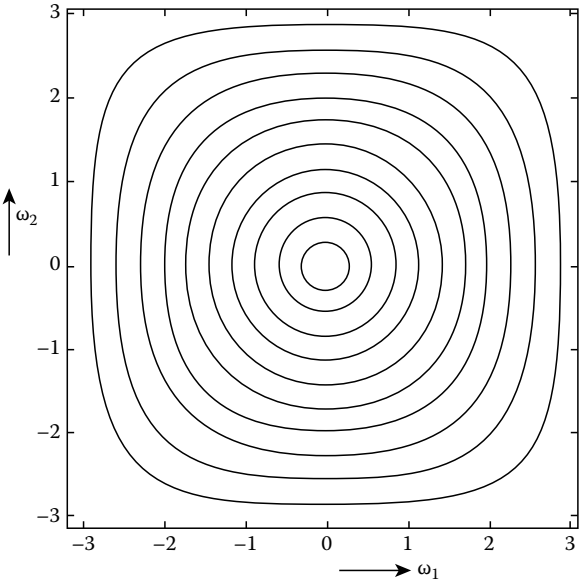


FIGURE 22.10 Contour plot of the McClellan transformation, $F(\omega_1, \omega_2) = 0.5\cos(\omega_1) + 0.5\cos(\omega_2) + 0.5\cos(\omega_1)\cos(\omega_2) - 0.5$.

the origin. It can be seen that the deviation from the circularity, expressed as a fraction of the radius, decreases with the radius. In other words, the distortion from a circular response is larger for large radii. It is observed from Figure 22.10 that, with the above choice of parameters, A, B, C, D, and E, the transformation is bounded ($|F(\omega_1, \omega_2)| \leq 1$), which implies that $H(\omega_1, \omega_2)$ can take only the values that are taken by the 1-D prototype filter, $H_1(\omega)$. Since $|\cos(\omega)| \leq 1$, the transformation, $F(\omega_1, \omega_2)$, which replaces $\cos(\omega)$ in Equation 22.41 must also take values between 1 and -1. If a particular transformation does not obey these bounds, then it can be scaled such that the scaled transformation satisfies the bounds.

If the transformation, $F(\omega_1, \omega_2)$, is real (it is real in Equation 22.42) then the 2-D filter, $H(\omega_1, \omega_2)$, will also be real or, in other words, it will be a zero-phase filter. Furthermore, it can be shown that the 2-D filter, $H(\omega_1, \omega_2)$, is an FIR filter with a support containing $(2M_1N + 1) \times (2M_2N + 1)$ coefficients, if the transformation, $F(\omega_1, \omega_2)$, is an FIR filter with $(2M_1 + 1) \times (2M_2 + 1)$ coefficients, and the order of the 1-D prototype filter is $2N + 1$. In (19.42) $M_1 = M_2 = 1$. As it can be intuitively guessed, one can design a 2-D approximately circularly symmetric low-pass (high-pass) (bandpass) filter with the above McClellan transformation by choosing the 1-D prototype filter, $H_1(\omega)$, a low-pass (high-pass) (bandpass) filter.

We will present some examples to demonstrate the effectiveness of the McClellan transformation.

Example 22.4

2-D window design by transformations [53]: In this example we design 2-D windows by using the McClellan transformation. Actually, we briefly mentioned this technique in Section 22.3. The 1-D prototype filter is chosen as an arbitrary 1-D symmetric window centered at the origin. Let $w(n)$ be the 1-D window of size $2N + 1$, and $W(\boldsymbol{\omega}) = \sum_{n=-N}^{N} w(n) \exp(-j\boldsymbol{\omega}n)$ be its frequency response. The transformation, $F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$, is chosen as in Equation 22.42 with the parameters $A = -(1/2)$, $B = C = (1/2)$, $D = E = (1/4)$, of Figure 22.10. This transformation, $F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$, can be shown to be equal to

$$F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = 0.5 \cos(\boldsymbol{\omega}_1) + 0.5 \cos(\boldsymbol{\omega}_2) + 0.5 \cos(\boldsymbol{\omega}_1) \cos(\boldsymbol{\omega}_2) - 0.5. \tag{22.43}$$

The frequency response of the McClellan window, $H_t(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$, is given by

$$H_t(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = W(\boldsymbol{\omega})|_{\cos(\boldsymbol{\omega}) = F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)}. \tag{22.44}$$

The resultant 2-D zero-phase window, $w_t(n_1, n_2)$, is centered at the origin and of size $(2N + 1) \times (2N + 1)$ because $M_1 = M_2 = 1$. The window coefficients can be computed either by using the inverse Chebyshev relation,* or by using the inverse Fourier transform of Equation 22.44. The frequency response of a 2-D window constructed from a 1-D Hamming window of order 13 is shown in Figure 22.3c. The size of the window is 13×13.

Example 22.5

Let us consider the design of a circularly symmetric low-pass filter and a bandpass filter by using the transformation of Equation 22.43. In this case, if one starts with a 1-D low-pass (bandpass) filter as the prototype filter, then the resulting 2-D filter will be a 2-D circularly symmetric low-pass (bandpass) filter due to the almost circularly symmetric nature of the transformation. In this example, the Lagrange filter of order 7 considered in Section 22.2 is used as the prototype. The prototype 1-D bandpass filter of order 15 is designed by using the Parks–McClellan algorithm [41]. Frequency response and contour plots of the low-pass and bandpass filters are shown in Figures 22.11 and 22.12, respectively.

* $1 = T_0(x)$, $x = T_1(x) - T_0(x)$, $x^2 = (t_0(x) + T_2(x))$, $x^3 = (3T_1(x) + T_3(x))$ etc.

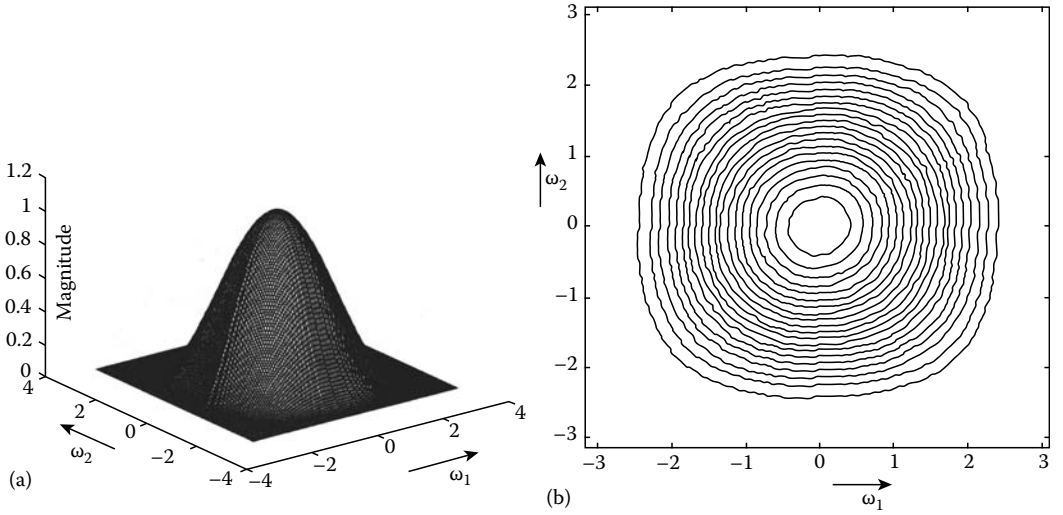


FIGURE 22.11 Frequency response and contour plots of the low-pass filter of Example 22.5.

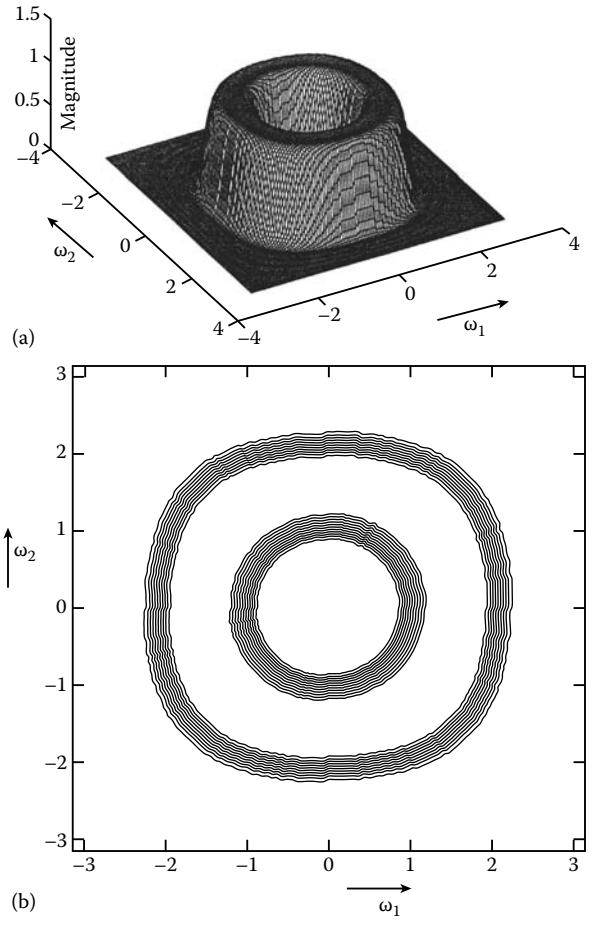


FIGURE 22.12 Frequency response and contour plots of the bandpass filter of Example 22.5.

It is seen from the above examples that filters designed by the transformation method appear to have better frequency responses than those designed by the windowing or frequency sampling methods. In other words, one can control the 2-D frequency response by controlling the frequency response of the 1-D prototype filter and choosing a suitable 2-D transformation. Furthermore, in some special cases it was shown that minimax optimal filters can be designed by the transformation method [20].

We have considered specific cases of the special transformations given by Equation 22.42. By varying the parameters in Equation 22.42 or expanding the transformation to include additional terms, a wider class of contours can be approximated. Ideally, the frequency transformation approach requires the simultaneous optimal selection of the transformation, $F(\omega_1, \omega_2)$, and the 1-D prototype filter $H_1(\omega)$ to approximate a desired 2-D frequency response. This can be posed as a nonlinear optimization problem. However, a suboptimal two-stage design by separately choosing $F(\omega_1, \omega_2)$ and $H_1(\omega)$ works well in practice. The transformation $F(\omega_1, \omega_2)$ should approximate $1(-1)$ in the passband (stopband) of the desired filter. The contour produced by the transformation corresponding to the 1-D passband (stopband) edge frequency, $\omega_p(\omega_s)$, should ideally map to the given passband (stopband) boundary in the 2-D specifications. However, this cannot be achieved in general given the small number of variable parameters in the transformation. The parameters are therefore selected to minimize a suitable norm of the error between actual and ideal (constant) values of the transformation over the boundaries.

Various transformations and design considerations are described in Refs. [37,38,40,42,43]. The use of this transformation in exact reconstruction filter bank design was proposed in Ref. [7].

Filters designed by the transformation method can be implemented in a computationally efficient manner [14,30]. The key idea is to implement Equation 22.41 instead of implementing the filter by using the direct convolution sum. By implementing the transformation, $F(\omega_1, \omega_2)$, which is an FIR filter of low-order, in a modular structure realizing Equation 22.41 is more advantageous than ordinary convolution sum [14,34].

In the case of circular passband design, it was observed that for low-order transformation, the transformation contours exhibit large deviations from circularity. A simple artifice to overcome this problem in approximating wideband responses is to use decimation of a 2-D narrowband filter impulse response [18]. The solution consists of transforming the specifications to an appropriate narrowband design, where the deviation from circularity is smaller. The narrow passband can be expanded by decimation while essentially preserving the circularity of the passband.

22.4.3 Design Using Nonrectangular Transformations and Sampling Rate Conversions

In some filter specifications the desired responses are characterized by ideal frequency responses in which passbands and stopbands are separated by straight-line boundaries that are not necessarily parallel to the frequency axes. Examples of these are the various kinds of fan filters [4,15,17,27] and diamond-shaped filters [6,48]. Other shapes with straight-line boundaries are also approximated [8,9,13,28,29,50]. Several design methods applicable in such cases have been developed and these methods are usually based on transformations related to concepts of sampling rate conversions. Often alternate frequency domain interpretations are used to explain the design manipulations. A detailed treatment of these methods is beyond the scope of this chapter. However some key ideas are described and a specific case of a diamond filter is used to illustrate the methods. The importance of these design methods stems from the implementation efficiency that results from a generalized notion of separable processing.

In the family of methods considered here, manipulations of a separable 2-D response using a combination of several steps are carried out. In the general case of designing filters with straight-line boundaries, it is difficult to describe a systematic procedure. However, in a given design problem, an appropriate set of steps in the design is suggested by the nature of the desired response.

Some underlying ideas can be understood by examining the problem of obtaining a filter with a parallelogram-shaped passband region. The sides of the parallelogram are assumed to be tilted with

respect to the frequency axes. One approach to solving this problem is to perform the following series of manipulations on a separable prototype filter with a rectangular passband. The prototype filter impulse response is upsampled on a *nonrectangular* grid. The upsampling is done by an integer factor greater than one and it is defined by a nondiagonal nonsingular integer matrix [39]. The upsampling produces a parallelogram by a rotation and compression of the frequency response of the prototype filter together with a change in the periodicity. The matrix elements are chosen to produce the desired orientation in the resulting response. Depending on the desired response, cascading to eliminate unwanted portions of the passband in the frequency response, along with possible shifts and additions, may be used. The nonrectangular upsampling is then followed by a rectangular decimation of the sequence to expand the passband out to the desired size. In some cases, the operations of the upsampling transformation and decimation can be combined by the use of nonrectangular decimation of impulse response samples. Results using such procedures produce efficient filter structures that are implemented with essentially 1-D techniques but where the orientations of processing are not parallel to the sample coordinates.

Consider the case of a diamond filter design shown in Figure 22.13. Note that the filter in Figure 22.13 can be obtained from the filter in Figure 22.14a by a transformation of variables. If $F_a(z_1, z_2)$ is the transfer function of the filter approximating the response in Figure 22.14a, then the diamond filter transfer function $D(z_1, z_2)$ given by

$$D(z_1, z_2) = F_a\left(z_1^{\frac{1}{2}} z_2^{\frac{1}{2}}, z_1^{-\frac{1}{2}} z_2^{\frac{1}{2}}\right) \tag{22.45}$$

will approximate the response in Figure 22.1a. The response in Figure 22.2a can be expressed as the sum of the two responses shown in Figure 22.2b and c. We observe that if $F_b(z_1, z_2)$ is the transfer function of the filter approximating the response in Figure 22.2b then

$$F_c(z_1, z_2) = F_b(-z_1, -z_2) \tag{22.46}$$

will approximate the response in Figure 22.14c. This is due to the fact that negating the arguments shifts the (periodic) frequency response of F_b by (π, π). The response in Figure 22.14b can be expressed as the product of two ideal 1-D low-pass filters, one horizontal and one vertical, which have the response shown in Figure 22.14d. This 1-D frequency response can be approximated by a half-band filter. Such an approximation will produce a response in which the transition band straddles both sides of the cutoff frequency boundaries in Figure 22.14a. If we wish to constrain the transition band to lie within the boundaries of the diamond-shaped region in Figure 22.13a, then we should choose a 1-D filter whose stopband interval is $(\pi/2, \pi)$. Let $H(z)$ be the transfer function of the prototype 1-D low-pass filter approximating the response in Figure 22.14d with a suitably chosen transition boundary. The transfer function $H(z)$ can be expressed as

$$H(z) = T_1(z^2) + zT_2(z^2). \tag{22.47}$$

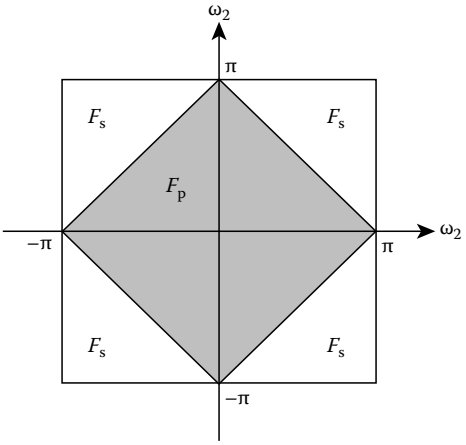


FIGURE 22.13 Ideal frequency response of a diamond filter.

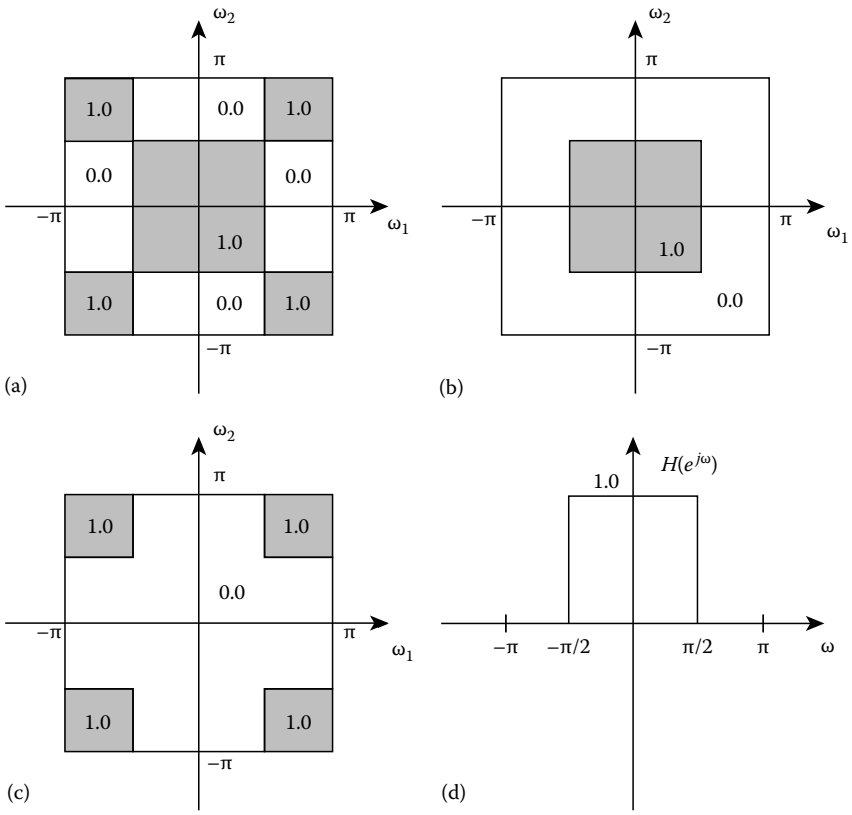


FIGURE 22.14 Ideal frequency responses of the filters: (a) $F_a(z_1, z_2)$, (b) $F_b(-z_1, -z_2)$, (c) $F_c(z_1, z_2)$, and (d) $H(z)$ in obtaining a diamond filter.

The transfer function F_a is given by

$$F_a(z_1, z_2) = H(z_1)H(z_2) + H(-z_1)H(-z_2). \tag{22.48}$$

Combining Equations 22.45, 22.47, and 22.48 we get

$$D(z_1, z_2) = 2T_1(z_1, z_2)T_1\left(z_1^{-1}z_2\right) + 2z_2T_2(z_1, z_2)T_2\left(z_1^{-1}z_2\right). \tag{22.49}$$

As mentioned before, $H(z)$ can be chosen to be a half-band filter with

$$T_1(z^2) = 0.5. \tag{22.50}$$

The filter T_2 can be either FIR or IIR. It should be noted that the result can also be obtained as a nonrectangular downsampling, by a factor of 2, of the impulse response of the filter $F_b(-z_1, -z_2)$.

Another approach that utilizes multirate concepts is based on a novel idea of applying frequency masking in the 2-D case [31].

22.5 2-D FIR Filter Implementation

The straightforward way to implement 2-D FIR filters is to evaluate the convolution sum given in Equation 22.1. Let us assume that the FIR filter has L nonzero coefficients in its region of support I. In order to get an output sample, L multiplications and L additions need to be performed. The number of arithmetic operations can be reduced by taking advantage of the symmetry of the filter coefficients, that is, $h(n_1, n_2) = h(-n_1, -n_2)$. For example, let the filter support be a rectangular region, $I = \{n_1 = -N_1, \ldots, 0, 1, \ldots, N_1, n_2 = -N_2, \ldots, 0, 1, \ldots, N_2\}$. In this case,

$$y(n_1, n_2) = \sum_{k_1=-N_1}^{N_1} \sum_{k_2=1}^{N_2} [h(k_1, k_2)x(n_1 - k_1, n_2 - k_2) + x(n_1 + k_1, n_2 + k_2)] + h(0,0)x(n_1, n_2) + \sum_{k_1=1}^{N_1} h_1(k_1, 0)[x(n_1 - k_1, n_2) + x(n_1 + k_1, n_2)], \quad (22.51)$$

which requires approximately half of the multiplications required in the direct implementation equation 22.1.

Any 2-D FIR filter can also be implemented by using an FFT algorithm. This is the direct generalization of 1-D FFT-based implementation [14,30]. The number of arithmetic operations may be less than the space domain implementation in some cases.

Some 2-D filters have special structures that can be exploited during implementation. As we pointed out in Section 22.4, 2-D filters designed by McClellan-type transformations can be implemented in an efficient manner [14,34,35] by building a network whose basic module is the transformation function which is usually a low order 2-D FIR filter.

2-D FIR filters that have separable system responses can be implemented in a cascade structure. In general, an arbitrary 2-D polynomial cannot be factored into subpolynomials due to the absence of a counterpart of fundamental theorem of algebra in two or higher dimensions (whereas in 1-D any polynomial can be factored into polynomials of lower orders). Since separable 2-D filters are constructed from 1-D polynomials, they can be factored and implemented in a cascade form. Let us consider Equation 22.35 where $H(x_1, x_2) = H_1(x_2)H_2(x_2)$ which corresponds to $h(n_1, n_2) = h_1(n_1)\, h_2(n_2)$ in space domain. Let us assume that orders of the 1-D filters $h_1(n)$ and $h_2(n)$ are $2N_1 + 1$ and $2N_2 + 1$, respectively. In this case the 2-D filter, $h(n_1, n_2)$, has the same rectangular support, I, as in Equation 22.51. Therefore,

$$y(n_1, n_2) = \sum_{k_2=-N_2}^{N_2} h_2(k_2) \sum_{k_1=-N_1}^{N_1} h(k_1)x(n_1 - k_1, n_2 - k_2). \quad (22.52)$$

The 2-D filtering operation in Equation 22.52 is equivalent to a two-stage 1-D filtering in which the input image, $x(n_1, n_2)$, is first filtered horizontally line by line by $h_1(n)$, then the resulting output is filtered vertically column by column by $h_2(n)$. In order to produce an output sample, the direct implementation requires $(2N_1 + 1) \times (2N_2 + 1)$ multiplications, whereas the separable implementation requires $(2N_1 + 1) + (2N_2 + 1)$ multiplications, which is computationally much more efficient than the direct form realization. This is achieved at the expense of memory space (separable implementation needs a buffer to store the results of first stage during the implementation). By taking advantage of the symmetric nature of h_1 and h_2, the number of multiplications can be further reduced.

Filter design methods by imposing structural constraints like cascade, parallel, and other forms are proposed by several researchers including Refs. [16,47]. These filters can be efficiently implemented because of their special structures. Unfortunately, the design procedure requires nonlinear optimization techniques which may be very complicated.

With advances in VLSI technology, the implementation of 2-D FIR filters using high-speed digital signal processors is becoming increasingly common in complex image processing systems.

22.6 Two-Dimensional Filter Banks

2-D subband decomposition of signals using filter banks (that implement a 2-D wavelet transform) find applications in a wide range of tasks including image and video coding, restoration, denoising, and signal analysis. For example, in recently finalized JPEG-2000 image coding standard an image is first processed by a 2-D filter bank. Data compression is then carried out in the subband domain. In this section we briefly discuss the case of four-channel separable filter banks. Chapter 24 provides a detailed description of 1-D filter banks.

In most cases, 2-D filter banks are constructed in a separable form with the use of the filters of 1-D filter banks, i.e., as a product of two 1-D filters [49,52]. We confine our attention to a 2-D four-channel filter bank obtained from a 1-D two-channel filter bank. Let h_0 and h_1 denote the analysis filters of a 1-D two-channel filter bank. The four analysis filters of the separable 2-D filter bank are given by

$$h_{i,j}(n_1, n_2) = h_i(n_1)h_j(n_2), \quad i, j = 0, 1. \tag{22.53}$$

The filters h_0 and h_1 can be either FIR or IIR. If they are FIR (IIR), then the 2-D filters, $h_{i,j}$, are also FIR (IIR). Frequency responses of these four filters, $H_{i,j}(\omega_1, \omega_2)$, $i,j = 0, 1$, are given by

$$H_{i,j}(\omega_1, \omega_2) = H_i(\omega_1)H_j(\omega_2), \quad i, j = 0, 1, \tag{22.54}$$

where $H_0(\omega_1)$ and $H_1(\omega_2)$ are the frequency responses of the 1-D low-pass (approximating an ideal cutoff frequency at $\pi/2$) and high-pass filters of a 1-D subband filter bank, respectively [51]. Any 1-D filter bank described in Chapter 24 can be used in Equation 22.53 to design 2-D filter banks. Feature-rich structures for 1-D filter banks are described in Ref. [6].

The 2-D signal is decomposed by partitioning its frequency domain support into four rectangular regions of equal areas. The ideal passband regions of the filters, $H_{i,j}(\omega_1, \omega_2)$, are shown in Figure 22.15. For example, the ideal passband of $H_{0,0}(\omega_1, \omega_2)$ is the square region $[-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$. The 2-D filter bank is shown in Figure 22.16.

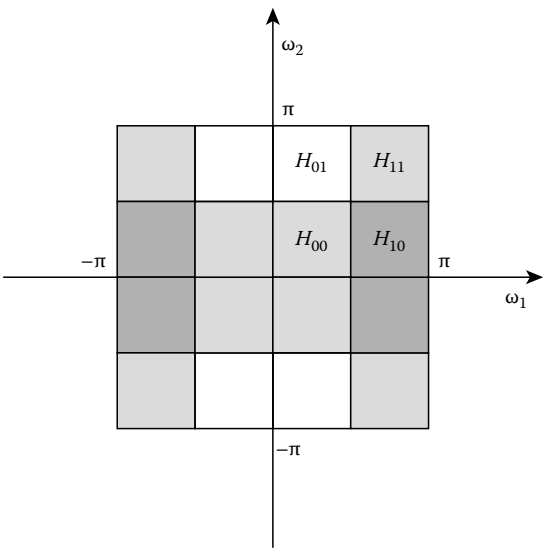


FIGURE 22.15 Ideal passband regions of the separable filters of a rectangular filter bank.

Corresponding 2-D synthesis filters are also constructed in a separable manner from the synthesis filters of the 1-D filter bank. If the 1-D filter bank has the perfect reconstruction (PR) property, then the 2-D filter bank also has the PR property. Subband decomposition filter banks (or filter banks implementing the 2-D wavelet transform) consist of analysis and synthesis filters, upsamplers, and downsamplers as discussed in Chapter 24. In the separable 2-D filter bank, downsampling is carried out both horizontally and vertically as follows:

$$x_0(n_1, n_2) = x_a(2n_1, 2n_2). \tag{22.55}$$

Here we consider the input 2-D signal x_a to be an image. The downsampled image x_0 is a quarter-size version of x_a. Only one sample out of four is retained in the downsampling operation described in Equation 22.55. The upsampling operation is the dual of the downsampling

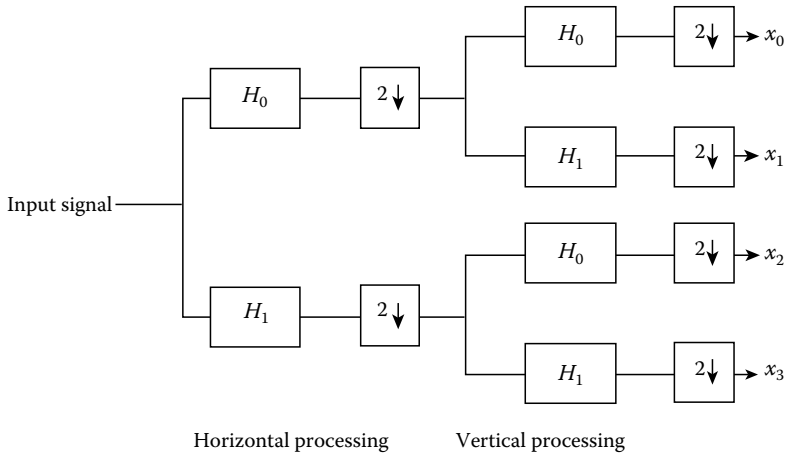


FIGURE 22.16 Block diagram of separable processing in 2D filter bank.

operation. In other words, a zero valued sample is inserted in upsampling corresponding to the location of each dropped sample during downsampling.

The implementation of the above filter bank can be carried out separably in a computationally efficient manner as described in Chapter 24 and Refs. [49,52]. The input image is first processed horizontally row by row by a 1-D filter bank with filters, h_0 and h_1. After the input signal is horizontally filtered with the 1-D two-channel filter bank, the signal in each channel is downsampled row-wise to yield two images. Each image is then filtered and downsampled vertically by the filter bank. As a result, four quarter-size subimages, x_i, $i = 0, 1, 2, 3$, are obtained. These images are the same as the images obtained by direct implementation of the analysis filter bank shown in Figure 22.16. The synthesis filter bank is also implemented in a separable manner.

Nonseparable 2-D filter banks [14] are not as computationally efficient as separable filter banks as discussed in Section 22.5.

In two or higher dimensions, downsampling and upsampling are not restricted to the rectangular grid, but can be carried out in a variety of ways. One example of this is quincunx downsampling where the downsampled image x_q is related to the input image x_a as follows:

$$x_q(n_1, n_2) = x_a(n_1 + n_2, n_2 - n_1). \tag{22.56}$$

In this case only the samples for which $n_1 + n_2$ is even are retained in the output. Filter banks employing quincunx and other downsampling strategies are described in Refs. [6,8,9,19,50,52] and in Chapter 25. Filter banks that employ quincunx downsampling have only two channels and the frequency support is partitioned in a diamond-shaped manner as shown in Figure 22.13. Filters of a quincunx filter bank which have diamond-shaped passbands and stopbands can be designed from a 1-D subband decomposition filter bank using the transformation method described in Equations 22.45 through 22.48. Chapter 25 provides a detailed discussion of the theory and applications of directional filter banks.

Acknowledgments

The authors would like to express their sincere thanks to Omer Nezih Gerek, Imad Abdel Hafez, and Ahmet Murat Bagci for the help they provided in preparing the figures for the chapter.

References

1. A. Abo-Taleb and M. M. Fahmy, Design of FIR two-dimensional digital filters by successive projections, *IEEE Transactions on Circuits and Systems*, CAS-31, 801–805, 1984.
2. V. R. Algazi, M. Suk, C.-S. Rim, Design of almost minimax FIR filters in one and two dimensional by WLS techniques, *IEEE Transactions on Circuits and Systems*, 33(6), 590–596, June 1986.
3. S. A. H. Aly and M. M. Fahmy, Symmetry in two-dimensional rectangularly sampled digital filters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29, 794–805, 1981.
4. R. Ansari, Efficient IIR and FIR fan filters, *IEEE Transactions on Circuits System*, CAS-34, 941–945, August 1987.
5. R. Ansari, A. E. Cetin, and S. H. Lee, Subband coding of images using nonrectangular filter banks, *Proceedings of the SPIE 32nd Annual International Technical Symposium: Applications of Digital Signal Processing*, San Diego, CA, Vol. 974, August 1988.
6. R. Ansari, C. W. Kim, and M. Dedovic, Structure and design of two-channel filter banks derived from a triplet of halfband filters, *IEEE Transactions on Circuits and Systems II*, 46, 1487–1496, December 1999.
7. R. Ansari and C. Guillemot, Exact reconstruction filter banks using diamond FIR filters, *Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing*, E. Arikan (Ed.), pp. 1412–1421, Elsevier, Holland, 1990.
8. R. H. Bamberger and M. J. T. Smith, Efficient 2-D analysis/synthesis filter banks for directional image component representation, *Proceedings of IEEE International Symposium on Circuits and Systems*, New Orleans, LA, pp. 2009–2012, May 1990.
9. R. H. Bamberger and M. J. T. Smith, A filter bank for the directional decomposition of images: Theory and design, *IEEE Transactions on Acoustics, Speech, Signal Processing*, 40(4), 882–892, April 1992.
10. C. Charalambous, The performance of an algorithm for minimax design of two-dimensional linear-phase FIR digital filters, *IEEE Transactions on Circuits and Systems,* CAS-32, 1016–1028, 1985.
11. A. E. Cetin and R. Ansari, An iterative procedure for designing two dimensional FIR filters, *Proceedings of IEEE International Symposium on Circuits and Systems* (*ISCAS*), Philadelphia, PA, pp. 1044–1047, 1987.
12. A. E. Cetin and R. Ansari, Iterative procedure for designing two dimensional FIR filters, *Electronics Letters, IEE,* 23, 131–133, January 1987.
13. T. Chen and P. P. Vaidyanathan, Multidimensional multirate filters and filter banks derived from one-dimensional filters, *IEEE Transactions on Signal Processing,* 41, 1035–1047, March 1993.
14. D. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
15. P. Embree, J. P. Burg, and M. M. Backus, Wideband velocity filtering—The pie slice process, *Geophysics,* 28, 948–974, 1963.
16. O. D. Faugeras and J. F. Abramatic, 2-D FIR filter design from independent 'small' generating kernels using a mean square and Tchebyshev error criterion, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Washington, DC, pp. 1–4, 1979.
17. A. P. Gerheim, Synthesis procedure for 90° fan filters, *IEEE Transactions on Circuits and Systems*, CAS30, 858–864, December 1983.
18. C. Guillemot and R. Ansari, Two-dimensional filters with wideband circularly symmetric frequency response, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 41(10), 703–707, October 1994.
19. C. Guillemot, A. E. Cetin, and R. Ansari, Nonrectangular wavelet representation of 2-D signals: Application to image coding, *Wavelets and Application to Image Coding*, M. Barlaud (Ed.), p. 2764, Elsevier Publications, Amsterdam, Holland, 1994.
20. D. B. Harris and R. M. Mersereau, A comparison of algorithms for minimax design of two-dimensional linear phase FIR digital filters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-25, 492–500, 1977.

21. J. V. Hu and L. R. Rabiner, Design techniques for two-dimensional digital filters, *IEEE Transactions on Audio Electroacoustics,* 20, 249–257, 1972.
22. T. S. Huang, Two-dimensional windows, *IEEE Transactions on Audio and Electroacoustics*, AU-20(1), 88–90, 1972.
23. T. S. Huang (Ed)., *Two-Dimensional Digital Signal Processing I: Linear Filters*, Springer-Verlag, New York, 1981.
24. Y. Kamp and J. P. Thiran, Maximally flat nonrecursive two-dimensional digital filters, *IEEE Transactions on Circuits and Systems,* CAS-21, 437–449, May 1974.
25. Y. Kamp and J. P. Thiran, Chebyshev approximation for two-dimensional nonrecursive digital filters, *IEEE Transactions on Circuits and Systems,* CAS-22, 208–218, 1975.
26. H. Kato and T. Furukawa, Two-dimensional type-preserving circular windows, *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-29, 926–928, 1981.
27. A. H. Kayran and R. A. King, Design of recursive and nonrecursive fan filters with complex transformations, *IEEE Transactions on Circuits and Systems*, CAS-30, 849–857, December 1983.
28. C. -L. Lau and R. Ansari, Two dimensional digital filter and implementation based on generalized decimation, *Princeton Conference*, Princeton, NJ, March 1986.
29. C. -L. Lau and R. Ansari, Design of two-dimensional filters using sampling rate alteration, *IEEE International Symposium on Circuits and Systems*, Montreal, Canada, pp. 474–477, 1984.
30. J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
31. Y. C. Lim and Y. Lian, The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* 40, 88–95, 1993.
32. J. H. Lodge and M. M. Fahmy, An efficient l_p optimization technique for the design of two-dimensional linear-phase FIR digital filters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28, 308–313, 1980.
33. J. H. McClellan, The design of two-dimensional filters by transformations, *Proceedings of the* 7*th Annual Princeton Conf. Information Sciences and Systems*, Princetone, NJ, 247–251, 1973.
34. J. H. McClellan and D. S. K. Chan, A 2-D FIR filter structure derived from the Chebyshev recursion, *IEEE Transactions on Circuits and Systems,* 24, 372–378, 1977.
35. W. F. G. Mecklenbrauker and R. M. Mersereau, McClellan transformation for 2-D digital filtering: II—Implementation, *IEEE Transactions on Circuits and Systems*, CAS-23, 414–422, 1976.
36. R. M. Mersereau, D. B. Harris, and H. S. Hersey, An efficient algorithm for the design of two-dimensional digital filters, *Proceedings of the* 1974 *International Symposium Circuits and Systems,* Newton, MA, pp. 443–446, 1975.
37. R. M. Mersereau, W. F. G. Mecklenbrauker, and T. F. Quatieri, Jr., McClellan transformation for 2-D digital filtering: I—Design, *IEEE Transactions on Circuits and Systems,* CAS-23, 405–414, 1976.
38. R. M. Mersereau, The design of arbitrary 2-D zero-phase FIR filters using transformations, *IEEE Transactions on Circuits and Systems*, 27, 372–378, 1980.
39. R. M. Mersereau and T. C. Speake, The processing of periodically sampled multidimensional signals, *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-31, 188–194, February 1983.
40. D. T. Nguyen and M. N. S. Swamy, Scaling free McClellan transformation for 2-D digital filters, *IEEE Transactions on Circuits and Systems*, CAS-33, 108–109, January 1986.
41. T. W. Parks and J. H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, *IEEE Transactions on Circuits Theory*, 19, 189–194, 1972.
42. S. -C. Pei and J. -J. Shyu, Design of 2-D FIR digital filters by McClellan transformation and least squares eigencontour mapping, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* 40, 546–555, 1993.
43. E. Z. Psarakis and G. V. Moustakides, Design of two-dimensional zero-phase fir filters via the generalized McClellan transform, *IEEE Transactions on Circuits and Systems*, CAS-38, 1355–1363, November 1991.

44. P. K. Rajan, H. C. Reddy, and M. N. S. Swamy, Fourfold rational symmetry in two-dimensional FIR digital filters employing transformations with variable parameters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31, 488–499, 1982.
45. V. Rajaravivarma, P. K. Rajan, and H. C. Reddy, Design of multidimensional FIR digital filters using the symmetrical decomposition technique, *IEEE Transactions on Signal Processing,* 42, 164–174, January 1994.
46. T. Speake and R. M. Mersereau, A comparison of different window formulas for two-dimensional FIR filter design, *Proceedings on IEEE International Conference on Acoustics, Speech, and Signal Processing*, Washington, DC, pp. 5–8, 1979.
47. S. Treitel and J. L. Shanks, The design of multistage separable planar filters, *IEEE Transactions on Geoscience Electronics*, GE-9, 10–27, 1971.
48. G. J. Tonge, The sampling of television images, Independent Broadcasting Authority, Experimental and Development Report 112/81.
49. M. Vetterli, A theory of multirate filter banks, *Signal Processing,* 6, 97–112, February 1984.
50. E. Viscito and J. P. Allebach, The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices, *IEEE Transactions on Circuits and Systems*, CAS-38, 29–41, January 1991.
51. J. W. Woods and S. D. O'Neill, Subband coding of images, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34, 1278–1288, 1986.
52. J. W. Wood (Ed)., *Subband Image Coding*, Kluwer Academic Publishers, Norwell, MA, 1990.
53. T.-H. Yu and S. K. Mitra, A new two-dimensional window, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33, 1058–1061, 1985.

Further Information

Most of the research articles describing the advances in 2-D FIR filter design methods appear in *IEEE Transactions on Signal Processing*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems*, and *Electronics Letters*, and *International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*) and *International Symposium on Circuits and Systems*.

23

Two-Dimensional IIR Filters

23.1 Introduction **23**-1
23.2 Transformation Techniques **23**-2
Analog Filter Transformations • Spectral Transformations
23.3 Design of Separable Product Filters **23**-11
Design of Quadrantally Symmetric Filters • Design of Half-Plane Symmetric Filters • Design of Filters Satisfying Prescribed Specifications
23.4 Design of Circularly Symmetric Filters **23**-18
Design of LP Filters • Realization of HP, BP, and BS Filters • Design of Filters Satisfying Prescribed Specifications
23.5 Design of 2-D IIR Filters by Optimization **23**-26
Design by Least *p*th Optimization • Design by Singular-Value Decomposition • Design Based on Two-Variable Network Theory
References **23**-47

A. G. Constantinides
Imperial College of Science, Technology, and Medicine

Xiaojian Xu
Beihang University

23.1 Introduction

A linear 2-D IIR digital filter can be characterized by its transfer function

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} = \frac{\sum_{i=0}^{N_2} \sum_{j=0}^{M_2} a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^{N_1} \sum_{j=0}^{M_1} b_{ij} z_1^{-i} z_2^{-j}} \tag{23.1}$$

where the sampling period $T_i = 2\pi/\omega_{si}$ for $i = 1, 2$ with ω_{si} and the sampling frequencies a_{ij} and b_{ij} are real numbers known as the coefficients of the filter. Without loss of generality we can assume $M_1 = M_2 = N_1 = N_2 = M$ and $T_1 = T_2 = T$. Designing a 2-D filter is to calculate the filter coefficients a_{ij} and b_{ij} in such a way that the amplitude response and/or the phase response (group delay) of the designed filter approximates to some ideal responses while maintaining the stability of the designed filter. The latter requires that

$$D(z_1, z_2) \neq 0 \quad \text{for } |z_i| \geq 1,\ i = 1, 2 \tag{23.2}$$

The amplitude response of the 2-D filter is expressed as

$$M(\omega_1, \omega_2) = \left| H(e^{j\omega_1 T}, e^{j\omega_2 T}) \right| \tag{23.3}$$

the phase response as

$$\phi(\omega_1, \omega_2) = \arg H(e^{j\omega_1 T}, e^{j\omega_2 T}) \tag{23.4}$$

and the two group delay functions as

$$\tau_i(\omega_1, \omega_2) = \frac{d\phi(\omega_1, \omega_2)}{d\omega_i}, \; i = 1, 2 \tag{23.5}$$

Equation 23.1 is the general form of transfer functions of the nonseparable numerator and denominator 2-D IIR filters. It can involve two subclasses, namely, the separable product transfer function

$$\begin{aligned} H(z_1, z_2) &= H_1(z_1)H_2(z_2) \\ &= \frac{\sum_{i=0}^{N_2} a_{1i} z_1^{-i} \sum_{j=0}^{M_2} a_{2i} z_2^{-j}}{\sum_{i=0}^{N_1} b_{1i} z_1^{-i} \sum_{j=0}^{M_1} b_{2i} z_2^{-j}} \end{aligned} \tag{23.6}$$

and the separable denominator, nonseparable numerator transfer function given by

$$H(z_1, z_2) = \frac{\sum_{i=0}^{N_2} \sum_{j=0}^{M_2} a_{ij} z_1^{-i} z_2^{-j}}{\sum_{i=0}^{N_1} b_{1i} z_1^{-i} \sum_{j=0}^{M_1} b_{2j} z_2^{-j}} \tag{23.7}$$

The stability constraints for the above two transfer functions are the same as those for the individual two 1-D cases. These are easy to check and correspondingly the transfer function is easy to stabilize if the designed filter is found to be unstable. Therefore, in the design of the above two classes, in order to reduce the stability problem to that of the 1-D case, the denominator of the 2-D transfer function is chosen to have two 1-D polynomials in z_1 and z_2 variables in cascade. However, in the general formulation of nonseparable numerator and denominator filters, this oversimplification is removed. The filters of this type are generally designed either through transformation of 1-D filters, or through optimization approaches, as is discussed in the following.

23.2 Transformation Techniques

23.2.1 Analog Filter Transformations

In the design of 1-D analog filters, a group of analog filter transformations of the form $s = g(s')$ is usually applied to normalized continuous low-pass transfer functions like those obtained by using the Bessel, Butterworth, Chebyshev, and elliptic approximations. These transformations can be used to design low-pass, high-pass, bandpass, or bandstop filters satisfying piecewise-constant amplitude response specifications. Through the application of the bilinear transformation, corresponding 1-D digital filters can be designed, and since 2-D digital filters can be designed in terms of 1-D filters, these transformations are of considerable importance in the design of 2-D digital filters as well. In the 2-D cases, the transformations have a form of

$$s = g(s_1, s_2) \tag{23.8}$$

As a preamble, in this section two groups of transformations of interest in the design of 2-D digital filters are introduced, which essentially produce 2-D continuous transfer functions from 1-D ones.

23.2.1.1 Rotated Filter

The first group of transformations, suggested by Shanks, Treitel, and Justice [1], are of the form

$$g_1(s_1, s_2) = -s_1 \sin\beta + s_2 \cos\beta \tag{23.9a}$$

$$g_2(s_1, s_2) = s_1 \cos\beta + s_2 \sin\beta \tag{23.9b}$$

They map 1-D into 2-D filters with arbitrary directionality in a 2-D frequency response plane. These filters are called rotated filters because they are obtained by rotating 1-D filters.

If $H(s)$ is a 1-D continuous transfer function, then a corresponding 2-D continuous transfer function can be generated as

$$H_{D1}(s_1, s_2) = H(s)\Big|_{s=g_1(s_1,s_2)} \tag{23.10a}$$

$$H_{D2}(s_1, s_2) = H(s)\Big|_{s=g_2(s_1,s_2)} \tag{23.10b}$$

by replacing the s in $H(s)$ with $g_1(s_1, s_2)$ and $g_2(s_1, s_2)$, respectively.

It is easy to show [2] that a transformation of $g_1(s_1, s_2)$ or $g_2(s_1, s_2)$ will give rise to a contour in the amplitude response of the 2-D analog filter that is a straight line rotated by an angle β with respect to the s_1 or s_2 axis, respectively. Figure 23.1 illustrates an example of 1-D to 2-D analog transformation by (23.10a) for $\beta = 0^\circ$ and $\beta = 45^\circ$.

The rotated filters are of special importance in the design of circularly symmetric filters, as will be discussed in Section 23.4.

23.2.1.2 Transformation Using a Two-Variable Reactance Function

The second group of transformations is based on the use of a two-variable reactance function. One of the transformations was suggested by Ahmadi, Constantinides, and King [3,4]. This is given by

$$g_3(s_1, s_2) = \frac{a_1 s_1 + a_2 s_2}{1 + b s_1 s_2} \tag{23.11}$$

where a_1, a_2, and b are all positive constants.

Let us consider a 2-D filter designed by using a 1-D analog low-pass filter with cutoff frequency Ω_c. Equation 23.11 results in

$$\Omega_2 = \frac{\Omega_c - a_1 \Omega_1}{a_2 - b\Omega_c \Omega_1} \tag{23.12}$$

The mapping of $\Omega = \Omega_c$ onto the (Ω_1, Ω_2) plane for various values of b is depicted in Figure 23.2 [5]. The cutoff frequencies along the Ω_1 and Ω_2 axes can be adjusted by simply varying a_1 and a_2. On the other hand, the convexity of the boundary can be adjusted by varying b. We note that b must be greater than zero to preserve stability. Also, it should be noted that $g_3(s_1, s_2)$ becomes a low-pass to bandpass transformation along $s_1 = s_2$, and therefore the designed filter will behave like a bandstop filter along $\Omega_1 = \Omega_2$. This problem can be overcome by using a guard filter of any order.

King and Kayran [6] have extended the above technique by using a higher order reactance function of the form

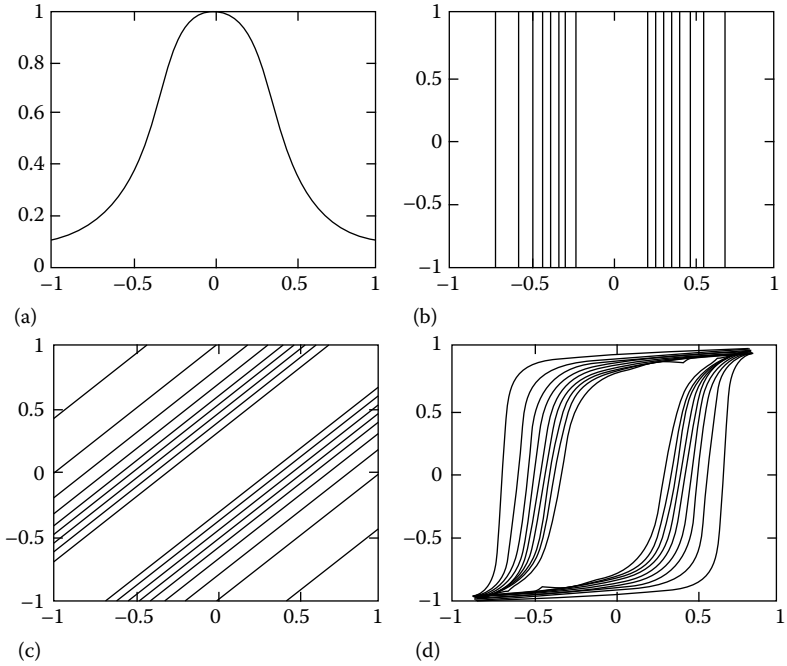


FIGURE 23.1 1D to 2D analog filter transformation. (a) Amplitude response of 1D analog filter. (b) Contour plot of 2D filter, $\beta = 0°$. (c) Contour plot of 2D filter, $\beta = 45°$. (d) Contour plot of 2D filter, $\beta = 45°$, after applying double bilinear transformation.

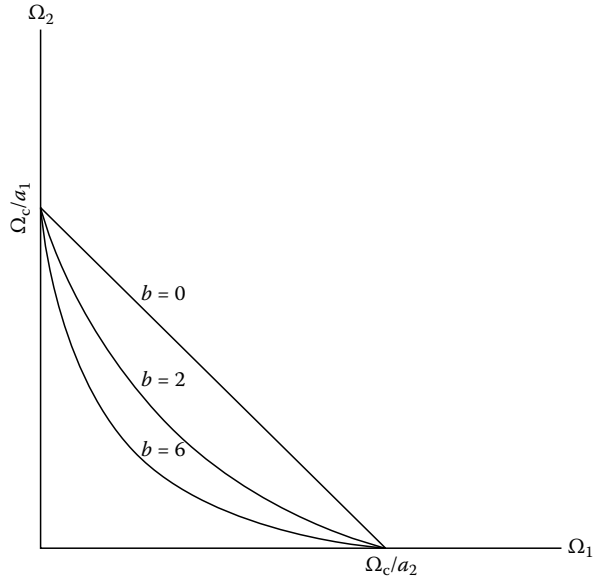


FIGURE 23.2 Plot of Equation 23.12 for different values of b.

$$g_4(s_1, s_2) = \frac{a_1 s_1 + a_2 s_2}{1 + b_1(s_1^2 + s_2^2) + b_2 s_1 s_2} \tag{23.13}$$

and they proved that the stability of $g_4(s_1, s_2)$ is ensured if

$$b_1 > 0 \tag{23.14}$$

and

$$b_1 > \frac{b_2^2}{4} - b_1^2 > 0 \tag{23.15}$$

However, it is necessary as earlier to include a guard filter, which may have the simple form of

$$G(z_1, z_2) = \frac{(1 + z_1)(1 + z_2)}{(d_1 + z_1)(d_2 + z_2)} \tag{23.16}$$

in order to remove the high-pass regions along all radii except the coordinate axes.

Then, through an optimization procedure, the coefficients of $g_4(s_1, s_2)$ and $G(z_1, z_2)$ are calculated subject to the constraints of Equations 23.14 and 23.15, so that the cutoff frequency of the 1-D filter is mapped into a desired cutoff boundary in the (Ω_1, Ω_2) plane.

23.2.2 Spectral Transformations

Spectral transformation is another kind of important transformation in the design of both 1-D and 2-D IIR filters. In this section, three groups of spectral transformations are discussed. Among them, the linear transformations map frequency axes onto frequency axes in the (Ω_1, Ω_2) plane, the complex transformation is of wide applications to the design of fan filters, and the Constantinides transformations transform a discrete function into another discrete function and through which any transformation of a low-pass filter to another low-pass, high-pass, bandpass, or bandstop filter becomes possible.

23.2.2.1 Linear Transformations

Consider a group of linear transformations that map frequency axes onto themselves in the (Ω_1, Ω_2) plane. There are eight possible such transformations [7,8] and they have the algebraic structure of a finite group under the operation of multiplication [2]; each transformation can be expressed as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} := D(T) \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{23.17}$$

where $D(T)$ is a 2×2 unitary matrix representing transformation $T\delta$. The eight transformations and their effect on the frequency response of the digital filter are as follows with a multiplication table being illustrated in Table 23.1 [2].

1. Identity (I):

$$D(I) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2. Reflection about the ω_1 axis ($\rho_{\omega 1}$):

$$D(\rho_{\omega 1}) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

TABLE 23.1 Multiplication Table of Group

	I	$\rho\omega 1$	$\rho\omega 2$	$\rho\psi 1$	$\rho\psi 2$	R_4	R_4^2	R_4^3
I	I	$\rho\omega 1$	$\rho\omega 2$	$\rho\psi 1$	$\rho\psi 2$	R_4	R_4^2	R_4^3
$\rho\omega 1$	$\rho\omega 1$	I	R_4^2	R_4^3	R_4	$\rho\psi 2$	$\rho\omega 2$	$\rho\psi 1$
$\rho\omega 2$	$\rho\omega 2$	R_4^2	I	R_4	R_4^3	$\rho\psi 1$	$\rho\omega 1$	$\rho\psi 2$
$\rho\psi 1$	$\rho\psi 1$	R_4	R_4^3	I	R_4^2	$\rho\omega 1$	$\rho\psi 2$	$\rho\omega 2$
$\rho\psi 2$	$\rho\psi 2$	R_4^3	R_4	R_4^2	I	$\rho\omega 2$	$\rho\psi 1$	$\rho\omega 1$
R_4	R_4	$\rho\psi 1$	$\rho\psi 2$	$\rho\omega 2$	$\rho\omega 1$	R_4^2	R_4^3	I
R_4^2	R_4^2	$\rho\omega 2$	$\rho\omega 1$	$\rho\psi 2$	$\rho\psi 1$	R_4^3	I	R_4
R_4^3	R_4^3	$\rho\psi 2$	$\rho\psi 1$	$\rho\omega 1$	$\rho\omega 2$	I	R_4	R_4^2

3. Reflection about the ω_2 axis ($\rho_{\omega 2}$):

$$D(\rho_{\omega 2}) = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

4. Reflection about the ($\rho_{\psi 1}$):

$$D(\rho_{\psi 1}) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

5. Reflection about the ($\rho_{\psi 2}$):

$$D(\rho_{\psi 2}) = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

6. Counterclockwise rotation by 90° (R_4):

$$D(R_4) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

7. Counterclockwise rotation by 180° (R_4^2):

$$D\left(R_4^2\right) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

8. Counterclockwise rotation by 270° (R_4^3):

$$D\left(R_4^3\right) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

In the above symbolic representation, ψ_1 and ψ_2 represent axes that are rotated by 45° in the counterclockwise sense relative to the ω_1 and ω_2 axes, respectively, and R_k denotes rotation by $360°/k$ in the counterclockwise sense. These transformations could equivalently be defined in the (z_1, z_2) domain by complex conjugating and/or interchanging the complex variables z_1 and z_2 in the filter transfer function.

An important property of the group is that each transformation distributes over a product of functions of ω_1 and ω_2, that is,

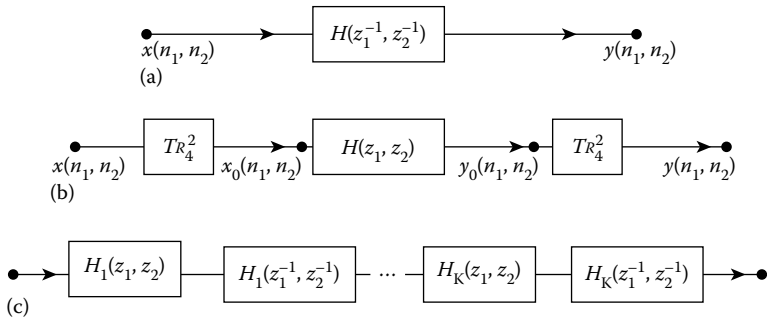


FIGURE 23.3 Design of zero-phase filters: (a) noncausal filter; (b) equivalent causal implementation; (c) cascade zero-phase filter.

$$T_\delta\left[\prod_{i=1}^{K} F_i(\omega_1, \omega_2)\right] = \prod_{i=1}^{K} T_\delta[F_i(\omega_1, \omega_2)] \tag{23.18}$$

where $T\delta$ represents any of the eight transformation operators. The validity of this property follows the definition of the transformations [8].

In the (z_1, z_2) domain, if $H(z_1, z_2)$ represents a causal filter with impulse response $h(n_1, n_2)$, then the filter represented by $H(z_1^{-1}, z_2^{-1})$will have an impulse response $h(-n_1, -n_2)$, and is therefore, noncausal, since $h(-n_1, -n_2) \neq 0$ for $n_1 < 0$, $n_2 < 0$. Such a filter can be implemented in terms of causal transfer function $H(z_1, z_2)$, i.e., by rotating the n_1 and n_2 axes of the input signal by 180°, processing the rotated signal by the causal filter, and then rotating the axes of the output signal by 180°, as illustrated in Figure 23.3b.

Noncausal filters can be used for the realization of zero-phase filters by cascading K pairs of filters whose transfer functions are $H_i(z_1, z_2)$ and $H(z_1^{-1}, z_2^{-1})$ for $i = 1, 2, \ldots, K$, as depicted in Figure 23.3c.

23.2.2.2 Complex Transformation and 2-D Fan Filters

Complex Transformation. A complex transformation is of the form [9]

$$z = e^{\phi} z_1^{\alpha_1/\beta_1} z_2^{\alpha_2/\beta_1} \tag{23.19}$$

by which a 2-D filter $H(z_1, z_2)$ can be derived from 1-D filter $H_1(z)$. The corresponding frequency transformation of Equation 23.19 is

$$\exp(j\omega) \rightarrow \exp\left\{j\left(\phi + \omega_1 \frac{\alpha_1}{\beta_1} + \omega_2 \frac{\alpha_2}{\beta_2}\right)\right\} \tag{23.20}$$

or

$$\omega \rightarrow \phi + \omega_1 \frac{\alpha_1}{\beta_1} + \omega_2 \frac{\alpha_2}{\beta_2} \tag{23.21}$$

There are three major effects of transformation Equation 23.20 on the resulting filter:

1. Frequency shifting along the ω_1 axis. The frequency response of the resulting filter will be shifted by ϕ along the ω_1 axis.
2. Rotation of the frequency response. The angle of rotation is

$$\theta = \arctan\left(\frac{\alpha_2}{\beta_2}\right) \tag{23.22}$$

Since the original filter is 1-D and a function of z_1, the angle of rotation will be defined by the fractional power of z_2.

3. Scaling the frequency response along the ω_1 axis. The fractional power of z_1 will scale the frequency response by a factor β_1/α_1. However, the periodicity of the frequency response will be $(\alpha_1/\beta_1)2\pi$ instead of 2π. Other effects may also be specified [9].

The complex transformation is of importance in the design of fan filters. By using a prototype lowpass filter with a cutoff frequency at $\omega_c = \pi/2$, and the transformation Equation 23.19, one obtains the shifted, scaled, and rotated characteristics in the frequency domain. We denote the transformed filter by

$$H\left(z_1, z_2; \frac{\alpha_1}{\beta_1}, \frac{\alpha_2}{\beta_2}\right) = H_1(z)\Big|_{z=e^{\phi} z_1^{\alpha_1/\beta_1} z_2^{\alpha_2/\beta_2}} \tag{23.23}$$

In general, the filter coefficients in function H will be complex and the variables z_1 and z_2 will have rational noninteger powers. However, appropriate combinations of transformed filters will remove both of these difficulties, as will be shown in the following.

Symmetric Fan Filters. An ideal symmetric fan filter has the specification of

$$H_{f1}(e^{j\omega_1 T}, e^{j\omega_2 T}) = \begin{cases} 1 & \text{for } |\omega_1| \geq |\omega_2| \\ 0 & \text{otherwise} \end{cases} \tag{23.24}$$

We introduce transfer function $\hat{H}_1(z_1, z_2)$, $\hat{H}_2(z_1, z_2)$, $\hat{H}_3(z_1, z_2)$, and $\hat{H}_4(z_1, z_2)$, of four filters generated by Equation 23.23 with $(\alpha_1/\beta_1, \alpha_1/\beta_2) = (1/2, 1/2)$, $(-1/2, 1/2)$, $(1/2, -1/2)$, and $(-1/2, -1/2)$, respectively, and $\phi = \pi/2$. The responses of the transformed filters, $\hat{H}_i(z_1, z_2)$, $i = 1, 2, 3, 4$, can be found in Figure 23.4 together with the prototype filter.

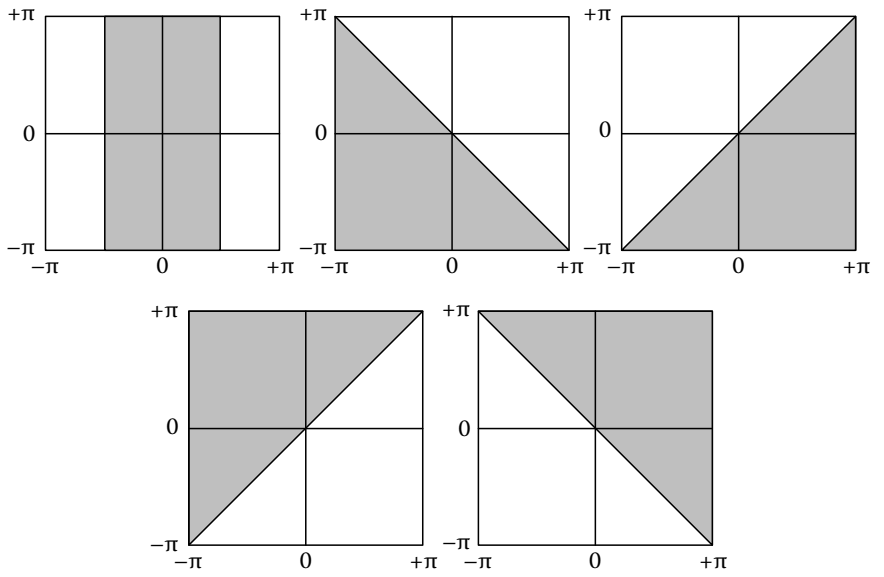


FIGURE 23.4 Basic building blocks of symmetric fan filters.

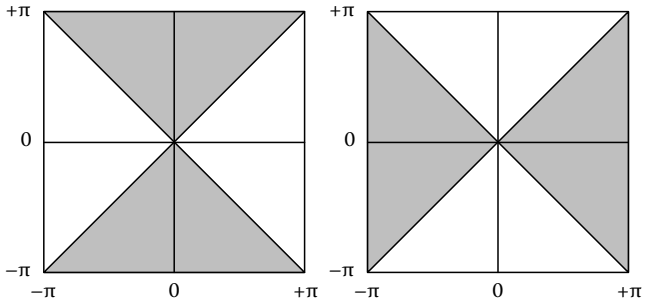


FIGURE 23.5 Amplitude characteristics of $G_{11}(z_1, z_2)$ and $G_{22}(z_1, z_2)$.

In this design procedure, the filters in Figure 23.4 will be used as the basic building blocks for a fan filter specified by Equation 23.24. One can construct the following filter characteristics:

$$G_{11}(z_1,z_2) = \hat{H}_1(z_1,z_2)\hat{H}_1^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_2(z_1,z_2)\hat{H}_2^{\star}\left(z_1^{-1},z_2^{-1}\right) + \hat{H}_3(z_1,z_2)\hat{H}_3^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_4(z_1,z_2)\hat{H}_4^{\star}\left(z_1^{-1},z_2^{-1}\right) \tag{23.25a}$$

$$G_{22}(z_1,z_2) = \hat{H}_1(z_1,z_2)\hat{H}_1^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_3(z_1,z_2)\hat{H}_3^{\star}\left(z_1^{-1},z_2^{-1}\right) + \hat{H}_2(z_1,z_2)\hat{H}_2^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_4(z_1,z_2)\hat{H}_4^{\star}\left(z_1^{-1},z_2^{-1}\right) \tag{23.25b}$$

which are shown in Figure 23.5.

Quadrant Fan Filters. The frequency characteristic of a quadrant fan filter is specified as

$$H_{f2}(e^{j\omega_1 T}, e^{j\omega_2 T}) = \begin{cases} 1 & \text{for } \omega_1\omega_2 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{23.26}$$

We consider the same ideal prototype filter. Then the transformed filters $\hat{H}_{14}(z_1, z_2)$, $\hat{H}_{12}(z_1, z_2)$, $\hat{H}_{23}(z_1, z_2)$, and $\hat{H}_{34}(z_1, z_2)$, are obtained via Equation 23.23 with $(\alpha_1/\beta_1, \alpha_2/\beta_2)$ equal to (1, 0), (0, 1), $(-1, 0)$, and $(0, -1)$, respectively, and $\phi = \pi/2$. The subscripts on $\hat{H}$ refer to quadrants to which the low-pass filter characteristics have been shifted. Figure 23.6 illustrates the amplitude responses of these transformed filters together with the prototype.

The filters in Figure 23.6 will be used as the basic building blocks for fan filters specified by Equation 23.26. In a similar manner to Equations 23.25a and b, the filter characteristics $G_{13}(z_1, z_2)$ and $G_{23}(z_1, z_2)$ can be constructed as follows:

$$G_{13}(z_1,z_2) = \hat{H}_{12}(z_1,z_2)\hat{H}_{12}^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_{14}(z_1,z_2)\hat{H}_{14}^{\star}\left(z_1^{-1},z_2^{-1}\right) + \hat{H}_{23}(z_1,z_2)\hat{H}_{23}^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_{34}(z_1,z_2)\hat{H}_{34}^{\star}\left(z_1^{-1},z_2^{-1}\right) \tag{23.27a}$$

$$G_{24}(z_1,z_2) = \hat{H}_{12}(z_1,z_2)\hat{H}_{12}^{\star}\left(z_1^{-1},z_2^{-1}\right)\hat{H}_{23}(z_1,z_2)\hat{H}_{23}^{\star}\left(z_1^{-1},z_2^{-1}\right) + \hat{H}_{14}(z_1,z_2)\hat{H}_{14}^{\star}\left(z_1^{-1},\ z_2^{-1}\right)\hat{H}_{34}(z_1,z_2)\hat{H}_{34}^{\star}\left(z_1^{-1},z_2^{-1}\right) \tag{23.27b}$$

whose amplitude responses are depicted in Figure 23.7.

23.2.2.3 Constantinides Transformations

The so-called Constantinides [10,19] are of importance in the design of 1-D digital filters, and are of the form

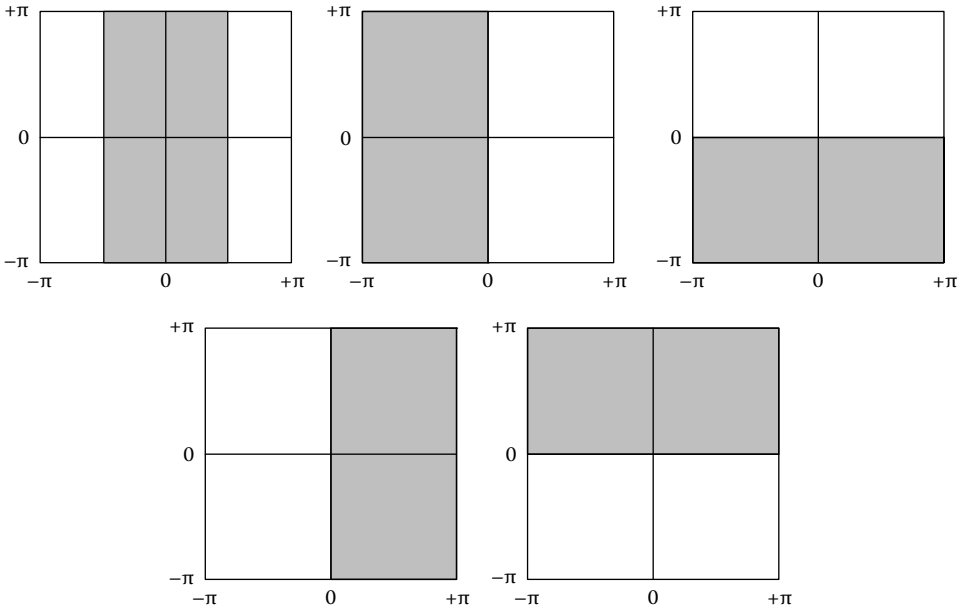


FIGURE 23.6 Basic building blocks for quadrant fan filter design.

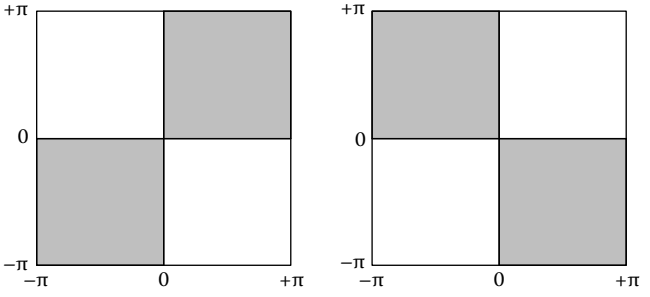


FIGURE 23.7 Two quadrant pass filters.

$$z = f(\bar{z}) = e^{jl\pi} \prod_{t=1}^{m} \frac{\bar{z} - a_t^{\star}}{1 - a_t \bar{z}} \tag{23.28}$$

where

l and m are integers

$a_t^{\star}$ is the complex conjugate of a_t

Pendergrass, Mitra, and Jury [11] showed that, in the decision of 2-D IIR filters, this group of transformations is as useful as in 1-D cases. By choosing the parameters l, m, and a_t in Equation 23.28 properly, a set of four specific transformations can be obtained that can be used to transform a low-pass transfer function into a corresponding low-pass, high-pass, bandpass, or a bandstop transfer function. These transformations are summarized in Table 23.2, where subscript i is included to facilitate the application of the transformation to 2-D discrete transfer functions.

Let Ω_i and ω_i for $i = 1, 2$ be the frequency variables in the original and transformed transfer function, respectively. Suppose $H_L(z_1, z_2)$ is a low-pass transfer function with respect to z_i, if each of z_1 and z_2 is

TABLE 23.2 Constantinides Transformations

Type	Transformation	Parameters
LP to LP	$z_i = \dfrac{\bar{z}_i - \alpha_i}{1 - \alpha_i \bar{z}_i}$	$\alpha_i = \dfrac{\sin\left[(\Omega_{pi} - \omega_{pi})T_i/2\right]}{\sin\left[(\Omega_{pi} + \omega_{pi})T_i/2\right]}$
LP to HP	$z_i = \dfrac{\bar{z}_i - \alpha_i}{1 - \alpha_i \bar{z}_i}$	$\alpha_i = \dfrac{\cos\left[(\Omega_{pi} - \omega_{pi})T_i/2\right]}{\cos\left[(\Omega_{pi} + \omega_{pi})T_i/2\right]}$
LP to BP	$z_i = -\dfrac{\bar{z}_i^2 - \dfrac{2\alpha_i k_i}{k_i+1}\bar{z}_i + \dfrac{k_i-1}{k_i+1}}{1 - \dfrac{2\alpha_i k_i}{k_i+1}\bar{z}_i + \dfrac{k_i-1}{k_i+1}\bar{z}_i^2}$	$\alpha_i = \dfrac{\cos\left[(\omega_{p2i} + \omega_{p1i})T_i/2\right]}{\cos\left[(\omega_{p2i} + \omega_{p1i})T_i/2\right]}$ $k_i = \tan\dfrac{\Omega_{pi}T_i}{2}\cot\dfrac{(\omega_{p2i} - \omega_{p1i})T_i}{2}$
LP to BS	$z_i = -\dfrac{\bar{z}_i^2 - \dfrac{2\alpha_i}{1+k_i}\bar{z}_i + \dfrac{1-k_i}{1+k_i}}{1 - \dfrac{2\alpha_i}{1+k_i}\bar{z}_i + \dfrac{1-k_i}{1+k_i}\bar{z}_i^2}$	$\alpha_i = \dfrac{\cos\left[(\omega_{p2i} + \omega_{p1i})T_i/2\right]}{\cos\left[(\omega_{p2i} + \omega_{p1i})T_i/2\right]}$ $k_i = \tan\dfrac{\Omega_{pi}T_i}{2}\tan\dfrac{(\omega_{p2i} - \omega_{p1i})T_i}{2}$

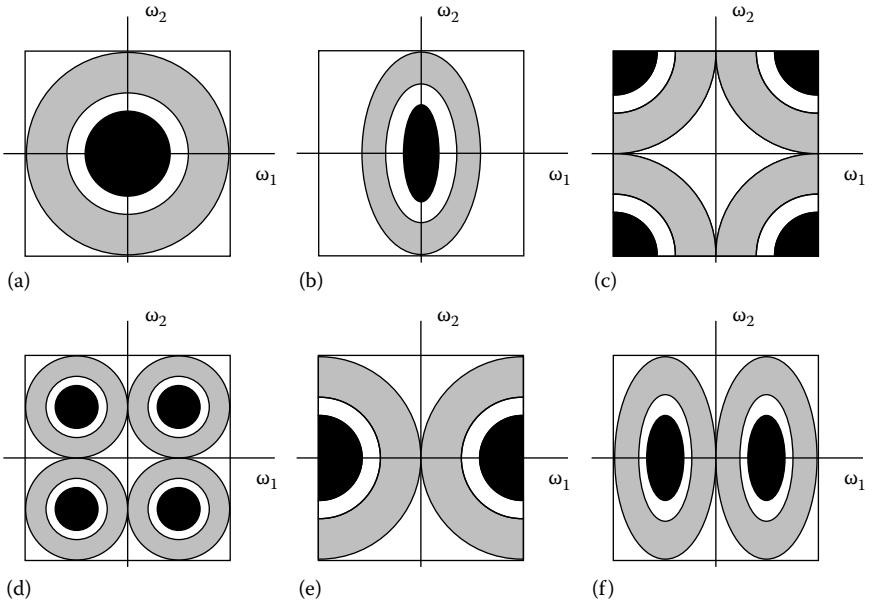


FIGURE 23.8 Application of Constantinides transformations to 2D IIR filters. (a) Circularly symmetric low-pass filter. (b) LP to LP for z_1 and z_2. (c) LP to HP for z_1 and z_2. (d) LP to BP for z_1 and z_2. (e) LP to HP for z_1 and LP to LP for z_2. (f) LP to BP for z_1 and LP to LP for z_2.

transformed to be low-pass, bandpass, or high-pass, then a number of different 2-D filter combinations can be achieved. As an example, some of the possible amplitude responses are illustrated in Figure 23.8a through f [2].

23.3 Design of Separable Product Filters

A 2-D IIR filter is characterized by a separable product transfer function of the form

$$H(z_1, z_2) = H_1(z_1)H_2(z_2) \tag{23.29}$$

if its passband or stopband is of the shape of a rectangular domain. The design of such a class of filters can be accomplished by using the method proposed by Hirano and Aggarwal [12]. The method can be used to design filters with quadrantal or half-plane symmetry.

23.3.1 Design of Quadrantally Symmetric Filters

A 2-D filter is said to be quadrantally symmetric if its amplitude response satisfies the equality

$$|H(z_1, z_2)| = |H(z_1^\star, z_1^\star)| = |H(z_1^\star, z_2)| = |H(z_1, z_1^\star)| \tag{23.30}$$

Consider two 1-D bandpass filters specified by two transfer functions $H_1(z)$ and $H_2(z)$, respectively, and let $z = z_1$ in the first one and $z = z_2$ in the second. If their frequency responses can be expressed by

$$\left|H_1(e^{j\omega_1 T_1})\right| = \begin{cases} 1 & \omega_{12} \le \omega_1 < \omega_{13} \\ 0 & 0 \le \omega_1 < \omega_{11} \text{ or } \omega_{14} \le \omega_1 < \infty \end{cases} \tag{23.31a}$$

and

$$\left|H_2(e^{j\omega_2 T_2})\right| = \begin{cases} 1 & \omega_{22} \le \omega_2 < \omega_{23} \\ 0 & 0 \le \omega_2 < \omega_{21} \text{ or } \omega_{24} \le \omega_2 \le \infty \end{cases} \tag{23.31b}$$

respectively, and since

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \left|H_1(e^{j\omega_1 T_1})\right|\left|H_2(e^{j\omega_2 T_2})\right|$$

Equations 23.31a and b give

$$\left|H(e^{j\omega_1 T_1}, e^{j\omega_2 T_2})\right| = \begin{cases} 1 & \omega_{12} \le \omega_1 < \omega_{13} \text{ and } \omega_{22} \le \omega_2 < \omega_{23} \\ 0 & \text{otherwise} \end{cases}$$

Evidently, the 2-D filter obtained will pass frequency components that are in both passband of $H_1(z_1)$ and $H_2(z_2)$; that is, the passband $H_1(z_1, z_2)$ will be a rectangle with sides $\omega_{13} - \omega_{12}$ and $\omega_{23} - \omega_{22}$. On the other hand, frequency components that are in the stopband of either the first filter or the second filter will be rejected. Hence, the stopband of $H(z_1, z_2)$ consists of the domain obtained by combining the stopbands of the two filters.

By a similar method, if each of the two filters is allowed to be a low-pass, bandpass, or high-pass 1-D filter, then nine different rectangular passbands can be achieved, as illustrated in Figure 23.9. The cascade arrangement of any two of those filters may be referred to as a generalized bandpass filter [12].

Another typical simple characteristic is specified by giving the rectangular stopband region, which is referred to as the rectangular stop filter. Also, there are nine possible types of such a filter which are complementary to that of the generalized bandpass filter shown in Figure 23.9 (i.e., considering the shadowed region as the stopband). This kind of bandstop filter can be realized in the form of

$$H(z_1, z_2) = H_A(z_1)H_A(z_2) - e^{jk\pi}[H_1(z_1)H_2(z_2)]^2 \tag{23.32}$$

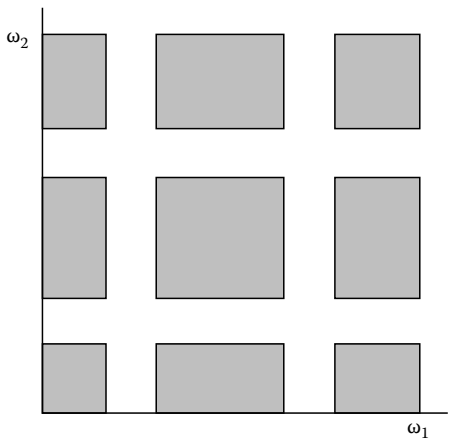


FIGURE 23.9 Idealized amplitude response of generalized bandpass filter.

where $H_1(z_1)H_2(z_2)$ is a generalized bandpass filter described above, and $H_A(z_1)$ and $H_A(z_2)$ are allpass 1-D filters [13] whose poles are poles of $H_1(z_1)$ and $H_2(z_2)$, respectively. Equation 23.32 can be referred to as a generalized bandstop filter.

Extending the principles discussed above, if a 2-D filter is constructed by cascading K 2-D filters with passbands P_i, stopbands S_i, and transfer function $H_i(z_1, z_2)$, the overall transfer function is obtained as

$$H(z_1, z_2) = \prod_{i=1}^{k} H_i(z_i, z_2)$$

and the passband P and stopband S of the cascaded 2-D filter are defined by

$$P = \bigcap P_i \quad \text{and} \quad S = \bigcup_{i=1}^{K} S_i$$

that is, the only frequency components not to be rejected will be those that will be passed by each and every filter in the cascade arrangement.

On the other hand, if a 2-D filter is constructed by connecting K 2-D filters in parallel, then

$$H(z_1, z_2) = \sum_{i=1}^{K} H_i(z_1, z_2)$$

Assuming that all the parallel filters have the same phase shift, and the passband $P_i = 1$ of the various filters are not overlapping, then the passband and stopband of the parallel arrangement is given by

$$P = \bigcup_{i=1}^{K} P_i \quad \text{and} \quad S = \bigcap S_i$$

Parallel IIR filters are more difficult to design than cascade ones, due to the requirement that the phase shifts of the various parallel filters be equal. However, if all the data to be filtered are available at the start of the processing, zero-phase filters can be used.

By combining parallel and cascade subfilters, 2-D IIR filters whose passbands or stopbands are combinations of rectangular regions can be designed, as illustrated by the following example.

Example 23.1

Design a 2-D IIR filter whose passband is the area between two overlapping rectangles, as depicted in Figure 23.10a.

Solution

1. Construct a first 2-D low-pass filter with rectangular passband ($\boldsymbol{\omega}_{12}$, $\boldsymbol{\omega}_{22}$) by using 1-D low-pass filters, as shown in Figure 23.10a.
2. Construct a second 2-D low-pass filter with rectangular passband ($\boldsymbol{\omega}_{11}$, $\boldsymbol{\omega}_{21}$) by 1-D low-pass filters. Then using Equation 23.32 to construct a 2-D high-pass filter with rectangular stopband ($\boldsymbol{\omega}_{11}$, $\boldsymbol{\omega}_{21}$), as shown in Figure 23.10c.
3. Cascade the first 2-D low-pass filter with the 2-D high-pass filter to obtain the required filter. The amplitude response of a practically designed 2-D filter is in Figure 23.10d.

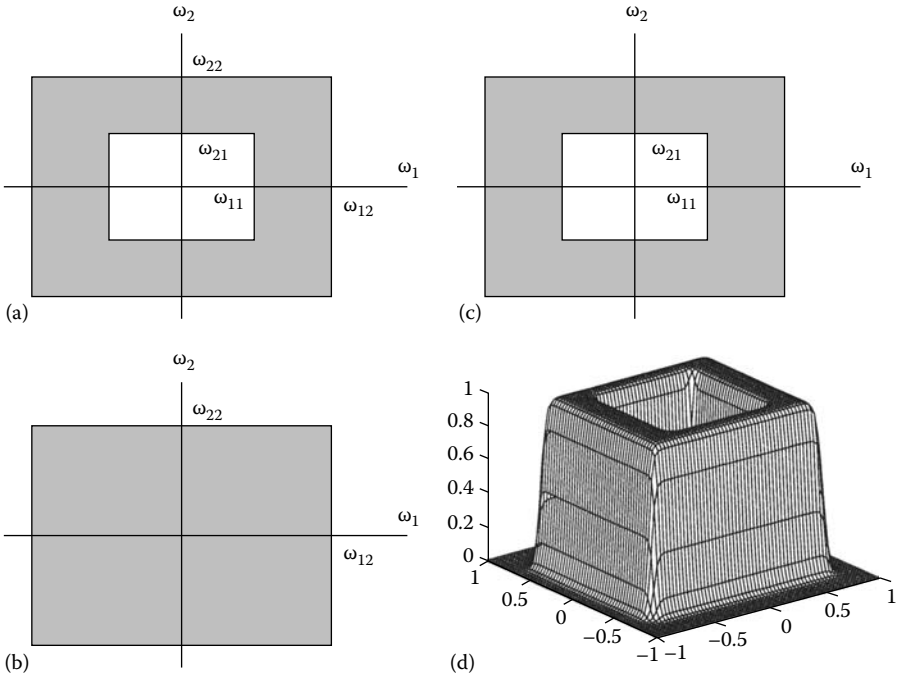


FIGURE 23.10 Amplitude response for the filter of Example 23.1: (a) specified response; (b) response of the 2D low-pass filter; (c) response of the 2D high-pass filter; (d) 3D plot of the amplitude response of a real 2D IIR filter.

23.3.2 Design of Half-Plane Symmetric Filters

A 2-D filter is said to be half-plane symmetric if its amplitude response satisfies

$$\left|H(z_1, z_2)\right| = \left|H(z_1^*, z_2^*)\right| \tag{23.33}$$

but

$$|H(z_1, z_2)| \neq |H(z_1^*, z_2)| \neq |H(z_1, z_2^*)| \tag{23.34}$$

that is, half-plane symmetry does not imply quadrantal symmetry.

The design of those filters can be accomplished by cascading two quadrant pass filters derived previously in Section 23.2.2.2 with quadrantally symmetric filters, as is demonstrated in Example 23.2.

Example 23.2

Design a 2-D half-plane symmetric filter whose passband is defined in Figure 23.11a.

Solution

1. Follow the steps 1 to 3 in Example 23.1 to construct a 2-D bandpass filter with the passband being the area between two overlapping rectangles, as shown in Figure 23.10a.

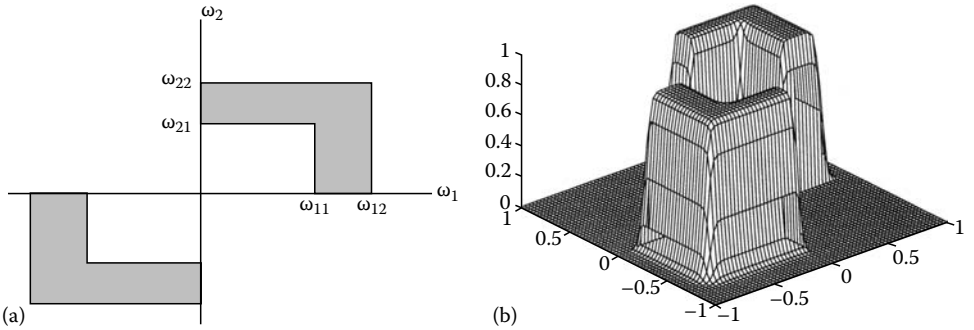


FIGURE 23.11 Amplitude response for the filter of Example 23.2: (a) specified response; (b) 3D plot of the amplitude response of a real 2D IIR filter.

2. Construct a two quadrant (first and third quadrants) pass filter as depicted in Figure 23.7a.
3. Cascade the 2-D bandpass filter with the two quadrant pass filters to obtain the required filter.

The amplitude response of a practically designed filter is shown in Figure 23.11b.

23.3.3 Design of Filters Satisfying Prescribed Specifications

The problem of designing filters that satisfy prescribed specifications to a large extent has been solved for the case of 1-D filters, and by extending the available methods, 2-D IIR filters satisfying prescribed specifications can also be designed.

Assume that we are to design a 2-D low-pass filter with the following specification:

$$|H(\omega_1, \omega_2)| = \begin{cases} 1 \pm \Delta p & \text{for } 0 \le |\omega_i| \le \omega_{pi} \\ \Delta a & \text{for } \omega_{ai} \le |\omega_i| \le \omega_{si}/2, \quad i = 1, 2 \end{cases} \tag{23.35}$$

where ω_{pi} and ω_{ai}, $i = 1, 2$ are passband and stopband edges along the ω_1 and ω_2 axes, respectively. The two 1-D filters that are cascaded to form a 2-D one are specified by ω_{pi}, ω_{ai}, δ_{pi}, and δ_{ai} $(i = 1, 2)$ as the passband edge, stopband edge, passband ripple, and stopband loss, respectively. From Equation 23.29 we have

$$\max\{M(\omega_1, \omega_2)\} = \max\{M_1(\omega_1)\} \max\{M_2(\omega_2)\}$$

and

$$\min\{M(\omega_1, \omega_2)\} = \min\{M_1(\omega_1)\} \min\{M_2(\omega_2)\}$$

Hence the derived 2-D filter will satisfy the specifications of Equation 23.35 if the following constraints are satisfied

$$(1 + \delta_{p1})(1 + \delta_{p2}) \le 1 + \Delta p \tag{23.36}$$

$$(1 - \delta_{p1})(1 - \delta_{p2}) \ge 1 - \Delta p \tag{23.37}$$

$$(1 + \delta_{p1})\delta_{a2} \le \Delta a \tag{23.38}$$

$$(1 + \delta_{p2})\delta_{a1} \le \Delta a \tag{23.39}$$

$$\delta_{a1}\delta_{a2} \le \Delta a \tag{23.40}$$

Constraints 23.36 and 23.37 can be expressed respectively in the alternative form

$$\delta_{p1} + \delta_{p2} + \delta_{p1}\delta_{p2} \leq \Delta p \tag{23.41}$$

and

$$\delta_{p1} + \delta_{p2} - \delta_{p1}\delta_{p2} \leq \Delta p \tag{23.42}$$

Hence if Equation 23.41 is satisfied, Equation 23.42 is also satisfied. Similarly, Constraints 23.38 through 23.40 will be satisfied if

$$\max\{(1 + \delta_{p1})\delta_{a2}, (1 + \delta_{p2})\delta_{a1}\} \leq \Delta a \tag{23.43}$$

since $(1 + \delta_{p1}) \gg \delta_{a1}$ and $(1 + \delta_{p2}) \gg \delta_{a2}$. Now if we assume that $\delta_{p1} = \delta_{p2} = \delta_p$ and $\delta_{a1} = \delta_{a2} = \delta_a$, then we can assign

$$\delta_p = (1 + \Delta p)^{1/2} - 1 \tag{23.44}$$

and

$$\delta_a = \frac{\Delta a}{(1 + \Delta p)^{1/2}} \tag{23.45}$$

so as to satisfy Constraints 23.36 through 23.40. And since $\Delta p \ll 1$, we have

$$\delta_p \approx \frac{\Delta p}{2} \tag{23.46}$$

$$\delta_a \approx \Delta a \tag{23.47}$$

Consequently, if the maximum allowable passband and stopband errors Δp and Δa are specified, the maximum passband ripple A_p and the maximum stopband attenuation A_a and dB for the two 1-D filters can be determined as

$$A_p = 20\log\frac{1}{1 - \delta_p} = 20\log\frac{2}{2 - \Delta p} \tag{23.48}$$

and

$$A_a = 20\log\frac{1}{\delta_p} = 20\log\frac{1}{\Delta a} \tag{23.49}$$

Finally, if the passband and stopband edges ω_{pi} and ω_{ai} are also specified, the minimum order and the transfer function of each of the two 1-D filters can readily be obtained using any of the approaches of the previous sections. Similar treatments for bandpass, bandstop, and high-pass filters can be carried out.

Example 23.3 [12]

Design a zero-phase filter whose amplitude response is specified in Figure 23.12a, with $A_p = 3$ dB, $A_a = 20$ dB.

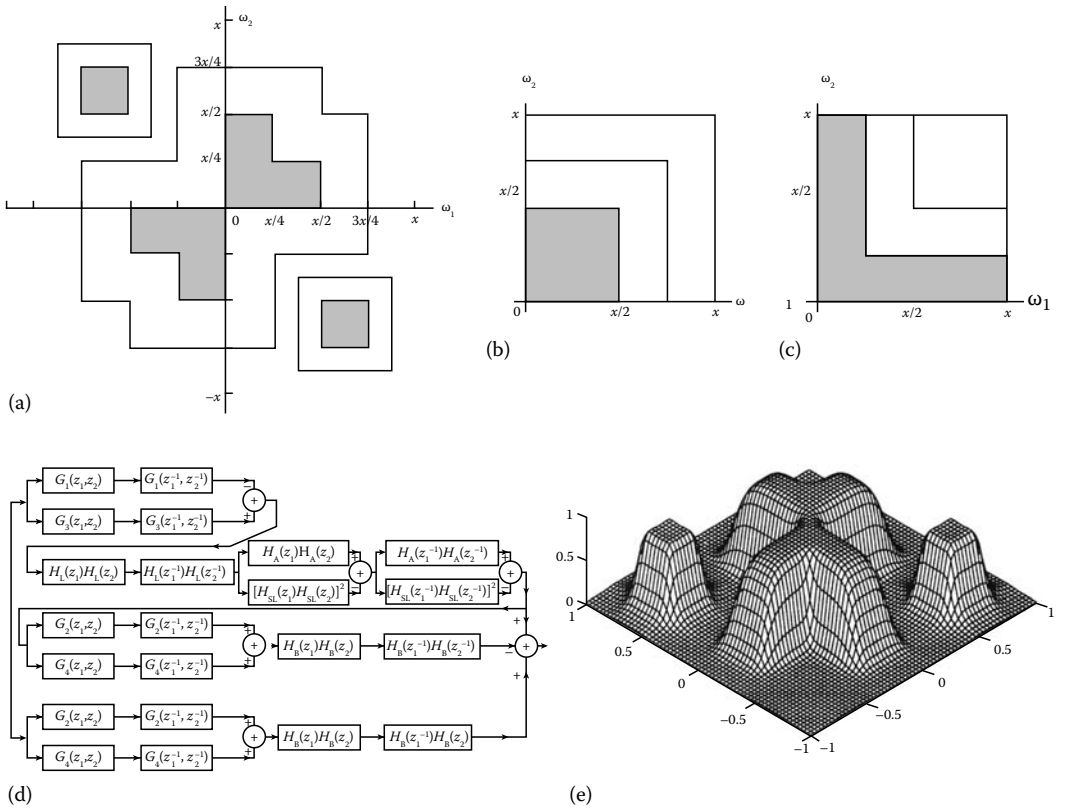


FIGURE 23.12 Amplitude responses of the filters in Example 23.3: (a) the given characteristics; (b) rectangular pass subfilter; (c) rectangular step subfilter; (d) final configuration of the 2D filter; (e) 3D plot of the amplitude response of the resulting 2D filter.

Solution

1. Decomposition: Because there are no contiguous passbands between the characteristics of the first and third quadrants and that of the second and fourth quadrants, in the first step of design, the required 2-D filter can be decomposed into two subfilters $H_{13}(z_1, z_2)$ and $H_{24}(z_1, z_2)$, which represent the characteristics of the first and third quadrants and that of the second and fourth quadrants, respectively. By connecting $H_{13}(z_1, z_2)$ and $H_{24}(z_1, z_2)$, in parallel, the required characteristics of the 2-D filter specified in Figure 23.12a can then be realized.

$$H(z_1, z_2) = H_{13}(z_1, z_2) + H_{24}(z_1, z_2)$$

   (a) Decomposition of $H_{13}(z_1, z_2)$: To accomplish the design of $H_{13}(z_1, z_2)$, further decompositions should be made. The characteristics of the first and third quadrants can be realized by cascading three subfilters, i.e., a two quadrant filter $G_{13}(z_1, z_2)$ [as shown in Figure 23.7a], a low-pass filter $H_L(z_1, z_2)$ [as shown in Figure 23.12b], and a rectangular stop filter $H_S(z_1, z_2)$ [as shown in Figure 23.12c],

$$H_{13}(z_1, z_2) = G_{13}(z_1, z_2)H_L(z_1, z_2)H_S(z_1, z_2)$$

   furthermore, the rectangular stop filter $H_S(z_1, z_2)$ can be designed by allpass filter $H_A(z_1, z_2)$ and low-pass filter $H_{SL}(z_1, z_2)$, using Equation 23.32

$$H_S(z_1, z_2) = H_A(z_1, z_2) - [H_{SL}(z_1, z_2)]^2$$

(b) Decomposition of $H_{24}(z_1, z_2)$: Similarly, $H_{24}(z_1, z_2)$ can be decomposed into the cascade of a two quadrant filter $G_{24}(z_1, z_2)$ [as is shown in Figure 23.7b], and a bandpass filter $H_B(z_1, z_2)$ which can be realized by using two 1-D bandpass filters for both directions.

$$H_{24}(z_1, z_2) = G_{24}(z_1, z_2)H_B(z_1, z_2)$$

The final configuration of the desired filter $H(z_1, z_2)$ is illustrated in Figure 23.12d, with the final transfer function being of the form

$$H(z_1, z_2) = H_{13}(z_1, z_2) + H_{24}(z_1, z_2) - H_{13}(z_1, z_2)H_{24}(z_1, z_2)$$

where the purpose of the term $H_{13}(z_1, z_2)H_{24}(z_1, z_2)$, is to remove the overlap that may be created by adding $H_{13}(z_1, z_2)$ and $H_{24}(z_1, z_2)$.

(2) Design of all the subfilters: At this point, the problem is to derive the two quadrant subfilters $G_{13}(z_1, z_2)$ and $G_{24}(z_1, z_2)$, the low-pass subfilters $H_L(z_1, z_2)$ and $H_{SL}(z_1, z_2)$, the allpass subfilter $H_A(z_1, z_2)$, and the bandstop filter $H_B(z_1, z_2)$.
Note the symmetry of the given characteristics, the identical 1-D sections can be used to develop all the above 2-D subfilters, and the given specifications can easily be combined into the designs of all the 1-D sections.

(3) By connecting all the 2-D subfilters in cascade or parallel as specified in Figure 23.12d, the required 2-D filter is obtained. The 3-D plot of the amplitude response of the final resulting 2-D filter is depicted in Figure 23.12e.

23.4 Design of Circularly Symmetric Filters

23.4.1 Design of LP Filters

As mentioned in Section 23.2.1, rotated filters can be used to design circularly symmetric filters. Costa and Venetsanopoulos [14] and Goodman [15] proposed two methods of this class, based on transforming an analog transfer function or a discrete one by rotated filter transformation, respectively. The two methods lead to filters that are, theoretically, unstable but by using an alternative transformation suggested by Mendonca et al. [16], this problem can be eliminated.

23.4.1.1 Design Based on 1-D Analog Transfer Function

Costa and Venetsanopoulos [14] proposed a method to design circularly symmetric filters. In their method, a set of 2-D analog transfer functions is first obtained by applying the rotated filter transformation in Equation 23.9a for several different values of the rotation angle β to a 1-D analog low-pass transfer function. A set of 2-D discrete low-pass functions are then deduced through the application of the bilinear transformation. The design is completed by cascading the set of 2-D digital filters obtained. The steps involved are as follows.

Step 1. Obtain a stable 1-D analog low-pass transfer function

$$H_{Al}(s) = \frac{Ns}{Ds} = K_0 \frac{\prod_{i=1}^{M}(s - z_{ai})}{\prod_{i=1}^{N}(s - p_{ai})} \tag{23.50}$$

where

z_{ai} and p_{ai} for $i = 1, 2, \ldots,$ are the zeros and poles of $H_{A1}(s)$, respectively

K_0 is a multiplier constant

Step 2. Let β_k for $k = 1, 2, \ldots, K$ be a set of rotation angles defined by

$$\beta_k = \begin{cases} \left(\dfrac{2k-1}{2K} + 1\right)\pi & \text{for even } K \\ \left(\dfrac{k-1}{K} + 1\right)\pi & \text{for even } K \end{cases} \tag{23.51}$$

Step 3. Apply the transformation of Equation 23.9a to obtain a 2-D analog transfer function as

$$|H_{A2k}(s_1, s_2) = H_{Al}(s)|_{s=-s_1 \sin\beta_k + s_2 \cos\beta_k} \tag{23.52}$$

for each rotation angle β_k identified in Step 2.

Step 4. Apply the double bilinear transformation to $H_{A2k}(s_1, s_2)$ to obtain

$$H_{D2k}(z_1, z_2) = |H_{A2k}(s_1, s_2)|_{s_i = 2(z_i - 1)/T_i(z_i + 1)}, \quad i = 1, 2 \tag{23.53}$$

Assuming that $T_1 = T_2 = T$, Equations 23.50 and 23.53 yield

$$H_{D2k}(z_1, z_2) = K_1 \prod_{i=1}^{M_0} H_{2i}(z_1, z_2) \tag{23.54}$$

where

$$H_{2i}(z_1, z_2) = \frac{a_{11i} + a_{21i} z_1 + a_{12i} z_2 + a_{22i} z_1 z_2}{b_{11i} + b_{21i} z_1 + b_{12i} z_2 + b_{22i} z_1 z_2} \tag{23.55}$$

$$K_1 = K_0 \left(\frac{T}{2}\right)^{N-M} \tag{23.56}$$

$$\begin{aligned} a_{11i} &= -\cos\beta_k + \sin\beta_k - \frac{Tz_{ai}}{2} \\ a_{21i} &= -\cos\beta_k - \sin\beta_k - \frac{Tz_{ai}}{2} \quad \text{for } 1 \le i \le M \\ a_{12i} &= \cos\beta_k + \sin\beta_k - \frac{Tz_{ai}}{2} \\ a_{22i} &= \cos\beta_k - \sin\beta_k - \frac{Tz_{ai}}{2} \\ a_{11i} &= a_{12i} = a_{21i} = a_{22i} = 1 \quad \text{for } M \le i \le M_0 \end{aligned} \tag{23.57a}$$

$$\begin{aligned} b_{11i} &= -\cos\beta_k + \sin\beta_k - \frac{Tp_{ai}}{2} \\ b_{21i} &= -\cos\beta_k - \sin\beta_k - \frac{Tp_{ai}}{2} \quad \text{for } 1 \le i \le N \\ b_{12i} &= \cos\beta_k + \sin\beta_k - \frac{Tp_{ai}}{2} \\ b_{22i} &= \cos\beta_k - \sin\beta_k - \frac{Tp_{ai}}{2} \\ b_{11i} &= b_{12i} = b_{21i} = b_{22i} = 1 \quad \text{for } N \le i \le M_0 \end{aligned} \tag{23.57b}$$

and

$$M_0 = \max(M, N)$$

Step 5. Cascade the filters obtained in Step 4 to yield an overall transfer function

$$H(z_1, z_2) = \prod_{k=1}^{K} H_{D2k}(z_1, z_2)$$

It is easy to find that, at point $(z_1, z_2) = (-1, -1)$, both the numerator and denominator polynomials of $H_{2i}(z_1, z_2)$ assume the value of zero. And thus each $H_{2i}(z_1, z_2)$ has nonessential singularity of the second kind on the unit bicircle

$$U^2 = \{(z_1, z_2):\ |z_1| = 1, |z_2| = 1\}$$

The nonessential singularity of each $H_{2i}(z_1, z_2)$ can be eliminated and, furthermore, each subfilter can be stabilized by letting

$$b'_{12i} = b_{12i} + \varepsilon b_{11i} \tag{23.58a}$$

$$b'_{22i} = b_{22i} + \varepsilon b_{21i} \tag{23.58b}$$

where ε is a small positive constant. With this modification, the denominator polynomial of each $H_{2i}(z_1, z_2)$ is no longer zero and, furthermore, the stability of the subfilter can be guaranteed if

$$\mathrm{Re}(p_{ai}) < 0 \tag{23.59}$$

and

$$270^\circ < \beta_k < 360^\circ \tag{23.60}$$

As can be seen in Equation 23.51, half of the rotation angles are in the range $180^\circ < \beta_k < 270^\circ$ and according to the preceding stable conditions they yield unstable subfilters. However, the problem can easily be overcome by using rotation angles in the range given by Equation 23.60 and the rotating the transfer function of the subfilter by -90° using linear transformations described in Section 23.2.1. For example, an effective rotation angle $\beta_k = 225^\circ$ is achieved by rotating the input data by 90°, filtering using a subfilter rotated by 315°, and then rotating the output data by -90°, as shown in Figure 23.13.

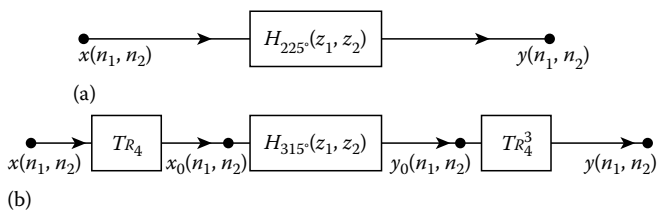


FIGURE 23.13 Realization of subfilter for rotation angle in the range $180^\circ < \beta_k < 270^\circ$. (a) An effective rotation angle $\beta_k = 225^\circ$ achieved by rotating the input by 90°. (b) Filtering using a subfilter rotated by 315°, and then rotating the output data by –90°.

In addition, a 2-D zero-phase filter can be designed by cascading subfilters for rotation angles $\pi + \beta_k$ for $k = 1, 2, \ldots, N$. The resulting transfer function is given by

$$H(z_1, z_2) = \prod_{k=1}^{K} H_{D2k}(z_1, z_2) H_{D2k}(z_1, z_2)$$

where the noncausal sections can be realized as illustrated in Figure 23.3.

23.4.1.2 Design Based on 1-D Discrete Transfer Function

The method proposed by Goodman [15] is based on the 1-D discrete transfer function transformation. In the method, a 1-D discrete transfer is first obtained by applying the bilinear transformation to a 1-D analog transfer function. Then, through the application of an allpass transformation that rotates the contours of the amplitude of the 1-D discrete transfer function, a corresponding 2-D transfer function is obtained. The steps involved are as follows.

Step 1. Obtain a stable 1-D analog low-pass transfer function $H_{A1}(s)$ of the form given by Equation 23.50.

Step 2. Apply the bilinear transformation to $H_{A1}(s)$ to obtain

$$H_{D1}(z) = H_{Al}(s)|_{s=2(z-1)/T(z+1)} \tag{23.61}$$

Step 3. Let β_k for $k = 1, 2, \ldots, K$ be a set of rotation angles given by Equation 23.51.

Step 4. Apply the allpass transformation defined by

$$z = f_k(z_1, z_2) = \frac{1 + c_k z_1 + d_k z_2 + e_k z_1 z_2}{e_k + d_k z_1 + c_k z_2 + z_1 z_2} \tag{23.62}$$

where

$$c_k = \frac{1 + \sin\beta_k + \cos\beta_k}{1 - \sin\ \beta_k + \cos\beta_k} \tag{23.63a}$$

$$d_k = \frac{1 - \sin\ \beta_k - \cos\ \beta_k}{1 - \sin\beta_k + \cos\beta_k} \tag{23.63b}$$

$$e_k = \frac{1 + \sin\beta_k - \cos\beta_k}{1 - \sin\beta_k + \cos\beta_k} \tag{23.63c}$$

to obtain the 2-D discrete transfer function

$$H_{D2k}(z_1, z_2) = H_{D1}(z)\Big|_{z=f_k(z_1,z_2)} \tag{23.64}$$

for $k = 1, 2, \ldots, K$. The procedure yields the 2-D transfer function of Equation 23.64, as can be easily demonstrated, and by cascading the rotated subfilters $H_{D2k}(z_1, z_2)$ the design can be completed.

The method of Goodman is equivalent to that of Costa and Venetsanopoulos and consequently, the resulting filter is subject to the same stability problem due to the nonessential singularity of the second kind at point $(z_1, z_2) = (-1, -1)$. To achieve a stable design, Goodman suggested that the transfer

function $H_{D2k}(z_1, z_2)$ for $k = 1, 2, \ldots, K$ be obtained directly by minimizing an appropriate objective function subject to the constraints

$$\begin{aligned} c_k + d_k - e_k &\leq 1 - \varepsilon \\ c_k - d_k + e_k &\leq 1 - \varepsilon \\ -c_k + d_k + e_k &\leq 1 - \varepsilon \\ -c_k - d_k - e_k &\leq 1 - \varepsilon \end{aligned}$$

through an optimization procedure. If ε is a small positive constant, the preceding constraints constitute necessary and sufficient conditions for stability and, therefore, such an approach will yield a stable filter.

23.4.1.3 Elimination of Nonessential Singularities

To eliminate the nonessential singularities in the preceding two methods, Mendonca et al. [16] suggested a new transformation of the form

$$s = g_5(s_1, s_2) = \frac{\cos\beta_k s_1 + \sin\beta_k s_2}{1 + cs_1 s_2} \tag{23.65}$$

by combining the transformations in Equations 23.9a and 23.11 to replace the transformation Equation 23.9a. If we ensure that

$$\cos\beta_k > 0, \quad \sin\beta_k > 0, \quad \text{and} \quad c > 0$$

then the application of this transformation followed by the application of the double bilinear transformation yields stable 2-D digital filters that are free of nonessential singularities of the second kind. If, in addition

$$c = \frac{1}{\omega_{\max}^2}$$

then local-type preservation can be achieved on the set Ω_2 given by

$$\Omega_2 = \{(\omega_1, \omega_2):\ \omega_1 \geq 0,\ \omega_2 \geq 0,\ \omega_1\omega_2 \leq \omega_{\max}\}$$

and if $\omega_{\max} \to \infty$, then a global-type preservation can be approached as closely as desired.

By using the transformation of Equation 23.65 instead of that in Equation 23.9a in the method of Costa and Venetsanopoulos, the transfer function of Equation 23.54 becomes

$$H_{D2k}(z_1, z_2) = K_1 P_{D2}(z_1, z_2) \times \prod_{i=1}^{M_0} \frac{a_{11i} + a_{21i}z_1 + a_{12i}z_2 + a_{22i}z_1z_2}{b_{11i} + b_{21i}z_1 + b_{12i}z_2 + b_{22i}z_1z_2} \tag{23.66}$$

where

$$K_1 = K_0\left(\frac{T}{2}\right)^{N-M} \tag{23.67a}$$

$$P_{D2}(z_1, z_2) = \left[1 + \frac{4c}{T^2} + \left(1 - \frac{4c}{T^2}\right)z_1 + \left(1 - \frac{4c}{T^2}\right)z_2 + \left(1 + \frac{4c}{T^2}\right)z_1z_2\right]^{N-M} \tag{23.67b}$$

and

$$
\begin{aligned}
a_{11i} &= -\cos\beta_k - \sin\beta_k - \left(\frac{T}{2} + \frac{2c}{T}\right) z_{ai} \\
a_{21i} &= \cos\beta_k - \sin\beta_k - \left(\frac{T}{2} + \frac{2c}{T}\right) z_{ai} \quad \text{for } 1 \le i \le M \\
a_{12i} &= -\cos\beta_k + \sin\beta_k - \left(\frac{T}{2} - \frac{2c}{T}\right) z_{ai} \\
a_{22i} &= \cos\beta_k + \sin\beta_k - \left(\frac{T}{2} + \frac{2c}{T}\right) z_{ai} \\
a_{11i} &= a_{21i} = a_{12i} = a_{22i} = 1 \quad \text{for } M \le i \le M_0 \\
b_{11i} &= -\cos\beta_k - \sin\beta_k - \left(\frac{T}{2} + \frac{2c}{T}\right) p_{ai} \\
b_{21i} &= \cos\beta_k - \sin\beta_k - \left(\frac{T}{2} - \frac{2c}{T}\right) p_{ai} \quad \text{for } 1 \le i \le N \\
b_{12i} &= -\cos\beta_k + \sin\beta_k - \left(\frac{T}{2} - \frac{2c}{T}\right) p_{ai} \\
b_{22i} &= \cos\beta_k + \sin\beta_k - \left(\frac{T}{2} - \frac{2c}{T}\right) p_{ai} \\
b_{11i} &= b_{21i} = b_{12i} = b_{22i} = 1 \quad \text{for } N \le i \le M_0 \\
M_0 &= \max(M, N)
\end{aligned}
$$

An equivalent design can be obtained by applying the allpass transformation of Equation 23.62 in Goodman's method with

$$
\begin{aligned}
c_k &= \frac{1 + \cos\beta_k - \sin\beta_k - 4c/T^2}{1 - \cos\beta_k - \sin\beta_k + 4c/T^2} \\
d_k &= \frac{1 - \cos\beta_k + \sin\beta_k - 4c/T^2}{1 - \cos\beta_k - \sin\beta_k + 4c/T^2} \\
c_k &= \frac{1 + \cos\beta_k + \sin\beta_k + 4c/T^2}{1 - \cos\beta_k - \sin\beta_k + 4c/T^2}
\end{aligned}
$$

23.4.2 Realization of HP, BP, and BS Filters

Consider two zero-phase rotated subfilters that were obtained from a 1-D analog high-pass transfer function using rotating angles $-\beta_1$ and β_1, where $0^\circ < \beta_1 < 90^\circ$. The idealized contour plots of the two subfilters are shown in Figure 23.14a and b. If these two subfilters are cascaded, the amplitude response of the combination is obtained by multiplying the amplitude responses of the subfilters at corresponding points. The idealized contour plot of the composite filter is thus obtained as illustrated in Figure 23.14c. As can be seen, the contour plot does not represent the amplitude response of a 2-D circularly symmetric high-pass filter, and, therefore, the design of high-pass filters cannot readily be achieved by simply cascading rotated subfilters as in the case of low-pass filters. However, the design of these filters can be accomplished through the use of a combination of cascade and parallel subfilters [16].

If the above rotated subfilters are connected in parallel, we obtain a composite filter whose contour plot is shown in Figure 23.14d. By subtracting the output of the cascade filter from the output of the parallel filter, we achieve an overall filter whose contour plot is depicted in Figure 23.14e. Evidently, this plot

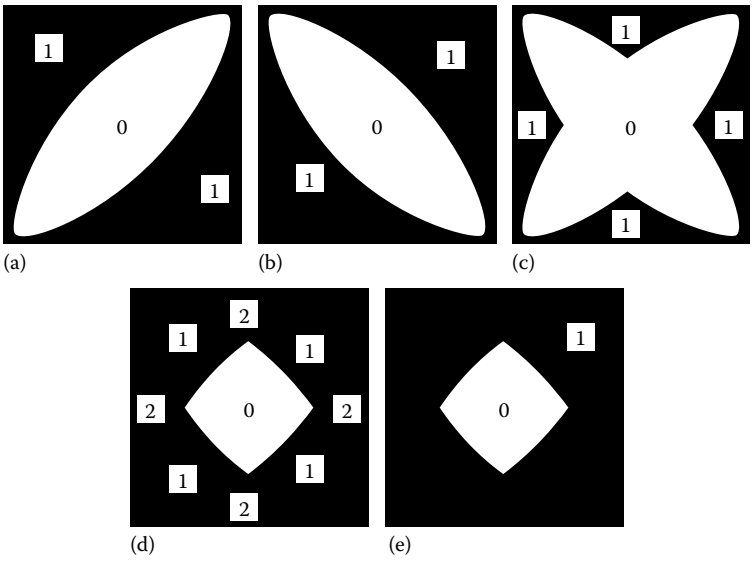


FIGURE 23.14 Derivation of 2D high-pass filter. (a) Contour plot of subfilter rotated by $-\beta_1$. (b) Contour plot of subfilter rotated by β_1. (c) Contour plot of subfilters in (a) and (b) connected in cascade. (d) Contour plot of subfilters in (a) and (b) connected in parallel. (e) Contour plot obtained by subtracting the amplitude response in (c) from that of (d).

resembles the idealized contour plot of a 2-D circularly symmetric high-pass filter, and, in effect, following this method a filter configuration is available for the design of high-pass filters.

The transfer function of the 2-D high-pass filter is then given by

$$\hat{H}_1 = \overline{H}_1 = H^{\beta_1} + H^{-\beta_1} - H^{\beta_1}H^{-\beta_1} \tag{23.68}$$

where

$$H^{\beta_1} = H_1(z_1, z_2)H_1(z_1^{-1}, z_2^{-1})$$

and

$$H^{-\beta_1} = H_1\left(z_1, z_2^{-1}\right)H_1\left(z_1^{-1}, z_2\right)$$

represent zero-phase subfilters rotated by angle β_1 and $-\beta_1$, respectively.

The above approach can be extended to two or more rotation angles in order to improve the degree of circularity. For N rotation angles, $\hat{H}_N$ is given by the recursive relation

$$\hat{H}_N = \hat{H}_{N-1} + \overline{H}_N - \hat{H}_{N-1}\overline{H}_N \tag{23.69}$$

where

$$\overline{H}_N = H^{\beta_N} + H^{-\beta_N} - H^{\beta_N}H^{-\beta_N}$$

and $\hat{H}_{N-1}$ can be obtained from $\overline{H}_{N-1}$ and $\hat{H}_{N-2}$. The configuration obtained is illustrated in Figure 23.15, where the realization of $\hat{H}_{N-1}$ is of the same as that of $\hat{H}_N$.

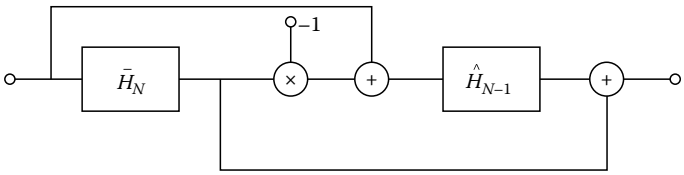


FIGURE 23.15 2D high-pass filter configuration.

As can be seen in Figure 23.15, the complexity of the high-pass configuration tends to increase rapidly with the number of rotations, and consequently, the number of rotations should be kept to a minimum. It should also be mentioned that the coefficients of the rotated filters must be properly adjusted, by using Equation 23.65, to ensure that the zero-phase is approximated. However, the use of this transformation leads to another problem: the 2-D digital transfer function obtained has spurious zeros at the Nyquist points. These zeros are due to the fact that the transformation in Equation 23.65 does not have type preservation in the neighborhoods of the Nyquist points but their presence does not appear to be of serious concern.

With the availability of circularly symmetric low-pass and high-pass filters, bandpass and bandstop filters with circularly symmetric amplitude responses can be readily obtained. A bandpass filter can be obtained by connecting a low-pass filter and a high-pass filter with overlapping passbands in cascade, whereas a bandstop filter can be realized by connecting a low-pass filter and a high-pass filter with overlapping passbands in parallel.

23.4.3 Design of Filters Satisfying Prescribed Specifications

A similar approach to that described in Section 23.3.3 can be used for the design of circularly symmetric filters satisfying prescribed specifications. Assume that the maximum/minimum passband and the maximum stopband gain of the 2-D filter are $(1 \pm \Delta p)$ and Δa, respectively, if K rotated filter sections are cascaded where half of the rotations are in the range of 180°–270° and the other half are in the range 270°–360°. Then, we can assign the passband ripple δ_p and the stopband loss δ_a to be [2]

$$\delta_p = \frac{\Delta p}{K} \tag{23.70}$$

and

$$\delta_a = \Delta a^{2/K} \tag{23.71}$$

The lower (or upper) bound of the passband gain would be achieved if all the rotated sections were to have minimum (or maximum) passband gains at the same frequency point. Although it is possible for all the rotated sections to have minimum (or maximum) gains at the origin of the (ω_1, ω_2) plane, the gains are unlikely to be maximum (or minimum) together at some other frequency point and, in effect, the preceding estimate for δ_p is low. A more realistic value for δ_p is

$$\delta_p = \frac{2\Delta p}{K} \tag{23.72}$$

If Δp and Δa are prescribed, then the passband ripple and minimum stopband attenuation of the analog filter can be obtained from Equations 23.72 and 23.71 as

$$A_p = 20 \log \left[\frac{K}{K - 2\Delta p} \right] \tag{23.73}$$

and

$$A_a = \frac{40}{K} \log\left[\frac{1}{\Delta a}\right] \tag{23.74}$$

If the passband Ω_p and stopband Ω_a are also prescribed, the minimum order and the transfer function of the analog filter can be determined using the method in preceding sections.

Example 23.4

Using the method of Costa and Venetsanopoulos, design a circularly symmetric low-pass filter satisfying the following specifications:

$$\omega_{s1} = \omega_{s2} = 2\pi \text{ rad/s}$$
$$\omega_p = 0.4\pi \text{ rad/s}, \quad \omega_a = 0.6\pi \text{ rad/s}$$
$$\delta_p = \delta_a = 0.1$$

Solution

The filter satisfying prescribed specifications can be designed through the following steps:

1. Select a prototype of approximation and suitably select the number of rotations K.
2. Calculate rotation angles by Equation 23.51.
3. Determine A_p and A_a from δ_p and δ_a, respectively, using Equations 23.73 and 23.74. Calculate the prewarped Ω_p and Ω_a, from ω_p and ω_a, respectively.
4. Use above calculated specifications to obtain the prewarped 1-D analog transfer function.
5. Apply the transformations of Equations 23.52, 23.53, 23.57, and 23.58 to obtain K rotated subfilters.
6. Cascade all the rotated subfilters.

The 3-D plot of the amplitude response of the resulting filter is shown in Figure 23.16, where $K = 10$.

23.5 Design of 2-D IIR Filters by Optimization

In the preceding sections, several methods for the solution of approximation problems in 2-D IIR filters have been described. These methods lead to a complete description of the transfer function in closed form, either in terms of its zeros and poles or its coefficients. They are, as a consequence, very efficient and lead to very precise designs. Their main advantage is that they are applicable only for the design of filters with piecewise-constant amplitude responses. In the following sections, the optimization methods for the design of 2-D IIR filters are considered. In these methods, a discrete transfer function is assumed and an error function is formulated on the basis of some desired amplitude and/or phase response. These methods are iterative and, as a result, they usually involve a large amount of computation. However, unlike the closed-form methods, they are suitable for the design of filters having arbitrary amplitude or phase responses.

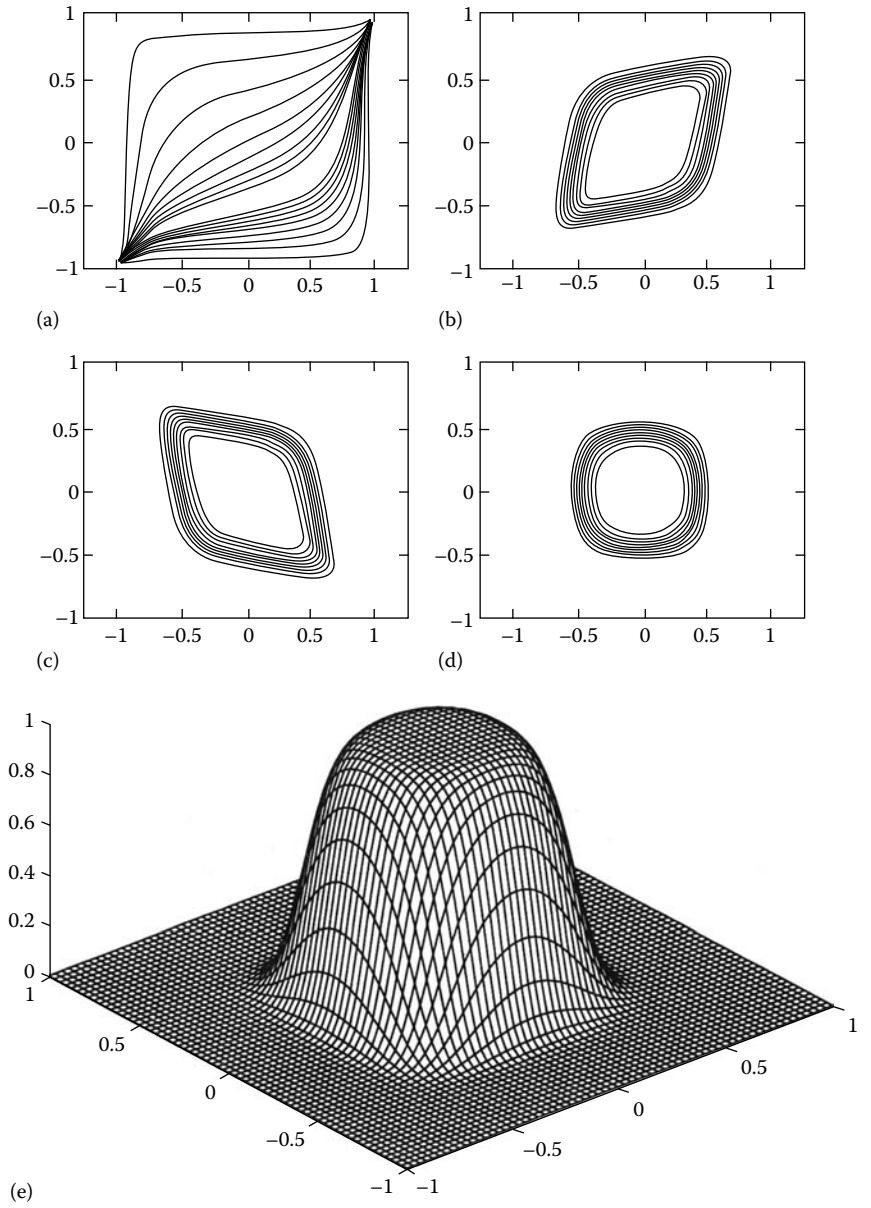


FIGURE 23.16 Amplitude response of a circularly symmetric filter in Example 23.4. (a) Subfilter for rotation angle of 243°. (b) Subfilters for rotation angles of 189°, 207°, 225°, 243°, and 261° in cascade. (c) Subfilters for rotation angles of 279°, 297°, 315°, 333°, and 351° in cascade. (d) All 10 subfilters in cascade. (e) 3D plot of the amplitude response of the resulting 2D low-pass filter.

23.5.1 Design by Least *p*th Optimization

The least *p*th optimization method has been used quite extensively in the past in a variety of applications. In this approach, an objective function in the form of a sum of elemental error functions, each raised to the *p*th power, is first formulated and is then minimized using any one of the available unconstrained optimization methods [17].

23.5.1.1 Problem Formulation

Consider the transfer function

$$H(z_1, z_2) = H_0 \prod_{k=1}^{K} \frac{N_k(z_1, z_2)}{D_k(z_1, z_2)} = H_0 \prod_{k=1}^{K} \frac{\sum_{l=0}^{L_{1k}} \sum_{m=0}^{M_{1k}} a_{lm}^{(k)} \, z_1^{-1} \, z_2^{-m}}{\sum_{l=0}^{L_{2k}} \sum_{m=0}^{M_{2k}} b_{lm}^{(k)} \, z_1^{-1} \, z_2^{-m}} \tag{23.75}$$

where $N_k(z_1, z_2)$ and $D_k(z_1, z_2)$ are polynomials of order equal to or less than 2 and H_0 is a constant, and let

$$\mathbf{x} = [\mathbf{a}^{\mathrm{T}} \quad \mathbf{b}^{\mathrm{T}} \quad H_0]^{\mathrm{T}} \tag{23.76}$$

where

$$\mathbf{a} = \left[a_{10}^{(1)} \quad a_{20}^{(1)} \quad \ldots \quad a_{L_{11}M_{11}}^{(1)} \quad a_{10}^{(2)} \quad a_{20}^{(2)} \quad \ldots \quad a_{L_{12}M_{12}}^{(2)} \quad \ldots \quad a_{L_{1K}M_{1K}}^{(\mathrm{K})} \right]^T$$

and

$$\mathbf{b} = \left[b_{10}^{(1)} \quad b_{20}^{(1)} \quad \ldots \quad b_{L_{11}M_{11}}^{(1)} \quad b_{10}^{(2)} \quad a_{20}^{(2)} \quad \ldots \quad b_{L_{12}M_{12}}^{(2)} \quad \ldots \quad b_{L_{2K}M_{2K}}^{(\mathrm{K})} \right]^T$$

are row vectors whose elements are the coefficients of $N_k(z_1, z_2)$ and $D_k(z_1, z_2)$, respectively. An objective function can be defined in terms of the L_p norm of $\mathbf{E}(\mathbf{x})$ as

$$J(\mathbf{x}) = L_p = \| \mathbf{E}(\mathbf{x}) \|_p = \left[\sum_{i=1}^{K} |E_{mn}(\mathbf{x})|^p \right]^{1/p} \tag{23.77}$$

where p is an even positive integer

$$E_{mn}(\mathbf{x}) = \mathbf{M}(\mathbf{m}, \mathbf{n}) - \mathbf{M}_1(m, n) \tag{23.78}$$

and

$$M(m, n) = \left| H(e^{j\omega_{1m}T_1}, e^{j\omega_{2n}T_2}) \right|, \quad m = 1, \ldots, M, \; n = 1, \ldots, N$$

are samples of the amplitude response of the filter at a set of frequency pairs (ω_{1m}, ω_{2n}) ($m = 1, \ldots, M$, $n = 1, \ldots, N$) with

$$\omega_{1m} = \frac{\omega_{s1}(m-1)}{2(M-1)} \quad \text{and} \quad \omega_{2n} = \frac{\omega_{s2}(n-1)}{2(N-1)}$$

$M_1(m, n)$ represents the desired amplitude response at frequencies (ω_{1m}, ω_{2n}).

Several special cases of the L_p norm are of particular interest. The L_1 norm, namely

$$L_1 = \sum_{i=1}^{K} \left| E_{mn}(x) \right|$$

is the sum of the magnitudes of the elements of $\mathbf{E}(\mathbf{x})$; the L_2 norm given by

$$L_2 = \left[\sum_{i=1}^{K} |E_{mn}(\mathbf{x})|^2\right]^{1/2}$$

is the well-known Euclidean norm; and L_2^2 is the sum of the squares of the elements of $\boldsymbol{E}(\mathbf{x})$. In the case where $p \to \infty$ and

$$E_M(\mathbf{x}) = \max_{m,n} \{E_{mn}(\mathbf{x})\} \neq 0$$

we can write

$$\begin{aligned} L_\infty &= \lim_{\mathrm{p}\to\infty} \left\{\sum_{k=1}^{K} |E_{mn}(\mathbf{x})|^p\right\}^{1/\mathrm{p}} \\ &= E_M(\mathbf{x}) = \lim_{p\to\infty} \left\{\sum_{k=1}^{K} \left|\frac{E_{mn}(\mathbf{x})}{E_M(\mathbf{x})}\right|^p\right\}^{1/\mathrm{p}} \\ &= E_M(\mathbf{x}) \end{aligned} \tag{23.79}$$

The design task at hand amounts to finding a parameter vector $\mathbf{x}$ that minimizes the least pth objective function $J(\mathbf{x})$ defined in Equation 23.77. If $J(\mathbf{x})$ is defined in terms of L_2^2, a least-squares solution is obtained; if the L_∞ norm is used, a so-called minimax solution is obtained, since in this case the largest element in $E(\mathbf{x})$ is minimized.

23.5.1.2 Quasi-Newton Algorithms

The design problem described above can be solved by using any one of the standard unconstrained optimization algorithms. A class of such algorithms that has been found to be very versatile, efficient, and robust is the class of quasi-Newton algorithms [17–19]. These are based on the principle that the minimum point $\mathbf{x}^*$ of a quadratic convex function $J(\mathbf{x})$ of N variables can be obtained by applying the correction

$$\delta = -\mathbf{H}^{-1}\gamma$$

to an arbitrary point $\mathbf{x}$, that is

$$\mathbf{x}^\star = \mathbf{x} + \mathbf{d}$$

where vector

$$\mathbf{g} = \nabla J(\mathbf{x}) = \left[\frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2}, \ldots, \frac{\partial J}{\partial x_N}\right]^{\mathrm{T}}$$

and $N \times N$ matrix

$$\mathbf{H} = \begin{vmatrix} \dfrac{\partial^2 J(\mathbf{x})}{\partial x_1^2} & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_1 \partial x_N} \\ \dfrac{\partial^2 J(\mathbf{x})}{\partial x_2 \partial x_1} & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_2^2} & \cdots & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_2 \partial x_N} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial^2 J(\mathbf{x})}{\partial x_N \partial x_1} & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_N \partial x_2} & \cdots & \dfrac{\partial^2 J(\mathbf{x})}{\partial x_N^2} \end{vmatrix}$$

are the gradient vector and Hessian matrix of $J(\mathbf{x})$ at point $\mathbf{x}$, respectively.

The basic quasi-Newton algorithm as applied to the 2-D IIR filter design problem is as follows [20].

ALGORITHM 1: Basic Quasi-Newton Algorithm

Step 1. Input $\mathbf{x}_0$ and ε. Set $\mathbf{S}_0 = \mathbf{I}_N$, where $\mathbf{I}_N$ is the $N \times N$ unity matrix and N is the dimension of $\mathbf{x}$, and set $k = 0$. Compute $\mathbf{g}_0 = -\mathbf{J}(\mathbf{x}_0)$.
Step 2. Set $\mathbf{d}_k = -\mathbf{S}_k\,\mathbf{g}_k$ and find α_k, the value of α that minimizes $\mathbf{J}(\mathbf{x}_k + \alpha\mathbf{d}_k)$, using a line search.
Step 3. Set $\mathrm{d}_k = \alpha_k\mathbf{d}_k$ and $\mathrm{x}_{k+1} = \mathbf{x}_k + \mathrm{d}_k$.
Step 4. If $\|\delta_K\| < \varepsilon$, then output $\mathbf{x}^* = \mathbf{x}_{k+1}$, $\mathbf{J}(\mathbf{x}^*) = \mathbf{J}(\mathbf{x}_{k+1})$ and stop, else go to step 5.
Step 5. Compute $\mathbf{g}_{k+1} = J(\mathbf{x}_{k+1})$ and set $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.
Step 6. Compute $\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{C}_k$, where $\mathbf{C}_k$ is a suitable matrix correction.
Step 7. Check $\mathbf{S}_{k+1}$ for positive definiteness and if it is found to be nonpositive definite force it to become positive definite.
Step 8. Set $k = k + 1$ and go to step 2.

The correction matrix C_k required in step 6 can be computed by using either the Davidon–Fletcher–Powell (DFP) formula

$$\mathbf{C}_k = \frac{\mathbf{d}_k\mathbf{d}_k^{\mathrm{T}}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{g}_k}\,\frac{\mathbf{S}_k\mathbf{g}_k\mathbf{g}_k^{\mathrm{T}}\mathbf{S}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{S}_k\mathbf{g}_k} \tag{23.80}$$

or the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula

$$\mathbf{C}_k = \left(1 + \frac{\mathbf{g}_k^{\mathrm{T}}\mathbf{S}_k\mathbf{g}_k}{\mathbf{d}_k^{\mathrm{T}}\mathbf{d}_k}\right)\quad\frac{\mathbf{d}_k\mathbf{d}_k^{\mathrm{T}}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k} - \frac{\mathbf{d}_k\mathbf{g}_k^{\mathrm{T}}\mathbf{S}_k + \mathbf{S}_k\mathbf{g}_k\mathbf{d}_k^{\mathrm{T}}}{\mathbf{g}_k^{\mathrm{T}}\mathbf{d}_k} \tag{23.81}$$

Algorithm 1 eliminates the need to calculate the second derivatives of the objective function; in addition, the matrix inversion is unnecessary. However, matrices $\mathbf{S}_1$, $\mathbf{S}_2, \ldots,$ $\mathbf{S}_k$ need to be checked for positive definiteness and may need to be manipulated. This can be easily done in practice by diagonalizing $\mathbf{S}_{k+1}$ and then replacing any nonpositive diagonal elements with corresponding positive ones. However, this would increase the computational burden quite significantly. The amount of computation required to complete a design is usually very large, due with the large numbers in 2-D digital filters and the large number of sample points needed to construct the objective function. Generally, the computational load can often be reduced by starting with an approximate design based on some closed-form solution. For example, the design of circularly or elliptical symmetric filters may start with filters that have square or rectangular passbands and stopbands.

Example 23.5 [20]

Design a circularly symmetric low-pass filter of order (2, 2) with $\boldsymbol{\omega}_{p1} = \boldsymbol{\omega}_{p2} = 0.08\boldsymbol{\pi}$ rad/s and $\boldsymbol{\omega}_{a1} = \boldsymbol{\omega}_{a2} = 0.12\boldsymbol{\pi}$ rad/s, assuming that $\boldsymbol{\omega}_{s1} = \boldsymbol{\omega}_{s2} = 2\boldsymbol{\pi}$ rad/s.

Solution

1. Construct the ideal discrete amplitude response of the filter

$$M_I(m,n) = \begin{cases} 1 & \text{for } \left(\boldsymbol{\omega}_{1m}^2 + \boldsymbol{\omega}_{2n}^2\right) \le 0.008\boldsymbol{\pi} \\ 0.5 & \text{for } 0.08\boldsymbol{\pi} \le \left(\boldsymbol{\omega}_{1m}^2 + \boldsymbol{\omega}_{2n}^2\right) \le 0.12\boldsymbol{\pi} \\ 0 & \text{otherwise} \end{cases}$$

where

$$\{\boldsymbol{\omega}_{1m}\} = \{\boldsymbol{\omega}_{2n}\} = 0, 0.02\boldsymbol{\pi}, 0.04\boldsymbol{\pi}, 0.2\boldsymbol{\pi}, 0.4\boldsymbol{\pi}, 0.6\boldsymbol{\pi}, 0.8\boldsymbol{\pi}, \boldsymbol{\pi}$$

2. To reduce the amount of computation, a 1-D low-pass filter with passband edge $\boldsymbol{\omega}_p = 0.08\boldsymbol{\pi}$ and stopband edge $\boldsymbol{\omega}_a = 0.1\boldsymbol{\pi}$ is first obtained with the 1-D transfer function being

$$H_1(z) = 0.11024 \frac{1 - 1.64382z^{-1} + z^{-2}}{1 - 1.79353z^{-1} + 0.84098z^{-2}}$$

and then a 2-D transfer function with a square passband is obtained as

$$H(z_1, z_2) = H_1(z_1)H_1(z_2)$$

3. Construct the objective function of Equation 23.77, using algorithm 1 to minimize the objective function $J(\mathbf{x})$. After 20 more iterations the algorithm converges to

$$H(z_1, z_2) = 0.00895 \times \frac{\begin{bmatrix} 1 & z_1^{-1} & z_1^{-2} \end{bmatrix} \begin{vmatrix} 1.0 & -1.62151 & 0.99994 \\ -1.62151 & 2.6370 & -1.62129 \\ 0.99994 & -1.62129 & 1.00203 \end{vmatrix} \begin{vmatrix} 1 \\ z_2^{-1} \\ z_2^{-2} \end{vmatrix}}{\begin{bmatrix} 1 & z_1^{-1} & z_1^{-2} \end{bmatrix} \begin{vmatrix} 1.0 & -1.78813 & 0.82930 \\ -1.78813 & 3.20640 & -1.49271 \\ 0.82930 & -1.49271 & 0.69823 \end{vmatrix} \begin{vmatrix} 1 \\ z_2^{-1} \\ z_2^{-2} \end{vmatrix}}$$

The amplitude response of the final optimal filter is depicted in Figure 23.17.

23.5.1.3 Minimax Algorithms

Least pth Minimax Algorithm. When a objective function is formulated in terms of the L_p norm of the error function and then minimizing $J(\mathbf{x})$ for increasing values of p, such an objective function can be obtained as

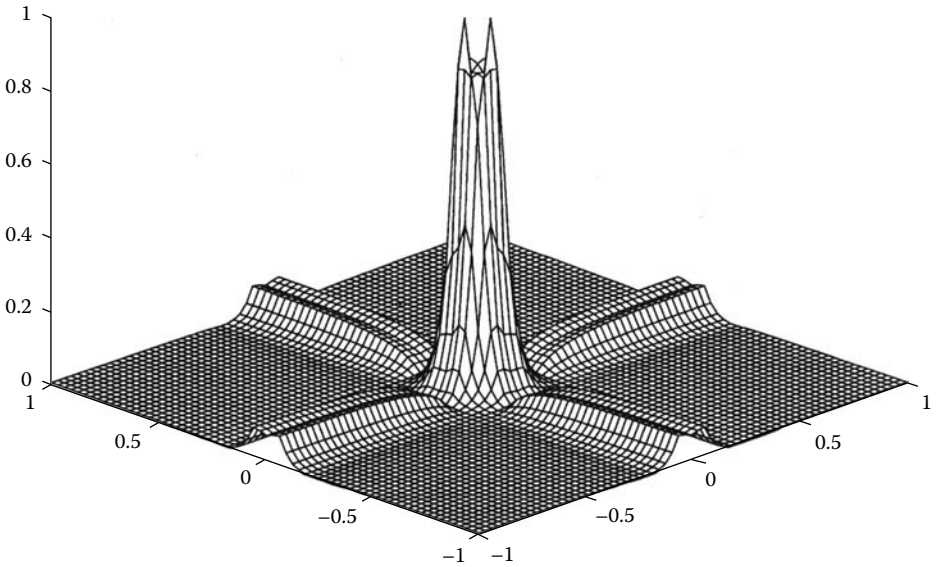


FIGURE 23.17 Amplitude response of the 2D optimal filter in Example 23.5.

$$J(x) = E_M(x)\left\{\sum_{i=1}^{m}\left|\frac{E(x,\omega_{1i},\omega_{2i})}{E_M(x)}\right|^p\right\}^{1/p} \tag{23.82}$$

where

$$E_M(\mathbf{x}) = \max_{1\le i\le m}\{E_i(\mathbf{x})\} = \max_{1\le i\le m}\{E_i(\mathbf{x},\omega_{1i},\omega_{2i})\} \tag{23.83}$$

A minimax algorithm based on $J(\mathbf{x})$ is as follows [21].

ALGORITHM 2: Least-*p*th Minimax Algorithm

Step 1. Input $\mathbf{x}_0$ and ε. Set $k=1$, $p=1$, $\mu=2$, $\mathbf{E}_0=10^{99}$.
Step 2. Initialize frequencies ω_{1i}, ω_{2i} for $i=1, 2,\ldots, m$.
Step 3. Using point $\mathbf{x}_{k-1}$ as initial point, minimize $J(\mathbf{x})$ with respect to $\mathbf{x}$ to obtain $\mathbf{x}_k$. Set $\mathbf{E}_k = E(\mathbf{x}_k)$.
Step 4. If $|E_{k-1} - E_k| < \varepsilon$, then output $\mathbf{x}^\star = \mathbf{x}_k$ and $\mathbf{E}_k$ and stop. Else, set $p=\mu p$, $k=k+1$ and go to step 3.

The minimization in step 3 can be carried out using algorithm 1 or any other unconstrained optimization algorithms.

Charalambous Minimax Algorithm. The preceding algorithm gives excellent results except that it requires a considerable amount of computation. An alternative and much more efficient algorithm is the minimax algorithm proposed by Charalambous and Antoniou [22,23]. This algorithm is based on principles developed by Charalambous [24] and involves the minimization of the objective function $J(\mathbf{x}, \zeta, \lambda)$, defined by

$$J(\mathbf{x},\zeta,\lambda) = \sum_{i\in I_1}\frac{1}{2}\lambda_i[J_i(\mathbf{x},\zeta)]^2 + \sum_{i\in I_2}\frac{1}{2}[J_i(\mathbf{x},\zeta)]^2 \tag{23.84}$$

where ζ and λ_i for $i=1, 2,\ldots, m$ are constants,

$$J_i(\mathbf{x},\zeta) = E_i(\mathbf{x}) - \zeta$$
$$I_1 = \{i: J_i(\mathbf{x},\zeta) > 0 \text{ and } \lambda_i > 0\}$$

and

$$I_2 = \{i: J_i(\mathbf{x},\zeta) > 0 \text{ and } \lambda_i > 0\}$$

The factor 1/2 in Equation 23.84 is included for the purpose of simplifying the gradient that is given by

$$\nabla J(\mathbf{x},\zeta,\lambda) = \sum_{i\in I_1}\lambda_i J_i(\mathbf{x},\zeta)\nabla J_i(x,\zeta) + \sum_{i\in I_2} J_i(\mathbf{x},\zeta)\nabla J_i(\mathbf{x},\zeta) \tag{23.85}$$

It can be shown that, if

1. The second-order sufficient conditions for a minimum hold at $x^\star$,
2. $\lambda_i = \lambda_i^*$, $i=1, 2,\ldots, m$, where λ_i^* are the minimax multipliers corresponding to a minimum optimum solution $\mathbf{x}^\star$, and
3. $E(\mathbf{x}^\star) - \xi$ is sufficiently small then $\mathbf{x}^\star$ is a strong local minimum point of $J(\mathbf{x}, \xi, \lambda)$.

Condition 1 is usually satisfied in practice. Therefore, a local minimum point $\mathbf{x}^*$ can be found by forcing λ_i to approach λ_i^* $(i = 1, 2, \ldots, m)$ and making $E(\mathbf{x}^*) - \xi$ sufficiently small. These two constraints can be simultaneously satisfied by applying the following algorithm.

ALGORITHM 3: Charalambous Minimax Algorithm

Step 1. Set $\xi = 0$ and $\lambda_i = 1$ for $i = 1, 2, \ldots, m$. Initialize $\mathbf{x}$.
Step 2. Minimize function $J(\mathbf{x}, \xi, \lambda)$ to obtain $\mathbf{x}$.
Step 3. Set

$$S = \sum_{i \in I_1} \lambda_i f_i(\mathbf{x}, \xi) + \sum_{i \in I_2} J_i(\mathbf{x}, \xi)$$

and update λ_i and ξ as

$$\lambda_i = \begin{cases} \lambda_i J_i(\mathbf{x}, \xi)/S & \text{if } \mathbf{J}_i(\mathbf{x}, \xi) \geq 0,\ \lambda_i \geq 0 \\ J_i(\mathbf{x}, \xi)/S & \text{if } \mathbf{J}_i(\mathbf{x}, \xi) \geq 0,\ \lambda = 0 \\ 0 & \text{if } \mathbf{J}_i(\mathbf{x}, \xi) < 0 \end{cases}$$

$$\xi = \sum_{i=1}^{\mathrm{m}} \lambda_i \mathbf{E}_i(\mathbf{x})$$

Step 4. Stop if

$$\frac{E_M(\mathbf{x}) - \xi}{E_M(\mathbf{x})} \leq \varepsilon$$

otherwise go to step 2.

The parameter ε is a prescribed termination tolerance. When the algorithm converges, conditions 2 and 3 are satisfied and $\mathbf{x} = \mathbf{x}^*$. The unconstrained optimization in step 2 can be accomplished by applying a quasi-Newton algorithm.

Example 23.6 [23]

Design a 2-D circularly symmetric filter with the same specifications as in Example 23.5, using algorithm 3.

Solution

1. Construct the ideal discrete amplitude response of the filter. Since the passband and stopband contours are circles, the sample points can be placed on arcs of a set of circles centered at the origin. Five circles with radii

$$r_1 = 0.3\boldsymbol{\omega}_p, \quad r_2 = 0.6\boldsymbol{\omega}_p, \quad r_3 = 0.8\boldsymbol{\omega}_p, \quad r_4 = 0.9\boldsymbol{\omega}_p, \quad \text{and} \quad r_5 = \boldsymbol{\omega}_p$$

are placed in the passband and five circles with radii

$$r_6 = \boldsymbol{\omega}_a, \quad r_7 = \boldsymbol{\omega}_a + 0.1(\boldsymbol{\pi} - \boldsymbol{\omega}_a), \quad r_8 = \boldsymbol{\omega}_a + 0.2(\boldsymbol{\pi} - \boldsymbol{\omega}_a), \quad r_9 = \boldsymbol{\omega}_a + 0.55(\boldsymbol{\pi} - \boldsymbol{\omega}_a), \quad \text{and} \quad r_{10} = \boldsymbol{\pi}$$

are placed in the stopband.

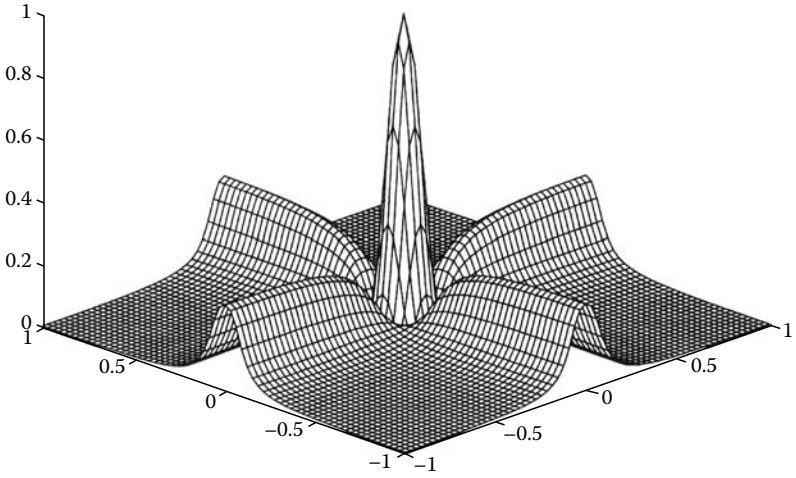


FIGURE 23.18 3D plot of the amplitude response of the filter in Example 23.6.

For circularly symmetric filters, the amplitude is uniquely specified by the amplitude response in the sector [0°, 45°]. Therefore, six equally spaced points on each circle described above between 0° and 45° are chosen. These points plus the origin $(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = (0,0)$ form a set of 61 sample points.

2. Select the 2-D transfer function. Because a circularly symmetric filter has a transfer function with separable denominator [24], we can select the transfer function to be of the form

$$H(z_1, z_2) = H_0(z_1, z_2)^{-1} \times \prod_{k=1}^{k} \frac{z_1 z_2 + z_1^{-1} z_2^{-1} + a_k\left(z_1 + z_1^{-1} + z_2 + z_2^{-1}\right) + z_1^{-1} z_2 + z_1 z_2^{-1} + b_k}{(1 + c_k z_1^{-1} + d_k z_1^{-2})(1 + c_k z_2^{-1} + d_k z_2^{-2})} \tag{23.86}$$

with parameter H_0 fixed as $H_0 = (0.06582)^2$, $K = 1$, $\varepsilon = 0.01$.

3. Starting from

$$a_1^{(0)} = -1.514, \quad b_1^{(0)} = (a_1^{(0)})^2, \quad c_1^{(0)} = -1.784, \quad d_1^{(0)} = 0.8166$$

and algorithm 3 yields the solution

$$a_1^* = 1.96493, \quad b_1^* = -10.9934, \quad c_1^* = -1.61564, \quad d_1^* = 0.66781$$

$$E_M(\mathbf{x}) = 0.37995$$

The 3-D plot of the amplitude response of the resulting filter is illustrated in Figure 23.18.

23.5.2 Design by Singular-Value Decomposition

As will be seen, an important merit of the design methods of 2-D IIR filters based on singular-value decomposition (SVD) is that the required 2-D filter is decomposed into a set of 1-D digital subfilters, which are much easier to design by optimization than the original 2-D filters.

23.5.2.1 Problem Formulation

In a quadrantally symmetric filter, $H(z_1, z_2)$ has a separable denominator [25]. Therefore, it can be expressed as

$$H(z_1, z_2) = \sum_{i=1}^{K} f_i(z_1) g_i(z_2) \tag{23.87}$$

In effect, a quadrantally symmetric filter can always be realized using a set of K parallel sections where the ith section is a separable subfilter characterized by the transfer function $f_i(z_1)\, g_i(z_2)$.

Consider the desired amplitude response sample of 2-D filter $H(z_1, z_2)$, we form a 2-D amplitude specification matrix $\mathbf{A}$ as

$$\mathbf{A} = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \cdots & a_{2L} \\ \vdots & \vdots & & \vdots \\ a_{M1} & a_{M1} & \cdots & a_{ML} \end{vmatrix} \tag{23.88}$$

where $\{a_{ml}\}$ is a desired amplitude response sampled at frequencies $(\omega_{1l}, \omega_{2m}) = (\pi\mu_l/T_i, \pi\nu_m/T_2)$, with

$$\mu_l = \frac{l-1}{L-1}, \quad \nu_m = \frac{m-1}{M-1} \quad \text{for } 1 \le l \le L, \quad 1 \le m \le M$$

that is,

$$a_{ml} = \left| H\left(e^{j\pi\mu_1},\ e^{j\pi\nu_m}\right) \right|$$

If the matrix $\mathbf{A}$ can be decomposed into the form of

$$\mathbf{A} = \sum_{i=1}^{r} \mathbf{F}_i \mathbf{G}_i \tag{23.89}$$

then, by using the column vectors $\mathbf{F}_i$ and row vectors $\mathbf{G}_i$, we can construct matrices

$$\mathbf{F} = [\mathbf{F}_1 \quad \mathbf{F}_2 \quad \cdots \quad \mathbf{F}_r] \tag{23.90}$$

$$\mathbf{G} = [\mathbf{G}_1 \quad \mathbf{G}_2 \quad \cdots \quad \mathbf{G}_r]^{\mathrm{T}} \tag{23.91}$$

If all elements of $\mathbf{F}$ and $\mathbf{G}$ are nonnegative then they can be regarded as the amplitude specifications matrices of an r-input/1-output 1-D filter $F(z_1)$ and a 1-input/r-output 1-D filter $G(z_2)$,

$$F(z_1) = [f_1(z_1), f_2(z_1), \ldots, f_r(z_1)] \tag{23.92}$$

$$G(z_2) = [g_1(z_2),\ g_2(z_2), \ldots, g_r(z_2)]^{\mathrm{T}} \tag{23.93}$$

Therefore, the 2-D filter of Equation 23.87 can be approximated by

$$H(z_1, z_2) = F(z_1)G(z_2) \tag{23.94}$$

In this section, two design procedures are described that can be applied to the design of 2-D IIR filters whose amplitude responses are quadrantally symmetric.

23.5.2.2 Method of Antoniou and Lu

Antoniou and Lu proposed a method of 2-D IIR filter based on the SVD of the amplitude response matrix $\mathbf{A}$ [26]. The SVD of matrix $\mathbf{A}$ gives [27]

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} = \sum_{i=1}^{r} \mathbf{f}_i \mathbf{g}_i^{\mathrm{T}} \tag{23.95}$$

where σ_i are the singular values of $\mathbf{A}$ such that $\sigma_1 \le \sigma_2 \le \cdots \le \sigma_r \le 0$ is the rank of $\mathbf{A}$, $\mathbf{u}_i$, and $\mathbf{v}_i$ are the ith eigenvector of $\mathbf{AA}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, respectively, $\boldsymbol{\phi}_i = \sigma_i^{1/2}\mathbf{u}_i$, $\boldsymbol{\gamma}_i = \sigma_i^{1/2}\mathbf{v}_i$, and $\{\Phi_i\colon 1 \le i \le r\}$ and $\{\gamma_i\colon 1 \le i \le r\}$ are sets of orthogonal L-dimensional and M-dimensional vectors, respectively.

An important property of the SVD can be stated as

$$\left\| \mathbf{A} - \sum_{i=1}^{K} \mathbf{F}_i \mathbf{g}_i^{\mathbf{T}} \right\| = \min_{\bar{\phi}, \bar{\gamma}} \left\| \mathbf{A} - \sum_{i=1}^{K} \overline{\mathbf{F}}_i \overline{\mathbf{g}}_i^{\mathrm{T}} \right\| \quad \text{for } 1 \le K \le r \tag{23.96}$$

where $\overline{\phi}_i \in R^L$, $\overline{\gamma}_i \in R^M$

To design a 2-D IIR filter by SVD, two steps are involved, namely

Step 1. Design of the main section.
Step 2. Design of the error correction sections as will be detailed below.

*Design of the Main Section***.** Note that Equation 23.95 can be written as

$$\mathbf{A} = \mathbf{f}_1 \mathbf{g}_1^{\mathrm{T}} + \varepsilon_1 \tag{23.97}$$

where $\varepsilon_1 = \Sigma_i^{\gamma} \Phi_i \gamma_i^{\mathrm{T}}$. And since all the elements of $\mathbf{A}$ are nonnegative, if follows that all elements of ϕ_1 and γ_1 are nonnegative.

On comparing Equation 23.97 with Equation 23.95 and assuming that $K = 1$ and that ϕ_1, γ_1 are sampled versions of the desired amplitude responses for the 1-D filters characterized by $f_1(z_1)$ and $g_1(z_2)$, respectively, a 2-D filter can be designed through the following procedures:

1. Design 1-D filters F_1 and G_1 characterized by $f_1(z_1)$ and $g_1(z_2)$.
2. Connect filters F_1 and G_1 in cascade, i.e.,

$$H_1(z_1, z_2) = f_1(z_1) g_1(z_2)$$

Step 1 above can be carried out by using an optimization algorithm such as the quasi-Newton algorithm or the minimax algorithm.

Since $f_1(z_1)g_1(z_2)$ corresponds to the largest singular value σ_1, the subfilter characterized by $f_1(z_1)\, g_1(z_2)$ is said to be the main section of the 2-D filter.

Design of the Error Correction Sections. The approximation error of $H_1(z_1, z_2)$ can be reduced by realizing more of the terms in Equation 23.95 by means of parallel filter sections. From Equation 23.97, we can write

$$\mathbf{A} = \mathbf{f}_1 \mathbf{g}_1^{\mathrm{T}} + \mathbf{f}_2 \mathbf{g}_2^{\mathrm{T}} + \varepsilon_{21} \tag{23.98}$$

Since ϕ_2 and λ_2 may have some negative components, a careful treatment in Equation 23.98 is necessary.

Let ϕ_2^- and γ_2^- be the absolute values of the most negative components of ϕ_2 and γ_2, respectively. If

$$\mathbf{e}_{\phi} = [1\ 1 \cdots 1]^{\mathrm{T}} \in R^L \quad \text{and} \quad \mathbf{e}_{\gamma} = [1\ 1 \cdots 1]^{\mathrm{T}} \in R^M$$

then all components of

$$\boldsymbol{\phi}_{2p} = \boldsymbol{\phi}_2 + \boldsymbol{\phi}_2^- \boldsymbol{\varepsilon}_{\phi}\ \boldsymbol{\alpha\nu\delta}\ \boldsymbol{\gamma}_{2p} = \boldsymbol{\gamma}_2 + \boldsymbol{\gamma}_2^- \boldsymbol{\varepsilon}_{\gamma}$$

are nonnegative. If it is possible to design 1-D linear-phase or zero-phase filters characterized by $f_1(z_1)$, $g_1(z_2)$, $f_{2p}(z_1)$, and $g_{2p}(z_2)$, such that

$$f_1(e^{j\pi\mu_i}) = \left|f_1(e^{j\pi\mu_i})\right| e^{j\alpha_1\mu_i}$$
$$g_1(e^{j\pi\nu_m}) = \left|g_1(e^{j\pi\nu_m})\right| e^{j\alpha_2\nu_m}$$

and

$$f_{2p}(e^{j\pi\mu_i}) = \left|f_{2p}(e^{j\pi\mu_i})\right| e^{j\alpha_1\mu_i}$$
$$g_{2p}(e^{j\pi\nu_m}) = \left|g_{2p}(e^{j\pi\nu_m})\right| e^{j\alpha_2\nu_m}$$

for $1 \le l \le L, \le m \le M$, where

$$\left|f_1(e^{j\pi\mu_1})\right| \approx \phi_{1l}$$
$$\left|g_1(e^{j\pi\nu_m})\right| \approx \gamma_{1m}$$
$$\left|f_{2p}(e^{j\pi\mu_l})\right| \approx \phi_{2lp}$$
$$\left|g_{2p}(e^{j\pi\nu_m})\right| \approx \gamma_{2mp}$$

In above ϕ_{1l}, ϕ_{2lp}, γ_{lm}, γ_{2mp} represent the lth component of ϕ_1, ϕ_{2p} and mth component of γ_1 and γ_{2p}, respectively. α_1 and α_2 are constants that are equal to zero if zero-phase filters are to be designed. Let

$$\alpha_1 = -\pi n_1, \quad \alpha_2 = -\pi n_2 \quad \text{with integers } n_1, n_2 \ge 0 \tag{23.99}$$

and define

$$f_2(z_1) = f_{2p}(z_1) - \phi_2^- z_1^{-n_1} \tag{23.100}$$

$$g_2(z_2) = g_{2p}(z_2) - \gamma_2^- z_2^{-n_2} \tag{23.101}$$

It follows that

$$f_2(e^{j\pi\mu_l}) = \left[f_{2p}(e^{j\pi\mu_l}) - \phi_2^-\right] e^{-j\pi n_1\mu_l} \approx \phi_{2l} e^{-j\pi\mu_l n_1}$$
$$g_2(e^{j\pi\nu_m}) = \left[g_{2p}(e^{j\pi\nu_m}) - \gamma_2^-\right] e^{-j\pi n_2\nu_m} \approx \gamma_{2m} e^{-j\pi\gamma_m n_2}$$

Furthermore, if we form

$$H_2(z_1, z_2) = f_1(z_1)g_1(z_2) + f_2(z_1)g_2(z_2) \tag{23.102}$$

then

$$\begin{aligned}\left|H_2(e^{j\pi\mu_l}, e^{j\pi\nu_m})\right| &= \left|f_1(e^{j\pi\mu_l})g_1(e^{j\pi\nu_m}) + f_2(e^{j\pi\mu_l})g_2(e^{j\pi\nu_m})\right| \\ &\approx |\phi_{1l}\gamma_{1m} + \phi_{2l}\gamma_{2m}|\end{aligned} \tag{23.103}$$

Follow this procedure, K–1 correction sections characterized by $k_2(z_1)g_2(x_2), \ldots, g_2(x_1)g_1(x_2)$ can be obtained, and $H_k(z_1, z_2)$ can be formed as

$$H_k(z_1, z_2) = \sum_{i=1}^{K} f_i(z_i)g_i(z_2) \tag{23.104}$$

and from Equation 23.96 we have

$$\left\|\mathbf{A} - \left|H_k(e^{j\pi\mu_l}, e^{j\pi\nu_m})\right|\right\| \approx \left\|\mathbf{A} - \sum_{i=1}^{K} \mathbf{f}_i \mathbf{g}_i^{\mathrm{T}}\right\|$$
$$\leq \left\|\varepsilon_K\right\| = \min_{\bar{\phi}_i, \bar{\gamma}_i} \left\| \sum_{i=1}^{K} \bar{\mathbf{f}}_i \bar{\mathbf{g}}_i{}^{\mathrm{T}} \right\| \tag{23.105}$$

In effect, a 2-D filter consisting of K sections is obtained whose amplitude response is a minimal mean-square-error approximation to the desired amplitude response.

The method leads to an asymptotically stable 2-D filter, provided that all 1-D subfilters employed are stable. The general configuration of the 2-D filter obtained is illustrated in Figure 23.19, where the various 1-D subfilters may be either linear-phase or zero-phase filters.

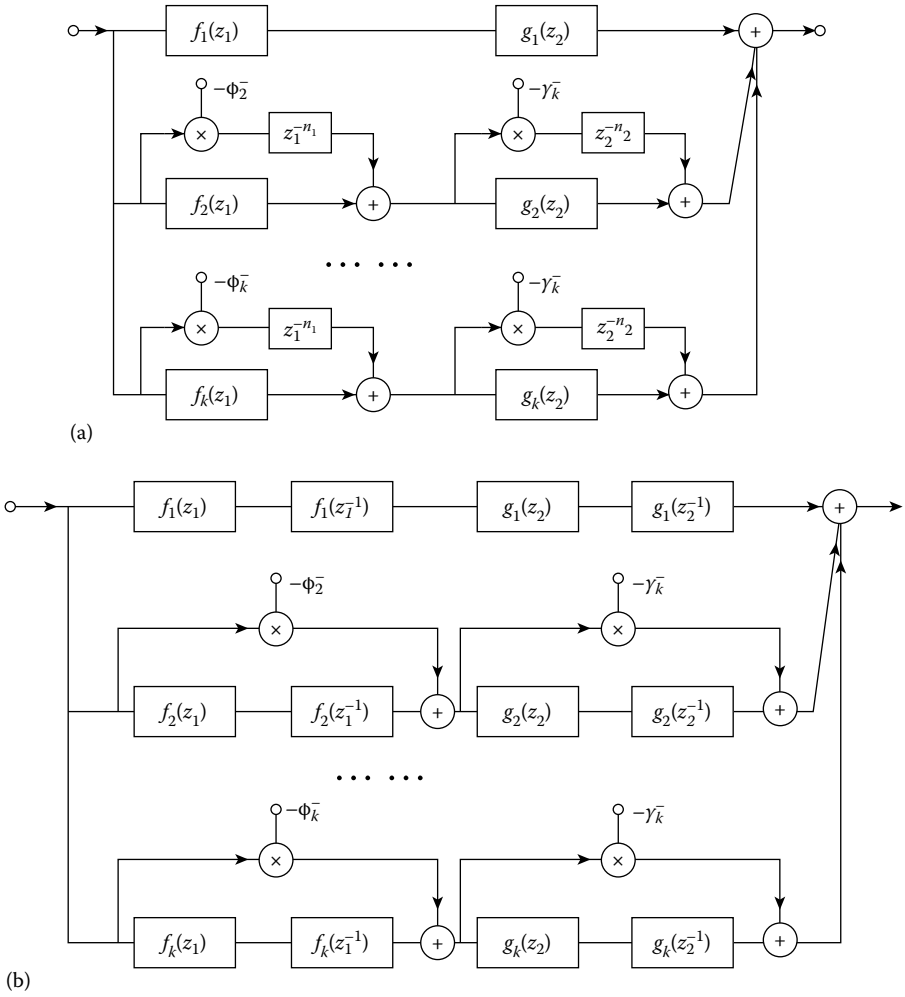


FIGURE 23.19 Configurations of 2D IIR filter by SVD. (a) General structure of 2D filter. (b) Structure using zero-phase IIR filters.

If linear-phase subfilters are to be used, the equalities in Equation 23.99 must be satisfied. This implies that the subfilters must have constant group delays. If zero-phase subfilters are employed, where $f_i(z_1)$ and $f_i(z_1^{-1})$,and $g_i(z_2)$ and $g_i(z_2^{-1})$contribute equally to the amplitude response of the 2-D filter. The design can be accomplished by assuming that the desired amplitude responses for subfilters F_i, G_i are $\phi_i^{1/2}$, $\gamma_i^{1/2}$, for $i = 1, 2, k, \ldots, K$, respectively.

Error Compensation Procedure. When the main section and correction sections are designed by an optimization procedure as described above, approximation errors inevitably occur that will accumulate and manifest themselves as the overall error. The accumulation of error can be reduced by the following compensation procedure.

When filters F_1 and G_1 are designed, the approximation error matrix $\mathbf{E}_1$ can be calculated as

$$\mathbf{E}_1 = \mathbf{A} - f_1(e^{j\pi\mu_l})g_1(e^{j\pi\nu_m}) \tag{23.106}$$

and then perform SVD on $\mathbf{E}_1$ to obtain

$$\mathbf{E}_1 = S_{22}\mathbf{f}_{22}\mathbf{g}_{22}^{\mathrm{T}} + \cdots + S_{r2}\mathbf{f}_{r2}\mathbf{g}_{r2}^{\mathrm{T}} \tag{23.107}$$

Data ϕ_{22} and γ_{22} can be used to deduce filters $f_2(z_1)$ and $g_2(z_2)$. Thus, the first correction section can be designed. Next, form the error matrix $\mathbf{E}_2$ as

$$\mathbf{E}_2 = \mathbf{E}_1 - S_{22}f_2(e^{j\pi\mu_l})g_2(e^{j\pi\nu_m}) \tag{23.108}$$

and then perform SVD on $\mathbf{E}_2$ to obtain

$$\mathbf{E}_2 = \mathbf{S}_{33}\ \mathbf{f}_{33}\ \mathbf{g}_{33}^{\mathrm{T}} + \cdots + S_{r3}\ \mathbf{f}_{r3}\ \mathbf{g}_{r3}^{\mathrm{T}} \tag{23.109}$$

and use data f_{33} and g_{33} to design the second correction section. The procedure is continued until the norm of the error matrix becomes sufficiently small that a satisfactory approximation to the desired amplitude response is reached.

Design of 1-D filters by using optimization can sometimes yield unstable filters. This problem can be eliminated by replacing poles outside the unit circle of the z plane by their reciprocals and simultaneously adjusting the multiplier constant to compensate for the change in gain [19].

Example 23.7 [26]

Design a circularly symmetric, zero-phase 2-D filter specified by

$$|H(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)| = \begin{cases} 1 & \text{for } \left(\boldsymbol{\omega}_1^2 + \boldsymbol{\omega}_2^2\right)^{1/2} < 0.35\boldsymbol{\pi} \\ 0 & \text{for} \left(\boldsymbol{\omega}_1^2 + \boldsymbol{\omega}_2^2\right)^{1/2} \geq 0.65\ \boldsymbol{\pi} \end{cases}$$

assuming that $\boldsymbol{\omega}_{s1} = \boldsymbol{\omega}_{s2} = 2\boldsymbol{\pi}$.

Solution

1. Construct a sampled amplitude response matrix. By taking $L = M = 21$ and assuming that the amplitude response varies linearly with the radius in the transition band, the amplitude response matrix can be obtained as

$$\mathbf{A} = \left| \begin{matrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{matrix} \right|_{21\times 21}$$

where

$$\mathbf{A} = \left| \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0.75 & 0.5 & 0.25 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0.75 & 0.75 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.75 & 0.75 & 0.5 & 0.5 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.75 & 0.5 & 0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right|$$

The ideal amplitude response of the filter is illustrated in Figure 23.20a.

2. Perform SVD to matrix **A** to obtain the amplitude response of the main section of the 2-D filter. It is worth noting that when a circularly symmetric 2-D filter is required, matrix **A** is symmetric and, therefore, Equation 23.95 becomes

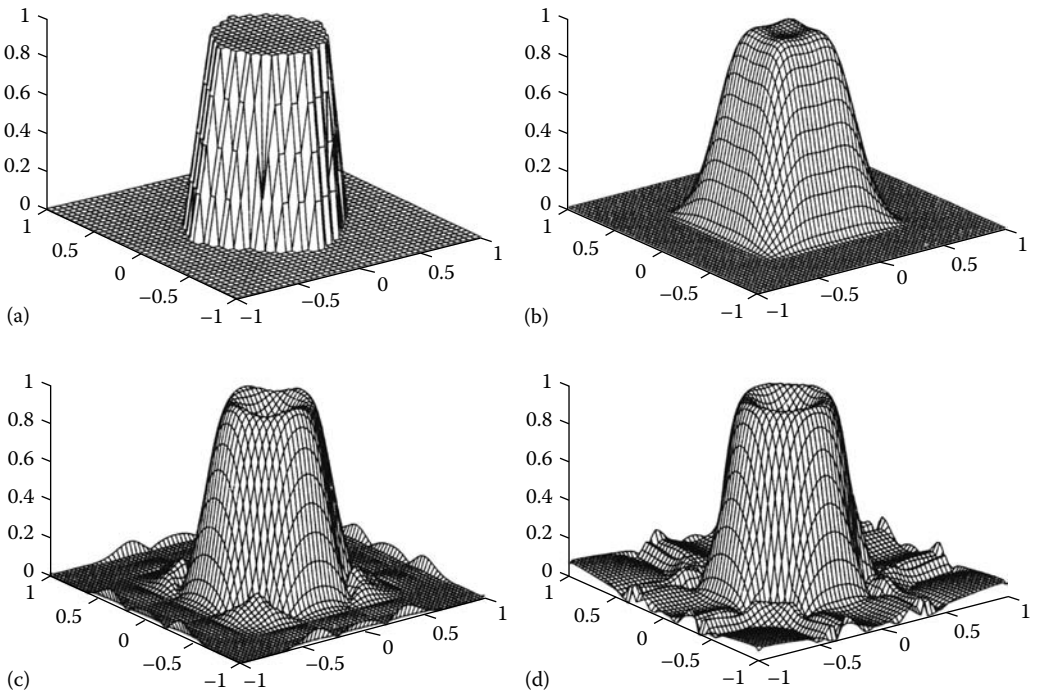


FIGURE 23.20 Amplitude responses of (a) the ideal circularly symmetric 2D filter; (b) the main section; (c) the main section plus the first correction; (d) the main section plus the first and second correction sections.

TABLE 23.3 Design Based on Fourth-Order Subfilters (Example 23.7)

The main section	$f_1(z) = 0.1255\dfrac{(z^2 + 0.7239z + 1)\ (z^2 + 1.6343z + 1)}{(z^2 + 0.1367z + 1)\ (z^2 - 0.5328z + 0.2278)}$
The first correction section	$f_2(z) = 0.6098\ \dfrac{(z^2 + 1.1618z + 0.1661)\ (z^2 + 0.8367z + 0.9958)}{(z^2 + 0.9953z)\ (z^2 + 0.5124z + 0.32)}$
	$S_2 = -1, \phi_2^- = 0.6266$
The second correction section	$f_3(z) = 0.4630\dfrac{(z^2 + 1.5381z + 0.4456)\ (z^2 - 1.397z + 1.1191)}{(z^2 + 2.0408z - 1)\ (z^2 - 0.7092z + 0.6961)}$
	$S_3 = +1, \phi_3^- = 0.2764$

$$\mathbf{A} = \sum_{i=1}^{r} S_i \mathbf{f}_i \mathbf{f}_i^T \tag{23.110}$$

where $S_1 = 1$ and $S_i = \pm 1$ or -1, for $2 \le i \le r$. This implies that each parallel section requires only one 1-D subfilter to be designed. As a consequence the design work is reduced by 50%.

When vector f_1 is obtained, by selecting a fourth-order approximation and after optimization, the transfer function of the main section $f_1(z)$ is obtained.

3. Design the correction sections. Successively perform SVD to the error $\mathbf{E}_1$ and $\mathbf{E}_2$, and apply the preceding design technique; the transfer functions of the first and second correction sections can be obtained.

   The transfer function of the main section and the first and second correction sections are listed in Table 23.3. And the amplitude responses of (1) the main section, (2) the main section plus the first correction, and (3) the main section plus the first and second correction sections are depicted in Figure 23.20b through d.

23.5.2.3 Method of Deng and Kawamata

In decomposing 2-D amplitude specifications into 1-D ones, the conventional SVD cannot avoid the problem that the 1-D amplitude specifications that result are often negative. Since negative values cannot be viewed as amplitude response, the problem of 1-D digital filter design becomes intricate. Deng and Kawamata [28] proposed a procedure that guarantees all the decomposition results to be always nonnegative and thus simplifies the design of correction sections.

The method decomposes the matrix **A** into the form

$$\mathbf{A} = \sum_{i=1}^{r} S_i \mathbf{F}_i \mathbf{G}_i \tag{23.111}$$

where all the elements of $\mathbf{F}_i$ and $\mathbf{G}_i$ are nonnegative and the decomposition error

$$\mathbf{E} = \| \mathbf{A} - \sum_{i=1}^{r} \mathbf{S}_i \mathbf{F}_i \mathbf{G}_i \| \tag{23.112}$$

is sufficiently small, and $S_i = 1$ or -1 for $i = 1, 2, \ldots, r$.

The design procedure can be desired as follows.

Step 1. Let $\mathrm{A}_1' = \mathrm{A}, \mathrm{A}_1^- = 0$, and perform the SVD on A_1^+ as

$$\mathbf{A}_1^+ = \sum_{i=1}^{r_i} \sigma_{1i} \mathbf{u}_{1i} \mathbf{v}_{1i} \approx \mathbf{F}_1^+ \mathbf{G}_1^+ \tag{23.113}$$

where σ_{1i} is the ith singular value of $\mathbf{A}_1^+(\sigma_{11} \geq \sigma_{12} \geq \cdots \geq \sigma_{1r_1})$ and $\mathbf{F}_1^+ = u_{11}\sigma_{11}^{1/2}, \mathbf{G}_1^+ = u_{11}\sigma_{11}^{1/2}$.

Let

$$\mathbf{F}_1 = \mathbf{F}_1^+, \quad \mathbf{G}_1 = \mathbf{G}_1^+, \quad \text{and} \quad S_1 = 1 \tag{23.114}$$

all the elements of $\mathbf{F}_1$ and $\mathbf{G}_1$ are nonnegative.

Step 2. Calculate the approximation error matrix $\mathbf{A}_2$ and decompose it into the sum of A_2^+ and A_2^- as

$$\mathbf{A}_2 = \mathbf{A} - S_1\mathbf{F}_1\mathbf{G}_1 = \mathbf{A}_2^+ + \mathbf{A}_2^- \tag{23.115}$$

where

$$\mathbf{A}_2^+(m,n) = \begin{cases} A_2(m,n) & \text{if } A_2(m,n) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{23.116}$$

and

$$\mathbf{A}_2^-(m,n) = \begin{cases} A_2(m,n) & \text{if } A_2(m,n) \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{23.117}$$

To determine S_2 and $\mathbf{F}_2$ and $\mathbf{G}_2$ for approximating $\mathbf{A}_2$ as accurately as possible, the following three steps are involved.

1. Perform the SVD on A_2^1 and approximate it as

$$\mathbf{A}_2^+ = \sum_{i=1}^{r_2} \mathbf{s}_{2i}\mathbf{u}_{2i}\mathbf{v}_{2i} \approx \mathbf{F}_2^+\mathbf{G}_2^+ \tag{23.118}$$

   where $\mathbf{F}_2^+ = \mathbf{u}_{21}\sigma_{21}^{1/2}, \mathbf{G}_2^+ = \sigma_{21}^{1/2}\mathbf{v}_{21}$. All the elements of F_2^1 and G_2^1 and G^* are nonnegative. If $\mathbf{F}_2 = \mathbf{F}_2^+, \mathbf{G}_2 = \mathbf{G}_2^+$, and $S_2 = 1$, the approximation error is

$$\mathbf{E}_2^+ = \| \mathbf{A} - \sum_{i=1}^{2} \mathbf{S}_i\mathbf{F}_i\mathbf{G}_i \| \tag{23.119}$$

2. Perform the SVD on $-\mathbf{A}_2^-$ and approximate it as

$$-\mathbf{A}_2^- = \sum_{i=1}^{r_{2-}} \sigma_{2i-}\mathbf{u}_{2i-}\mathbf{v}_{2i-} \approx \mathbf{F}_2^-\mathbf{G}_2^- \tag{23.120}$$

   where $\mathbf{F}_2^- = \mathbf{u}_{21-}\sigma_{21-}^{1/2}, \mathbf{G}_2^- = \sigma_{21-}^{1/2}\mathbf{v}_{21-}$, and r_{2-} is the rank of A_2^-. All the elements of $\mathbf{F}_2^-$ and $\mathbf{G}_2^-$ are nonnegative. If $\mathbf{F}_2 = \mathbf{F}_2^-, \mathbf{G}_2 = \mathbf{G}_2^-$, and $S_2 = 1$, the approximation error $\mathbf{E}_2^-$ is

$$\mathbf{E}_2^- = \| \mathbf{A} - \sum_{i=1}^{2} S_i\mathbf{F}_i\mathbf{G}_i \| \tag{23.121}$$

3. According to the results from steps 1 and 2, the optimal vectors $\mathbf{F}_2$ and $\mathbf{G}_2$ for approximating $\mathbf{A}$ are determined as

$$\begin{aligned} &\mathbf{F}_2 = \mathbf{F}_2^+, \quad \mathbf{G}_2 = \mathbf{G}_2^+, \quad S_2 = 1 \quad \text{if } \mathbf{E}_2^+ \leq \mathbf{E}_2^- \\ &\mathbf{F}_2 = \mathbf{F}_2^-, \quad \mathbf{G}_2 = \mathbf{G}_2^-, \quad S_2 = -1 \quad \text{if } \mathbf{E}_2^+ \geq \mathbf{E}_2^- \end{aligned}$$

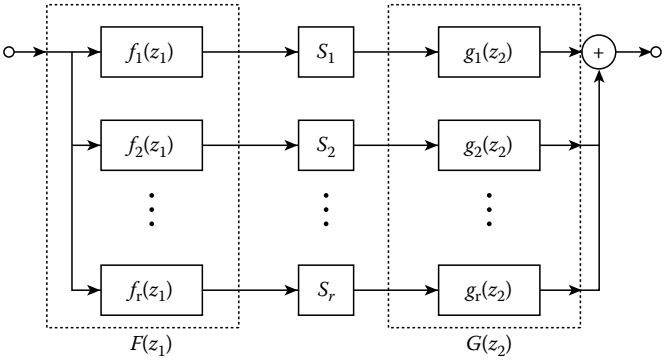


FIGURE 23.21 2D filter realization based on iterative SVD.

Successively decomposing the approximation error matrices $\mathbf{A}j$ $(j = 3, 4, \ldots, r)$ in the same way above described, a good approximation of matrix **A** can be obtained as in Equation 23.111.

Step 3. With the matrix **A** being decomposed into nonnegative vectors, the 1-D subfilters are designed through an optimization procedure, and a 2-D filter can then be readily realized, as shown in Figure 23.21.

It is noted that, in addition to SVD based methods as described in this section, design method based on other decomposition is also possible [29].

23.5.3 Design Based on Two-Variable Network Theory

Ramamoorthy and Bruton proposed a design method of 2-D IIR filters that always guarantees the stability of a filter and that involves the application of two-variable (2-*V*) strictly Hurwitz polynomials [30]. A 2 *V* polynomial $b(s_1, s_2)$ is said to be strictly Hurwitz if

$$b(s_1, s_2) \neq 0 \quad \text{for } \mathrm{Re}\{s_1\} \geq 0 \text{ and } \mathrm{Re}\{s_2\} \geq 0 \tag{23.122}$$

In their method, a family of 2-*V* strictly Hurwitz polynomials is obtained by applying network theory [31,32] to the frequency-independent, 2-*V* lossless network illustrated in Figure 23.22. The network has

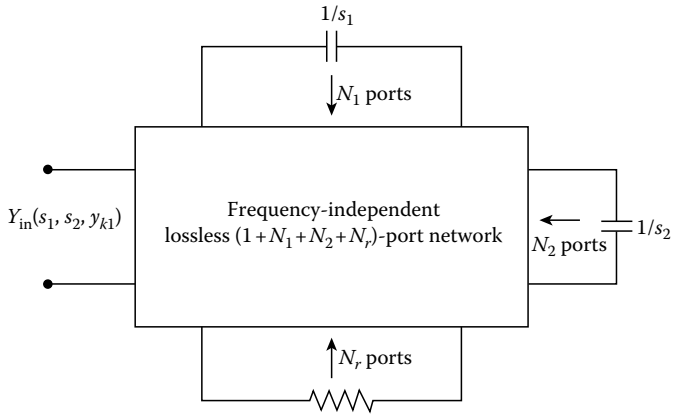


FIGURE 23.22 A $(1 + N_1 + N_2 + N_r)$-port 2-*V* lossless network.

$1 + N_1 + N_2 + N_r$ ports and N_1 and N_2 are terminated in unit capacitors in complex variables s_1 and s_2, respectively, and N_r is terminated in unit resistors. Since the network is lossless and frequency independent, its admittance matrix $\mathbf{Y}$ is a real and skew-symmetric matrix given by

$$\mathbf{Y} = \begin{vmatrix} 0 & y_{12} & y_{12} & \cdots & y_{1N} \\ -y_{12} & 0 & y_{23} & \cdots & y_{2N} \\ -y_{13} & -y_{23} & 0 & \cdots & y_{3N} \\ \vdots & \vdots & \vdots & & \vdots \\ -y_{1N} & -y_{2N} & -y_{3N} & \cdots & 0 \end{vmatrix} = \begin{vmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ -\mathbf{Y}_{12}^T & \mathbf{Y}_{22} \end{vmatrix} \quad N = 1 + N_1 + N_2 + N_r \tag{23.123}$$

If we define

$$\mathbf{Y}_{22}(s_1, s_2, y_{kl}) = \mathbf{Y}_{22} + \text{diag}\ \{\overset{N_r}{1 \cdots 1}\ \overset{N_1}{s_1 \cdots s_1}\ \overset{N_2}{s_2 \cdots s_2}\} \tag{23.124}$$

and

$$\Delta(s_1, s_2, y_{kl}) = \det\left[\Psi_{22}(s_1, s_2, y_{kl})\right] \tag{23.125}$$

where diag$(1 \cdots 1\ s_1 \cdots s_1\ s_2 \cdots s_2)$ represents a diagonal matrix in which each of the first N_r elements is unity, each of the next N_1 elements is s_1, and each of the last N_2, elements is s_2. Then, from the network theory, the input admittance at port 1 is given by

$$\mathbf{Y}_{\text{in}}(s_1, s_2, y_{kl}) = \frac{\mathbf{Y}_{12}\text{adj}\ [\mathbf{Y}_{22}(s_1, s_2, y_{kl})]\mathbf{Y}_{12}^{\text{T}}}{\det\left[\mathbf{Y}_{22}(s_1, s_2, y_{kl})\right]} = \frac{p(s_1, s_2, y_{kl})}{\Delta(s_1, s_2, y_{kl})} \tag{23.126}$$

where $\Delta(s_1, s_2, y_{kl})$ is defined by Equation 23.125 and is a strictly Hurwitz polynomial for any set of real values of the $(N-1)\ (N-2)/2$ independent parameters $\{y_{kl}: 1 < k < l \leq N\}$. Table 23.4 lists polynomial $\Delta(s_1, s_2, y_{kl})$ for $N_r = 1$ $(N_1, N_2) = (2, 1)$ and $(N_1, N_2) = (2, 2)$ [30].

Having obtained the parameterized strictly Hurwitz polynomial $\Delta(s_1, s_2, y_{kl})$ the design procedure of a 2-D IIR filter can be summarized as follows.

TABLE 23.4 2-*V* Strictly Hurwitz Polynomials

N_1	N_2	N	$\Delta(s_1, s_2, y_{kl})$
1	1	4	$s_1 s_2 + y_{24}^2 s_1 + y_{23}^2 s_2 + y_{34}^2$
2	1	5	$s_1^2 s_2 + y_{25}^2 s_1^2 + (y_{23}^2)(y_{23}^2 + y_{24}^2) s_1 s_2 + (y_{35}^2 + y_{45}^2) s_1 + y_{34}^2 s_2 + (y_{23}y_{46} - y_{24}y_{35} + y_{25}y_{34})^2$
2	2	6	$s_1^2 s_2^2 + (y_{23}^2 + y_{24}^2) s_1 s_2^2 + (y_{25}^2 + y_{26}^2) s_1^2 s_2 + y_{56}^2 s_1^2 + y_{34}^2 s_2^2 + (y_{35}^2 + y_{36}^2 - y_{45}^2 + y_{46}^2) s_1 s_2 + y_{34}^2 s_2 + [(y_{23}y_{56} - y_{25}y_{36} + y_{26}y_{35})^2 + (y_{24}y_{56} - y_{25}y_{46} + y_{26}y_{45})^2] s_1 + [(y_{23}y_{45} - y_{24}y_{35} + y_{25}y_{34})^2 + (y_{23}y_{46} - y_{24}y_{36} + y_{26}y_{34})^2] s_2 + (y_{34}y_{56} - y_{35}y_{46} + y_{36}y_{45})^2$

Step 1. Construct a parameterized analog transfer function of the 2-D IIR filter, by using the Hurwitz polynomial $\Delta(s_1, s_2, y_{kl})$

$$H(s_1, s_2, y_{kl}, a_{ij}) = \frac{p(s_1, s_2)}{\Delta(s_1, s_2, y_{kl})} \tag{23.127}$$

where

$$p(s_1, s_2) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} a_{ij} s_1^i s_2^j$$

is an arbitrary 2-V polynomial in s_1 and s_2 with degree in each variable not greater than the corresponding degree of the denominator.

Step 2. Perform the double bilinear transformation to the parameterized analog transfer function obtained in step 1.

$$H(z_1, z_2, y_{kl}, a_{ij}) = \frac{p(s_1, s_2)}{\Delta(s_1, s_2, y_{kl})}\Big|_{s_i = 2(z_i - 1)/T_i(z_i + 1),\, i = 1, 2} \tag{23.128}$$

Step 3. Construct an objective function according to the given design specifications and the parameterized discrete transfer function obtained in step 2.

$$J(\mathbf{x}) = \sum_{n_1} \sum_{n_2} [M(n_1, n_2) - M_I(n_1, n_2)]^p \tag{23.129}$$

where p is an even positive, $M(n_1, n_2)$ and $M_I(n_1, n_2)$ are the actual and desired amplitude responses, respectively, of the required filter at frequencies $(\omega_{1n1}, \omega_{1n2})$, and $\mathbf{x}$ is the vector consisting of parameters $\{y_{kl}: 1 < k < l \leq N\}$ and $\{a_{ij}, 0 \leq i \leq N_1, 0 \leq j \leq N_2\}$.

Step 4. Apply an optimization algorithm to find the optimal vector $\mathbf{x}$ that minimizes the objective function and substitute the resulting x into Equation 23.128 to obtain the required transfer function $H(z_1, z_2)$.

Example 23.8 [33]

By using the preceding approach, design a 2-D circularly symmetric low-pass filter of order (5, 5) with $\omega_p = 0.2\pi$, assuming that $\omega_{s1} = \omega_{s2} = 1.2\pi$.

Solution

1. Construct the desired amplitude response of the desired filter

$$M_I(\omega_{1n_1}, \omega_{2n_2}) = \begin{cases} 1 & \text{for } \left(\omega_{1n_1}^2 + \omega_{2n_2}^2\right) \leq 0.2\pi \\ 0 & \text{otherwise} \end{cases}$$

where

$$\omega_{1n_1} = \begin{cases} 0.01\pi n_2 & \text{for } 0 \leq n_1 \leq 20 \\ 0.01\pi n_1 & \text{for } 21 \leq n_1 \leq 24 \end{cases}$$

and

$$\boldsymbol{\omega}_{2n_2} = \boldsymbol{\omega}_{1(24-n_2)} \quad \text{for } 0 \le n_2 \le 24$$

2. Construct the 2-D analog transfer function and perform double bilinear transformation to obtain the discrete transfer function. The analog transfer function at hand is assumed to be an all-pole transfer function of the form

$$\mathbf{H}(s_1, s_2, \mathbf{x}) = \frac{1}{\mathbf{D}(s_1, s_2, y_{kl})}$$

Therefore, the corresponding discrete transfer function can be written as

$$H(z_1, z_2, \mathbf{x}) = \frac{A(z_1 + 1)^5 \ (z_2 + 1)^5}{\sum_{i=0}^{5} \sum_{j=0}^{5} b_{ij} \ z_1^i z_2^j} \tag{23.130}$$

TABLE 23.5 Coefficients of Transfer Function in Equation 23.130
$[A = 0.28627, b_{ij}: 0 \le i \le 5, \ 0 \le j \le 5]$

0.0652	−0.6450	3.3632	−4.8317	0.3218	−0.1645
−0.7930	7.8851	−25.871	23.838	3.4048	3.4667
4.2941	−28.734	61.551	−29.302	−13.249	−25.519
−6.3054	28.707	−33.487	−7.2275	−22.705	83.011
0.7907	1.4820	−7.4214	−33.313	136.76	−128.43
−0.4134	6.0739	−36.029	101.47	140.20	78.428

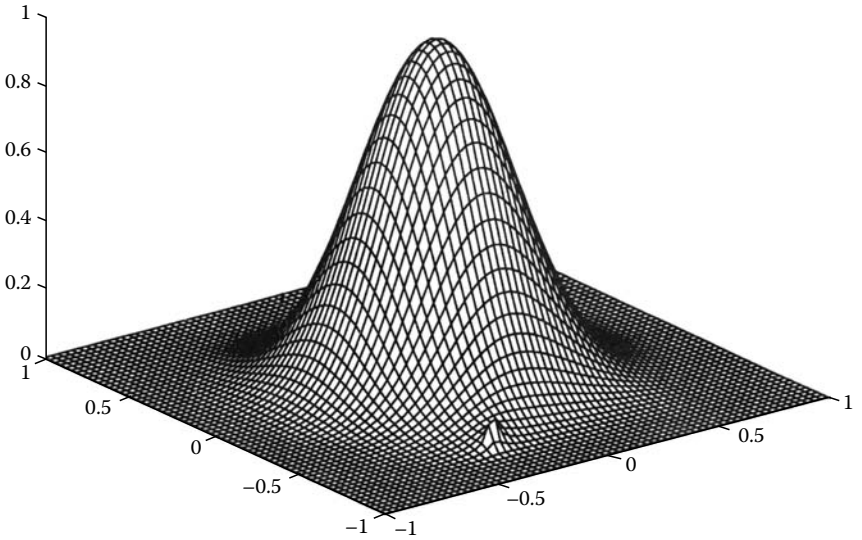


FIGURE 23.23 Amplitude response of circularly symmetric low-pass filter of Example 23.8.

where

$$\sum_{i=0}^{5}\sum_{j=0}^{5} b_{ij} z_1^i z_2^j = (z_1+1)^5 (z_2+1)^5 \Delta(s_1, s_2, y_{kl})\big|_{s_i=(z_i-1/z_i+1),\, i=1,2}$$

contains $(N-1)(N-2)/2 = 36$ parameters.

3. Optimization: A conventional quasi-Newton algorithm has been applied to minimize the objective function in Equation 23.129 with $p = 2$. The resulting coefficients are listed in Table 23.5. The amplitude response of the resulting filter is depicted in Figure 23.23.

References

1. J. L. Shanks, S. Treitel, and J. H. Justice, Stability and synthesis of two-dimensional recursive filters, *IEEE Trans. Audio Electroacoustic.*, AU-20, 115–128, June 1972.
2. W. S. Lu and A. Antoniou, *Two-Dimensional Digital Filters*, New York: Marcel Dekker, 1992.
3. M. Ahmadi, A. G. Constantinides, and R. A. King, Design technique for a class of stable two-dimensional recursive digital filters, in *Proc.* 1976 *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Philadelphia, Pennsylvania, 1976, 145–147.
4. A. M. Ali, A. G. Constantinides, and R. A. King, On 2-variable reactance functions for 2-dimensional recursive filters, *Electron. Lett.*, 14, 12–13, January 1978.
5. R. King et al., *Digital Filtering in One and Two Dimensions: Design and Applications*, New York: Plenum, 1989.
6. R. A. King and A. H. Kayran, A new transformation technique for the design of 2-dimensional stable recursive digital filters, in *Proc. IEEE Int. Symp. Circuits Syst.*, Chicago, pp. 196–199, April 1981.
7. J. M. Costa and A. N. Venetsanopoulos, A group of linear spectral transformations for two-dimensional digital filters, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-24, 424–425, October 1976.
8. K. P. Prasad, A. Antoniou, and B. B. Bhattacharyya, On the properties of linear spectral transformations for 2-dimensional digital filters, *Circuits Syst. Signal Process.*, 2, 203–211, 1983.
9. A. H. Kayran and R. A. King, Design of recursive and nonrecursive fan filters with complex transformations, *IEEE Trans. Circuits Syst.*, CAS-30, 849–857, 1983.
10. A. G. Constantinides, Spectral transformations for digital filters, *IEEE Proc.*, 117, 1585–1590, August 1970.
11. N. A. Pendergrass, S. K. Mitra, and E. I. Jury, Spectral transformations for two-dimensional digital filters, *IEEE Trans. Circuits Syst.*, CAS-23, 26–35, January 1976.
12. K. Hirano and J. K. Aggarwal, Design of two-dimensional recursive digital filters, *IEEE Trans. Circuits Syst.*, CAS-25, 1066–1076, December 1978.
13. S. K. Mitra and K. Hirano, Digital all-pass networks, *IEEE Trans. Circuits Syst.*, CAS-21, 688–700, September 1974.
14. J. M. Costa and A. N. Venetsanopoulos, Design of circularly symmetric two-dimensional recursive filters, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-22, 432–443, December 1974.
15. D. M. Goodman, A design technique for circularly symmetric low-pass filters, *IEEE Trans. Acoust., Speech, Signal Process.*, ASSP-26, 290–304, August 1978.
16. G. V. Mendonca, A. Antoniou, and A. N. Venetsanopoulos, Design of two-dimensional pseudorotated digital filters satisfying prescribed specifications, *IEEE Trans. Circuits Syst.*, CAS-34, 1–10, January 1987.
17. R. Fletcher, *Practical Methods of Optimization*, 2nd edn., New York: Wiley, 1990.
18. S. Chakrabarti and S. K. Mitra, Design of two-dimensional digital filters via spectral transformations, *Proc. IEEE*, 6, 905–914, June 1977.

19. A. Antoniou, *Digital Filters: Analysis, Design and Applications*, 2nd edn., New York: McGraw-Hill, 1993.
20. G. A. Maria and M. M. Fahmy, An l_p design technique for two-dimensional digital recursive filters, *IEEE Trans. Acoust., Speech, Signal Process.,* ASSP-22, 15–21, February 1974.
21. C. Charalambous, A unified review of optimization, *IEEE Trans. Microwave Theory Tech.,* MTT-22, 289–300, March 1974.
22. C. Charalambous and A. Antoniou, Equalization of recursive digital filters, *IEE Proc., Pt. G*, 127, 219–225, October 1980.
23. C. Charalambous, Design of 2-dimensional circularly-symmetric digital filters, *IEE Proc*., *Pt. G*, 129, 47–54, April 1982.
24. C. Charalambous, Acceleration of the least *p*th algorithm for minimax optimization with engineering applications, *Math Program***,** 17, 270–297, 1979.
25. P. K. Rajan and M. N. S. Swamy, Quadrantal symmetry associated with two-dimensional digital transfer functions, *IEEE Trans. Circuit Syst*., CAS-29, 340–343, June 1983.
26. A. Antoniou and W. S. Lu, Design of two-dimensional digital filters by using the singular value decomposition, *IEEE Trans. Circuits Syst*., CAS-34, 1191–1198, Oct. 1987.
27. G. W. Stewart, *Introduction to Matrix Computations*, New York: Academic, 1973.
28. T. B. Deng and M. Kawamata, Frequency-domain design of 2-D digital filters using the iterative singular value decomposition, *IEEE Trans. Circuits Syst*., CAS-38, 1225–1228, 1991.
29. T. B. Deng and T. Soma, Successively linearized non-negative decomposition of 2-D filter magnitude design specifications, *Digital Signal Process*., 3, 125–138, 1993.
30. P. A. Ramamoorthy and L. T. Bruton, Design of stable two-dimensional analog and digital filters with applications in image processing, *Circuit Theory Appl*., 7, 229–245, 1979.
31. T. Koga, Synthesis of finite passive networks with prescribed two-variable reactance matrices, *IEEE Trans. Circuit Theory,* CT-13, 31–52, 1966.
32. H. G. Ansel, On certain two-variable generalizations of circuit theory, with applications networks of transmission lines and lumped reactance, *IEEE Trans. Circuit Theory*, CT-11, 214–233, 1964.
33. P. A. Ramamoorthy and L. T. Bruton, Design of stable two-dimensional recursive filters, in *Topics in Applied Physics*, 42, T. S. Huang, Ed., New York: Springer-Verlag, 1981, pp. 41–83.

24

1-D Multirate Filter Banks

24.1 Introduction: Why Use Filter Banks? **24**-1
24.2 2-Band Filter Bank **24**-3
24.3 Multirate Filtering **24**-5
Multirate Filtering Theorem • General Results for *M*:1 Subsampling
24.4 Binary Filter Trees **24**-6
Transformation of the Filter Tree
24.5 Wavelets and Scaling Functions **24**-8
24.6 Good FIR Filters and Wavelets **24**-12
Perfect Reconstruction Condition • Good Filters/Wavelets • Some Simple Filters/Wavelets (Haar and LeGall) • Filters with Balanced *H* and *G* Frequency Responses (but Nonlinear Phase Responses)—Daubechies Wavelets • Filters with Linear Phase and Nearly Balanced Frequency Responses • Smoother Wavelets
24.7 IIR Filter Banks **24**-23
All-Pass Filter Design Methods • Transformation-Based Design Methods
24.8 Polyphase Representations and the Lifting Scheme **24**-28
Basic Polyphase Concepts • Orthogonal Lattice Structures • Linear Phase Structures • Lifting Scheme
24.9 Nonlinear Filter Banks **24**-39
24.10 *M*-Band Filter Banks **24**-41
24.11 Hilbert Pairs of Filter Banks (the Dual Tree) **24**-43
Dual-Tree Wavelet Transform • Common-Factor Dual-Tree Filter Design • Q-Shift Dual-Tree Filter Design • Metrics for Shift Dependence of a Filter Bank
References **24**-52

Nick G. Kingsbury
University of Cambridge

David B. H. Tay
Latrobe University

24.1 Introduction: Why Use Filter Banks?

An important class of digital filter system is the multirate filter bank. In this chapter we shall only be considering the one-dimensional (1-D) type of filter bank, as might be applied to typical signals that evolve with time, such as audio waveforms and communications signals.

The reason that filter banks are important is that we can often achieve useful functionality by separating signals into various different frequency bands and applying different processing to each

band. Typical examples of this are the widely used MP3 digital audio compression systems. In order to achieve a low coded data file size for digital music tracks, the MP3 coding standard specifies that the audio signal should be split into many frequency bands and that separate adaptive quantization should be applied to each band. The quantization is designed to take maximum advantage of the noise masking properties of the human auditory system, such that frequency bands containing substantial audio energy are quantized quite coarsely (because the quantizing noise gets masked by the signal here), whereas bands with low levels of audio are quantized more finely (since the masking is only effective at frequencies close to those containing most of the audio energy).

Filter banks can operate with filtered outputs being sampled at the same rate as the input signal. However, with many filters operating in parallel, this can lead to an unacceptably large amount of output data being generated. It is therefore sensible to subsample the outputs of the filters in a filter bank so that the total output data rate from all the filters is similar to that of the input. Such filters are called multirate filters, and the complete system is a multirate filter bank. When signals are subsampled, aliasing can occur and cause degradation of signal quality, but, with careful design, aliasing effects can be eliminated in multirate filter banks as long as the total output data rate is no less than the input rate. Multirate filters can be implemented with much less computational cost than the equivalent full rate filters, and, in the case of compression systems, they generate much less data to be coded by the adaptive quantizers.

An alternative way to view the advantages of filter banks is that, with careful design, they can encourage signal sparsity, i.e., most of the energy of an input signal can be concentrated in a small proportion of the output samples from the filter bank. Sparsity has been shown to be a key element in successfully performing many signal processing tasks, such as compression, denoising, signal separation, and other enhancement techniques. Sparsity can be achieved if, at any given time, the input signal can be well approximated by a weighted sum of the impulse responses from just a few of the filters in the filter bank. This occurs when the filters are matched to typical components of the signal.

Digital filter banks have been actively studied since the 1960s. However their use achieved a considerable boost with the development of wavelet theory in the 1980s. The theory of wavelet transforms was developed principally by French and Belgian mathematicians, notably A. Grossman, J. Morlet, Y. Meyer, I. Daubechies, and S. Mallat, and efficient implementation of the wavelet transform is usually achieved with multirate filter banks. The two topics are now firmly linked and are of great importance for signal analysis and compression.

The discrete wavelet transform (DWT) may be used to analyze a wide variety of signals, particularly those that combine long low-frequency events with much shorter high-frequency events (e.g., transients). It has perhaps achieved its greatest success with images. Although these are two-dimensional (2-D) signals, the 1-D filters considered in this chapter are still highly relevant since 2-D wavelet transforms are usually achieved using separable 1-D processing along the rows and then down the columns of the image (or vice versa).

In this chapter we shall be introducing some of the basic ideas behind 1-D filter banks, and then will concentrate much of our coverage on the 2-band multirate filter bank, which is the workhorse of the DWT. In the final sections we shall extend the discussion to M-band ($M > 2$) filter banks and also to Hilbert pairs of filter banks, which lead to the dual-tree complex wavelet transform.

Most of the following discussions in this chapter will assume that the filters are finite-impulse-response (FIR) filters and that all samples of the input signal are available in the memory of the signal processing hardware. Hence causality of the filters is not an issue and filter taps corresponding to negative delays (positive powers of z in the z-transform) pose no implementation problems. In this situation, the most natural way to design filters is for zero overall delay (the zero-phase condition).

If these assumptions are not valid (e.g., in the case of a continuously evolving audio signal), then appropriate delays can usually be inserted so that the filters are still implementable. Only in the case of recursive infinite-impulse-response (IIR) filters is the causality issue a real potential problem; and so more care is required on this issue in Section 24.7. Elsewhere we shall ignore problems of causality.

24.2 2-Band Filter Bank

A simple analysis filter bank with just two bands is shown in Figure 24.1a. It comprises two filters H_0 and H_1 which split the input signal X into its lower and higher frequency components, respectively. If X is sampled at a rate of f_s Hz, then its frequency spectrum can occupy the range 0 to $\frac{1}{2} f_S$, while still satisfying Nyquist's rule; and so H_0 will normally be designed to pass frequencies from 0 to $\frac{1}{4} f_S$, and H_1 from $\frac{1}{4} f_S$ to $\frac{1}{2} f_S$. Hence, H_0 will be a low-pass filter and H_1 will be high-pass.

Let us initially consider the simplest practical form of 2-band filter bank in which H_0 and H_1 are both 2-tap FIR filters with coefficient vectors $\mathbf{h}_0 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$ and $\mathbf{h}_1 = [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$ (corresponding to the well-known Haar wavelet basis). Hence the filter outputs may be expressed as

$$\begin{aligned} y_0(n) &= \frac{1}{\sqrt{2}} x(n-1) + \frac{1}{\sqrt{2}} x(n) \\ y_1(n) &= \frac{1}{\sqrt{2}} x(n-1) - \frac{1}{\sqrt{2}} x(n) \end{aligned} \tag{24.1}$$

As z-transforms, Equation 24.1 becomes

$$\begin{aligned} Y_0(z) &= H_0(z)X(z) \quad \text{where} \quad H_0(z) = \frac{1}{\sqrt{2}}(z^{-1} + 1) \\ Y_1(z) &= H_1(z)X(z) \quad \text{where} \quad H_1(z) = \frac{1}{\sqrt{2}}(z^{-1} - 1) \end{aligned} \tag{24.2}$$

(By substituting $z = e^{j\omega}$, the reader may check that these filters are indeed low-pass and high-pass, respectively. We shall later extend the filters to be more complicated.)

In practice, $y_0(n)$ and $y_1(n)$ are only calculated at alternate (say even) values of n so that the total number of samples in vectors $\mathbf{y}_0$ and $\mathbf{y}_1$ is the same as in the input vector $\mathbf{x}$. This is shown by the 2:1 downsamplers on the right in Figure 24.1a.

It is straightforward to invert Equation 24.1 to obtain the two samples of x back from the filter output samples, as follows:

$$\begin{aligned} x(n-1) &= \frac{1}{\sqrt{2}} y_0(n) + \frac{1}{\sqrt{2}} y_1(n) \\ x(n) &= \frac{1}{\sqrt{2}} y_0(n) - \frac{1}{\sqrt{2}} y_1(n) \end{aligned} \quad \text{for } n \text{ even.} \tag{24.3}$$

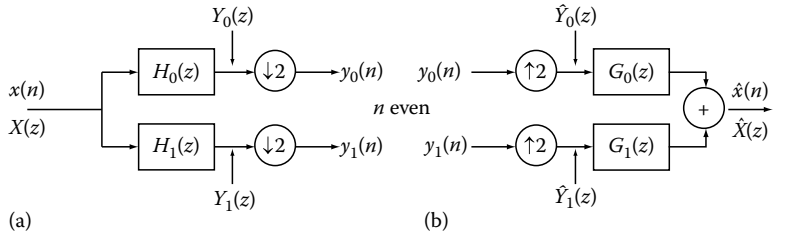


FIGURE 24.1 Two-band filter banks for analysis (a) and reconstruction (b).

Assuming that the missing samples of $\mathbf{y}_0$ and $\mathbf{y}_1$ are zero at odd values of n, we may combine the equations in Equation 24.3 into a single expression for $x(n)$, valid for all n:

$$x(n) = \frac{1}{\sqrt{2}}[y_0(n+1) + y_0(n)] + \frac{1}{\sqrt{2}}[y_1(n+1) - y_1(n)] \tag{24.4}$$

or as z-transforms

$$X(z) = G_0(z)Y_0(z) + G_1(z)Y_1(z) \tag{24.5}$$

where

$$G_0(z) = \frac{1}{\sqrt{2}}(z+1) \quad \text{and} \quad G_1(z) = \frac{1}{\sqrt{2}}(z-1) \tag{24.6}$$

Note that the factors of $\frac{1}{\sqrt{2}}$ in the coefficients of the H and G filters are chosen to ensure that the l_2-norm (energy) of each vector of filter coefficients is unity, so that total energy is preserved from the X-domain to the Y-domain and vice versa.

In Equation 24.5 the signals $Y_0(z)$ and $Y_1(z)$ are not really the same as $Y_0(z)$ and $Y_1(z)$ in Equation 24.2 because those in Equation 24.2 have not had alternate samples set to zero. Also, in Equation 24.5 $X(z)$ is the reconstructed output whereas in Equation 24.2 it is the input signal.

To avoid confusion we shall use $\hat{X}$, $\hat{Y}_0$, and $\hat{Y}_1$ for the signals in Equation 24.5, so it becomes

$$\hat{X}(z) = G_0(z)\hat{Y}_0(z) + G_1(z)\hat{Y}_1(z) \tag{24.7}$$

We may show this reconstruction operation as upsampling followed by two filters, as in Figure 24.1b, forming a 2-band reconstruction filter bank.

If $\hat{Y}_0$ and $\hat{Y}_1$ are not the same as Y_0 and Y_1, it is important to know how they do relate to each other. Now

$$\hat{y}_0(n) = y_0(n) \quad \text{for } n \text{ even}, \quad \hat{y}_0(n) = 0 \quad \text{for } n \text{ odd} \tag{24.8}$$

Therefore its z-transform, $\hat{Y}_0(z)$, is a polynomial in z, comprising *only* the terms in even powers of z from $Y_0(z)$. This may be written as

$$\hat{Y}_0(z) = \sum_{\text{even } n} y_0(n)z^{-n} = \sum_{\text{all } n} \frac{1}{2}[y_0(n)z^{-n} + y_0(n)(-z)^{-n}] = \frac{1}{2}[Y_0(z) + Y_0(-z)] \tag{24.9}$$

Similarly

$$\hat{Y}_1(z) = \frac{1}{2}[Y_1(z) + Y_1(-z)] \tag{24.10}$$

This is our general model for downsampling by two, followed by upsampling by two as defined in Equation 24.8.

Substituting Equations 24.9 and 24.10 into Equation 24.7 and then using Equation 24.2, we get

$$\begin{aligned}\hat{X}(z) &= \frac{1}{2}G_0(z)[Y_0(z) + Y_0(-z)] + \frac{1}{2}G_1(z)[Y_1(z) + Y_1(-z)] \\ &= \frac{1}{2}G_0(z)H_0(z)X(z) + \frac{1}{2}G_0(z)H_0(-z)X(-z) \\ &\quad + \frac{1}{2}G_1(z)H_1(z)X(z) + \frac{1}{2}G_1(z)H_1(-z)X(-z) \\ &= \frac{1}{2}X(z)[G_0(z)H_0(z) + G_1(z)H_1(z)] \\ &\quad + \frac{1}{2}X(-z)[G_0(z)H_0(-z) + G_1(z)H_1(-z)] \end{aligned} \tag{24.11}$$

This result will be used in Section 24.6.

24.3 Multirate Filtering

In order to be able to calculate the characteristics of more complicated configurations, for example, comprising cascaded filter banks such as used for wavelet transforms, it is necessary to derive some key results for multirate filter systems.

24.3.1 Multirate Filtering Theorem

To calculate the impulse and frequency responses for a multistage filter network with downsampling/upsampling between stages, we derive an important theorem for multirate filters.

THEOREM 24.1

The downsample–filter–upsample operation of Figure 24.2a *is equivalent to either the filter–downsample–upsample operation of Figure* 24.2b *or the downsample–upsample–filter operation of Figure* 24.2c, *if the filter is changed from* $H(z)$ *to* $H(z^2)$.

PROOF: Expressing the convolution and down + upsampling of Figure 24.2a in full:

$$\begin{aligned}\hat{y}(n) &= \sum_i x(n-2i)h(i) \quad \text{for } n \text{ even} \\ &= 0 \qquad\qquad\qquad\quad \text{for } n \text{ odd}\end{aligned} \tag{24.12}$$

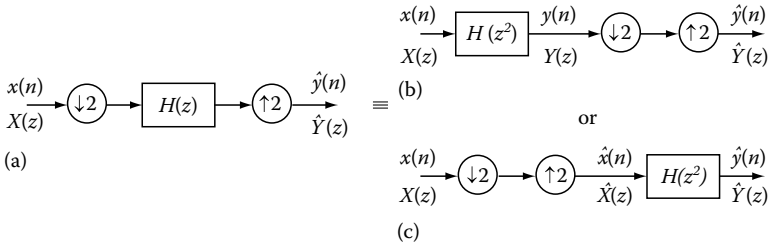


FIGURE 24.2 Multirate filtering (a). This figure shows the result of shifting a filter ahead of a downsampling operation (b) or after an upsampling operation (c).

Taking z-transforms

$$\hat{Y}(z) = \sum_n \hat{y}(n) z^{-n} = \sum_{\text{even } n} \sum_i x(n-2i) h(i) z^{-n} \tag{24.13}$$

Reversing the order of summation and letting $m = n - 2i$:

$$\begin{aligned}
\hat{Y}(z) &= \sum_i h(i) \sum_{\text{even } m} x(m) z^{-m} z^{-2i} \\
&= \sum_i h(i) z^{-2i} \sum_{\text{even } m} x(m) z^{-m} \\
&= H(z^2) \frac{1}{2} [X(z) + X(-z)] \qquad (24.14) \\
&= \frac{1}{2} [H(z^2) X(z) + H((-z)^2) X(-z)] \quad \text{since } (-z)^2 = z^2 \\
&= \frac{1}{2} [Y(z) + Y(-z)] \quad \text{where } Y(z) = H(z^2) X(z) \qquad (24.15)
\end{aligned}$$

Equation 24.15 describes the operations of Figure 24.2b. Hence the first result is proved. The result from Equation 24.14 gives

$$\hat{Y}(z) = H(z^2) \frac{1}{2} [X(z) + X(-z)] = H(z^2) \hat{X}(z) \tag{24.16}$$

This shows that the filter $H(z^2)$ may be placed after the down + upsampler as in Figure 24.2c, which proves the second result.

24.3.2 General Results for *M*:1 Subsampling

The results above may be extended to the case of M:1 down and upsampling as follows:

- $H(z)$ becomes $H(z^M)$ if shifted ahead of an M:1 downsampler or following an M:1 upsampler. (These two results are known as the Noble identities.)
- M:1 down + upsampling of a signal $X(z)$ produces

$$\hat{X}(z) = \frac{1}{M} \sum_{m=0}^{M-1} X(z e^{j2\pi m/M}) \tag{24.17}$$

These results will now be used to analyze binary filter trees.

24.4 Binary Filter Trees

For applications such as signal compression, the purpose of the 2-band filter bank is to compress most of the signal energy into the samples representing the low-frequency half of the signal band. Hence 50% of the filter-bank output samples (the low-pass half) may well contain 90% or more of the signal energy (if it is a signal with dominant low-frequency components such as an image or audio signal).

We may achieve greater compression if the low band is further split into two. This may be repeated a number of times to give the binary filter tree. An example with four levels of decomposition is shown in Figure 24.3.

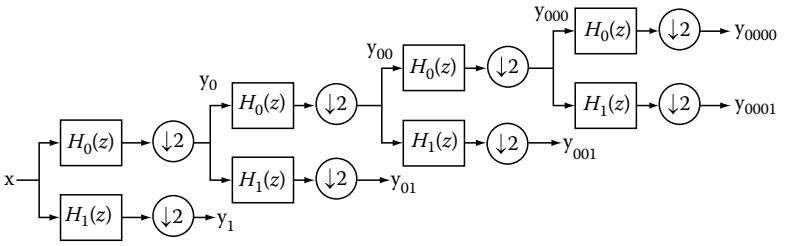


FIGURE 24.3 Extension of the 2-band filter bank into a binary filter tree. H_0 and H_1 are low-pass and high-pass filters.

For an N-sample input vector $\mathbf{x}$, sampled at a frequency f_s, the sizes and bandwidths of the signals of the 4-level filter tree are

Signal	No. of Samples	Approximate Pass Band
$\mathbf{x}$	N	$0 \rightarrow \frac{1}{2}f_s$
$\mathbf{y}_1$	$N/2$	$\frac{1}{4} \rightarrow \frac{1}{2}f_s$
$\mathbf{y}_{01}$	$N/4$	$\frac{1}{8} \rightarrow \frac{1}{4}f_s$
$\mathbf{y}_{001}$	$N/8$	$\frac{1}{16} \rightarrow \frac{1}{8}f_s$
$\mathbf{y}_{0001}$	$N/16$	$\frac{1}{32} \rightarrow \frac{1}{16}f_s$
$\mathbf{y}_{0000}$	$N/16$	$0 \rightarrow \frac{1}{32}f_s$

Because of the downsampling by 2 at each level, the total number of output samples equals N, regardless of the number of levels in the tree; so the process is nonredundant.

The H_0 filter is normally designed to be a low-pass filter with a passband from 0 to approximately $\frac{1}{4}$ of the input sampling frequency for that stage; and H_1 is a high-pass (bandpass) filter with a pass band approximately from $\frac{1}{4}$ to $\frac{1}{2}$ of the input sampling frequency.

When formed into a 4-level tree, the filter outputs have the approximate pass bands given in the above table. The final output $\mathbf{y}_{0000}$ is a low-pass signal, while the other outputs are all bandpass signals, each covering a band of approximately one octave.

An inverse tree, mirroring Figure 24.3, may be constructed using filters G_0 and G_1 instead of H_0 and H_1, as shown for just one level in Figure 24.1b. If the PR conditions of Equations 24.40 and 24.41 are satisfied, then the output of each level will be identical to the input of the equivalent level in Figure 24.3, and the final output will be a PR of the input signal.

24.4.1 Transformation of the Filter Tree

Using the result of Equation 24.15, Figure 24.3 can be redrawn as in Figure 24.4 with all downsamplers moved to the outputs. (Note Figure 24.4 requires much more computation than Figure 24.3,

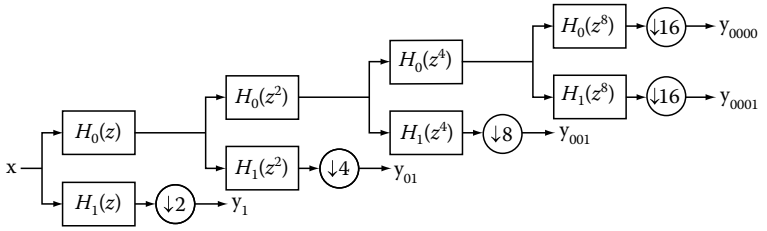


FIGURE 24.4 Binary filter tree, transformed so that all downsampling operations occur at the outputs.

but is a useful analysis aid.) We can now calculate the transfer function to each output (before the downsamplers) as

$$\begin{aligned} H_{01}(z) &= H_0(z)H_1(z^2) \\ H_{001}(z) &= H_0(z)H_0(z^2)H_1(z^4) \\ H_{0001}(z) &= H_0(z)H_0(z^2)H_0(z^4)H_1(z^8) \\ H_{0000}(z) &= H_0(z)H_0(z^2)H_0(z^4)H_0(z^8) \end{aligned} \tag{24.18}$$

In general the transfer functions to the two outputs at level k of the tree are given by

$$H_{k,1} = \left(\prod_{i=0}^{k-2} H_0(z^{2^i})\right) H_1(z^{2^{k-1}}) \quad \text{and} \quad H_{k,0} = \prod_{i=0}^{k-1} H_0(z^{2^i}) \tag{24.19}$$

For the Haar filters of Equation 24.2, the transfer functions to the outputs of the 4-level tree become

$$\begin{aligned} H_{01}(z) &= \frac{1}{2}[(z^{-3} + z^{-2}) - (z^{-1} + 1)] \\ H_{001}(z) &= \frac{1}{2\sqrt{2}}[(z^{-7} + z^{-6} + z^{-5} + z^{-4}) - (z^{-3} + z^{-2} + z^{-1} + 1)] \\ H_{0001}(z) &= \frac{1}{4}[(z^{-15} + z^{-14} + z^{-13} + z^{-12} + z^{-11} + z^{-10} + z^{-9} + z^{-8}) \\ &\quad - (z^{-7} + z^{-6} + z^{-5} + z^{-4} + z^{-3} + z^{-2} + z^{-1} + 1)] \\ H_{0000}(z) &= \frac{1}{4}(z^{-15} + z^{-14} + z^{-13} + z^{-12} + z^{-11} + z^{-10} + z^{-9} + z^{-8} \\ &\quad + z^{-7} + z^{-6} + z^{-5} + z^{-4} + z^{-3} + z^{-2} + z^{-1} + 1) \end{aligned} \tag{24.20}$$

These transfer functions are illustrated in Figure 24.5, simply by interpreting the coefficients of the z-transform polynomials as samples of the corresponding impulse responses; and by substituting $z = e^{j2\pi f/f_s}$ into the transfer functions and evaluating their magnitudes as a function of frequency, f, to obtain the frequency responses.

24.5 Wavelets and Scaling Functions

The process of creating the outputs $\mathbf{y}_1$ to $\mathbf{y}_{0000}$ from $\mathbf{x}$ in Figure 24.3 is known as the DWT; and the reconstruction process is the inverse DWT [1—3]. The term wavelet refers to the impulse response of the cascade of filters which leads to a given bandpass output.

Since the frequency responses of the bandpass bands are scaled down by 2:1 at each level (see Figure 24.5), their impulse responses become longer by the same factor at each level, but their shapes remain very similar. The basic impulse response wave shape is almost independent of scale and is known as the mother wavelet. The impulse response to a low-pass output $H_{k,0}$ is called the scaling function at level k.

Figure 24.5 shows these effects using the impulse responses and frequency responses for the five outputs of the 4-level tree of Haar filters, based on the z-transforms given in Equation 24.20. Notice the abrupt transitions in the midddle and at the ends of the Haar wavelets. These result in noticeable blocking artifacts in decompressed images and other signal types. For this reason we are interested in developing a more general theory of wavelets, with improved impulse and frequency responses.

There are several forms of the wavelet transform but the focus here will be on the form that is related to the 2-band filter bank. This form, which is the most popular in applications, is known as the DWT. Unless otherwise stated, it is assumed that the coefficients of all low-pass filters are scaled such that they

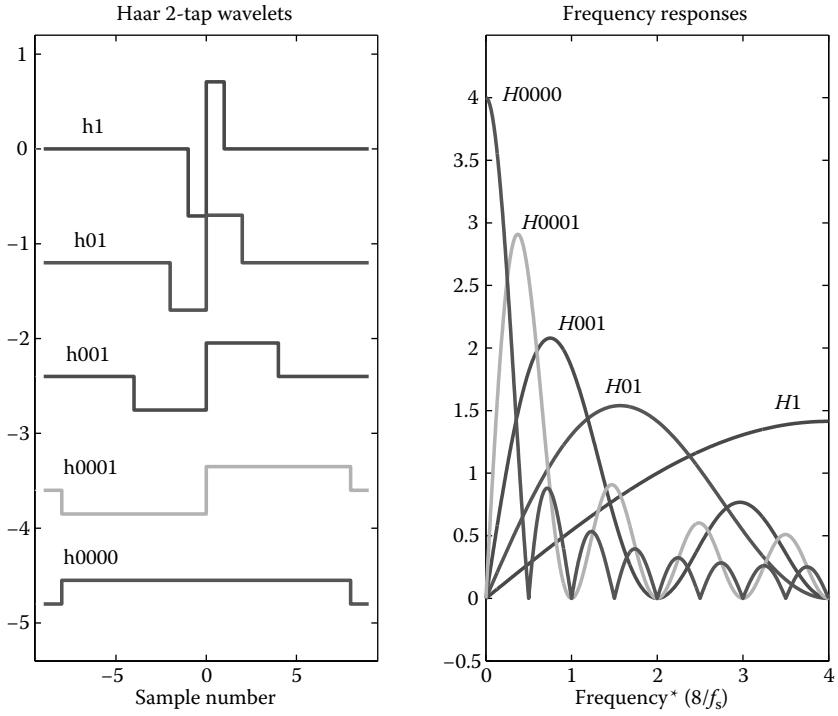


FIGURE 24.5 Impulse responses and frequency responses of the 4-level tree of Haar filters.

sum to the value $\sqrt{2}$, i.e., the DC frequency response is $H_0(e^{j0}) = H_0(1) = \sum_n h_0(n) = \sqrt{2}$. (This scaling has the effect of making the DWT tend to be an energy preserving transformation.)

Central to the principle of the DWT are the two-scale equations that link the discrete time filters of the filter bank to continuous time functions known as the scaling function and the mother wavelet. These equations exist both for the analysis side of the filter bank and for the synthesis side:

1. Analysis side equations:

$$\tilde{\phi}(t) = \sqrt{2}\sum_k h_0(k)\tilde{\phi}(2t-k) \tag{24.21}$$

$$\tilde{\psi}(t) = \sqrt{2}\sum_k h_1(k)\tilde{\phi}(2t-k) \tag{24.22}$$

where $\tilde{\phi}(t)$ and $\tilde{\psi}(t)$ are the analysis scaling function and mother wavelet, respectively. The coefficients of filters $H_0(z)$ and $H_1(z)$ are $h_0(k)$ and $h_1(k)$:
i.e.,

$$H_0(z) = \sum_k h_0(k)z^{-k} \quad \text{and} \quad H_1(z) = \sum_k h_1(k)z^{-k} \tag{24.23}$$

2. Synthesis side equations:

$$\phi(t) = \sqrt{2}\sum_k g_0(k)\phi(2t-k) \tag{24.24}$$

$$\psi(t) = \sqrt{2}\sum_k g_1(k)\phi(2t-k) \tag{24.25}$$

where $\phi(t)$ and $\psi(t)$ are the synthesis scaling function and mother wavelet, respectively. The coefficients of filters $G_0(z)$ and $G_1(z)$ are $g_0(k)$ and $g_1(k)$:
i.e.,

$$G_0(z) = \sum_k g_0(k)z^{-k} \quad \text{and} \quad G_1(z) = \sum_k g_1(k)z^{-k} \tag{24.26}$$

Note that the scaling functions ($\tilde{\phi}(t)$ or $\phi(t)$) are effectively the result of a cascade of low-pass filters (H_0 or G_0) across very many progressively finer scales. The mother wavelets ($\tilde{\psi}(t)$ or $\psi(t)$) are the result of cascading H_1 or G_1 at the coarsest scale, with very many low-pass filters at finer scales.

In the Fourier domain on the analysis side, this gives the infinite product formulae, achieved by transforming Equations 24.21 and 24.22 as follows:

$$\begin{aligned}
\tilde{\Phi}(\omega) &= \sqrt{2}\sum_k h_0(k)\int_{-\infty}^{\infty}\tilde{\phi}(2t-k)e^{-j\omega t}\,dt \\
&= \sqrt{2}\sum_k h_0(k)\int_{-\infty}^{\infty}\tilde{\phi}(\tau)e^{-j\omega\tau/2}e^{-j\omega k/2}\frac{d\tau}{2} \quad \text{if } \tau = 2t-k \\
&= \frac{1}{\sqrt{2}}H_0(e^{j\omega/2})\tilde{\Phi}(\omega/2) \\
&= \frac{1}{\sqrt{2}}H_0(e^{j\omega/2})\frac{1}{\sqrt{2}}H_0(e^{j\omega/4})\tilde{\Phi}(\omega/4) \\
&\vdots \\
&= \prod_{k=1}^{\infty}\left\{\frac{1}{\sqrt{2}}H_0\left(e^{j\omega/2^k}\right)\right\}\tilde{\Phi}(0)
\end{aligned} \tag{24.27}$$

Similarly

$$\begin{aligned}
\tilde{\Psi}(\omega) &= \frac{1}{\sqrt{2}}H_1(e^{j\omega/2})\tilde{\Phi}(\omega/2) \\
&= \frac{1}{\sqrt{2}}H_1(e^{j\omega/2})\prod_{k=2}^{\infty}\left\{\frac{1}{\sqrt{2}}H_0\left(e^{j\omega/2^k}\right)\right\}\tilde{\Phi}(0)
\end{aligned} \tag{24.28}$$

where $\tilde{\Phi}(\omega)$ and $\tilde{\Psi}(\omega)$ are the Fourier transforms of $\tilde{\phi}(t)$ and $\tilde{\psi}(t)$, respectively.

The above equations explicitly define the analysis scaling function and mother wavelet in terms of the analysis filters. Similar infinite product formulae exist linking the synthesis scaling function and wavelet to the synthesis filters.

Equations 24.27 and 24.28 are reminiscent of the equivalent transfer functions defined in Equation 24.19 for the binary filter tree. The relationships can be made more precise as follows. The transfer functions (repeated here for convenience) for the k level tree are

$$H_{k,0}(z) \equiv \prod_{i=0}^{k-1}H_0(z^{2^i}) = \sum_n h_{k,0}(n)z^{-n} \tag{24.29}$$

$$H_{k,1}(z) \equiv H_1(z^{2^{k-1}}) \prod_{i=0}^{k-2} H_0(z^{2^i}) = \sum_n h_{k,1}(n) z^{-n} \tag{24.30}$$

where $h_{k,0}(n)$ and $h_{k,1}(n)$ are the coefficients of $H_{k,0}(z)$ and $H_{k,1}(z)$.

From the coefficients of the equivalent filters, we construct the following piecewise constant functions:

$$\tilde{\phi}^{(k)}(t) \equiv 2^{k/2} h_{k,0}(n), \quad \frac{n}{2^k} \le t < \frac{n+1}{2^k} \tag{24.31}$$

$$\tilde{\psi}^{(k)}(t) \equiv 2^{k/2} h_{k,1}(n), \quad \frac{n}{2^k} \le t < \frac{n+1}{2^k} \tag{24.32}$$

Each coefficient defines a rectangular pulse of width 2^{-k}, and both $\tilde{\phi}^{(k)}(t)$ and $\tilde{\psi}^{(k)}(t)$ are made up of sequences of such pulses. The width of the pulse is halved with each increase in tree level (scale) k but the $2^{k/2}$ amplitude normalization ensures that the total energy of each function remains constant. When the number of levels k tends to infinity, the scaling function and mother wavelet are obtained:

$$\tilde{\phi}(t) = \lim_{k\to\infty} \tilde{\phi}^{(k)}(t) \quad \text{and} \quad \tilde{\psi}(t) = \lim_{k\to\infty} \tilde{\psi}^{(k)}(t) \tag{24.33}$$

i.e., the shape of the scaling function and wavelet are the shape of the impulse response of equivalent filters of the binary tree as $k \to \infty$. In practice, the shape of the scaling function and wavelet is obtained quite accurately after $k \simeq 6$ levels. Similar relationships exist on the synthesis side.

The discussion above involving infinite products assumes convergence but this is not always guaranteed. The filters must be properly designed to ensure convergence. One condition that is necessary for convergence is

$$H_0(e^{j\pi}) = H_0(-1) = 0 \quad \text{and} \quad G_0(e^{j\pi}) = G_0(-1) = 0 \tag{24.34}$$

i.e., the low-pass filter responses must vanish at the aliasing (half-sampling) frequency. This means that $(1 + z^{-1})$ must be a factor of both $H_0(z)$ and $G_0(z)$.

A simple degree-1 factor, i.e., $(1 + z^{-1})$, may not necessarily guarantee convergence; and even if convergence is achieved, the resultant scaling and wavelet functions may not be smooth. In general, higher order factors are usually imposed on the low-pass filters, such that

$$H_0(z) = 2^{-L_H}(1 + z^{-1})^{L_H} R_H(z) \tag{24.35}$$

$$G_0(z) = 2^{-L_G}(1 + z^{-1})^{L_G} R_G(z) \tag{24.36}$$

where $R_H(z)$ and $R_G(z)$ are the remainder factors of $H_0(z)$ and $G_0(z)$, respectively.

The orders L_H and L_G determine the numbers of vanishing moments (VM) of the corresponding wavelet function, such that

$$\int_{-\infty}^{+\infty} t^n \tilde{\psi}(t) \mathrm{d}t = 0 \quad \text{for } n = 0, \ldots, L_H - 1 \tag{24.37}$$

$$\int_{-\infty}^{+\infty} t^n \psi(t) \mathrm{d}t = 0 \quad \text{for } n = 0, \ldots, L_G - 1 \tag{24.38}$$

In general the higher the number of VM, the smoother will be the scaling and wavelet functions.

24.6 Good FIR Filters and Wavelets

24.6.1 Perfect Reconstruction Condition

One of the most important requirements for most filter banks is to be able to reconstruct perfectly the input signal $X(z)$ at the reconstruction filter bank output $\hat{X}(z)$ (see Figure 24.1).

We now wish to find the constraints on arbitrary filters, $\{H_0, H_1, G_0, G_1\}$, such that perfect reconstruction (PR) occurs.

Repeating the result from Equation 24.11, the input–output relationship for the pair of filter banks in Figure 24.1 is

$$\begin{aligned}\hat{X}(z) = &\frac{1}{2}X(z)[G_0(z)H_0(z) + G_1(z)H_1(z)] \\ &+ \frac{1}{2}X(-z)[G_0(z)H_0(-z) + G_1(z)H_1(-z)]\end{aligned} \tag{24.39}$$

If we require $\hat{X}(z) \equiv X(z)$—the PR condition—then

$$G_0(z)H_0(z) + G_1(z)H_1(z) \equiv 2 \tag{24.40}$$

and

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) \equiv 0 \tag{24.41}$$

Equation 24.41 is known as the antialiasing condition because the term in $X(-z)$ in Equation 24.39 is the unwanted aliasing term caused by the 2 : 1 downsampling of $\mathbf{y}_0$ and $\mathbf{y}_1$.

It is straightforward to show that the expressions for $\{H_0, H_1, G_0, G_1\}$, given in Equations 24.2 and 24.6 for the filters based on the Haar wavelet basis, satisfy Equations 24.40 and 24.41. They are the simplest set of filters which do.

24.6.2 Good Filters/Wavelets

Our main aim now is to search for better filters which result in wavelets and scaling functions that are smoother than the Haar functions (i.e., which avoid the discontinuities evident in the waveforms of Figure 24.5).

We start our search with the two PR identities, Equations 24.40 and 24.41.

The usual way of satisfying the antialiasing condition Equation 24.41, while permitting H_0 and G_0 to have low-pass responses (passband where $\mathrm{Re}[z] > 0$) and H_1 and G_1 to have high-pass responses (passband where $\mathrm{Re}[z] < 0$), is with the following relations in which k must be an odd integer:

$$H_1(z) = z^{-k}G_0(-z) \quad \text{and} \quad G_1(z) = z^k H_0(-z) \tag{24.42}$$

Hence:

$$\begin{aligned}G_0(z)H_0(-z) + G_1(z)H_1(-z) &= G_0(z)H_0(-z) + z^k H_0(-z)(-z)^{-k}G_0(z) \\ &= G_0(z)H_0(-z) + (-1)^{-k}H_0(-z)G_0(z) \\ &\equiv 0 \quad \text{if } k \text{ is odd}\end{aligned} \tag{24.43}$$

and so Equation 24.41 is satisfied, as required.

Now we define the low-pass product filter:

$$P(z) = H_0(z)G_0(z) \tag{24.44}$$

and substitute relations in Equation 24.42 into Equation 24.40 to get

$$\begin{aligned} G_0(z)H_0(z) + G_1(z)H_1(z) &= G_0(z)H_0(z) + H_0(-z)G_0(-z) \\ &= P(z) + P(-z) \equiv 2 \end{aligned} \tag{24.45}$$

Hence satisfying Equation 24.40 requires that all $P(z)$ terms in even powers of z be zero, except the z^0 term which must be 1. However the $P(z)$ terms in odd powers of z may take any desired values since they cancel out in Equation 24.45. If P is low-pass, this type of filter is often known as a halfband filter since $P(e^{j\omega}) + P(e^{j(\omega+\pi)}) =$ a constant, and so the bandwidth of P must be half of the input bandwidth.

A further commonly applied constraint on $P(z)$ is that it should be zero phase,* in order to minimize the magnitude of any distortions due to samples from the high-pass filters being suppressed (perhaps as a result of quantization or denoising). Hence $P(z)$ should be of the form:

$$P(z) = \cdots + p_5 z^5 + p_3 z^3 + p_1 z + 1 + p_1 z^{-1} + p_3 z^{-3} + p_5 z^{-5} + \cdots \tag{24.46}$$

The design of a set of PR filters H_0, H_1 and G_0, G_1 can now be summarized as

1. Choose a set of coefficients $p_1, p_3, p_5 \cdots$ to give a zero-phase low-pass product filter $P(z)$ with desirable characteristics. (This is nontrivial and is discussed below.)
2. Factorize $P(z)$ into $H_0(z)$ and $G_0(z)$, preferably so that the two filters have similar low-pass frequency responses.
3. Calculate $H_1(z)$ and $G_1(z)$ from Equation 24.42.

It can help to simplify the tasks of choosing $P(z)$ and factorizing it if, based on the zero-phase requirement, we transform $P(z)$ into $P_t(Z)$ such that

$$P(z) = P_t(Z) = 1 + p_{t,1}Z + p_{t,3}Z^3 + p_{t,5}Z^5 + \cdots \quad \text{where} \quad Z = \frac{1}{2}(z + z^{-1}) \tag{24.47}$$

To calculate the frequency response of $P(z) = P_t(Z)$, let $z = e^{j\omega T_s}$.

Therefore

$$\therefore\ Z = \frac{1}{2}(e^{j\omega T_s} + e^{-j\omega T_s}) = \cos(\omega T_s) \tag{24.48}$$

This is a purely real function of ω, varying from 1 at $\omega = 0$ to -1 at $\omega T_s = \pi$ (half the sampling frequency). Hence we may substitute $\cos(\omega T_s)$ for Z in $P_t(Z)$ to obtain its frequency response directly.

24.6.3 Some Simple Filters/Wavelets (Haar and LeGall)

As discussed in Section 24.5, in order to achieve smooth wavelets after many levels of the binary tree, the low-pass filters $H_0(z)$ and $G_0(z)$ must both have a number of zeros at half the sampling frequency (at $z = -1$). These will also be zeros of $P(z)$, and so $P_t(Z)$ will have zeros at $Z = -1$ (each of which will correspond to a pair of zeros at $z = -1$).

* See the end of Section 24.1 for a discussion of causality assumptions related to this.

The simplest case is a single zero at $Z = -1$, so that $P_t(Z) = 1 + Z$.

$$\therefore\ P(z) = \frac{1}{2}(z + 2 + z^{-1}) = \frac{1}{2}(z + 1)(1 + z^{-1}) = G_0(z)H_0(z)$$

which gives the familiar Haar filters.

As we have seen in Figure 24.5, the Haar wavelets have significant discontinuities so we need to add more zeros at $Z = -1$. However to maintain PR, we must also ensure that all terms in even powers of Z in $P_t(Z)$ are zero, so the next more complicated P_t must be third-order and of the form

$$\begin{aligned} P_t(Z) &= (1 + Z)^2(1 + aZ) = 1 + (2 + a)Z + (1 + 2a)Z^2 + aZ^3 \\ &= 1 + \frac{3}{2}Z - \frac{1}{2}Z^3 \quad \text{if } a = -\frac{1}{2} \text{ to suppress the term in } Z^2 \end{aligned} \tag{24.49}$$

Allocating the factors of P_t such that $(1 + Z)$ gives H_0 and $(1 + Z)(1 + aZ)$ gives G_0:

$$\begin{aligned} H_0(z) &= \frac{1}{2}(z + 2 + z^{-1}) \\ G_0(z) &= \frac{1}{8}(z + 2 + z^{-1})(-z + 4 - z^{-1}) \\ &= \frac{1}{8}(-z^2 + 2z + 6 + 2z^{-1} - z^{-2}) \end{aligned} \tag{24.50}$$

Using Equation 24.42 with $k = 1$, the corresponding high-pass filters then become

$$\begin{aligned} G_1(z) &= zH_0(-z) = \frac{1}{2}z(-z + 2 - z^{-1}) \\ H_1(z) &= z^{-1}G_0(-z) = \frac{1}{8}z^{-1}(-z^2 - 2z + 6 - 2z^{-1} - z^{-2}) \end{aligned} \tag{24.51}$$

This is often known as the *LeGall 3,5-tap filter set*, since it was first published in the context of 2-band filter banks by Didier LeGall in 1988.

The wavelets of the LeGall 3,5-tap filters, H_0 and H_1 above, and their frequency responses are shown in Figure 24.6. The scaling function (bottom left) converges to a pure triangular pulse and the wavelets are the superposition of two triangular pulses of opposing polarity.

The triangular scaling function produces linear interpolation between consecutive lowband coefficients and also causes the wavelets to be linear interpolations of the coefficients of the H_1 filter, $-1, -2, +6, -2, -1$ (scaled appropriately).

These wavelets have quite desirable properties for signal compression (note the absence of waveform discontinuities and the much lower sidelobes of the frequency responses), and they are one of the simplest useful wavelet types. Unfortunately there is one drawback—the inverse wavelets are not very good. These are formed from the LeGall 5,3-tap filter pair, G_0 and G_1 above, whose wavelets and frequency responses are shown in Figure 24.7

The main problem here is that the wavelets do not converge after many levels to a smooth function and hence the frequency responses have large unwanted sidelobes. For example, the jaggedness of the scaling function and wavelets causes highly visible coding artifacts if these filters are used for reconstruction of a compressed image.

However the allocation of the factors of $P_t(Z)$ to H_0 and G_0 is a free design choice, so we may swap the factors (and hence swap G and H) from the choice made in Equation 24.50 in order that the

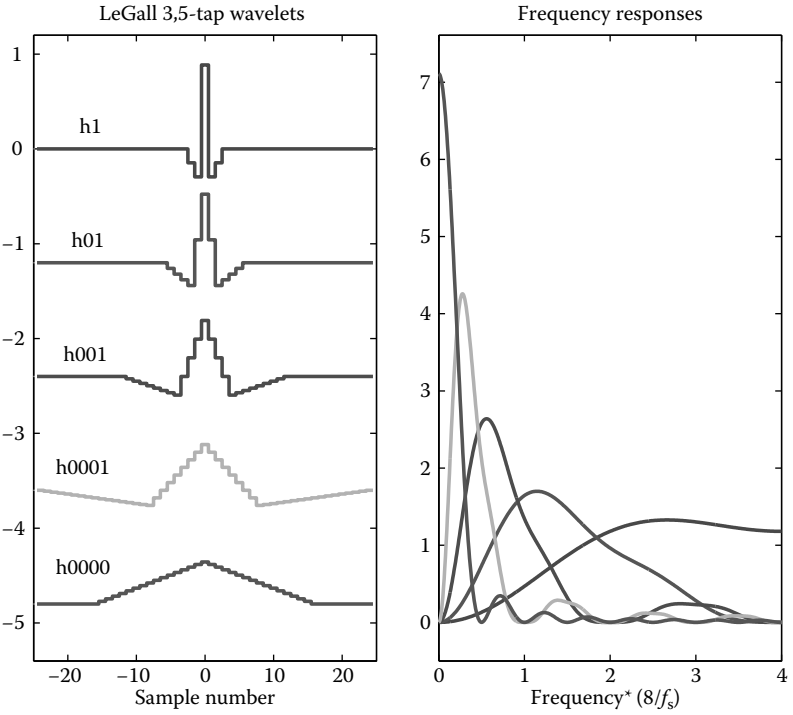


FIGURE 24.6 Impulse responses and frequency responses of the 4-level tree of LeGall 3,5-tap filters.

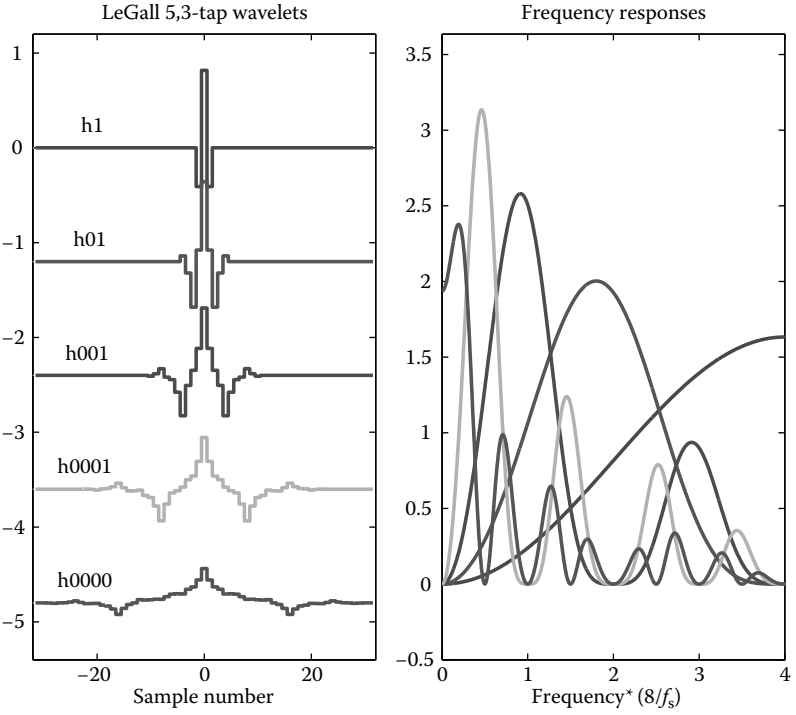


FIGURE 24.7 Impulse responses and frequency responses of the 4-level tree of LeGall 5,3-tap filters.

smoother 3,5-tap filters become G_0, G_1 and are used for reconstruction. It turns out that this leads to a good low-complexity solution for image compression and that the jaggedness of the analysis filters is not critical.

Unbalance between analysis and reconstruction filters/wavelets is nevertheless often regarded as being undesirable, particularly as it prevents the filtering process from being represented as an orthonormal transformation of the input signal and hence from preserving energy through the filter bank. An unbalanced PR filter system is often termed a biorthogonal filter bank.

We now consider ways to reduce this unbalance.

24.6.4 Filters with Balanced *H* and *G* Frequency Responses (but Nonlinear Phase Responses)—Daubechies Wavelets

In the above analysis, we used the factorization of $P_t(Z)$ to give us H_0 and G_0. This always gives unbalanced factors if terms of P_t in even powers of Z are zero, since $P_t(Z)$ must then be of odd-order.

However, each of these factors in Z may itself be factorized into a pair of factors in z, since

$$\begin{aligned}(\alpha z + 1)(1 + \alpha z^{-1}) &= \alpha z + (1 + \alpha^2) + \alpha z^{-1} \\ &= (1 + \alpha^2) + 2\alpha Z \qquad \text{if } Z = \frac{1}{2}(z + z^{-1}) \\ &= (1 + \alpha^2)(1 + \beta Z) \quad \text{where } \beta = \frac{2\alpha}{1 + \alpha^2}\end{aligned} \tag{24.52}$$

For each factor of $P_t(Z)$, we may allocate one of its z subfactors to $H_0(z)$ and the other to $G_0(z)$. Where roots of $P_t(Z)$ are complex, the subfactors must be allocated in conjugate pairs so that H_0 and G_0 remain purely real.

Since the subfactors occur in reciprocal pairs (roots at $z = \alpha$ and α^{-1} for any given β, as shown above), we find that

$$G_0(z) = H_0(z^{-1}) \tag{24.53}$$

This means that the impulse response of G_0 is the time-reverse of that of H_0.

Therefore, using the substitution $z = e^{j\omega T_s}$, their frequency responses are related by

$$G_0(e^{j\omega T_s}) = H_0(e^{-j\omega T_s}) \tag{24.54}$$

Hence, if H_0 and G_0 are real functions of z, the magnitudes of their frequency responses are the same, and their phases are opposite. It may be shown that this is sufficient to obtain orthogonal wavelet basis functions and scaling functions, but unfortunately the separate filters can no longer be zero (or linear) phase, apart from the unique case of the Haar wavelet. When the filters satisfy Equation 24.53, in addition to Equations 24.42 and 24.45, the filter bank is known as a conjugate quadrature filter bank (CQF).

The well-known Daubechies wavelets may be generated in this way, if we include the added constraint that the maximum number of zeros of $P_t(Z)$ are placed at $Z = -1$ (producing pairs of zeros of $P(z)$ at $z = -1$), consistent with terms in even powers of Z in $P_t(Z)$ being zero.

If $P_t(Z)$ is of order $2K - 1$, then it may have K zeros at $Z = -1$ such that

$$P_t(Z) = (1 + Z)^K R_t(Z) \tag{24.55}$$

where $R_t(Z)$ is of order $K - 1$ and its $K - 1$ roots may be chosen such that terms of $P_t(Z)$ in the $K - 1$ even powers of Z are zero.

Equation 24.49 is the $K = 2$ solution to Equation 24.55.

In this case, $R_t(Z) = 1 - \frac{1}{2}Z$, so $\beta = -\frac{1}{2}$ and, from Equation 24.52, the factors of $R(z)$ are

$$R(z) = \frac{(\alpha z + 1)(1 + \alpha z^{-1})}{1 + \alpha^2} \quad \text{where } \alpha = -2 \pm \sqrt{3}$$

since α must be a solution of $1 + \alpha^2 = 2\alpha/\beta$.

Also

$$(1 + Z)^2 = \frac{1}{2}(z + 1)^2 \frac{1}{2}(1 + z^{-1})^2$$

Hence

$$\begin{aligned} H_0(z) &= \frac{1}{2\sqrt{1 + \alpha^2}}(1 + z^{-1})^2\ (1 + \alpha z^{-1}) \\ &= 0.4830 + 0.8365z^{-1} + 0.2241z^{-2} - 0.1294z^{-3} \end{aligned} \tag{24.56}$$

and $G_0(z)$ is the time-reverse of this (replacing z^{-1} by z). Therefore

$$\begin{aligned} H_1(z) &= z^{-3}G_0(-z) = z^{-3}H_0(-z^{-1}) \\ &= 0.1294 + 0.2241z^{-1} - 0.8365z^{-2} + 0.4830z^{-3} \end{aligned} \tag{24.57}$$

The wavelets and frequency responses for these 4-tap filters are shown in Figure 24.8. It is clear that the wavelets and scaling function are no longer linear phase (symmetric about a central point) and

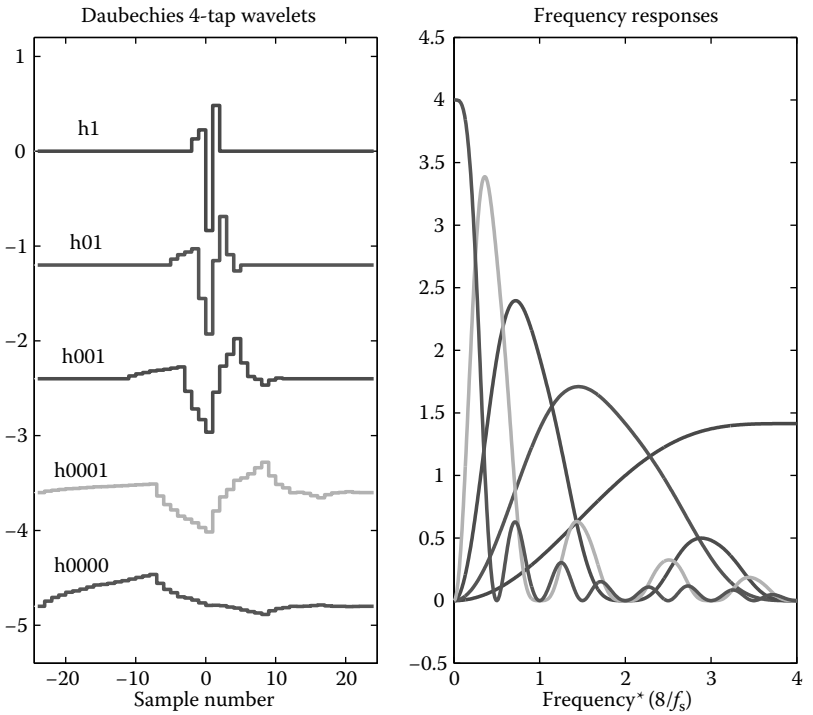


FIGURE 24.8 Impulse responses and frequency responses of the 4-level tree of Daubechies 4-tap filters.

are less smooth than those for the LeGall 3,5-tap filters. The frequency responses also show worse sidelobes. The G_0, G_1 filters give the time-reverse of these wavelets and identical frequency (magnitude) responses.

Higher order Daubechies filters achieve smoother wavelets but they still suffer from nonlinear phase. This tends to result in more visible image coding artefacts than linear phase filters, which distribute any artefacts equally on either side of sharp edges in the image.

When there is more than one root of the residual polynomial $R_t(Z)$, the designer may choose how the factors, $(1 + \alpha z)$ or $(1 + \alpha z^{-1})$, for each root are distributed to $H_0(z)$ and $G_0(z)$. While the distribution may be chosen to give maximum-phase or minimum-phase H_0 and G_0 filters, it is usual to attempt the distribution so as to approximate linear phase more closely. Depending on the choice of product filter order $2K - 1$, different levels of approximation to linear phase become possible.

A fairly good approximation to linear phase, often known as the Daubechies symlet, is obtained for the case when $K = 4$. This gives four zeros of $P_t(Z)$ at $Z = -1$ (and hence four pairs of zeros of $P(z)$ at $z = -1$, which can be shared equally between $H_0(z)$ and $G_0(z)$). The remainder polynomial $R_t(Z)$ is then third order. When converted back to being a function of z, the product filter is given by

$$\begin{aligned} P(z) = H_0(z)G_0(z) &= \frac{z^{-4}}{2048}(z+1)^8 R(z) \\ \text{where } R(z) &= 5z^3 - 40z^2 + 131z - 208 + 131z^{-1} - 40z^{-2} + 5z^{-3} \end{aligned} \tag{24.58}$$

$R(z)$ has real zeros at 0.3289 and 3.0407, and complex zeros at $0.2841 \pm 0.2432j$ and $2.0311 \pm 1.7390j$. To get the closest approximation to linear phase, we group the smaller conjugate pair with the larger real zero, and combine these with four of the zeros at $z = -1$ to obtain $H_0(z)$, while the remaining zeros of $P(z)$ give $G_0(z)$ (or vice versa). H_0 and G_0 are then both seventh-order polynomials (8-tap filters). The impulse and frequency responses of the wavelets and scaling functions from these filters are shown in Figure 24.9, from which we see the somewhat improved symmetry of the waveforms about their highest point (hence the name symlet), as compared with the 4-tap waveforms in Figure 24.8, as well as their greater smoothness and lower sidelobes in the frequency domain.

A useful way to obtain equations of the form of Equation 24.55, for arbitrary odd order $N \geq 2K - 1$ of the transformed product filter $P_t(z)$, is to use the Bernstein polynomial [4,5]:

$$B_N(x;\alpha) = \sum_{i=0}^{(N-1)/2} (1-\alpha_i)\binom{N}{i} x^i (1-x)^{N-i} + \sum_{i=(N+1)/2}^{N} \alpha_{N-i}\binom{N}{i} x^i (1-x)^{N-i} \tag{24.59}$$

where $\alpha = [\alpha_0 \ \alpha_1 \ \ldots \ \alpha_{(N-1)/2}]^{\mathrm{T}}$.

This polynomial has the property that, for any choice of x and α

$$B_N(x;\alpha) + B_N(1-x;\alpha) = 1 \tag{24.60}$$

For PR, from Equations 24.45 and 24.47, it is required that

$$P_t(Z) + P_t(-Z) = 2 \tag{24.61}$$

Hence, if we let $P_t(Z) = 2B_N(x;\alpha)$, where $x = (1 - Z)/2$ so that x goes from 0 to 1 as Z goes from $+1$ to -1, then $P_t(-Z) = 2B_N(1 - x;\alpha)$ and, from Equation 24.60, $P_t(Z)$ will satisfy the above PR condition (Equation 24.61) for any choice of α.

To obtain K zeros of $P_t(Z)$ at $Z = -1$, we require $B_N(x;\alpha)$ to have K zeros at $x = 1$, which may be easily obtained by setting $\alpha_i = 0$ for $i = 0 \ldots K - 1$, as long as $N \geq 2K - 1$.

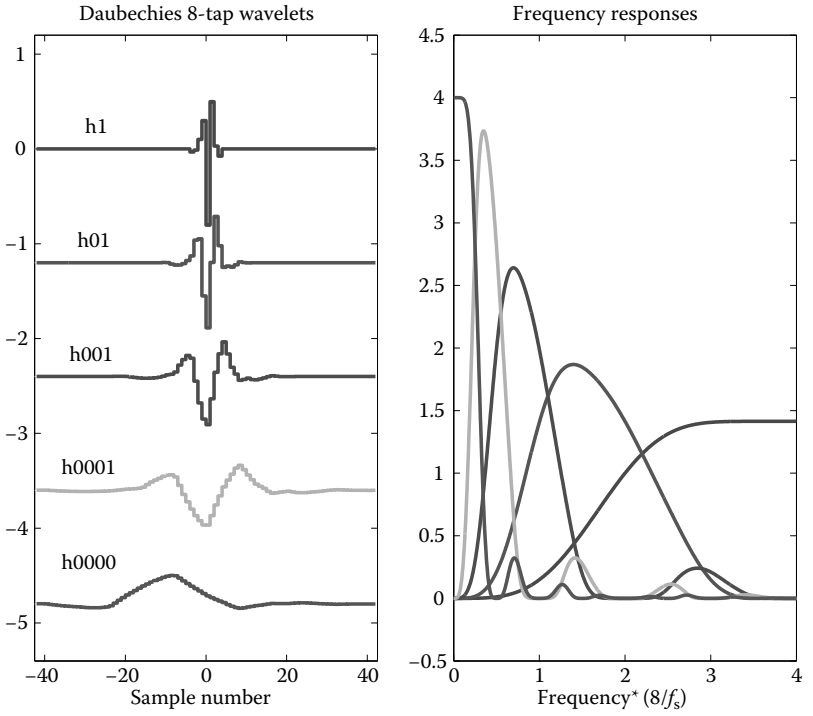


FIGURE 24.9 Impulse responses and frequency responses of the 4-level tree of Daubechies 8-tap symlet filters.

When $N = 2K - 1$, we have the maximum number of zeros at $Z = -1$ for a given N, as assumed above for the cases $K = 2$ and $K = 4$. Then are then no further degrees of freedom available to the designer. We shall see below that, in some situations, at least one additional degree of freedom can be very helpful, and this is achieved when $N = 2K + 1$ and the free parameter is α_K. More degrees of freedom may be achieved by increasing N in steps of two while keeping K constant.

24.6.5 Filters with Linear Phase and Nearly Balanced Frequency Responses

We have seen how to create filters with balanced frequency responses and an approximation to linear phase. Now we consider filters with exact linear phase and approximately balanced frequency responses.

Linear phase filters allow an elegant technique, known as symmetric extension, to be used at the outer edges of finite datasets such as images, where wavelet filters would otherwise require the size of the transformed data to be increased to allow for convolution with the filters. Symmetric extension assumes that the data is reflected by mirrors at each endpoint or edge. For images in 2-D, an infinitely tesselated plane of reflected images is generated. Reflections avoid unwanted edge discontinuities. If the filters are linear phase, then the resulting DWT coefficients also form reflections at the endpoints and no increase in size of the transformed data is necessary to accommodate convolution effects.

To ensure that the filters H_0, H_1 and G_0, G_1 are linear phase, the factors in Z must be allocated to H_0 or G_0 as a whole and not be split, as was done for the Daubechies filters. In this way, the symmetry between z and z^{-1} is preserved in all filters. Perfect balance of frequency responses between H_0 and G_0 is then not possible, if PR is preserved.

A popular linear-phase option, which works very well for image coding and is part of the JPEG 2000 standard, is a linear phase factorization of the same $P(z)$ product filter, used for the 8-tap Daubechies symlet and given in Equation 24.58. This factorization still distributes the eight zeros at $z = -1$ equally

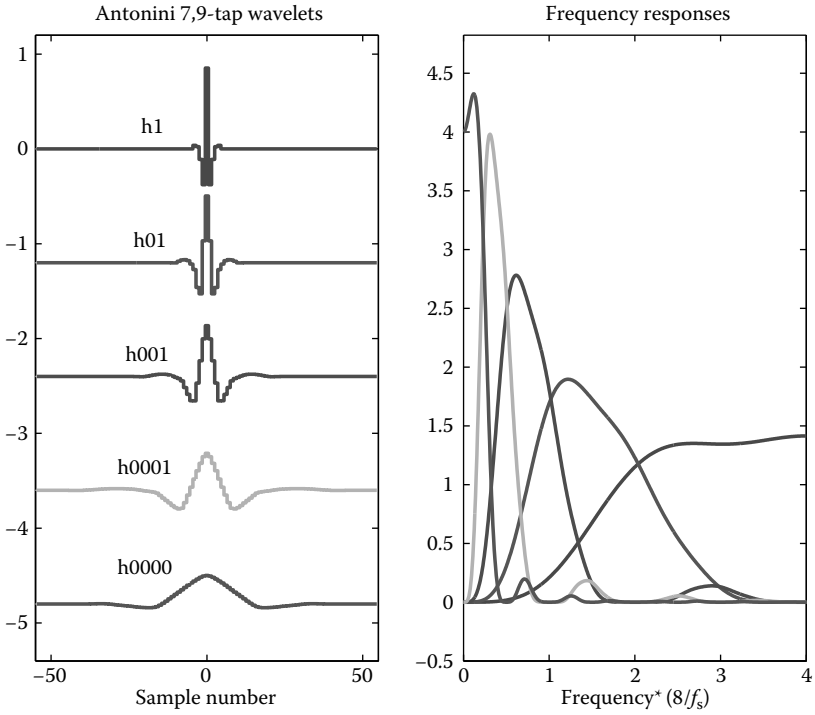


FIGURE 24.10 Impulse responses and frequency responses of the 4-level tree of Daubechies 7,9-tap filters.

between H_0 and G_0; but, to get linear phase and keep the filter coefficients purely real, it is necessary to allocate the two real zeros of $R(z)$ to one filter (say G_0) and the four complex zeros of $R(z)$ to the other filter (H_0). Linear phase is obtained because the zeros are in reciprocal pairs, and real coefficients are obtained because the complex zeros are also in conjugate pairs. The resulting 7-tap and 9-tap filters are known as the Daubechies biorthogonal 7,9-tap wavelet filters (and also known as Antonini filters) [6]. These 7,9-tap and 9,7-tap wavelets and scaling functions are shown in Figures 24.10 and 24.11.

The 7-tap filter from this pair tends to be somewhat smoother than the 9-tap one, and so is normally chosen for the reconstruction low-pass filter G_0 (and hence also generates the analysis high-pass filter H_1, Equation 24.42). In compression applications, where many higher frequency wavelet coefficients are set to zero, this tends to result in a smoother reconstructed signal. It may be seen that the smoother waveforms, with lower sidelobes in the frequency domain, occur in Figure 24.10 while the less smooth option is in Figure 24.11.

We have found a factorization of $P_t(Z)$ which achieves a much closer balance of the responses, by reducing K the number of zeros at $Z = -1$. This is

$$P_t(Z) = (1 + Z)(1 + aZ + bZ^2)(1 + Z)(1 + cZ) \tag{24.62}$$

This is a fifth order polynomial, and if the terms in Z^2 and Z^4 are to be zero, there are two constraints on the three unknowns [*a, b, c*] so that one of them (say *c*) may be regarded as a free parameter. These constraints require that

$$a = -\frac{(1 + 2c)^2}{2(1 + c)^2} \quad \text{and} \quad b = \frac{c(1 + 2c)}{2(1 + c)^2} \tag{24.63}$$

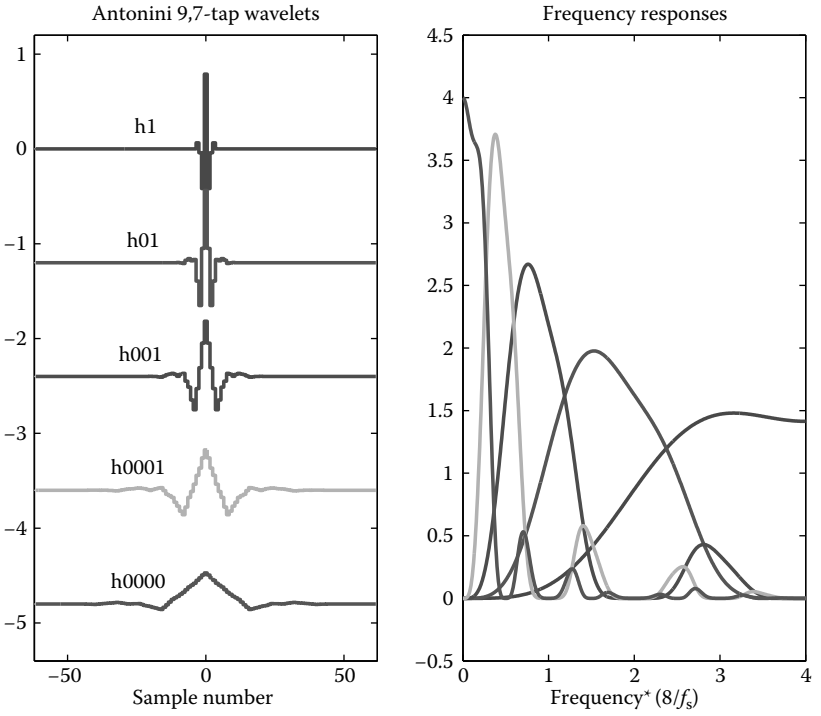


FIGURE 24.11 Impulse responses and frequency responses of the 4-level tree of Daubechies 9,7-tap filters.

c may then be adjusted to give maximum similarity between the left and right pairs of factors in Equation 24.62 as Z varies from 1 to -1 (i.e., as ωT_s varies from 0 to π).

It turns out that $c = -\frac{2}{7}$ gives good similarity and when substituted into Equations 24.62 and 24.63 gives

$$\begin{aligned} P_t(Z) &= \frac{1}{50}(1+Z)(50-9Z-6Z^2)\frac{1}{7}(1+Z)(7-2Z) \\ &= \frac{1}{50}(50+41Z-15Z^2-6Z^3)\frac{1}{7}(7+5Z-2Z^2) \end{aligned} \tag{24.64}$$

We get $G_0(z)$ and $H_0(z)$ by substituting $Z = \frac{1}{2}(z+z^{-1})$ into these two polynomial factors. This results in 5,7-tap filters whose wavelets and frequency responses are shown in Figure 24.12.

The near balance of the analysis and reconstruction responses may be seen from Figure 24.13 which shows the alternative 7,5-tap versions (i.e., with H and G swapped). It is quite difficult to spot the minor differences between these figures and it makes little difference which way round the filters are allocated.

These filters may also be obtained using the Bernstein polynomial method of equation (Equation 24.59), using $N = 5$, $K = 2$, and α_K as the free parameter. The value which corresponds to $c = -\frac{2}{7}$ is $\alpha_2 = \frac{477}{1750} = 0.2726$.

24.6.6 Smoother Wavelets

In all of the above designs we have used the substitution $Z = \frac{1}{2}(z+z^{-1})$. However other substitutions may be used to create improved wavelets. To preserve PR, the substitution should contain only odd powers of z (so that odd powers of Z produce only odd powers of z); and to produce zero phase, the coefficients of the substitution should be symmetric about z^0.

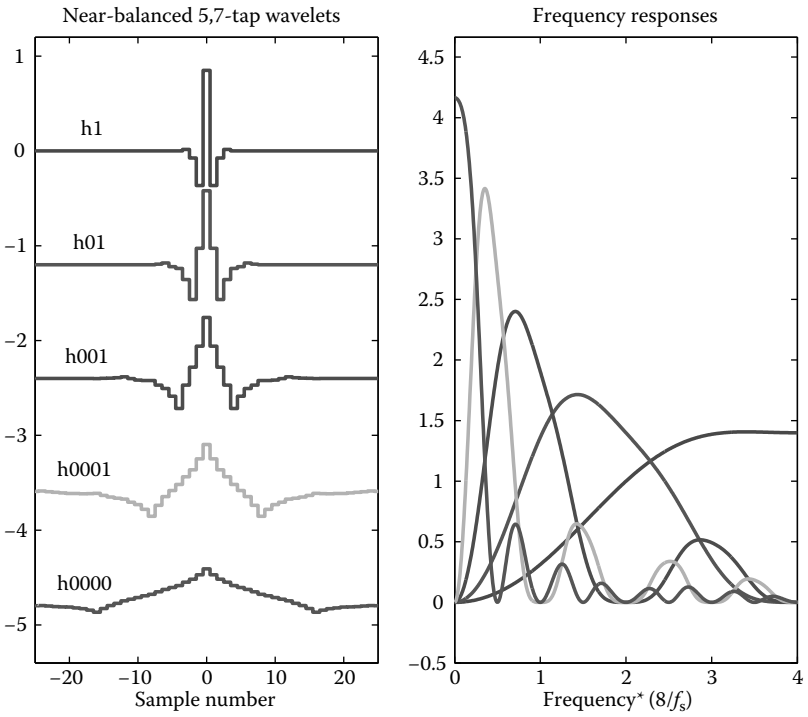


FIGURE 24.12 Impulse responses and frequency responses of the 4-level tree of near-balanced 5,7-tap filters.

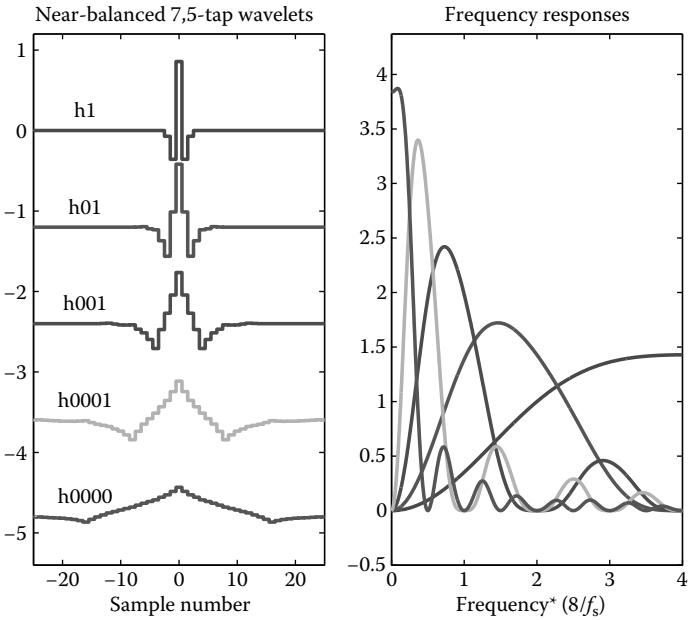


FIGURE 24.13 Impulse responses and frequency responses of the 4-level tree of near-balanced 7,5-tap filters.

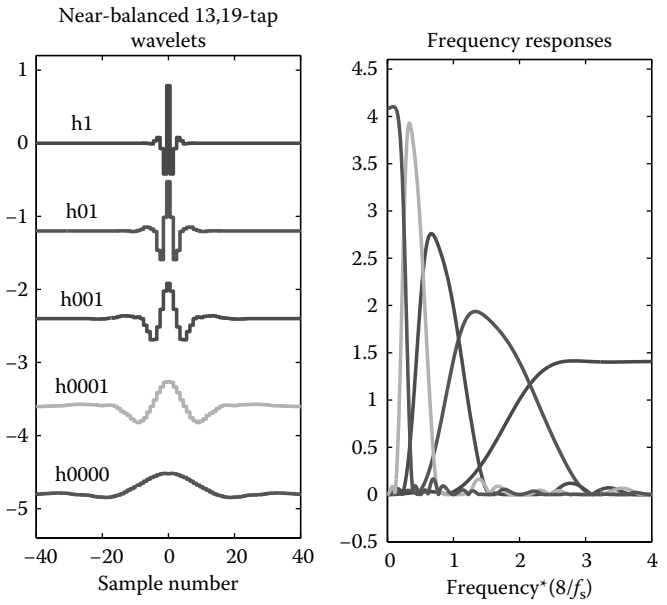


FIGURE 24.14 Impulse responses and frequency responses of the 4-level tree of near-balanced 13,19-tap filters.

A substitution, which can give much greater flatness near $z = \pm 1$ while still satisfying $Z = \pm 1$ when $z = \pm 1$, is

$$Z = pz^3 + (\tfrac{1}{2} - p)(z + z^{-1}) + pz^{-3} \tag{24.65}$$

Z then becomes the following function of frequency when $z = e^{j\omega T_s}$:

$$Z = (1 - 2p)\cos(\omega T_s) + 2p\cos(3\omega T_s) \tag{24.66}$$

Maximal flatness is achieved near $\omega T_s = 0$ and π, when $p = -\frac{1}{16}$. This is equivalent to more zeros at $z = -1$ for each $(Z + 1)$ factor than when $Z = \frac{1}{2}(z + z^{-1})$ is used. A high degree of flatness (with some ripple) and sharper transition bands is obtained if $p = -\frac{3}{32}$ instead.

The second order factor of $P_t(Z)$ in Equation 24.64 now produces terms from z^6 to z^{-6} and the third-order factor produces terms from z^9 to z^{-9}. Hence the filters become 13- and 19-tap filters, although two taps of each are zero and the outer two taps of the 19-tap filter are very small ($\sim 10^{-4}$).

Figure 24.14 shows the wavelets and frequency responses of the 13,19-tap filters, obtained by substituting Equation 24.65 into Equation 24.64 with $P = -3/32$. Note the smoother wavelets and scaling function and the much lower sidelobes in the frequency responses from these higher order filters.

Figure 24.15 demonstrates that the near-balanced properties of Equation 24.64 are preserved in the higher order filters.

There are many other types of wavelets with varying features and complexities, but we have found the examples given to be near optimum for applications such as image compression.

24.7 IIR Filter Banks

In general, for a given frequency response specification (e.g., transition bandwidth, stopband ripple magnitude), an IIR filter can achieve that specification with a lower computational complexity (e.g., number of

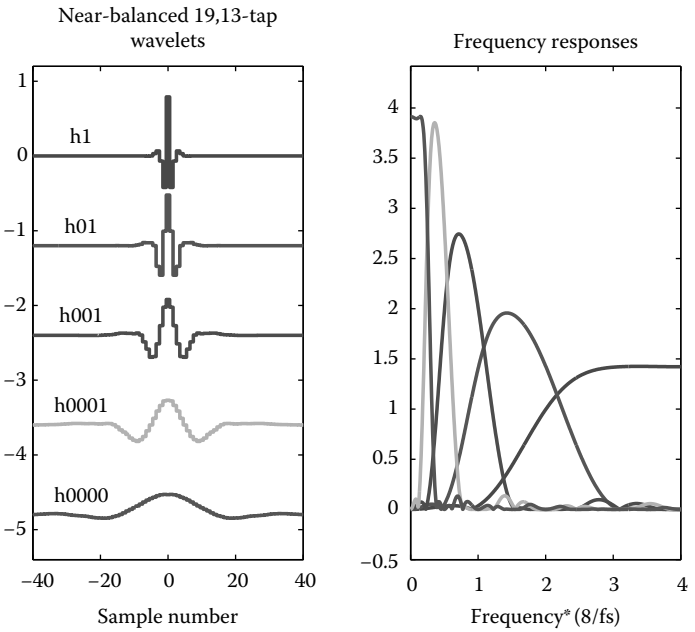


FIGURE 24.15 Impulse responses and frequency responses of the 4-level tree of near-balanced 19,13-tap filters.

multiplies per output sample) compared to an FIR filter. This is the potential advantage of IIR filter banks but the disadvantage is that their design and their implementation are usually more complicated. The main issue that needs to be considered when using IIR filters is stability. The well-known criterion for stability is that the poles of the IIR transfer function must lie inside the unit circle. This criterion, however, assumes that the IIR filter is implemented using a difference equation in the forward time direction, i.e., causal filtering. In some situations, in which the data is fully stored in memory and may therefore be processed backward as well as forward, anticausal filtering is feasible too. With anticausal filtering the poles need to be outside the unit circle for stability. There are therefore two types of IIR filter banks:

1. Causal stable IIR filter banks: used in situations where only causal filtering is allowed. Examples are real-time applications such as speech processing and echo-cancellation. All filters must then have poles inside the unit circle.
2. Noncausal IIR filter banks: used in situations which can allow off-line processing or when the signal is finite in duration (e.g., still images). Suppose an IIR filter $F(z)$ has poles both inside and outside the unit circle. The filter may be factored into a product of its minimum and maximum-phase components:

$$F(z) = F_{\min}(z)F_{\max}(z) \tag{24.67}$$

where the poles of $F_{\min}(z)$ are inside the unit circle, and those of $F_{\max}(z)$ are outside it. The filtering consists of two stages, in which causal filtering is first performed forward to implement $F_{\min}(z)$ and anticausal filtering is then performed backward to implement $F_{\max}(z)$.

Just as in FIR filter banks, the scaling functions and wavelets corresponding to IIR filter banks can be defined through the two-scale equation and infinite product formula. The main difference in the IIR case is that the corresponding continuous functions have support width that is infinite, i.e., the function

duration is infinite. For the causal stable case the function extends to positive infinity and for the noncausal case the function extends to infinity in both directions.

A versatile building block for IIR filters is the all-pass filter $a(z)$ which has the following two properties:

$$\frac{1}{a(z)} = a(z^{-1}) \quad \text{and} \quad |a(e^{j\omega})| = 1 \tag{24.68}$$

The all-pass filter $a(z)$ can be considered a generalization of the unit delay z^{-1} since $|e^{-j\omega}| = |a(e^{j\omega})| = 1$, i.e., both have the same magnitude response. The unit delay has a linear phase response but the all-pass filter can have a variety of phase responses. A wide family of IIR filters can be obtained by various interconnections of all-pass filters [7–9]. The simplest interconnection (described below) can give a large class of useful IIR filter banks.

24.7.1 All-Pass Filter Design Methods

Consider analysis filters obtained by the weighted sum and difference of two all-pass filters, a_0 and a_1:

$$\begin{aligned} H_0(z) &= \frac{1}{\sqrt{2}}[a_0(z) + a_1(z)] \\ H_1(z) &= \frac{1}{\sqrt{2}}[a_0(z) - a_1(z)] \end{aligned} \tag{24.69}$$

This can be considered an extension of the FIR Haar filter pair in Equation 24.2 to IIR filters. The terms 1 and z^{-1} (simple all-pass) have been replaced with a_0 and a_1, general all-pass filters. The Haar filters are quadrature mirror versions of each other. If the two all-pass filters are chosen to be of the form $a_0(z) = A_0(z^2), a_1(z) = z^{-1}A_1(z^2)$, then the analysis filters, given by

$$\begin{aligned} H_0(z) &= \frac{1}{\sqrt{2}}\left[A_0(z^2) + z^{-1}A_1(z^2)\right] \\ H_1(z) &= \frac{1}{\sqrt{2}}\left[A_0(z^2) - z^{-1}A_1(z^2)\right] \end{aligned} \tag{24.70}$$

will have the quadrature mirror characteristics:

$$|H_1(e^{j\omega})| = |H_0(e^{j(\pi-\omega)})|, \quad \text{i.e., be mirrored about } \omega = \frac{\pi}{2} \tag{24.71}$$

This means that if $H_0(e^{j\omega})$ has a low-pass frequency response, $H_1(e^{j\omega})$ will automatically have a high-pass response.

How should the all-pass filters be designed so that $H_0(e^{j\omega})$ is a good low-pass filter with half of the full bandwidth?

By denoting the phase response of the all-pass filters as $\phi_0(\omega) = \angle A_0(e^{j\omega})$ and $\phi_1(\omega) = \angle A_1(e^{j\omega})$, the magnitude response of the low-pass filter can be written as

$$\begin{aligned} |H_0(e^{j\omega})| &= \frac{1}{\sqrt{2}}\left|e^{j\phi_0(2\omega)} + e^{j(\phi_1(2\omega)-\omega)}\right| \\ &= \frac{1}{\sqrt{2}}\left|e^{j\phi_0(2\omega)}\right|\left|1 + e^{j(\phi_1(2\omega)-\phi_0(2\omega)-\omega)}\right| \\ &= \frac{1}{\sqrt{2}}\left|1 + e^{j(\theta(2\omega)-\omega)}\right| \quad \text{where} \quad \theta(\omega) \equiv \phi_1(\omega) - \phi_0(\omega) \end{aligned} \tag{24.72}$$

The term $e^{j(\theta(2\omega)-\omega)}$ can be considered a frequency dependent phasor that is added to the constant phasor 1. If the two phasors add constructively (i.e., if $e^{j(\theta(2\omega)-\omega)} \approx 1$), then $|H_0(e^{j\omega})| \approx \sqrt{2}$; and if the two phasors add destructively (i.e., if $e^{j(\theta(2\omega)-\omega)} \approx -1$), then $|H_0(e^{j\omega})| \approx 0$. Therefore the ideal criterion for the phase is

$$\theta(2\omega) - \omega = \begin{cases} 0, & 0 < \omega < \pi/2 \\ \pm\pi, & \pi/2 < \omega < \pi \end{cases} \tag{24.73}$$

What about the synthesis filters?

For the Haar filters, the synthesis filters were obtained from the analysis filters by replacing z^{-1} with z as shown in Equation 24.6, i.e., a reciprocal term is used. The situation now with all-pass filters (instead of delays) is, however, more complicated and there are two cases to consider:

1. No reciprocals are used and the synthesis filters are chosen as

$$\begin{aligned} G_0(z) &= \frac{1}{\sqrt{2}}\left[A_0(z^2) + z^{-1}A_1(z^2)\right] \\ G_1(z) &= -\frac{1}{\sqrt{2}}\left[A_0(z^2) - z^{-1}A_1(z^2)\right] \end{aligned} \tag{24.74}$$

   and are essentially the same as the analysis filters (apart from a sign change for G_1). Aliasing cancellation is achieved (because $G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0$), but not PR. The transfer function between the input and output of the entire filter bank, i.e., the reconstruction function, is given from Equation 24.11 by

$$T(z) \equiv \frac{\hat{X}(z)}{X(z)} = \frac{1}{2}[G_0(z)H_0(z) + G_1(z)H_1(z)] = z^{-1}A_0(z^2)A_1(z^2) \tag{24.75}$$

   which is an all-pass function. If phase-distortion can be tolerated, e.g., for speech, then this may be acceptable as it stands. Alternatively some postfiltering to equalize the phase of $T(z)$ may be needed.

   If the poles of $A_0(z)$ and $A_1(z)$ are inside the unit circle then the whole filter bank is causal stable.

2. If the synthesis filters are instead chosen as

$$\begin{aligned} G_0(z) &= \frac{1}{\sqrt{2}}\left[\frac{1}{A_0(z^2)} + z\frac{1}{A_1(z^2)}\right] = \frac{1}{\sqrt{2}}\left[A_0(z^{-2}) + zA_1(z^{-2})\right] \\ G_1(z) &= \frac{1}{\sqrt{2}}\left[\frac{1}{A_0(z^2)} - z\frac{1}{A_1(z^2)}\right] = \frac{1}{\sqrt{2}}\left[A_0(z^{-2}) - zA_1(z^{-2})\right] \end{aligned} \tag{24.76}$$

   then PR is achieved, i.e., $T(z) = 1$. However, the filter bank then becomes noncausal as the inversion of the all-pass filters causes poles that were originally inside the unit circle now to be outside the unit circle.

24.7.2 Transformation-Based Design Methods

The frequency transformation technique, described in Section 24.6 for FIR filter banks, can also be used to construct IIR filter banks [10].

Recall that in the transformation technique, given the prototype linear phase low-pass filters, $H_t(Z)$ and $F_t(Z)$, the following transformation is applied:

$$Z = M(z) \tag{24.77}$$

where $M(z) = pz^3 + (\frac{1}{2} - p)(z + z^{-1}) + pz^{-3}$ (from Equation 24.65) and $p = 0$ or $-\frac{3}{32}$ depending on the desired complexity/smoothness required. It is necessary that the transformation is of the form $M(z) \equiv zT(z^2)$ (i.e., it must have terms in only odd powers of z).

Hence the transformation function $M(z)$ for the FIR case is an FIR filter that is related to a halfband filter,* $H_{\text{HB}}(z) \equiv \frac{1}{2}[1 + M(z)]$. IIR filter banks can be obtained by using an IIR transformation function instead of an FIR function and this is equivalent to using an IIR halfband filter. The transformation function should ideally approximate $+1$ in the passband and -1 in the stopband, i.e.,

$$M(e^{j\omega}) = e^{j\omega}T(e^{j2\omega}) \approx \begin{cases} +1, & 0 < \omega < \pi/2 \\ -1, & \pi/2 < \omega < \pi \end{cases} \tag{24.78}$$

Therefore a straightforward choice is for $M(z)$ to be an all-pass filter approximating the following ideal phase condition:

$$\phi_{\text{ideal}}(\omega) = \begin{cases} 0, & 0 < \omega < \pi/2 \\ \pm\pi, & \pi/2 < \omega < \pi \end{cases} \tag{24.79}$$

If the poles of $M(z)$ are inside the unit circle then the IIR filter bank is causal stable. Furthermore, PR is achieved.

A simple example for $M(z)$ is given by

$$M(z) = z\frac{k_1 + z^{-2}}{1 + k_1 z^{-2}}\frac{k_2 + z^{-2}}{1 + k_2 z^{-2}} \tag{24.80}$$

with $k_1 = -0.0991$ and $k_2 = 0.5426$ and the phase response is shown in Figure 24.16.

In Equation 24.49, we saw that a simple $H_t(Z)$ and $F_t(Z)$ are given by

$$\begin{aligned} H_t(Z) &= (1 + Z) \\ F_t(Z) &= \frac{1}{2}(1 + Z)(2 - Z) \end{aligned} \tag{24.81}$$

Using an all-pass filter for $Z = M(z)$ in Equation 24.81 will however result in a 4 dB overshoot (bump) in the frequency response of the filter $F_0(z) = F_t(M(z))$. With a more general filter for $M(z)$ (non all-pass) the overshoot can be reduced but the design becomes more complicated.

Alternatively, a different set of prototype filters may lead to a lower overshoot. The following set of prototype filters from the product filter in Equation 24.64:

$$\begin{aligned} H_t(Z) &= \frac{1}{7}(1 + Z)(7 - 2Z) \\ F_t(Z) &= \frac{1}{50}(1 + Z)(50 - 9Z - 6Z^2) \end{aligned} \tag{24.82}$$

* Halfband filters are defined as filters that satisfy $H_{\text{HB}}(z) + H_{\text{HB}}(-z) = 1$. The even index coefficients are zero except for the center (zero index) coefficient, which is one half.

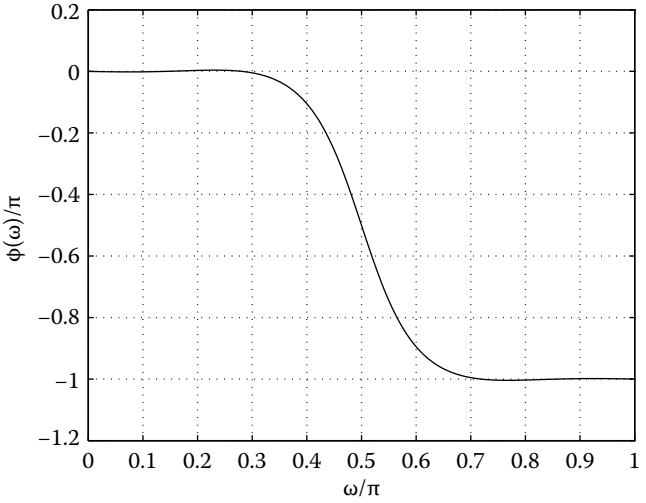


FIGURE 24.16 Phase response of IIR all-pass transformation in Equation 24.80.

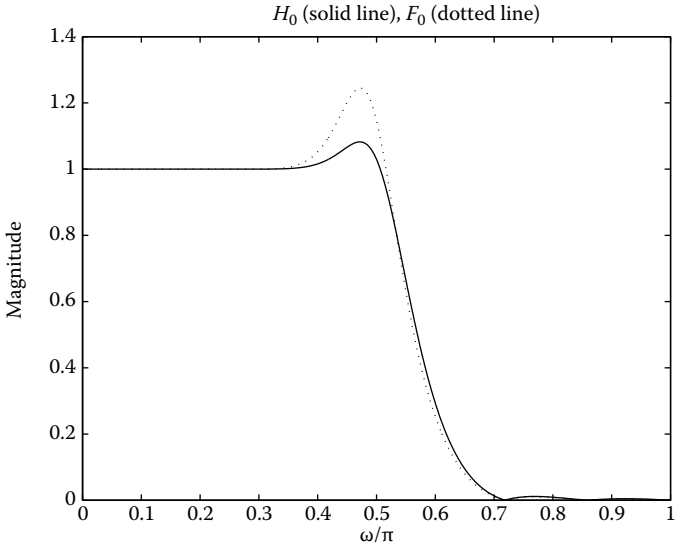


FIGURE 24.17 Magnitude responses of causal stable IIR filter bank given by Equation 24.82 with the transformation of Equation 24.80.

with an all-pass $M(z)$ will lead to IIR filters with overshoots of 0.68 dB and 1.90 dB, respectively. Using these prototypes with the transformation in Equation 24.80 gives low-pass filters with magnitude responses shown in Figure 24.17.

24.8 Polyphase Representations and the Lifting Scheme

The polyphase representation is an important tool in multirate systems, from both theoretical and application perspectives. Multirate systems are (periodically) linear time-varying systems due to the

presence of downsamplers and upsamplers. The polyphase decomposition allows a multirate system to be represented as a sum of linear time-invariant systems, thus simplifying its analysis and design [9,11,12]. The form of the representation depends on the downsampling/upsampling factor M, but the discussion here will focus on the simplest case of $M = 2$ which is relevant for 2-band filter-banks. Unless otherwise stated, all filters are assumed to be FIR in this section.

24.8.1 Basic Polyphase Concepts

The basic idea of an $M = 2$ polyphase representation is to separate the coefficients of a filter or a sequence into even-indexed and odd-indexed coefficients:

$$H(z) = \sum_n h(2n)z^{-2n} + \sum_n h(2n+1)z^{-2n-1} \tag{24.83}$$

By defining the polyphase components as

$$E_0(z) \equiv \sum_n h(2n)z^{-n} \quad \text{and} \quad E_1(z) \equiv \sum_n h(2n+1)z^{-n} \tag{24.84}$$

the filter function $H(z)$ can be written as

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2) \tag{24.85}$$

The separation of $H(z)$ into $E_0(z)$ and $E_1(z)$ is known as the polyphase decomposition of the filter. The decomposition in Equation 24.85 is known as a type 1 decomposition. An alternative is the type 2 decomposition defined as

$$H(z) = z^{-1}R_0(z^2) + R_1(z^2) = z^{-1}\left[R_0(z^2) + zR_1(z^2)\right] \tag{24.86}$$

The polyphase components of both types are related as follows:

$$R_0(z) = E_1(z) \quad R_1(z) = E_0(z)$$

i.e., type 2 is a permutation of type 1.

By applying the type 1 polyphase decomposition to the 2-band analysis filter bank of Figure 24.1a, the following is obtained:

$$\begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = \begin{bmatrix} E_{0,0}(z^2) & E_{0,1}(z^2) \\ E_{1,0}(z^2) & E_{1,1}(z^2) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \tag{24.87}$$

The matrix

$$\mathbf{E}(z) \equiv \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) \\ E_{1,0}(z) & E_{1,1}(z) \end{bmatrix}$$

is known as the analysis polyphase matrix of the filter bank. The first row contains the polyphase components of the low-pass analysis filter H_0, and the second row those of the high-pass filter H_1.

For the synthesis filters, the type 2 polyphase decomposition is applied:

$$[G_0(z)\ G_1(z)] = [z^{-1}\ 1]\begin{bmatrix} R_{0,0}(z^2) & R_{1,0}(z^2) \\ R_{0,1}(z^2) & R_{1,1}(z^2) \end{bmatrix} \tag{24.88}$$

and the synthesis polyphase matrix is defined as

$$\mathbf{R}(z) \equiv \begin{bmatrix} R_{0,0}(z) & R_{1,0}(z) \\ R_{0,1}(z) & R_{1,1}(z) \end{bmatrix}$$

The first column contains the polyphase components of the low-pass synthesis filter G_0, and the second column those of the high-pass filter G_1.

Now suppose the input signal is decomposed as

$$X(z) = X_0(z^2) + zX_1(z^2)$$

which is similar to the type 2 polyphase decomposition except for absence of the delay z^{-1} (compare with Equation 24.86). The output of the analysis filter bank can then be written compactly as

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) \\ E_{1,0}(z) & E_{1,1}(z) \end{bmatrix}\begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} = \mathbf{E}(z)\begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix} \tag{24.89}$$

By using a type 2 polyphase decomposition of the output signal

$$\hat{X}(z) = z^{-1}\hat{X}_0(z^2) + \hat{X}_1(z^2)$$

the output of the synthesis filter bank can be written compactly as

$$\begin{bmatrix} \hat{X}_0(z) \\ \hat{X}_1(z) \end{bmatrix} = \begin{bmatrix} R_{0,0}(z) & R_{1,0}(z) \\ R_{0,1}(z) & R_{1,1}(z) \end{bmatrix}\begin{bmatrix} Y_0(z) \\ Y_1(z) \end{bmatrix} = \mathbf{R}(z)\begin{bmatrix} Y_0(z) \\ Y_1(z) \end{bmatrix} \tag{24.90}$$

Equations 24.89 and 24.90 are the equivalent time-invariant representation of the multirate filter bank. Figure 24.18 shows the equivalent representation of the filter bank in polyphase form. The combination of the delay chain and downsampler on the analysis side provides a polyphase decomposition of the input signal. The combination of the upsamplers, delay chain, and adder on the synthesis side provides a reconstruction of the output signal from the polyphase components. The polyphase filter bank is operating at the lower sampling rate of the system and is thus more efficient than the original filter bank.

Combining Equations 24.89 and 24.90 gives the polyphase form of the input/output relationship of the entire filter bank:

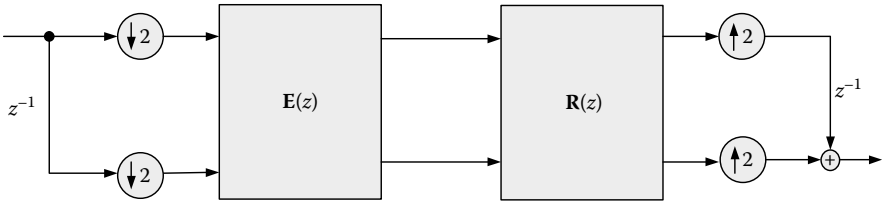


FIGURE 24.18 Polyphase form of filter bank.

$$\begin{bmatrix} \hat{X}_0(z) \\ \hat{X}_1(z) \end{bmatrix} = \mathbf{R}(z)\mathbf{E}(z) \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}$$

The total system polyphase matrix is defined as

$$\mathbf{P}(z) \equiv \mathbf{R}(z)\mathbf{E}(z)$$

and if

$$\mathbf{P}(z) = \mathbf{R}(z)\mathbf{E}(z) = \mathbf{I} \tag{24.91}$$

then

$$\begin{bmatrix} \hat{X}_0(z) \\ \hat{X}_1(z) \end{bmatrix} = \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}$$

Therefore

$$\hat{X}(z) = z^{-1}X_0(z^2) + X_1(z^2) = z^{-1}\left[X_0(z^2) + zX_1(z^2)\right] = z^{-1}X(z)$$

and PR (with one sample delay) is achieved. This is a convenient alternative formulation for PR to that given in Equations 24.40 and 24.41.

Equation 24.91 provides a concise condition for PR and the result is useful in the theoretical study and formulation of design techniques for filter banks. Conceptually, if synthesis filter bank is chosen such that

$$\mathbf{R}(z) = \mathbf{E}^{-1}(z) = \frac{\text{adj}(\mathbf{E}(z))}{\det(\mathbf{E}(z))} \tag{24.92}$$

where adj(·) and det(·) denote the adjoint and determinant of the matrix, respectively, then PR is assured. However, in general, this would lead to IIR synthesis filters which may be unstable (poles outside the unit circle) due to the presence of det (**E**(*z*)) in the denominator. Therefore to ensure that all filters are FIR in the filter bank, it is required that

$$\det(\mathbf{E}(z)) = cz^K \tag{24.93}$$

where c is a nonzero constant and K an integer.

An important property of determinants is that

$$\text{if } \mathbf{A} = \mathbf{A}_1\mathbf{A}_2 \quad \text{then } \det(\mathbf{A}) = \det(\mathbf{A}_1)\det(\mathbf{A}_2)$$

Therefore, a straightforward way to build polyphase matrices satisfying Equation 24.93 is through a cascade (product) of simpler matrices that satisfy Equation 24.93, i.e.,

$$\mathbf{E}(z) = \mathbf{E}_1(z)\mathbf{E}_2(z)\cdots\mathbf{E}_L(z) \tag{24.94}$$

where

$$\det(\mathbf{E}_i(z)) = c_i z^{K_i} \quad i = 1, \ldots, L$$

The cascade of polyphase matrices is the essence of virtually all design techniques of filter banks that are polyphase based.

24.8.2 Orthogonal Lattice Structures

This type of structure is also known as the lossless or paraunitary structure [13,14]. The polyphase matrix is given by

$$\mathbf{E}(z) = \mathbf{R}_L \mathbf{D}(z) \mathbf{R}_{L-1} \cdots \mathbf{D}(z) \mathbf{R}_0 = \mathbf{R}_L \prod_{i=0}^{L-1} [\mathbf{D}(z) \mathbf{R}_i] \tag{24.95}$$

where

$$\mathbf{D}(z) \equiv \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \quad \text{and} \quad \mathbf{R}_i \equiv \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$$

and $\theta_0 \ldots \theta_{L-1}$ are the L free parameters that define the filters. In the design process, these parameters are optimized with respect to some criterion, e.g., minimizing the stopband energy of the low-pass filter, which is given by

$$J = \int_{\omega_s}^{\pi} |H_0(e^{j\omega})|^2 \mathrm{d}\omega$$

where ω_s is the stopband edge. The matrices $\mathbf{R}_i$ are also known as rotation matrices and are orthogonal, i.e., $\mathbf{R}_i \mathbf{R}_i^{\mathrm{T}} = \mathbf{R}_i^{\mathrm{T}} \mathbf{R}_i = \mathbf{I}$ (where T denotes transpose).

Since $\mathbf{D}(z)\mathbf{D}^{\mathrm{T}}(z^{-1}) = \mathbf{D}^{\mathrm{T}}(z^{-1})\mathbf{D}(z) = \mathbf{I}$, the polyphase matrix satisfies

$$\mathbf{E}(z)\mathbf{E}^{\mathrm{T}}(z^{-1}) = \mathbf{E}^{\mathrm{T}}(z^{-1})\mathbf{E}(z) = \mathbf{I}$$

Therefore the synthesis polyphase matrix can be obtained as

$$\mathbf{R}(z) = \mathbf{E}^{\mathrm{T}}(z^{-1}) = \mathbf{R}_0^{\mathrm{T}} \mathbf{D}(z^{-1}) \cdots \mathbf{R}_{L-1}^{\mathrm{T}} \cdots \mathbf{D}(z^{-1}) \mathbf{R}_L^{\mathrm{T}}$$

The structure for such filter banks is shown in Figure 24.19

24.8.3 Linear Phase Structures

There are two types of structure that lead to linear phase filters [15,16]:

1. Type A: where all filters in the bank have even length. The low-pass analysis filter coefficients are symmetric, while the synthesis coefficients are antisymmetric.
   The polyphase matrix is given by

$$\mathbf{E}(z) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{D}(z) \mathbf{S}_{L-1} \cdots \mathbf{D}(z) \mathbf{S}_0 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \prod_{i=0}^{L-1} [\mathbf{D}(z) \mathbf{S}_i] \tag{24.96}$$

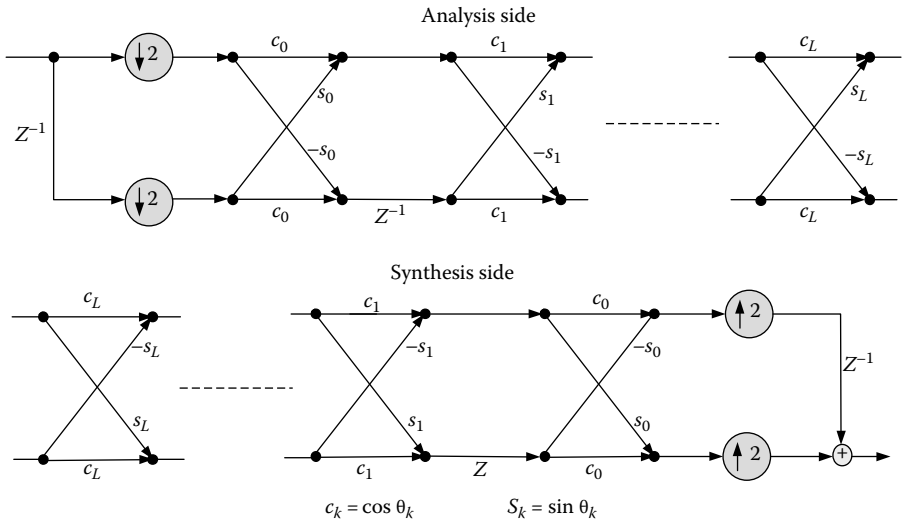


FIGURE 24.19 Orthogonal lattice structure.

where

$$\mathbf{D}(z) \equiv \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \quad \text{and} \quad \mathbf{S}_i \equiv \begin{bmatrix} 1 & \alpha_i \\ \alpha_i & 1 \end{bmatrix}$$

and $\alpha_i \neq 0$ are the free parameters that define the filters. The structure here is similar to the orthogonal lattice structure except for a sign change in one of the element of $\mathbf{S}_i$.

2. Type B: where all filters in the bank have odd length. All filter coefficients are then symmetric. The polyphase matrix is given by

$$\mathbf{E}(z) = \prod_{i=1}^{L} \mathbf{B}_i(z) \tag{24.97}$$

where

$$\mathbf{B}_i(z) = \begin{bmatrix} 1 + z^{-1} & \alpha_i \\ 1 + \beta_i z^{-1} + z^{-2} & \alpha_i(1 + z^{-1}) \end{bmatrix}$$

and $\alpha_i \neq 0, \beta_i \neq 2$ are the free parameters that define the filters.

This structure can be used to implement all of the linear-phase odd-length wavelet filter banks discussed in Sections 24.6.3, 24.6.5, and 24.6.6, such as the (5,3), (7,5), (9,7), and (19,13)-tap filters and their inverses.

24.8.4 Lifting Scheme

The basic idea behind lifting [17–19] is simple:

Whatever change was performed to the signal by some process of addition can be undone by the equivalent process of subtraction.

This idea ties in nicely with the concept of PR where the synthesis filter bank is supposed to undo whatever processing was done by the analysis filter bank. There are two attractive features of lifting: (1) the construction of filter banks, because PR can be guaranteed at every step of the design stage and (2) the implementation of filter banks, because computational efficiency gains of up to 2:1 are possible when compared with separate filters for each band of the filter bank. The former aspect will be considered first.

24.8.4.1 Construction of Filter Banks

The basic building block in lifting is called the lifting step and is related to concept of elementary matrices which is fundamental in matrix analysis. Elementary matrices can be used to perform elementary operations on an arbitrary matrix by either the pre- or post-multiplication of the former with the latter. In the context of filter banks, the arbitrary matrix is the analysis or synthesis polyphase matrix (which is a matrix of Laurent series*). There are three types of elementary matrices (operations) but the type that is relevant in lifting is the one that performs the addition to the elements of a certain row (or column) of a multiple (factor) of the elements of another row (or column). For the analysis polyphase matrix there two types of lifting step:

1. Primal lifting: with elementary matrix given by

$$\begin{bmatrix} 1 & p(z) \\ 0 & 1 \end{bmatrix}$$

   where $p(z)$ is an arbitrary FIR factor. Premultipication by this matrix will change the first row of an arbitrary matrix.
2. Dual lifting: with elementary matrix given by

$$\begin{bmatrix} 1 & 0 \\ d(z) & 1 \end{bmatrix}$$

   where $d(z)$ is an arbitrary FIR factor. Premultipication by this matrix will change the second row of an arbitrary matrix.

Note that both of these lifting matrices have determinants of unity.

Now suppose that there is an existing set of analysis filters with corresponding polyphase matrix $\mathbf{E}^0(z)$; i.e.,

$$\begin{bmatrix} H_0^0(z) \\ H_1^0(z) \end{bmatrix} = \mathbf{E}^0(z^2)\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} = \begin{bmatrix} E_{0,0}^0(z^2) & E_{0,1}^0(z^2) \\ E_{1,0}^0(z^2) & E_{1,1}^0(z^2) \end{bmatrix}\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \tag{24.98}$$

1. If the primal lifting is applied to $\mathbf{E}^0(z)$

$$\mathbf{E}^{\mathrm{p}}(z) \equiv \begin{bmatrix} 1 & p(z) \\ 0 & 1 \end{bmatrix}\mathbf{E}^0(z) \tag{24.99}$$

   then the new set of analysis filters becomes

$$H_0^p(z) = H_0^0(z) + H_1^0(z)p(z^2) \quad \text{and} \quad H_1^p(z) = H_1^0(z)$$

   i.e., the low-pass filter is changed but the high-pass is unchanged.

* Like polynomials but with both positive and negative powers of z.

2. If the dual lifting is applied to $\mathbf{E}^0(z)$

$$\mathbf{E}^{\mathrm{d}}(z) \equiv \begin{bmatrix} 1 & 0 \\ d(z) & 1 \end{bmatrix} \mathbf{E}^0(z) \tag{24.100}$$

then the new set of analysis filters becomes

$$H_0^{\mathrm{d}}(z) = H_0^0(z) \quad \text{and} \quad H_1^{\mathrm{d}}(z) = H_1^0(z) + H_0^0(z)d(z^2)$$

i.e., the high-pass filter is changed but the low-pass is unchanged.

Since the determinant of the lifting matrix is equal to one, the new polyphase matrix, $\mathbf{E}^p(z)$ or $\mathbf{E}^{\mathrm{d}}(z)$, will satisfy the PR condition (Equation 24.93) if the original matrix $\mathbf{E}^0(z)$ also satisfies the condition.

Starting with the trivial polyphase matrix $\mathbf{E}^0(z) = \mathbf{I}$ (identity), alternate primal and dual lifting can be used to construct a whole range of PR filter banks:

$$\begin{aligned} \mathbf{E}(z) &= \begin{bmatrix} K_0 & 0 \\ 0 & K_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ d_M(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & p_M(z) \\ 0 & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ d_1(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & p_1(z) \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} K_0 & 0 \\ 0 & K_1 \end{bmatrix} \prod_{i=M}^{1} \begin{bmatrix} 1 & 0 \\ d_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & p_i(z) \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{24.101}$$

The factors K_0 and K_1 are for normalizing the gains of the low-pass and high-pass filters, respectively.

Now consider the synthesis filter banks. First note that the inverse of the lifting step matrix can be easily obtained by negating the sign of the arbitrary factors, $p(z)$ or $d(z)$, since

$$\begin{bmatrix} 1 & p(z) \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -p(z) \\ 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 \\ d(z) & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -d(z) & 1 \end{bmatrix}$$

Suppose the initial synthesis polyphase matrix of the analysis and synthesis banks, $\mathbf{E}^0(z)$ and $\mathbf{R}^0(z)$, satisfy the PR condition:

$$\mathbf{E}^0(z)\mathbf{R}^0(z) = \mathbf{I}$$

If the equation above is pre- and post-multiplied, respectively, by the primal and inverse primal lifting matrix, then

$$\begin{aligned} \left(\begin{bmatrix} 1 & p(z) \\ 0 & 1 \end{bmatrix} \mathbf{E}^0(z) \right) \left(\mathbf{R}^0(z) \begin{bmatrix} 1 & -p(z) \\ 0 & 1 \end{bmatrix} \right) &= \begin{bmatrix} 1 & p(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -p(z) \\ 0 & 1 \end{bmatrix} = \mathbf{I} \\ \Rightarrow \mathbf{E}^p(z)\mathbf{R}^p(z) &= \mathbf{I} \end{aligned} \tag{24.102}$$

where $\mathbf{E}^p(z)$ is the new analysis polyphase matrix by primal lifting as in Equation 24.99 and $\mathbf{R}^p(z)$ is the new synthesis polyphase matrix by inverse primal lifting:

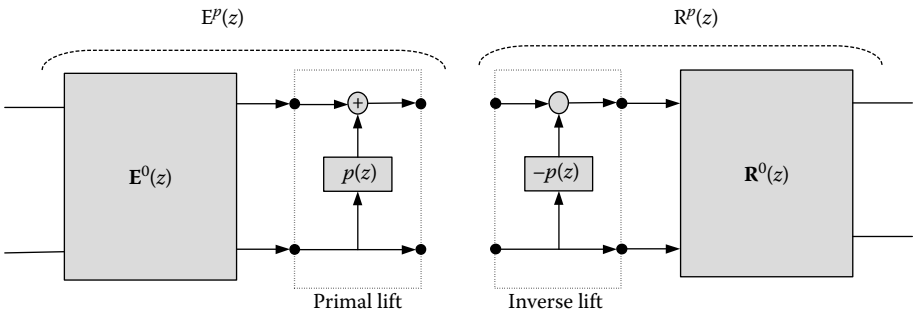


FIGURE 24.20 Primal lifting.

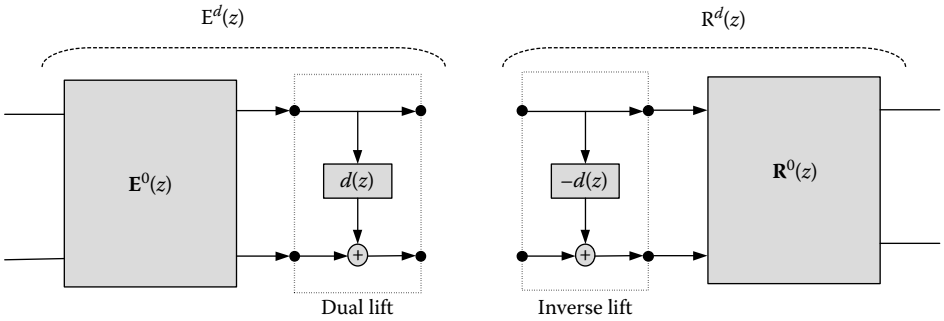


FIGURE 24.21 Dual lifting.

$$\mathbf{R}^p(z) \equiv \mathbf{R}^0(z)\begin{bmatrix} 1 & -p(z) \\ 0 & 1 \end{bmatrix} \tag{24.103}$$

With the combination of the primal/inverse-primal lifting, PR is preserved and this can also been easily seen in the structure in Figure 24.20: whatever was added in the analysis side has been exactly subtracted in the synthesis side.

Wherever the dual lifting is applied in the analysis side, as in Equation 24.100, then the inverse dual lifting should be applied in synthesis side:

$$\mathbf{R}^d(z) \equiv \mathbf{R}^0(z)\begin{bmatrix} 1 & 0 \\ -d(z) & 1 \end{bmatrix} \tag{24.104}$$

with the corresponding structure shown in Figure 24.21. Therefore, if the analysis polyphase matrix is as shown in Equation 24.101, the corresponding synthesis polyphase matrix is given by

$$\begin{aligned} \mathbf{R}(z) &= \begin{bmatrix} 1 & -p_1(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -d_1(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -p_M(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -d_M(z) & 1 \end{bmatrix}\begin{bmatrix} 1/K_0 & 0 \\ 0 & 1/K_1 \end{bmatrix} \\ &= \prod_{i=1}^{M}\begin{bmatrix} 1 & -p_i(z) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -d_i(z) & 1 \end{bmatrix} \cdot \begin{bmatrix} 1/K_0 & 0 \\ 0 & 1/K_1 \end{bmatrix} \end{aligned} \tag{24.105}$$

24.8.4.2 Implementation of Filter Banks

The lifting scheme is also useful for the efficient implementation of filter banks. The analysis or synthesis polyphase matrix of a PR filter bank can be factorized into a product of lifting matrices. The factorization can be achieved through the use of the Euclidean algorithm for Laurent series which is described next.

Given two Laurent series:

$$a(z) = \sum_{n=-L_a}^{M_a} a(n)z^{-n} \quad \text{and} \quad b(z) = \sum_{n=-L_b}^{M_b} b(n)z^{-n}$$

with degrees $|a(z)|$ and $|b(z)|$ defined as

$$|a(z)| = M_a + L_a \quad \text{and} \quad |b(z)| = M_b + L_b$$

The degree of a Laurent series in z is the length of the equivalent FIR filter minus one.

Without loss of generality, we assume $|a(z)| \geq |b(z)|$. Dividing $a(z)$ by $b(z)$ will yield a quotient $q(z)$ and remainder $r(z)$, such that

$$a(z) = b(z)q(z) + r(z) \quad \text{where } |q(z)| = |a(z)| - |b(z)| \text{ and } |r(z)| < |b(z)| \tag{24.106}$$

In Matlab, this operation may be performed using the function $[q, r] = \text{deconv}(a, b)$.

Unlike the case of regular polynomials (with only positive powers of z), the quotient and remainder in Equation 24.106 are not unique and there is some freedom in choosing $q(z)$, depending on how $r(z)$ is aligned within the range of the indices of $a(z)$. This means that in the following algorithm, there are multiple valid factorizations of a given pair of filter polynomials. The Matlab function, deconv(·), chooses to align $r(z)$ with the right-hand end of $a(z)$, assuming row vectors are used for the polynomial coefficients.

The Euclidean algorithm iteratively performs the division process on two Laurent series, $a(z)$ and $b(z)$, as follows:

$$\begin{bmatrix} a_{i+1}(z) & b_{i+1}(z) \end{bmatrix} = \begin{bmatrix} a_i(z) & b_i(z) \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \tag{24.107}$$

starting with $a_0(z) = a(z)$ and $b_0(z) = b(z)$. At each iteration, $i + 1$, the higher degree series a_i is divided by the lower degree series b_i and the remainder is recorded as b_{i+1} for the next iteration, while b_i becomes a_{i+1}. The process continues until iteration n when the remainder b_n becomes zero. The whole process can be expressed as

$$\begin{bmatrix} a_n(z) & 0 \end{bmatrix} = \begin{bmatrix} a(z) & b(z) \end{bmatrix} \prod_{i=1}^{n} \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \tag{24.108}$$

where $a_n(z)$ is the greatest common factor (gcf) of $a(z)$ and $b(z)$, i.e., $a_n(z) = \text{gcf}(a(z), b(z))$.

Inverting the terms in the product gives

$$\begin{aligned} \begin{bmatrix} a(z) & b(z) \end{bmatrix} &= \begin{bmatrix} a_n(z) & 0 \end{bmatrix} \prod_{i=n}^{1} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} a_n(z) & 0 \end{bmatrix} \prod_{i=n/2}^{1} \begin{bmatrix} q_{2i}(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_{2i-1}(z) & 1 \\ 1 & 0 \end{bmatrix} \end{aligned} \tag{24.109}$$

where n is assumed to be even and the product is grouped into pairs of even and odd terms. If n is odd, it is increased by 1 and we set $q_n(z) = 0$.

Each term in the product can be expressed in two forms that involve a primal or a dual lifting matrix

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix}$$

Using the first form for even terms in Equation 24.109 and the second form for odd terms gives

$$[\, a(z) \quad b(z) \,] = [\, a_n(z) \quad 0 \,] \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix} \tag{24.110}$$

For the special case when gcf($a(z)$, $b(z)$) is a constant (i.e., there is no common factor), the following factorization in terms of lifting steps exists:

$$[\, a(z) \quad b(z) \,] = [\, K \quad 0 \,] \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix} \tag{24.111}$$

Now suppose there is an analysis polyphase matrix, given by

$$\mathbf{E}(z) \equiv \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) \\ E_{1,0}(z) & E_{1,1}(z) \end{bmatrix}$$

and satisfying

$$\det(\mathbf{E}(z)) \equiv E_{0,0}(z)E_{1,1}(z) - E_{0,1}(z)E_{1,0}(z) = 1 \tag{24.112}$$

i.e., Equation 24.93 with $c = 1$ and $K = 0$ (no delay).* Equation 24.112 implies that gcf($E_{0,0}(z)$, $E_{0,1}(z)$) is a constant or else the determinant cannot be constant.

Using the Euclidean algorithm described above with $a(z) = E_{0,0}(z)$ and $b(z) = E_{0,1}(z)$ gives

$$[\, E_{0,0}(z) \quad E_{0,1}(z) \,] = [\, K \quad 0 \,] \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix}$$

Form the following matrix

$$\mathbf{E}^0(z) = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) \\ E^0_{1,0}(z) & E^0_{1,1}(z) \end{bmatrix} = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix}$$

The matrix $\mathbf{E}^0(z)$ has unit determinant ($\det(\mathbf{E}^0(z)) = 1$), i.e. PR, and the corresponding low-pass filter is the same as that in matrix $\mathbf{E}(z)$ under consideration. As explained earlier, the dual lifting (Equation 24.100) can be used to change the high-pass filter while leaving the low-pass filter unchanged. Therefore

* There is no loss of generality here as any general matrix satisfying Equation 24.93 can be made into a matrix with unit determinant by muliplying the first (second) row with z^{-K}/c. This is equivalent to the introducing a scaling factor of $1/c$ and the delay z^{-2K} in the low-pass (high-pass) filter which does not significantly change its filtering properties.

by applying an appropriate dual lifting step, $\mathbf{E}^0(z)$ can be changed to any valid $\mathbf{E}(z)$ which produces PR, using

$$\begin{aligned}\mathbf{E}(z) &= \begin{bmatrix} 1 & 0 \\ d(z) & 1 \end{bmatrix} \mathbf{E}^0(z) \\ &= \begin{bmatrix} 1 & 0 \\ d(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix} \\ &= \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ K^2 d(z) & 1 \end{bmatrix} \prod_{i=n/2}^{1} \begin{bmatrix} 1 & q_{2i}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i-1}(z) & 1 \end{bmatrix}\end{aligned} \tag{24.113}$$

Note that, if necessary, the scaling matrix can also be factorized into a sequence of lifting steps (but this is not usually helpful computationally):

$$\begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} = \begin{bmatrix} 1 & K - K^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1/K & 1 \end{bmatrix} \begin{bmatrix} 1 & K-1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{24.114}$$

Therefore it can be concluded that every perfect reconstructing polyphase matrix can be factorized into a sequence of lifting steps.

Example 24.1

The analysis polyphase matrix corresponding to the Haar wavelet filters and its factorization is given by

$$\mathbf{E}(z) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

which consist of one primal lift, one dual lift, and one scaling.

Example 24.2

The analysis polyphase matrix corresponding to the LeGall 3/5 filters and its factorization is given by

$$\mathbf{E}(z) = \begin{bmatrix} \frac{1}{2} & \frac{1}{4}(1+z) \\ -\frac{1}{4}(1+z^{-1}) & \frac{3}{4} - \frac{1}{8}(z^{-1}+z) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{4}(1+z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2}(1+z) \\ 0 & 1 \end{bmatrix}$$

which consist of one primal lift, one dual lift, and one scaling.

24.9 Nonlinear Filter Banks

In general it is difficult to achieve PR if the linear filters H_0, H_1, G_0, and G_1 are replaced with nonlinear filters. Furthermore, there is no generic framework which can be used to analyze the properties of filter banks with nonlinear filters. The lifting scheme described earlier, which works in the polyphase domain, however provides a convenient way to generalize linear filter banks to nonlinear ones. If the primal (dual) lifting filter $p(z)(d(z))$, which is linear, is replaced with a nonlinear filter, PR will still be

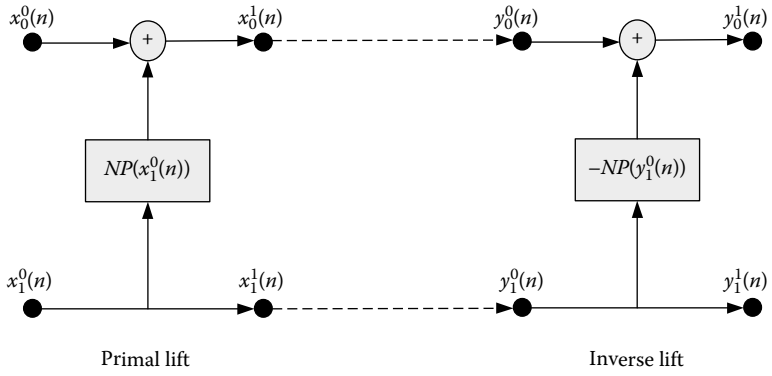


FIGURE 24.22 Nonlinear primal lifting.

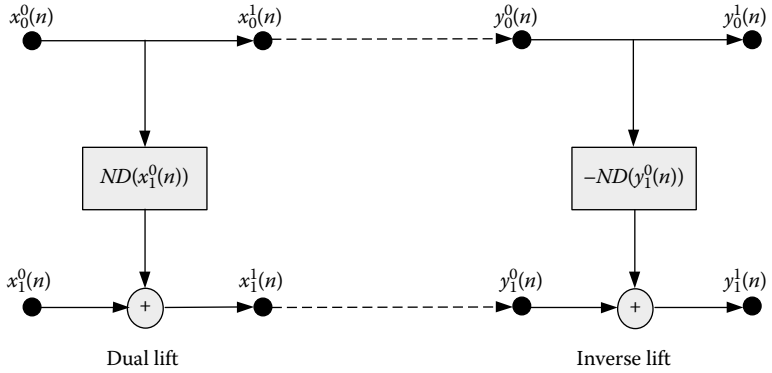


FIGURE 24.23 Nonlinear dual lifting.

maintained as can be easily seen in signal flow graph in Figures 24.22 and 24.23. Note that since the filter is nonlinear, the z-transform transfer function cannot (in general) be defined for the filter. The symbols $NP(\cdot)$ ($ND(\cdot)$) are used instead to denote the primal (dual) nonlinear operator on the signal in the time domain.

A particularly important type of nonlinear filter is the combination of a linear filter and a quantizer [20], such as

$$NP(x_1^0(n)) = \left\lfloor \sum_k p(k)x_1^0(n-k) + \frac{1}{2} \right\rfloor$$

$$ND(x_0^1(n)) = \left\lfloor \sum_k d(k)x_0^1(n-k) + \frac{1}{2} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the largest integer that is no bigger than x, i.e., the floor operation, and $p(k)$ ($d(k)$) are the coefficients of the linear primal (dual) filter. If the inputs $(x_0^0(n), x_1^0(n))$ are integer valued, then the outputs with the primal lifting are given by

$$x_0^1(n) = x_0^0(n) + \left\lfloor \sum_k p(k) x_1^0(n-k) + \frac{1}{2} \right\rfloor$$
$$x_1^1(n) = x_1^0(n)$$

will also be integer valued as the final addition operation involves integers. The same applies with the dual lifting and any combinations of primal and dual liftings. Any linear lifting-based filter bank can therefore be converted into a filter bank that maps integer inputs to integer outputs by using the floor operation to the outputs of the lifting filters and the transform is known as an integer wavelet transform (IWT). The IWT shares most properties of the parent (nonquantized) transform, as long as the floor operations do not contribute a significant level of distortion to the processed signals. This transform is particularly useful for implementing lossless compression schemes, since it retains its PR properties despite the rounding distortions of each primal or dual filter.

Some other nonlinear filters that can be useful in lifting stages are

Median filters (or more general order-statistic filters), which can provide robustness to noise and outliers in the data

Motion-compensated prediction filters, which can be used to improve the performance of video compression systems that employ wavelets along the time axis as well as in the two spatial directions.

24.10 *M*-Band Filter Banks

The 2-band filter bank can be generalized to an arbitrary number of channel as shown in Figure 24.24. The input signal is ideally split into M nonoverlapping contiguous frequency subbands with equal bandwidth. Since there is a factor of M reduction in the bandwidth, downsampling by a factor of M can be applied to preserve the data rate. M-band filter banks are most conveniently analyzed by using the polyphase representation [3,9,12].

The type 1 polyphase decomposition of the analysis filters is given by

$$H_i(z) = E_{i,0}(z^M) + z^{-1}E_{i,1}(z^M) + \cdots + z^{-(M-1)}E_{i,M-1}(z^M) \tag{24.115}$$

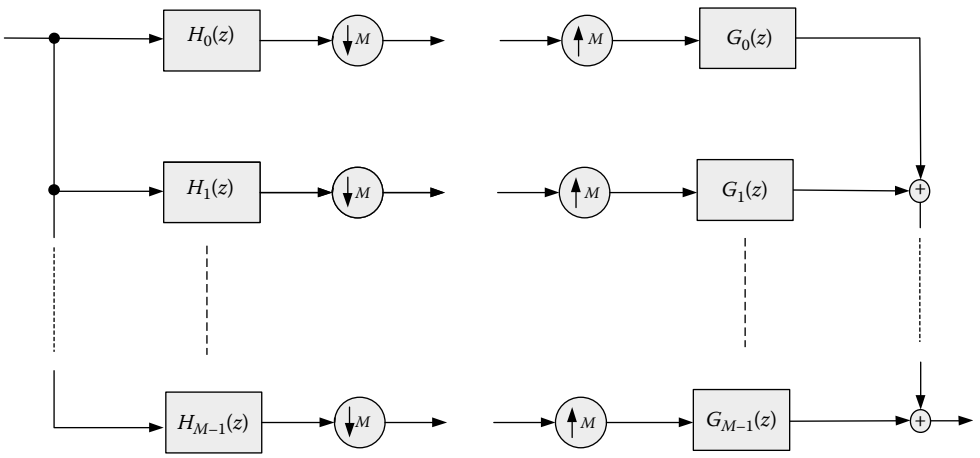


FIGURE 24.24 M-band filter bank.

for $i = 0, \ldots, M-1$; and the corresponding analysis polyphase matrix of size $M \times M$ is

$$\mathbf{E}(z) = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) & \ldots & E_{0,M-1}(z) \\ E_{1,0}(z) & E_{1,1}(z) & \ldots & E_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z) & E_{M-1,1}(z) & \ldots & E_{M-1,M-1}(z) \end{bmatrix} \tag{24.116}$$

The type 2 polyphase decomposition of the synthesis filters is given by

$$G_i(z) = z^{-(M-1)} R_{i,0}(z^M) + z^{-(M-2)} R_{i,1}(z^M) + \cdots + R_{i,M-1}(z^M) \tag{24.117}$$

for $i = 0, \ldots, M-1$; and the corresponding synthesis polyphase matrix of size $M \times M$ is

$$\mathbf{R}(z) = \begin{bmatrix} R_{0,0}(z) & R_{1,0}(z) & \ldots & R_{M-1,0}(z) \\ R_{0,1}(z) & R_{1,1}(z) & \ldots & R_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ R_{0,M-1}(z) & R_{1,M-1}(z) & \ldots & R_{M-1,M-1}(z) \end{bmatrix} \tag{24.118}$$

Figure 24.25 shows the equivalent representation of the M-band filter bank in polyphase form. Just as in the 2-band case, the total polyphase matrix is defined to be $\mathbf{P}(z) \equiv \mathbf{R}(z)\mathbf{E}(z)$ and the condition for PR is

$$\mathbf{P}(z) = \mathbf{R}(z)\mathbf{E}(z) = \mathbf{I} \tag{24.119}$$

For an FIR filter bank it is required that

$$\det(\mathbf{E}(z)) = cz^K \tag{24.120}$$

where c is a nonzero constant and K an integer.

As in the 2-band case, a product (cascade) of matrices satisfying Equation 24.120 will give a matrix that also satisfies Equation 24.120. Lifting can also be defined in the M-band setting but instead of just the

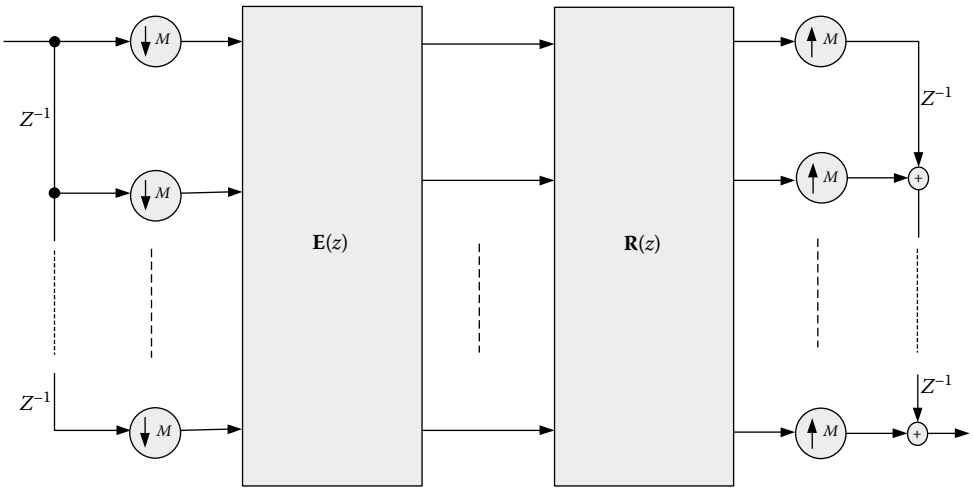


FIGURE 24.25 Polyphase form of the M-band filter bank.

primal and dual liftings, there are in general $M(M-1)$ types of lifting as there is a choice M bands that can be used to modify the signal in the other $M-1$ bands.

Special types of M-band filter banks are

1. Discrete fourier transform (DFT). Although the DFT is usually considered a transform that works on a block of data, it can also be viewed as an M-band filter bank. The analysis polyphase matrix in this case is the DFT matrix with elements given by

$$E_{i,k}(z) = \frac{1}{\sqrt{M}} W_M^{ik} \quad \text{where } W_M = \exp\left(-\frac{j2\pi}{M}\right) \quad \text{for } i,k = 0 \ldots M-1 \tag{24.121}$$

2. Discrete cosine transform (DCT). This can also be viewed as an M-band filter bank. The analysis polyphase matrix in this case is the DCT matrix with elements given by

$$\begin{aligned} E_{0,k}(z) &= \frac{1}{\sqrt{M}} && \text{for } k = 0 \ldots M-1 \\ E_{i,k}(z) &= \sqrt{\frac{2}{M}} \cos\left(\frac{(2k+1)i\pi}{2M}\right) && \text{for } \begin{cases} i = 1 \ldots M-1 \\ k = 0 \ldots M-1 \end{cases} \end{aligned} \tag{24.122}$$

3. Cosine modulated filter bank. The idea here is to design a single prototype low-pass filter for the low-pass subband and to use modulation on the prototype filter to obtain the other filters in the filter bank. The analysis/synthesis filter coefficients are given by

$$h_i(n) = g_i(n) = p(n)\sqrt{\frac{2}{M}} \cos\left(\frac{(2i+1)(2n+M+1)\pi}{4M}\right) \quad \text{for } i = 0 \ldots M-1 \tag{24.123}$$

The prototype filter is $p(n)$ and if it is of length $2M$ and satisfies

$$p(n) = p(2M-n) \quad \text{and} \quad p^2(n) + p^2(n+M) = 1 \tag{24.124}$$

then PR is achieved.

24.11 Hilbert Pairs of Filter Banks (the Dual Tree)

In this final section on 1-D multirate filter banks, we return to the case of the 2-band system, formed into a wavelet tree of filters, as shown in Figure 24.3. While the wavelet transform, implemented in this form, has been found to be remarkably effective as a tool for signal compression (coding) algorithms, it suffers from a rather serious drawback when used as a signal analysis tool. This problem is known as *shift dependence*, or a lack of shift invariance. It is caused by the 2:1 downsampling operations, shown in Figure 24.3, and manifests itself as aliased signal components which are generated by the downsamplers at each stage of the filter tree.

Although these aliased components do indeed cancel each other out when all the subbands are recombined in the reconstruction filter tree of the inverse wavelet transform (provided that the anti-aliasing condition of Equation 24.41 is satisfied), the aliased components are still very significant in the wavelet coefficients (subband outputs) themselves and tend to result in poor performance if the wavelet transform is used as a front-end for more complicated tasks such as signal classification and recognition, or for denoising.

One way to eliminate the aliasing distortion in each subband and obtain *shift invariance* is to use the modified filter tree of Figure 24.4, but without any of the downsamplers. This is known as the *à trous*

algorithm of Mallat [21]. However, for a K-level tree, it results in $(K+1)$ times as many wavelet coefficients being generated and also requires a similar increase in computation load to implement the filters, when compared with the standard nonredundant tree of Figure 24.3 (shown for $K = 4$). These factors become significantly worse for higher-dimensional signals such as images or 3-D datasets.

We now consider a more efficient alternative to the *à trous* algorithm.

24.11.1 Dual-Tree Wavelet Transform

The dual-tree wavelet transform [22] achieves approximate shift invariance with only 2:1 redundancy of coefficients and twofold increase in computation load, relative to the nonredundant tree. It simply employs a second tree of wavelet filters in parallel with the first tree, as shown in Figure 24.26.

The key to obtaining shift invariance from the dual-tree structure lies in designing the relative filter delays at each stage, such that the low-pass filter outputs in tree b are effectively sampled at points midway between the sampling points of the equivalent filters in tree a. This requires a delay difference between the a and b low-pass filters of 1 sample period at tree level 1, and of $\frac{1}{2}$ sample period at subsequent levels (because at these later levels half of the required delay has already been provided by earlier stages). At level 1 we can use any standard orthogonal or biorthogonal wavelet filters and produce the required delay shift trivially by insertion or deletion of unit delays. However, at subsequent levels, the $\frac{1}{2}$ sample delay difference is more difficult to achieve. In Ref. [23], Selesnick showed that these low-pass delay constraints produce a Hilbert pair relationship between the wavelet bases for the two trees. This leads naturally to the interpretation of the outputs from trees a and b as the real and imaginary parts, respectively, of complex wavelet coefficients. The two filter banks are then known as a Hilbert pair.

The $\pm\frac{1}{2}$ sample delay difference between filters $H_{0a}(z)$ and $H_{0b}(z)$ is difficult to express in the time domain as the filters are discrete, but it can be easily expressed in the frequency domain as

$$H_{0b}(e^{j\omega}) \simeq H_{0a}(e^{j\omega})e^{\pm j\omega/2} \tag{24.125}$$

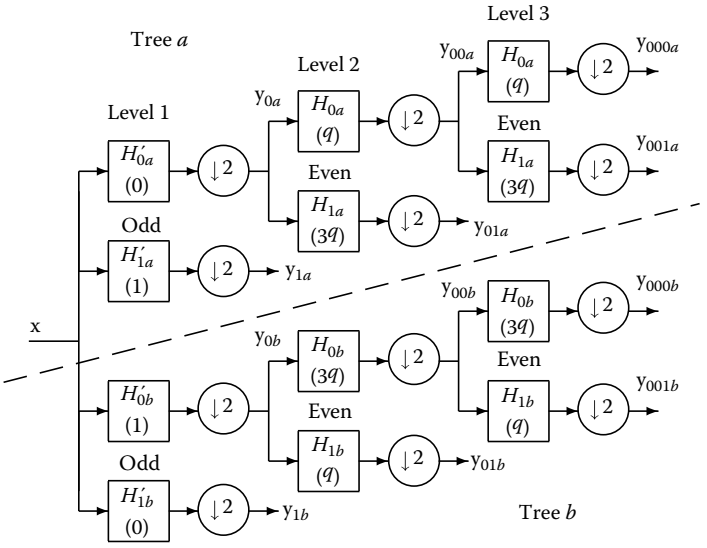


FIGURE 24.26 Dual-tree complex wavelet transform over three levels. The numbers in brackets indicate the delay of each filter (q = one quarter of a sample).

The approximation is necessary here assuming finite-order filters are being used. Solutions are possible if the phase responses are exact and the magnitude responses are approximate, or if the magnitude responses are exact and the phase responses are approximate. In the former case, odd-length and even-length linear-phase filters can be used for H_{0a} and H_{0b}, but better overall characteristics can usually be achieved with the latter case, which allows both filter banks to be CQFs, satisfying Equations 24.42, 24.45, and 24.53.

By convention, we select tree b to lag relative to tree a, so the polarity of the final exponent term in Equation 24.125 is selected to be negative.

24.11.2 Common-Factor Dual-Tree Filter Design

Selesnick [24] proposed a method of satisfying Equation 24.125 by letting

$$H_{0a}(z) = F(z)D(z) \quad \text{and} \quad H_{0b}(z) = F(z)z^{-L}D(z^{-1}) \tag{24.126}$$

where

$$A(z) = \frac{z^{-L}D(z^{-1})}{D(z)} \tag{24.127}$$

is an all pass filter, with group delay designed to approximate the required $\frac{1}{2}$ sample delay difference. Hence we require that

$$A(e^{j\omega}) \simeq e^{-j\omega/2} \tag{24.128}$$

Once $D(z)$ has been selected to satisfy Equations 24.127 and 24.128, we can design $F(z)$ such that both H_{0a} and H_{0b} result in CQFs. This is possible because the product filter $P(z)$ is the same in both cases, since

$$\begin{aligned} P_a(z) &= H_{0a}(z)H_{0a}(z^{-1}) \\ &= F(z)F(z^{-1})D(z)D(z^{-1}) \\ &= H_{0b}(z)H_{0b}(z^{-1}) = P_b(z) \end{aligned} \tag{24.129}$$

Hence $F(z)$ can be designed to contain the required number of zeros at $z = -1$ (to give the desired number of VM) and a factor $R(z)$ of sufficient order to eliminate all the nonzero even powers of z in $P(z)$. This is very similar to the method of choosing $R(z)$ used in Section 24.6.4.

There is a straightforward way to select $D(z)$ of a given order L using Thiran's filter:

$$D(z) = 1 + \sum_{n=1}^{L} \binom{L}{n} \left[\prod_{k=0}^{n-1} \frac{\tau - L + k}{\tau + 1 + k} \right] (-z)^{-n} \tag{24.130}$$

where $\tau = 0.5$ to get $\angle A(e^{j\omega}) \simeq \omega/2$. A range of filters designed with this method are given in Ref. [24].

The one disadvantage of this method of designing a Hilbert pair of filter banks is that the resulting complex basis functions will not be linear phase (i.e., conjugate-symmetric about their midpoint). To achieve this, $F(z)$ must be linear phase, which requires a biorthogonal factorization of $P(z)$ instead of the orthogonal CQF factorization, assumed above. The penalty then is that the reconstruction wavelets will differ from the analysis wavelets.

24.11.3 Q-Shift Dual-Tree Filter Design

A suitable way to employ CQFs and to obtain linear phase complex basis functions from the dual tree was proposed in Ref. [25]. The key here is to make the tree *b* filters equal to the time-reverse of the tree *a* filters. Hence

$$H_{0b}(z) = z^{-L} H_{0a}(z^{-1}) \tag{24.131}$$

The integer delay L is normally chosen so that H_{0a} and H_{0b} cover the same range of z indices.

The filters have real coefficients, so Equation 24.131 automatically ensures that

$$|H_{0b}(e^{j\omega})| = |H_{0a}(e^{j\omega})| \quad \text{and} \quad \angle H_{0b}(e^{j\omega}) = -L\omega - \angle H_{0a}(e^{j\omega})$$

It is also required, from Equation 24.125 (11.1), that $\angle H_{0b}(e^{j\omega}) - \angle H_{0a}(e^{j\omega}) \simeq -\frac{\omega}{2}$, thus giving

$$\angle H_{0a}(e^{j\omega}) \simeq -\frac{L\omega}{2} + \frac{\omega}{4} \quad \text{and} \quad \angle H_{0b}(e^{j\omega}) \simeq -\frac{L\omega}{2} - \frac{\omega}{4} \tag{24.132}$$

This shows that H_{0a} and H_{0b} must have delays that approximate $\pm\frac{1}{4}$ of a sample period relative to the midpoint of the filters at $L/2$ sample periods. Hence this type of 2-band filter system is called a Q-shift (quarter-shift) filter pair. A neat way to visualize the delay property of the Q-shift pair is shown in Figure 24.27, in which the filter taps of the two filters are interleaved to form a single smooth low-pass filter of even length and with symmetric filter coefficients (linear phase), given by

$$H_{L2}(z) = H_{0a}(z^2) + zH_{0b}(z^2) \tag{24.133}$$

The sample rate of the new filter H_{L2} is double that of H_{0a} and H_{0b} and its delay is therefore $(L - \frac{1}{2})$ sample periods. (In Figure 24.27, $L = 1$.)

As well as possessing the approximate Q-shift delay property, H_{0a} and H_{0b} must also satisfy the usual 2-band filter bank constraints of no aliasing and PR. Additionally it is advantageous in many emerging applications of complex wavelets that the Q-shift filters should be orthogonal (i.e., be CQFs) to make the transform a tight frame which conserves energy from the signal in the transform domain. This has the further advantage that the same filters (and their time-reverses) can be used in the forward and inverse transforms, as well as in the two trees *a* and *b*. Finally it is usually highly desirable that the filters have good smoothness properties when iterated over scale.

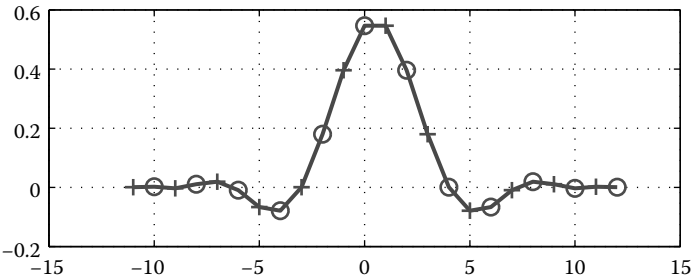


FIGURE 24.27 Impulse response of $H_{L2}(z)$ for $n = 6$. The H_{0a} and H_{0b} filters have $2n = 12$ taps each, shown as circles and crosses, respectively.

Two main design approaches have emerged to meet all of these constraints for Q-shift filter pairs:

Zero-forcing methods with search for good delay properties: The Daubechies orthonormal filters, described in Section 24.6.4, have the maximal number of zeros in the filters H_0 and G_0 at $z = -1$ and there are no further degrees of freedom. These are not able to produce filter pairs with good approximations to Q-shift delays. However the Bernstein method of equation (Equation 24.59) can produce much better approximations if $N = 2K + 1$, which allows one degree of freedom (the parameter α_K) that can be adjusted to optimize the Q-shift delays.

In Ref. [5], all even-length CQFs were evaluated from length 4 to 22. It was found that good approximations to the required Q-shift delay could be obtained for CQF lengths of 8, 12, 18, and 22. The optimal values of α_K in each case were found to be 0.0460, 0.1827, 0.2405, and 0.0245, respectively.

However, despite its good delay properties, the length-8 CQF, with 3 zeros at $z = -1$, was much less smooth than the longer filters, with 5, 8, and 10 zeros at $z = -1$, respectively. Figure 24.28 shows the impulse and frequency responses at level-4 of the DT-CWT, using length-12 Q-shift CQFs, designed with this method. Designs for other filter lengths are shown in Ref. [5]. In all the results of this section, the standard Daubechies 7,9-tap (Antonini) filters were used at level 1 of the dual tree (see Figure 24.10).

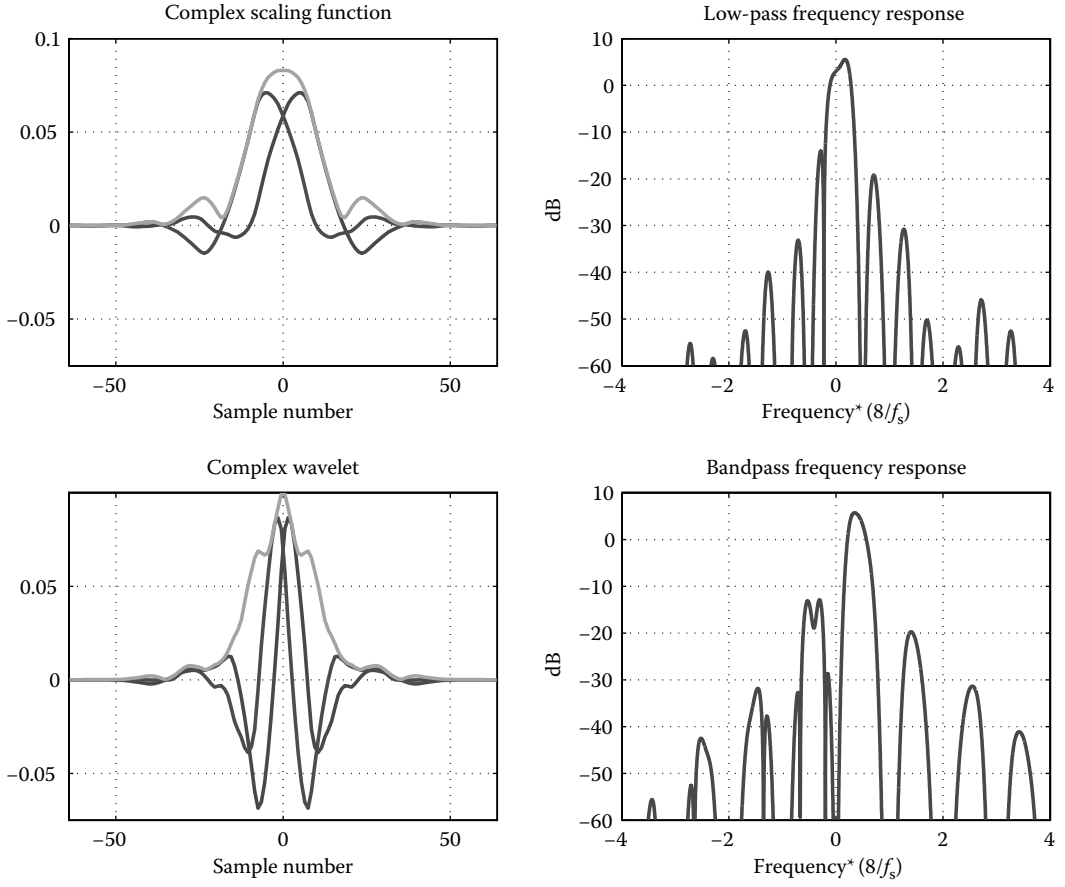


FIGURE 24.28 Impulse and frequency responses at level-4 of the DT-CWT, using length-12 Q-shift CQFs, designed with the zero-forcing method of Ref. [5]. In the impulse responses the two darker curves show the real and imaginary parts of the impulse responses, and the light gray curves show their magnitudes.

Iterative energy minimization methods: The other main design approach for Q-shift filter pairs has been presented in Ref. [26]. The aim is to design the filter $H_{0a}(z)$ (and hence also its time-reverse $H_{0b}(z)$) such that it satisfies the standard CQF conditions and also gives a smooth low-pass filter when formed into $H_{L2}(z)$ using Equation 24.133, which automatically produces the approximate $\frac{1}{4}$-sample (Q-shift) delay property if H_{L2} is smooth enough.

In Ref. [26] it is shown that the PR property of CQFs can be obtained if the product filter $H_{L2}(z)H_{L2}(z^{-1})$ has the property that the term in z^0 is unity and all terms in z^{4k} (k integer) are zero. Smoothness of the filter $H_{L2}(z)$ can be formulated as a need for all components of its spectrum to be approximately zero above about one-sixth of its sample rate ($\omega > \pi/3$). So the ideal design conditions for the length $4n$ symmetric low-pass filter H_{L2} have now been reduced to

1. Zero amplitude for all the terms of $H_{L2}(z)H_{L2}(z^{-1})$ in z^{4k} except the term in z^0, which must be unity
2. Zero (or near-zero) amplitude of $H_{L2}(e^{j\omega})$ for the stopband, $\frac{\pi}{3} \leq \omega \leq \pi$

Condition 1 is a set of quadratic constraints on the elements of the filter tap vector $\mathbf{h}_{L2}$, while condition 2 is a set of linear constraints on $\mathbf{h}_{L2}$, evaluated at a sufficiently fine set of frequencies covering the stopband. Together they form an overdetermined set of equations for the $2n$ unknowns that form one-half of the symmetric vector $\mathbf{h}_{L2}$ (where $2n$ is the length of filters H_{0a} and H_{0b}). If the constraints were all linear, the least mean square (LMS) error solution could be found in the standard way using the pseudo-inverse of the matrix which defines the equations.

To deal with the quadratic constraints, the problem is linearized using an iterative solution. If $\mathbf{h}_{L2}$ at iteration i is given by $\mathbf{h}_i = \mathbf{h}_{i-1} + \Delta\mathbf{h}_i$, then, since convolution (*) is commutative,

$$\begin{aligned}\mathbf{h}_i{*}\mathbf{h}_i &= (\mathbf{h}_{i-1} + \Delta\mathbf{h}_i){*}(\mathbf{h}_{i-1} + \Delta\mathbf{h}_i)\\ &= \mathbf{h}_{i-1}{*}(\mathbf{h}_{i-1} + 2\Delta\mathbf{h}_i) + \Delta\mathbf{h}_i{*}\Delta\mathbf{h}_i\end{aligned} \tag{24.134}$$

If the incremental update $\Delta\mathbf{h}_i$ is assumed to become small as i increases, the final term can be neglected and the convolution be expressed as a linear function of $\Delta\mathbf{h}_i$.

Hence the design problem can now be expressed as

$$\text{Solve for } \Delta\mathbf{h}_i \text{ such that:} \quad \mathbf{C}(\mathbf{h}_{i-1} + 2\Delta\mathbf{h}_i) = [\,0 \ldots 0 \quad 1\,]^{\mathrm{T}} \tag{24.135}$$

$$\mathbf{F}(\mathbf{h}_{i-1} + \Delta\mathbf{h}_i) \simeq [\,0 \ldots 0\,]^{\mathrm{T}} \tag{24.136}$$

where

C is a matrix which calculates every fourth term in the convolution with $\mathbf{h}_{i-1}$

F is a matrix which evaluates the Fourier transform at M discrete frequencies ω from $\frac{\pi}{3}$ to π (typically $M \simeq 8n$ to ensure that all sidelobe maxima and minima are captured reasonably accurately)

Note that only one side of the convolution is needed in **C**, since the result is symmetric about the central term. Also, the columns of **C** and **F** can be combined in pairs so that only the first half of the symmetric $\Delta\mathbf{h}_i$ need be solved for.

In typical applications of complex wavelets, it is often more important to ensure high accuracy in the PR condition than to produce highly smooth wavelets. This is why Equation 24.135 is shown as an equality while Equation 24.136 is only an approximation. Within an iterative LMS framework, high accuracy solutions to some equations can be produced by scaling these up by a factor, β_i, which is progressively increased with i. Hence the optimization may now be expressed as the iterative LMS solution of

$$\begin{bmatrix} 2\beta_i \mathbf{C} \\ \mathbf{F} \end{bmatrix} \Delta \mathbf{h}_i = \begin{bmatrix} \beta_i (\mathbf{c} - \mathbf{C}\ \mathbf{h}_{i-1}) \\ -\mathbf{F}\ \mathbf{h}_{i-1} \end{bmatrix} \tag{24.137}$$

$$\mathbf{h}_i = \mathbf{h}_{i-1} + \Delta \mathbf{h}_i \tag{24.138}$$

where $\mathbf{c} = [0 \ldots 01]^{\mathrm{T}}$. Typically one may choose $\beta_i = 2^i$, and iterate over about 20 iterations, so that reconstruction errors become of the order of $2^{-20} \simeq 10^{-6}$ of smoothness errors. There are two further refinements to this method, described in Ref. [26], which allow for transition band effects and can insert predefined zeros (e.g., VM) in the filters.

Figures 24.29 and 24.30 show some typical scaling functions and wavelets designed using the above method for filter lengths of 14 and 24, respectively. Note the high degree of smoothness that is achievable if the length of the filters is increased to 24. The frequency responses in Figures 24.28 through 24.30, with vertical scales in decibels, clearly show the much lower sidelobe levels and greater ability to reject negative frequencies, obtainable with longer filters. Many other intermediate results are possible with this energy minimization design method, and the method works satisfactorily for filter lengths up to 50 or more, which can be useful if sharp transition bands and/or very low sidelobe levels are required, such as for high-quality audio applications.

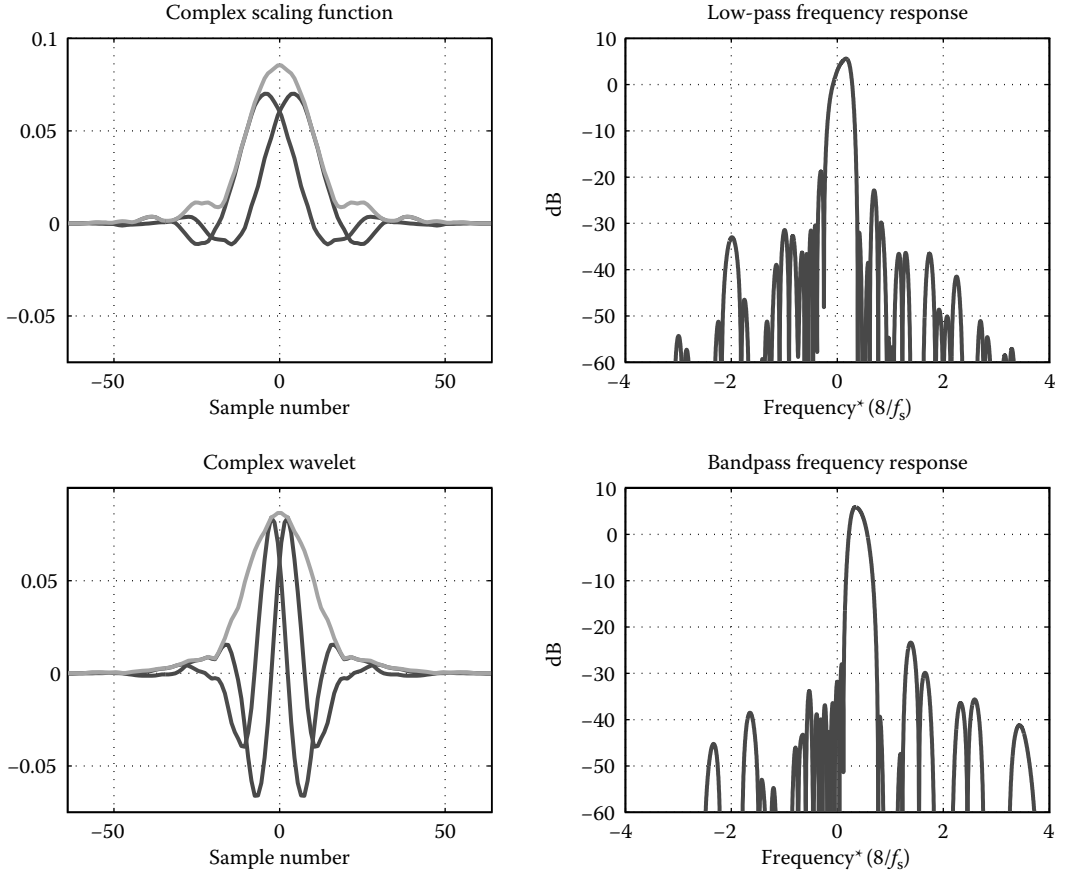


FIGURE 24.29 Impulse and frequency responses at level-4 of the DT-CWT, using length-14 Q-shift CQFs, designed with the energy minimization method of Kingsbury, N. G., Design of Q-shift complex wavelets for image processing using frequency domain energy minimization. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003, pp. 1013–1016.

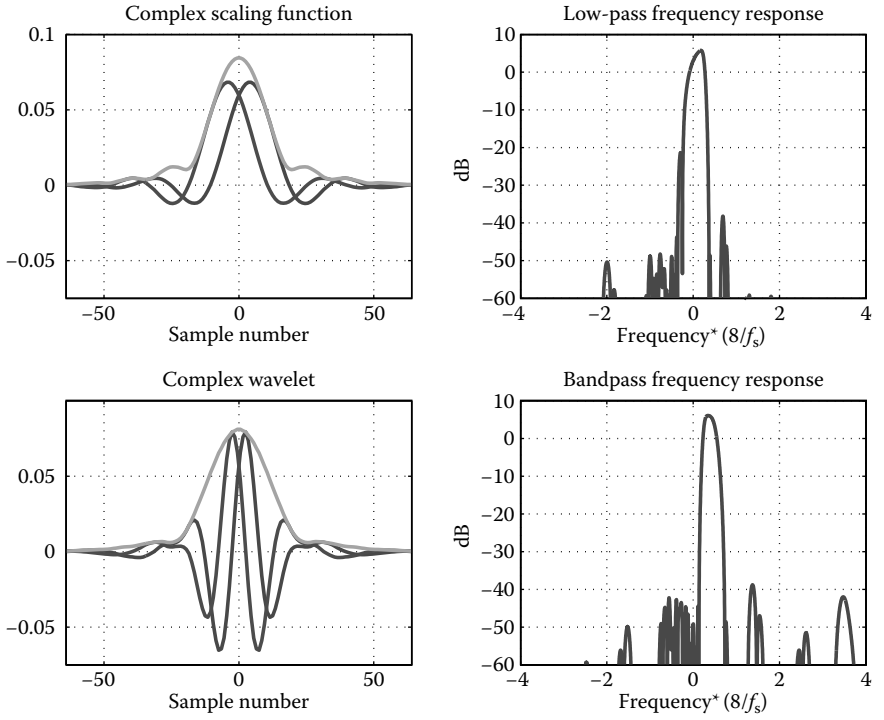


FIGURE 24.30 Impulse and frequency responses at level-4 of the DT-CWT, using length-24 Q-shift CQFs, designed with the energy minimization method of Kingsbury, N. G., Design of Q-shift complex wavelets for image processing using frequency domain energy minimization. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003, pp. 1013–1016.

Earlier papers on the dual-tree method [25,27] proposed some more heuristic methods for designing Q-shift filters. Most notable of these results are a very simple length-6 filter with fair performance and a length-14 filter with generally good performance (similar to Figure 24.29).

24.11.4 Metrics for Shift Dependence of a Filter Bank

Probably the most important feature of Hilbert pairs of wavelets, in addition to those of normal wavelet decompositions, is their approximate *shift invariance* or, expressed another way, their low level of *shift dependence*. This is discussed in detail in Ref. [27] and here we summarize the main technique from that paper for measuring shift dependence.

In order to examine the shift invariant properties of the dual tree, consider what happens when we choose to retain the coefficients of just one type (wavelet or scaling function) from just one level of the dual tree of Figure 24.26. For example, we might choose to retain only the level-3 wavelet coefficients y_{001a} and y_{001b}, and set all others to zero. If the signal $\hat{x}$, reconstructed from just these coefficients, is free of aliasing then we define the transform to be shift invariant at that level. This is because absence of aliasing implies that a given subband has a unique z-transfer function and so its impulse response is linear and time (shift) invariant. In this context we define a subband as comprising all coefficients from *both* trees at a given level and of a given type (either wavelet or scaling function).

Figure 24.31 shows the simplified analysis and reconstruction parts of the dual tree when coefficients of just one type and level are retained. All downsampling and upsampling operations are moved to the outputs of the analysis filter banks and the inputs of the reconstruction filter banks, respectively, using the transformation of Figure 24.4 on both trees, and the cascaded filter transfer functions are combined.

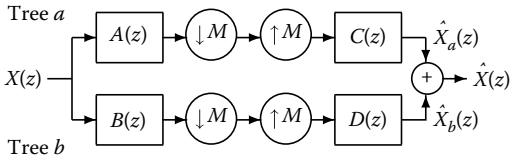


FIGURE 24.31 Basic configuration of the dual tree if either wavelet or scaling-function coefficients from just level m are retained ($M = 2^m$).

$M = 2^m$ is the total down/upsampling factor. For example if y_{001a} and y_{001b} from Figure 24.26 are the only sets of retained coefficients, then the subsampling factor $M = 8$, and $A(z) = H'_{0a}(z)H_{0a}(z^2)H_{1a}(z^4)$, the transfer function from x to y_{001a}. The transfer function $B(z)$ (from x to y_{001b}) is obtained similarly using $H_{\ldots b}(z)$; as are the inverse functions $C(z)$ and $D(z)$ from $G_{\ldots a}(z)$ and $G_{\ldots b}(z)$, respectively.

It is a standard result of multirate analysis that a signal $U(z)$, which is downsampled by M and then upsampled by the same factor (by insertion of zeros), becomes $\frac{1}{M}\sum_{k=0}^{M-1} U(W^k z)$, where $W = e^{j2\pi/M}$. Applying this result to Figure 24.31 gives

$$\hat{X}(z) = \hat{X}_a(z) + \hat{X}_b(z) = \frac{1}{M}\sum_{k=0}^{M-1} X(W^k z)\left[A(W^k z)C(z) + B(W^k z)D(z)\right] \tag{24.139}$$

The aliasing terms in this summation correspond to those for which $k \neq 0$, because only the term in $X(z)$ (when $k = 0$ and $W^k = 1$) corresponds to a linear time (shift) invariant response. For shift invariance, the aliasing terms must be negligible, so we must design $A(W^k z)C(z)$ and $B(W^k z)D(z)$ either to be very small or to cancel each other when $k \neq 0$. Now W^k introduces a frequency shift equal to kf_s/M to the filters A and B (where f_s is the input sampling frequency), so for larger values of k the shifted and unshifted filters have negligible passband overlap and it is quite easy to design the functions $B(W^k z)D(z)$ and $A(W^k z)C(z)$ to be very small over all frequencies, $z = e^{j\theta}$. But at small values of k (especially $k = \pm 1$) this becomes virtually impossible due to the significant width of the transition bands of short-support filters. In these cases it is necessary to design for cancellation when the two trees are combined, and this is what is achieved by the half-sample delay difference strategy outlined above.

A useful way of quantifying the shift dependence of a transform is to examine Equation 24.139 and determine the ratio of the total energy of the unwanted aliasing transfer functions (the terms with $k \neq 0$) to the energy of the wanted transfer function (when $k = 0$), as given by the aliasing energy ratio:

$$R_a = \frac{\sum_{k=1}^{M-1} \mathcal{E}\{A(W^k z)C(z) + B(W^k z)D(z)\}}{\mathcal{E}\{A(z)C(z) + B(z)D(z)\}} \tag{24.140}$$

where $\mathcal{E}\{U(z)\}$ calculates the energy, $\sum_r |u_r|^2$, of the impulse response of a z-transfer function, $U(z) = \sum_r u_r z^{-r}$. Note that $\mathcal{E}\{U(z)\}$ may alternatively be calculated as the integral of the squared magnitude of the frequency response, $\frac{1}{2\pi}\int_{-\pi}^{\pi} |U(e^{j\theta})|^2 d\theta$ from Parseval's theorem. Since R_a is an energy ratio, it is convenient to measure it in decibels.

In Table 3 of Ref. [27], values of R_a are given for various single-tree and dual-tree filter combinations. It is shown that R_a is typically only -3.5 dB for a single-tree DWT, whereas it may be improved to approximately -31 dB with 18-tap Q-shift filters in the dual-tree complex wavelet transform (DT-CWT). Longer filters can reduce the aliasing energy much further if needed, while shorter filters tend to produce more aliasing energy and hence somewhat greater shift dependence.

Other metrics for shift dependence of filter-bank transforms can be based on the amount of shift-dependent variation in the impulse or step responses of the transform, as illustrated in Figure 8

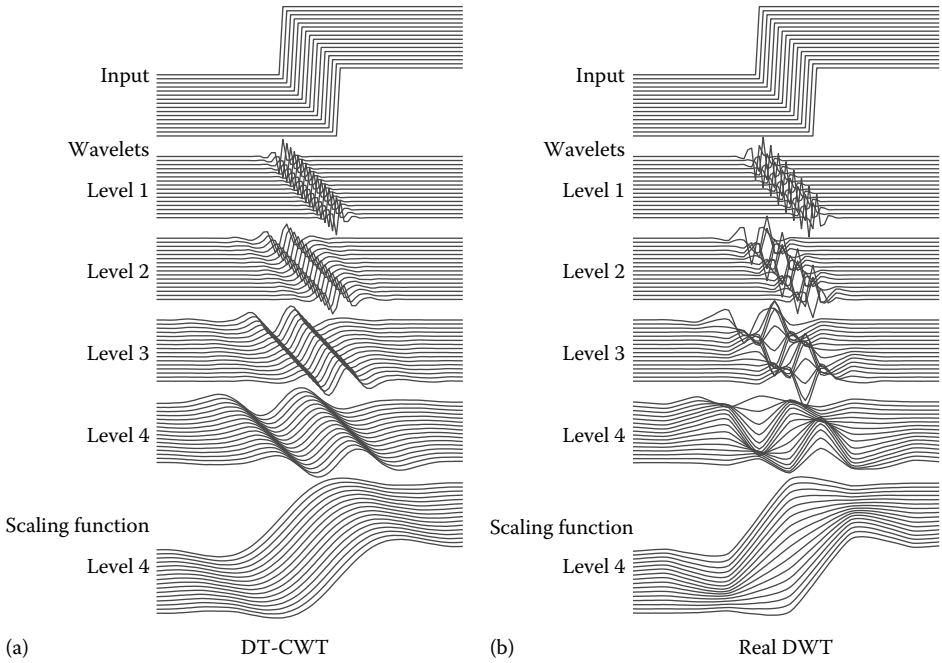


FIGURE 24.32 Wavelet and scaling-function components at levels 1–4 of 16 shifted step responses of the DT-CWT (a) with 18-tap Q-shift filters and the real DWT (b) with 13,19-tap filters.

of Ref. [27], part of which is reproduced here in Figure 24.32. Good transforms have responses that are visually almost identical (Figure 24.32a), whereas poor transforms (e.g., the DWT) have dramatically fluctuating responses (Figure 24.32b).

In this relatively short discussion of dual-tree/Hilbert pair ideas, we have not discussed the important extension of these ideas to two- and three-dimensional datasets, in which a second and very important advantage of the complex nature of the wavelet coefficients lies in their ability to produce strongly directionally selective wavelets while still retaining the computational advantages of separable filtering [22]. In addition, extension of the dual-tree ideas to M-band filter banks has been achieved by Chaux et al. [28], making even greater directional selectivity possible in two and three dimensions.

References

1. I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics Philadelphia, PA, 1992.
2. M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
3. G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.
4. H. Caglar and A. N. Akansu. A generalized parametric PR-QMF design technique based on Bernstein polynomial approximation. *IEEE Trans. Signal Proc.*, 41(7):2314–2321, July 1993.
5. D. B. H. Tay, N. G. Kingsbury, and M. Palaniswami. Orthonormal Hilbert-pair of wavelets with (almost) maximum vanishing moments. *IEEE Signal Proc. Lett.* 13(9):533–536, September 2006.
6. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 1(2):205–220, April 1992.

7. P. A. Regalia, S. K. Mitra, and P. P. Vaidyanathan. The digital all-pass filter: A versatile signal processing building block. *Proc. IEEE*, 76(1):19–37, January 1988.
8. M. J. T. Smith and S. L. Eddins. Analysis/synthesis techniques for sub-band coding. *IEEE Trans. Acoust. Speech Signal Proc.*, 38(8):1446–1456, August 1990.
9. P. P. Vaidyanathan. Multirate digital filters, filter banks, polyphase networks and applications: A tutorial. *IEEE*, 78(1):56–93, January 1990.
10. D. B. H. Tay and N. G. Kingsbury. Design of 2-D perfect reconstruction filter banks using transformations of variables: IIR case. *IEEE Trans. Circuits Syst.—II: Analog and Digital Signal Processing*, 43(3):274–279, March 1996.
11. R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1983.
12. P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
13. P. P. Vaidyanathan. Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect-reconstruction property. *IEEE Trans. Acoust. Speech Signal Proc.*, 35(4):476–492, April 1987.
14. P. P. Vaidyanathan and P. Q. Hoang. Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks. *IEEE Trans. Acoust. Speech Signal Proc.*, 36(1):81–94, January 1988.
15. T. Q. Nguyen and P. P. Vaidyanathan. Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters. *IEEE Trans. Acoust. Speech Signal Proc.*, 37(5):676–690, May 1989.
16. M. Vetterli and D. LeGall. Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE Trans. Acoust. Speech Signal Proc.*, 37(7):1057–1071, July 1989.
17. W. Sweldens. The lifting scheme: A custom-design of biorthogonal wavelets. *Appl. Comput. Harmonic Anal.*, 3(2):186–200, 1996.
18. W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.
19. I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):247–269, 1998.
20. A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo. Wavelet transforms that map integers to integers. *App. Comput. Harmonic Anal.*, 5(3):332–369, 1998.
21. S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, 1998.
22. I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Process. Mag.*, 22(6):123–151, November 2005.
23. I. W. Selesnick. Hilbert transform pairs of wavelet bases. *IEEE Signal Proc. Lett.*, 8(6):170–173, June 2001.
24. I. W. Selesnick. The design of approximate Hilbert transform pairs of wavelet bases. *IEEE Trans. Signal Proc.*, 50(5):1144–1152, May 2002.
25. N. G. Kingsbury. A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. In *Proc. IEEE Int. Conf. Image Processing*, Vancouver, September 2000, paper 1429.
26. N. G. Kingsbury. Design of Q-shift complex wavelets for image processing using frequency domain energy minimization. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003.
27. N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Appl. Comput. Harmonic Anal.*, 10(3):234–253, May 2001.
28. C. Chaux, L. Duval, and J.-C. Pesquet. Image analysis using a dual-tree M-band wavelet transform. *IEEE Trans. Image Proc.*, 15(8):2397–2412, August 2006.

25

Directional Filter Banks

25.1 Introduction **25**-1
25.2 Basic Theory of 2-D Multirate Systems **25**-2
2-D Decimation and Interpolation • Alias-Free 2-D Decimation • Effect of Unimodular Matrices in the Frequency Domain • 2-D Upsamplers and Interpolators
25.3 2-D Maximally Decimated Filter Banks **25**-7
Diamond and Fan Filter Banks
25.4 Bamberger Directional Filter Bank **25**-9
25.5 Design of 2-D Two-Channel Fan Filter Banks **25**-12
FFB Design Using 1-D Quadrature Mirror Filters • BDFB Design Using Ladder Structures
25.6 Undecimated Directional Filter Bank **25**-20
25.7 Bamberger Pyramids **25**-22
25.8 Applications **25**-24
Texture Analysis and Segmentation • Image Denoising Using Bamberger Pyramids • Fingerprint Enhancement and Recognition • Iris Recognition • Finite-Field DFBs • Velocity Selective Filter Banks
25.9 Closing Remarks **25**-30
References **25**-30

Jose Gerardo Rosiles
The University of Texas at EI Paso

Mark J. T. Smith
Purdue University

25.1 Introduction

Directional filter banks (DFBs), which were introduced by Bamberger in 1989 [1–3], are digital analysis–synthesis filter banks that allow an image to be represented as a collection of subbands. In the analysis section, filters succeeded by downsampling matrix operators decompose an image into a set of subbands. In the synthesis section, these subbands are upsampled via a complementary set of matrix operators, then filtered and merged to reconstruct the original image. Such a description is strikingly reminiscent of a conventional 2-D analysis–synthesis filter bank. The distinguishing feature is that the DFB subbands embody angular information (a different angle for each subband), as opposed to the traditional low-, mid-, and high-frequency information. The directional subbands in the DFB are maximally decimated so that the total number of pixels in the full set of subbands is equal to the number of pixels in the original image. Thus, the subbands are nonredundant or, equivalently stated, they form a critically sampled representation.

The DFB has the property that it can achieve exact reconstruction. That is, if the constituent filters are properly designed, the output of the synthesis section can reconstruct the original image from the

decimated subbands exactly. This property is important in applications where modifications are performed in the subband domain for the purpose of enhancing image quality in synthesis. In addition, DFBs are very efficient from a computational perspective and are typically designed for implementation in a separable two-hand filter bank tree structure. This is a noteworthy characteristic as it enables the DFB to have arithmetic complexity comparable to (though a little higher than) the popular 2-D transforms, like discrete Fourier transforms (DFTs) and discrete cosine transforms (DCTs).

DFBs are attractive for many image-processing applications that can benefit from directional analysis such as noise reduction, edge sharpening, feature enhancement, compression, object recognition, and texture synthesis, to mention a few. Furthermore, DFBs are also attractive from a visual information-processing perspective. Research has shown that the first layer of cells in the human visual cortex responds to different orientations and scales. Consequently, DFBs can model in an approximate way the human visual system. All of these reasons have motivated much of the study and interest in directional analysis–synthesis systems in recent years.

Since the introduction of the DFB in 1989, there have been a number of extensions and variations that have been considered, such as relaxation of the maximum decimation condition (leading to directional pyramids), multidimensional extensions of the DFB, nonseparable implementations, and mixed low-, mid-, high-frequency angular decompositions [4–7]. To avoid confusion among these variations, we will refer to the original DFB as the Bamberger DFB (BDFB) in our subsequent discussion and use the term DFB to refer more generically to the broad class of directional decompositions.

25.2 Basic Theory of 2-D Multirate Systems

To set the foundation for the later discussion on the DFB, we start by reviewing the basic concepts of 2-D multirate filter banks. The reader seeking a more comprehensive treatment on the subject is referred to Refs. [8,9].

First, we assume that the digital input signal $x[n_0, n_1]$ is a sampled version of a continuous time signal $x_a(t_0, t_1)$ such that

$$x[n_0, n_1] = x_a(n_0 T_0, n_1 T_1),$$

where T_0 and T_1 are the sampling periods. In matrix notation,* this equates to $x[\mathbf{n}] = x_a(\mathbf{Vn})$ where

$$\mathbf{n} = \begin{bmatrix} n_0 \\ n_1 \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} T_0 & 0 \\ 0 & T_1 \end{bmatrix}.$$

More generally, $\mathbf{V}$ can be any nonsingular real matrix, which implies that there are an infinite number of ways to sample a 2-D signal. Each sampling matrix gives rise to a different sampling geometry or lattice in the 2-D plane. The ''lattice'' generated by the sampling matrix $\mathbf{V}$ is the set of all points $\mathbf{t}$, which implies that $\mathbf{t} = \mathbf{Vn}$, where $\mathbf{n} \in \mathcal{N}$ (the set of all integer vectors). The lattice associated with $\mathbf{V}$ is denoted as $LAT(\mathbf{V})$. Such a lattice corresponds to all the ''integer'' linear combinations of the columns of $\mathbf{V}$, which can be written as $\mathbf{V} = [\mathbf{v}_0|\mathbf{v}_1]$. It should be noted that the relationship between $\mathbf{V}$ and $LAT(\mathbf{V})$ is not one-to-one. Hence different sampling matrices can give rise to the same lattice, a fact that will be useful in later discussion.

An important component of multirate 2-D systems is the unimodular matrix. An integer matrix $\mathbf{E}$ is unimodular if and only if its inverse is an integer matrix, or equivalently if $|\det \mathbf{E}| = 1$. A couple of useful unimodular matrix properties are summarized in the following theorem [8]:

* Boldface capital letters denote matrices and boldface lowercase letters denote vectors.

THEOREM 25.1

Let $\mathbf{V}$ *be a* 2×2 *real nonsingular matrix generating a lattice LAT(*$\mathbf{V}$*).*

1. *Let* $\hat{\mathbf{V}} = \mathbf{VE}$*, where* $\mathbf{E}$ *is a* 2×2 *unimodular matrix. Then* $LAT(\mathbf{V}) = LAT(\mathbf{VE}) = LAT(\hat{\mathbf{V}})$.
2. *Let* $\hat{\mathbf{V}}$ *also be a basis for LAT(*$\mathbf{V}$*). Then there exists a unimodular matrix* $\mathbf{E}$ *such that* $\hat{\mathbf{V}} = \mathbf{VE}$.

Note that $LAT(\mathbf{V}) = LAT(\hat{\mathbf{V}})$ does not imply $x_a(\mathbf{Vn}) = x_a(\hat{\mathbf{V}}n)$. Although the lattice points are the same for both matrices, they do not occur in the same order, i.e., the samples of $x_a(\mathbf{Vn})$ are rearranged samples of $x_a(\hat{\mathbf{V}}n)$. This property is useful and was employed in Ref. [3] as a "change-of-variables" procedure to simplify the DFB design.

To illustrate the idea of lattice resampling with unimodular matrices, consider the lattice generated by the matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = [\mathbf{v}_0 | \mathbf{v}_1]. \tag{25.1}$$

It generates what is known as the quincunx lattice, shown in Figure 25.1a, where the heavy dark dots correspond to elements of $LAT(\mathbf{V})$. The geometry of the lattice is established by vectors $\mathbf{v}_0$ and $\mathbf{v}_1$. A set of trivial unimodular matrices can be obtained by choosing variations of the matrices

$$\begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & \pm 1 \\ \pm 1 & 0 \end{bmatrix}. \tag{25.2}$$

Resampling with these matrices, we can produce rotations and reflections of any lattice. In particular, there are eight variations/permutations of the quincunx sampling matrix.

Another useful type of unimodular matrix is

$$\mathbf{E} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{25.3}$$

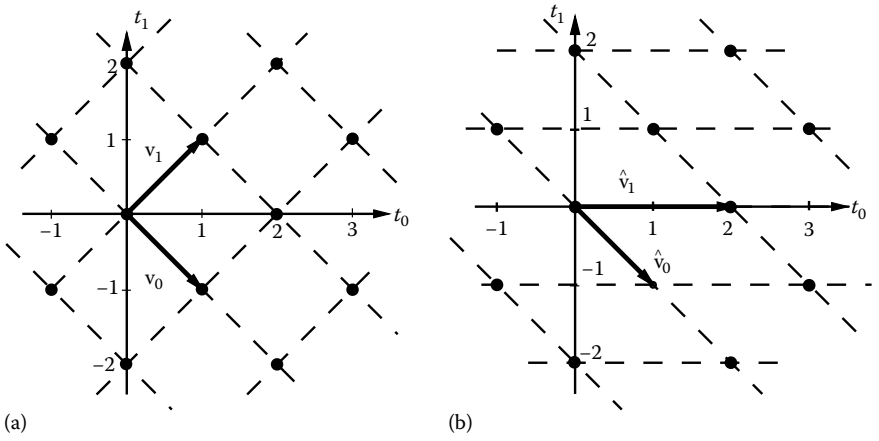


FIGURE 25.1 Examples of 2-D lattices. (a) Lattice generated by sampling with a quincunx matrix. (b) Lattice generated by resampling a quincunx matrix with a unimodular matrix. Note that the sample density of the lattice is same in both figures, but the geometry of the lattice is different.

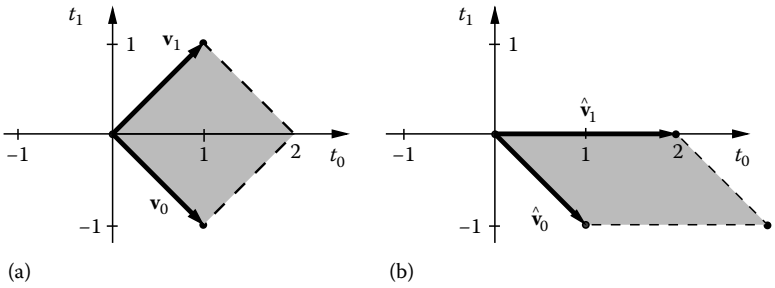


FIGURE 25.2 Fundamental parallepipeds for (a) the quincunx lattice and (b) the resampled quincunx lattice.

It generates the sampling matrix

$$\hat{\mathbf{V}} = \mathbf{VE} = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}, \tag{25.4}$$

whose lattice is illustrated in Figure 25.1. It is easy to see that $LAT(\mathbf{V}) = LAT(\hat{\mathbf{V}})$, but the geometry imposed by $\hat{\mathbf{v}}_0$ and $\hat{\mathbf{v}}_1$ is different than that of $\mathbf{V}$, i.e., the samples in the lattice have been permuted or reindexed. In the frequency domain, the reindexing of the lattice induces skewings and rotations in the frequency domain of the 2-D signal but does not alter the sample density.

The fundamental parallepiped of $\mathbf{V}$, denoted by $FPD(\mathbf{V})$, is defined by

$$FPD(\mathbf{V}) = \left\{\text{Set of all points } \mathbf{Vx} \text{ with } \mathbf{x} \in [0,1)^2\right\}. \tag{25.5}$$

Graphically, this is represented by the parallelogram formed by the sampling vectors $\mathbf{v}_0$ and $\mathbf{v}_1$, as shown in Figure 25.2 for $\mathbf{V}$ and $\hat{\mathbf{V}}$.

25.2.1 2-D Decimation and Interpolation

As in the 1-D case, decimation is performed by filtering (to reduce aliasing) followed by downsampling. Here, the downsampling is performed with a resampling matrix $\mathbf{M}$. Keep in mind that a 2-D resampling matrix does not only alter the sampling rate, but it also introduces a geometric modification to the sampling lattice, as illustrated in the previous figures. Thus a 2-D downsampler is implemented by

$$y[\mathbf{n}] = x[\mathbf{Mn}], \tag{25.6}$$

where $\mathbf{M}$ is a 2×2 nonsingular integer matrix. The signal $y[\mathbf{n}]$ retains only those samples of $x[\mathbf{n}]$ that reside on $LAT(\mathbf{M})$. The dark dots in Figure 25.3a illustrate this point for the case of quincunx resampling. The dark samples of $x[\mathbf{n}]$ are the ones retained by resampling with the quincunx matrix $\mathbf{Q}$ defined as

$$\mathbf{Q} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = [\mathbf{v}_0 | \mathbf{v}_1]. \tag{25.7}$$

Figure 25.3b illustrates how the samples are rearranged over the (n_0, n_1) plane. For this quincunx downsampling, the samples experience a clockwise rotation of 45°. For 2-D resampling matrices, the decimation ratio is the determinant of the matrix. Thus for the quincunx resampling case above, the decimation ratio is $|\det \mathbf{Q}| = 2$, as is evident from Figure 25.3.

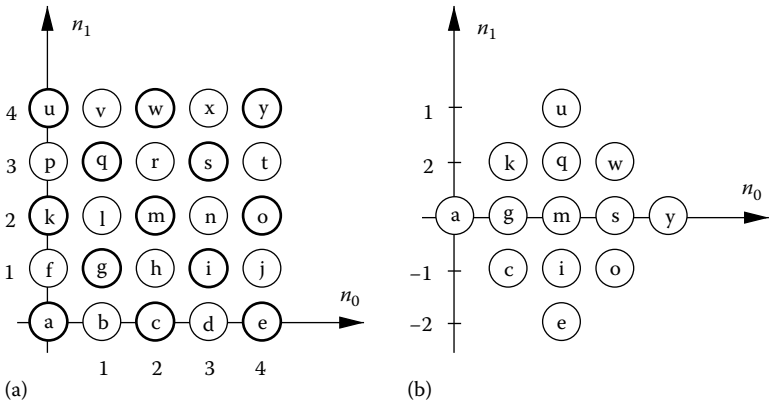


FIGURE 25.3 Downsampling with a quincunx matrix. (a) Original position of samples in the 2-D plane; samples in bold are retained by the downsampling operation. (b) Resulting data after quincunx downsampling with matrix **Q** in Equation 25.7.

To show that this property is true in general, consider the set

$$\mathcal{N}(\mathbf{M}) = \{\text{Set of integer vectors in } FPD(\mathbf{M})\} = \{\text{Set of integers of the form } \mathbf{Mx},\ \mathbf{x} \in [0,1)^2\}.$$

It is easy to see that the cardinality of the set $J(\mathbf{M})$ is $|\mathcal{N}(\mathbf{M})| = |\det \mathbf{M}|$. For instance, if we consider quincunx downsampling, then

$$\mathcal{N}(\mathbf{Q}) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\} = \{\mathbf{k}_0, \mathbf{k}_1\}. \tag{25.8}$$

The vectors $\mathbf{k}_i$ are known as coset vectors.

Assuming the signal $x(\mathbf{t})$ has been sampled with $\mathbf{V} = \mathbf{I}$ (i.e., a rectangular uniform grid over the 2-D plane), then any other sublattice $LAT(\mathbf{M})$ obtained by downsampling $LAT(\mathbf{I})$ is known as a sublattice of **I**. For matrix **M**, we can obtain $J(\mathbf{M})$ distinct, nonoverlapping sublattices from **I**, expressed as

$$x_i[\mathbf{n}] = x(\mathbf{Mn} + \mathbf{k}_i). \tag{25.9}$$

Each sublattice is also known as a coset of $LAT(\mathbf{M})$. Figure 25.3a shows graphically that there are two sublattices, $x_0[\mathbf{n}]$ and $x_1[\mathbf{n}]$ for the quincunx matrix **Q**. From a multirate systems perspective, cosets are the equivalent of polyphase components.

25.2.2 Alias-Free 2-D Decimation

In the 1-D case, an antialiasing filter is generally applied to the signal prior to downsampling. For decimation by a factor of M, the filter has an ideal cutoff frequency of π/M, where M is the decimation ratio. In two dimensions there is an analogous cutoff frequency requirement but also a prescribed region of support determined by the geometry of $LAT(\mathbf{M})$, where **M** is the downsampling matrix.

Let $X(\boldsymbol{\omega})$ be the 2-D DTFT of $x[\mathbf{n}]$, which is 2π-periodic in $\boldsymbol{\omega}$ over the 2-D frequency plane with $\boldsymbol{\omega} = [\omega_0 \omega_1]^{\mathrm{T}}$. For convenience, with abuse of notation, $X(\boldsymbol{\omega})$ will be used in place of $X(e^{j\omega_0}, e^{j\omega_1})$.

The region of support for an antialiasing filter needed to ideally downsample by **M** is given by [8]

$$\boldsymbol{\omega} = \pi \mathbf{M}^{-\mathrm{T}} \mathbf{x} + 2\pi \mathbf{m}, \quad \mathbf{x} \in [-1,1)^2, \quad \mathbf{m} \in \mathcal{N}. \tag{25.10}$$

The term "$2\pi\mathbf{m}$" represents the periodicity of the antialias region of support, while the term "$\pi\mathbf{M}^{-\mathrm{T}}\mathbf{x}$" corresponds to the fundamental frequency cell (or region of support) that is periodically replicated over the frequency plane. If

$$\mathbf{M} = \begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix}, \tag{25.11}$$

the frequency cell can be expressed as

$$-\pi \leq m_{00}\omega_0 + m_{10}\omega_1 < \pi \ \cap \ -\pi \leq m_{01}\omega_0 + m_{11}\omega_1 < \pi. \tag{25.12}$$

It is also illuminating to examine the 2-D decimator in the frequency domain. The key here is understanding the 2-D downsampling process. Suppose a signal $X(\boldsymbol{\omega})$ is downsampled with the matrix $\mathbf{M}$. The frequency-domain expression for the downsampled signal $Y(\boldsymbol{\omega})$ in terms of $X(\boldsymbol{\omega})$ is (analogous to the 1-D case)

$$Y(\boldsymbol{\omega}) = \frac{1}{J(\mathbf{M})} \sum_{\mathbf{k}\in\mathcal{N}(\mathbf{M}^T)} X(\mathbf{M}^{-\mathrm{T}}(\boldsymbol{\omega} - 2\pi\mathbf{k})), \tag{25.13}$$

where $X(\mathbf{M}^{-\mathrm{T}}\boldsymbol{\omega})$ is a "stretched" version of $X(\boldsymbol{\omega})$. For the case of the signal $X(\boldsymbol{\omega})$ shown in Figure 25.4a that undergoes quincunx downsampling, we have

$$Y(\boldsymbol{\omega}) = \frac{1}{2}X(\mathbf{Q}^{-\mathrm{T}}\boldsymbol{\omega}) + \frac{1}{2}X(\mathbf{Q}^{-\mathrm{T}}(\boldsymbol{\omega} - 2\pi\mathbf{k}_1)), \tag{25.14}$$

where $\mathbf{k}_1 = [1 \ \ 0]^{\mathrm{T}}$ is the second coset vector of $\mathbf{Q}$. Figure 25.4b shows the result after the downsampling operation. In this case, from Equation 25.10, the support for the term $X(\mathbf{Q}^{-\mathrm{T}}\boldsymbol{\omega})$ is $\boldsymbol{\omega} = \pi\mathbf{x} + 2\pi\mathbf{Q}^{\mathrm{T}}\mathbf{m}$,

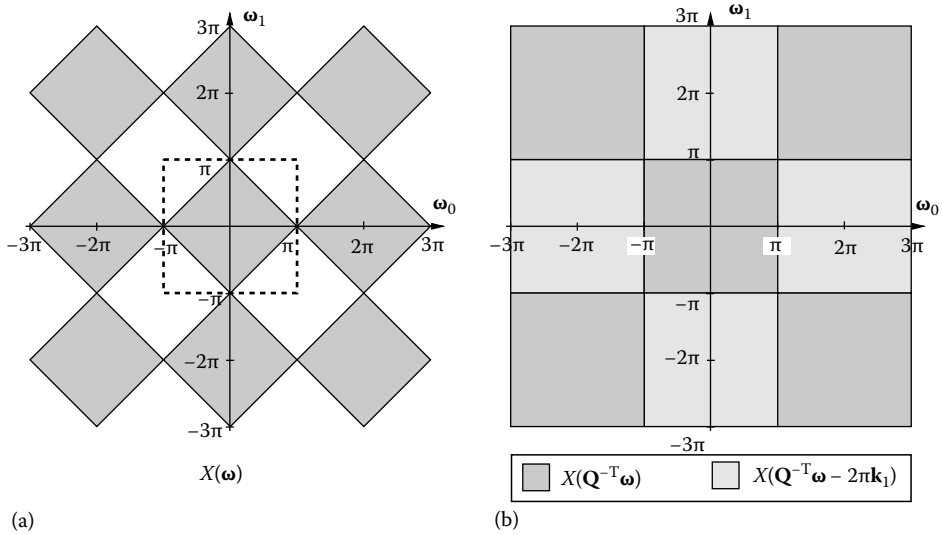


FIGURE 25.4 Effect of downsampling by $\mathbf{Q}$. (a) Diamond band limited signal. (b) The diamond region is stretched by a factor of two and rotated by 45° covering the full unit cell. The empty regions in (a) are covered by $X(\mathbf{Q}^{-\mathrm{T}}(\boldsymbol{\omega} - 2\pi\mathbf{k}_1))$.

which corresponds to the full $[-\pi, \pi)^2$ support periodically replicated with periodicity matrix $2\pi\mathbf{Q}^{\mathrm{T}}$. Hence the frequency support of $X(\boldsymbol{\omega})$ has been stretched by a factor of 2 and rotated by 45°. The complementary regions are filled by the term $X(\mathbf{Q}^{-\mathrm{T}}(\boldsymbol{\omega} - 2\pi\mathbf{k}_1))$.

25.2.3 Effect of Unimodular Matrices in the Frequency Domain

Previously, we used the unimodular matrix $\mathbf{E}$ from Equation 25.3 to modify $LAT(\mathbf{V})$. Similar properties follow for the case of resampling a discrete signal using unimodular matrices. For instance, we can modify the quincunx lattice generated by $\mathbf{Q}$ using $\mathbf{E}$ such that $LAT(\mathbf{QE})$ contains the same points as $LAT(\mathbf{Q})$ but with a different sample ordering. From Equation 25.12 the frequency support for a signal lying over $LAT(\mathbf{QE})$ is given by

$$-\pi \leq \omega_0 - \omega_1 < \pi \ \cap \ -\frac{\pi}{2} \leq \omega_0 < \frac{\pi}{2}. \tag{25.15}$$

This represents a skewing of the frequency cell associated with $LAT(\mathbf{Q})$.

25.2.4 2-D Upsamplers and Interpolators

Complementary to the decimation operation is the interpolation operation, which is composed of an upsampler followed by an anti-imaging filter. For a 2-D signal $x[\mathbf{n}]$, the upsampling operation with matrix $\mathbf{M}$ generates the signal

$$y[\mathbf{n}] = \begin{cases} x(\mathbf{M}^{-1}\mathbf{n}) & \text{if } \mathbf{n} \in LAT(\mathbf{M}), \\ 0 & \text{otherwise.} \end{cases} \tag{25.16}$$

This means that a matrix $\mathbf{M}$ with $J(\mathbf{M})$ coset vectors maps all the samples of $x[\mathbf{n}]$ to the sublattice produced by $\mathbf{k}_0$, while the remaining sublattices are populated by zeros.

The frequency domain expression for the interpolator is

$$Y(\boldsymbol{\omega}) = X(\mathbf{M}^{\mathrm{T}}\boldsymbol{\omega}), \tag{25.17}$$

which implies $Y(\omega)$ has a periodicity matrix $2\pi\mathbf{M}^{-\mathrm{T}}$, and there are $|\det \mathbf{M}|$ compressed images of $X(\omega)$ in the $[-\pi, \pi]^2$ frequency cell. Decimators and interpolators are the basic elements of multirate systems and will be used extensively in the discussions that follow.

25.3 2-D Maximally Decimated Filter Banks

A more general condition than Equation 25.10 to derive the support of the antialias filter is

$$\boldsymbol{\omega} = \mathbf{c} + \pi\mathbf{M}^{-\mathrm{T}}\boldsymbol{x} + 2\pi\mathbf{m}, \quad \boldsymbol{x} \in [-1,1)^2, \quad \mathbf{m} \in \mathcal{N}, \tag{25.18}$$

where $\mathbf{c}$ is some arbitrary constant vector. This implies there are an infinite number of possible antialiasing filter support regions for sampling matrix $\mathbf{M}$. There are, however, a few choices of $\mathbf{c}$ that are of more practical importance as our intuition can suggest.

Going back to the 1-D case, an M-channel maximally decimated filter bank requires having M nonoverlapping (ideal) filters with bandwidth $\frac{\pi}{M}$, so that the range $[-\pi, \pi)$ is completely covered by the M channels. This concept can be generalized to two dimensions by finding the nonoverlapping regions whose union covers the frequency cell $[-\pi, \pi)^2$. Such regions are obtained by choosing

$$\mathbf{c} = 2\pi\mathbf{M}^{-\mathrm{T}}\mathbf{k}, \quad \mathbf{k} \in \mathcal{N}(\mathbf{M}^{\mathrm{T}}) \tag{25.19}$$

in Equation 25.18. It is easy to show that for each of the $J(\mathbf{M})$ coset vectors $\mathbf{k}_i$, there is a different nonoverlapping frequency region. From the filter bank perspective, each of these regions can be used as the frequency-domain region for a subband whose information can be extracted with the properly designed subband filter. This concept is illustrated next with a couple of examples.

25.3.1 Diamond and Fan Filter Banks

Two popular 2-D filter banks are the diamond and fan filter banks (FFBs) shown in Figure 25.5. They form the building blocks for a broad class of directional decompositions, which we will discuss later. The diamond filter has a passband as shown in Figure 25.5a in the $[-\pi, \pi)^2$ region. These filters can provide alias-free decimation for the quincunx resampling matrix $\mathbf{Q}$. Using the corresponding coset vectors (Equation 25.8) in Equations 25.18 and 25.19, we obtain the support regions

$$\begin{aligned} \Omega_0&: \ -\pi \le \omega_0 - \omega_1 < \pi \ \cap \ -\pi \le \omega_0 + \omega_1 < \pi, \\ \Omega_1&: \ \pi \le \omega_0 - \omega_1 < 3\pi \ \cap \ -\pi \le \omega_0 + \omega_1 < \pi. \end{aligned} \tag{25.20}$$

Related to the diamond filter is the fan filter which has the passband shown in Figure 25.5b. It is identical to the diamond filter except that it is frequency shifted (either in ω_0 or ω_1) by π. We can show that fan support is a valid antialias passband for the quincunx matrix by finding the vector $\mathbf{c}$ that satisfies Equation 25.18. If we let

$$\mathbf{c}_0 = \begin{bmatrix} \pi \\ 0 \end{bmatrix}, \quad \mathbf{c}_1 = \begin{bmatrix} 0 \\ \pi \end{bmatrix},$$

we get the two fundamental complementary fan-shaped regions for $F_0(\omega)$ and $F_1(\omega)$:

$$\begin{aligned} \Omega_0&: \ 0 \le \omega_0 - \omega_1 < 2\pi \ \cap \ 0 \le \omega_0 + \omega_1 < 2\pi, \\ \Omega_1&: \ -2\pi \le \omega_0 - \omega_1 < 0 \ \cap \ 0 \le \omega_0 + \omega_1 < 2\pi. \end{aligned} \tag{25.21}$$

Again, the diamond filter bank and the FFB play a crucial role in the construction of DFBs.

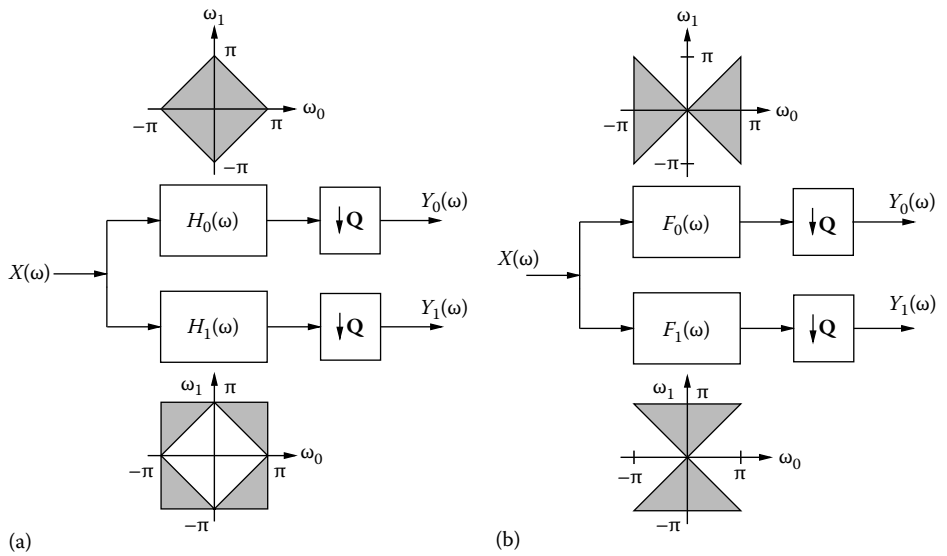


FIGURE 25.5 Block diagram for the diamond and FFBs showing the ideal support for the 2-D filters.

25.4 Bamberger Directional Filter Bank

In this chapter, we present the theory of the DFB introduced in Refs. [2,3]. The BDFB splits the frequency cell $[-\pi, \pi)^2$ into an even number of wedge-shaped regions as shown in Figure 25.6. The BDFB employs a tree-structured 2-D filter bank that diagrammatically is analogous to a 1-D tree-structured filter bank.

Using this approach, Bamberger introduced BDFBs with different numbers of subbands such as 6, 10, 18, and more [2]. However, the BDFBs that have received most attention over the years are the uniform N-stage tree structure filter banks that generate $M = 2^N$ subbands. Given the extensive number of applications in which $M = 2^N$ BDFBs are employed, we focus on this case in the reminder of this chapter.

The BDFB is implemented with a small set of well-defined building blocks, each with low computational complexity. Without loss of generality, we present the BDFB for the $M = 8 (N = 3)$ case, which achieves the frequency plane partitioning shown in Figure 25.6d. The extension to 16 bands, 32 bands, and higher follows by a straightforward extension of the tree structure. The block diagram for an eight-band BDFB analysis stage is shown in Figure 25.7. The primary building block is the two-channel FFB.

The third stage of the BDFB includes additional resampling matrices $\mathbf{U}_i$ and $\mathbf{B}_i$, which are unimodular. The matrices $\mathbf{U}_i$ resample the four subbands from the second stage to remap their frequency support to a fan-shaped region. This allows the use of the FFB on all stages. The function of the $\mathbf{B}_i$ matrices is to adjust the sampling lattice of the resulting subbands so they have a rectangular geometry (i.e., the overall sampling matrix is diagonal with a downsampling rate of M). The use of $\mathbf{U}_i$ matrices was introduced by

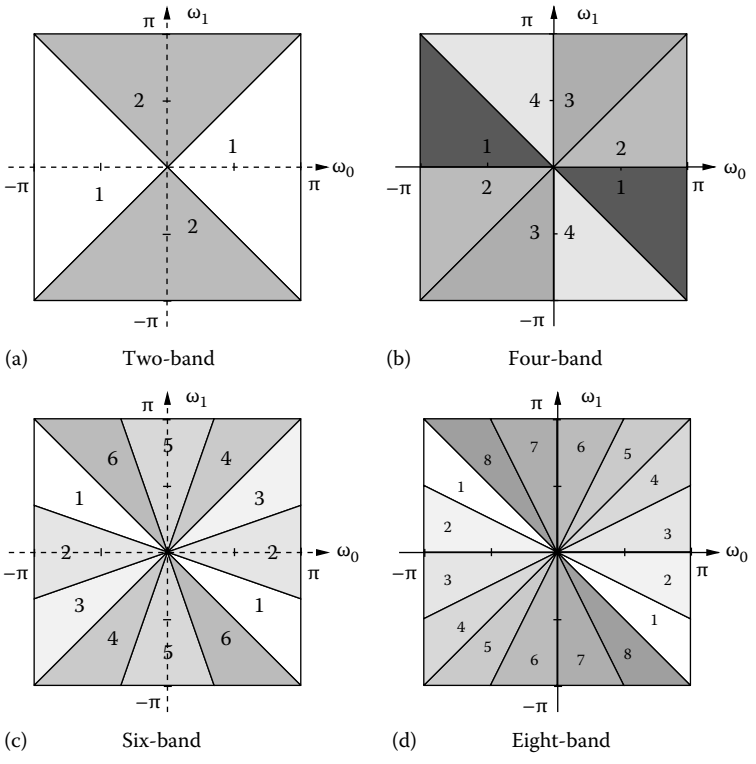


FIGURE 25.6 Frequency band partitions achieved by the BDFB. (a) Two-band. (b) Four-band. (c) Six-band. (d) Eight-band.

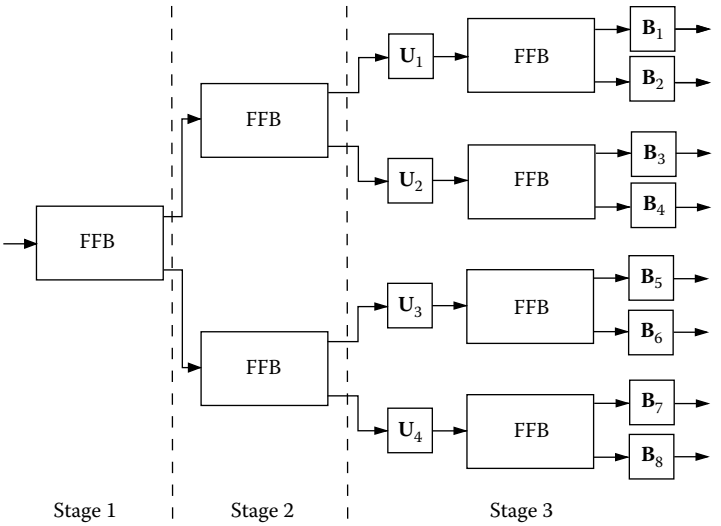


FIGURE 25.7 Implementation of an eight-band BDFB using a tree structure with FFBs and backsampling matrices.

Park et al. [10] as a way to achieve a more geometrically accurate subband representation. According to the selection rules developed in Ref. [10], we have chosen the following matrix values for $\mathbf{U}_i$:

$$\mathbf{U}_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{U}_3 = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{U}_4 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}. \tag{25.22}$$

Similarly, the values for the $\mathbf{B}_i$ matrices are given by

$$\begin{aligned} \mathbf{B}_1 = \mathbf{B}_2 &= \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{B}_3 = \mathbf{B}_4 = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}, \\ \mathbf{B}_5 = \mathbf{B}_6 &= \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{B}_7 = \mathbf{B}_8 = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}. \end{aligned} \tag{25.23}$$

A detailed stage-by-stage analysis of the BDFB structure is presented in Refs. [2,11]. Here, we analyze the BDFB by collapsing the tree structure into the M-channel parallel structure shown in Figure 25.8. Using the multirate identity that states "downsampling by $\mathbf{M}$ followed by $H(\omega)$ is equivalent to $H(\mathbf{M}^{\mathrm{T}}\omega)$ followed by downsampling by $\mathbf{M}$," it is possible to migrate all the filters in the tree structure to the left and the downsampling operations to the right of the tree structure. For instance, for an eight-band BDFB, the overall analysis filters are given by

$$G_\ell(\omega) = F_j(\omega)F_k(\mathbf{Q}_1^{\mathrm{T}}\omega)F_m(\mathbf{U}_i^{\mathrm{T}}\mathbf{Q}_2^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}\omega), \tag{25.24}$$

where

$j, k, m \in \{0, 1\}$
$i \in \{1, 2, 3, 4\}$
$\ell = 1, 2, \ldots, 8$

Looking at the tree structure as a binary tree, we have $\ell = j + 2k + 4m + 1$. The multiplication of the three frequency responses in Equation 25.24 results in the filters $G_\ell(\omega)$ with the wedge-shaped passbands

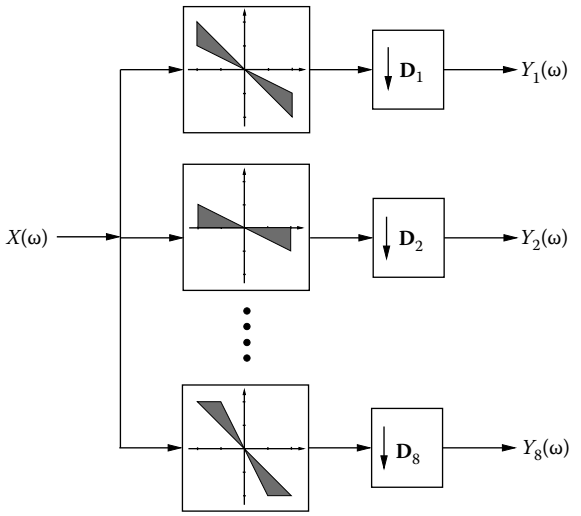


FIGURE 25.8 Parallel implementation of a maximally decimated eight-band DFB. Note that $|\det(\mathbf{D}_i)| = 8$.

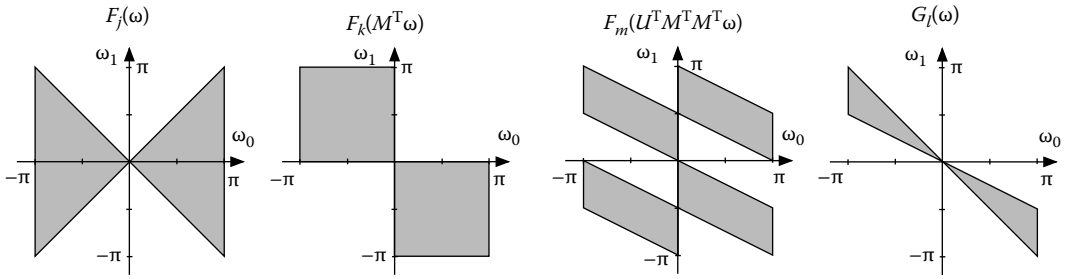


FIGURE 25.9 Illustration of the synthesis of wedge passband for an eight-band DFB. The wedge support is obtained by taking the product of the frequency responses as described by Equation 25.24.

shown in Figure 25.6d. As an illustration, Figure 25.9 shows graphically how the cascade of filters depicted in Equation 25.24 produces a wedge-shaped passband. The fan-shaped support for $F_0(\omega)$ and $F_1(\omega)$ is rotated, skewed, or scaled according to the resampling operations indicated by Equation 25.24. Analogous operations can be implemented to generate the remaining wedge filters.

Viewing the analysis tree structure as a parallel form structure, the eight-band BDFB has overall downsampling matrices given by

$$\mathbf{D}_\ell = \mathbf{Q}_1\mathbf{Q}_2\mathbf{U}_i\mathbf{Q}_3\mathbf{B}_i, \tag{25.25}$$

for $\ell = 1, 2, \ldots, 8$ and $i = \lceil \frac{\ell}{2} \rceil$. This implies that $\mathbf{D}_{2i-1} = \mathbf{D}_{2i}$. It is easy to show that $\mathbf{D}_\ell$ is diagonal with one of the following values:

$$\mathbf{C}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_2 = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}. \tag{25.26}$$

We can conclude three things from the above result:

1. Decimation ratio for all channels is $|\det(\mathbf{C}_1)| = |\det(\mathbf{C}_2)| = 8$, giving us a maximally decimated system as expected.
2. Since the $\mathbf{D}_\ell$ matrices are diagonal, the corresponding sampling grids and fundamental parallelepipeds are rectangular. We should note that this is a consequence of the judicious selection of the unimodular matrices $\mathbf{U}_i$ and $\mathbf{B}_i$. Having the data lie on a rectangular lattice makes further processing easier.
3. Half of the bands ($\ell = 1, 2, 3, 4$) are subsampled by two in the horizontal direction and by four in the vertical direction. The remaining bands ($\ell = 5, 6, 7, 8$) have the opposite structure.

For brevity we focussed on the analysis stage of the BDFB in our discussion. However, it should be noted that the same multirate concepts can be applied to the synthesis stage in an analogous manner.

The generation of BDFBs with $16, 32, \ldots, 2^N$ subbands is achieved by replicating the third stage in Figure 25.7. For an N-stage BDFB, the subbands will have an overall downsampling matrix given by

$$\mathbf{C}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2^{N-1} \end{bmatrix} \quad \text{or} \quad \mathbf{C}_2 = \begin{bmatrix} 2^{N-1} & 0 \\ 0 & 2 \end{bmatrix}.$$

In summary, we have shown how to construct directional filters using multirate operations and FFBs as building blocks. In Section 25.5, we show how to design the FFB filters, $F_0(\omega)$ and $F_1(\omega)$.

25.5 Design of 2-D Two-Channel Fan Filter Banks

Filter banks are usually designed to either satisfy aliasing-cancelation (AC) or perfect-reconstruction (PR) constraints in addition to the usual frequency response specifications. For the 2-D case, a commonly used approach for filter bank design is to derive separable filters from a 1-D systems. The typical consequence of separable filter banks is rectangular tilings of the frequency plane. Implementations of multidimensional filter banks for nonrectangular filters have been reported in the literature [8,9,12–15,58,59]. However, the resulting filters are, in general, nonseparable and require a high order to achieve good frequency selectivity. A more efficient approach is to use low-complexity FFBs. FFBs can be designed using the change-of-variable method, which involves a 1-D to 2-D mapping. Similar to separable implementations, we can take advantage of well-known 1-D filter bank design techniques. The resulting filter banks have efficient polyphase implementations that only require separable filtering operations.

25.5.1 FFB Design Using 1-D Quadrature Mirror Filters

Thus far, the BDFB theory has been developed using filters with ideal frequency responses. In practice, one designs the FFB filters to meet either AC or PR constraints while approximating good passband characteristics. Designing 2-D filter banks with diamond, fan, and other 2-D geometries has been studied by several authors [9,15–17,59,60] where different alias-free and PR 2-D filter banks have been proposed. Bamberger's approach [2] was different in that he presented a design method for FIR PR nonseparable systems using McClellan transformations. In addition, he generalized the change-of-variables scheme introduced by Ansari [18] to produce Quadrature Mirror Filter (QMF) finite impulse response (FIR) and infinite impulse response (IIR) implementations, which provides an efficient polyphase implementation. A drawback of this scheme is that the FIR implementations are not PR. More recently, Rosiles and Smith [19,20] reported the use of the ladder-based filter banks [16,17] to implement the BDFB with FIR PR filters. Ladder networks also have an efficient implementation and consequently are attractive. In this section, we discuss the QMF and ladder structure implementation of the FFB.

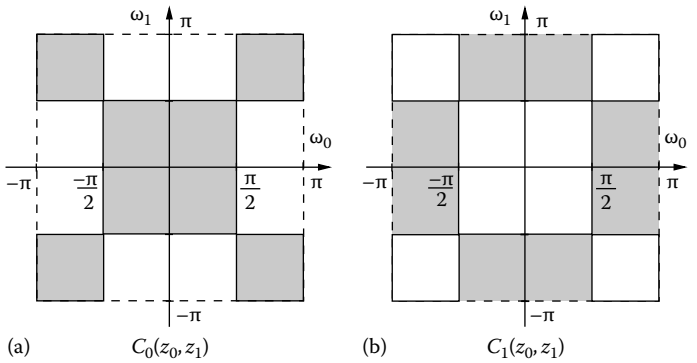


FIGURE 25.10 Ideal checkerboard magnitude responses needed for the generation of the FFB using 1-D QMFs.

The development that follows starts with the design of 2-D diamond filter banks. This system consists of two filters $H_0(\omega_0, \omega_1)$ and $H_1(\omega_0, \omega_1)$ with complementary diamond passband/stopband regions, as shown in Figure 25.5. Fan-shaped filters are obtained by a modulation operation, which in the frequency domain is expressed by $F_0(\omega_0, \omega_1) = H_0(\omega_0 - \pi, \omega_1)$ and $F_1(\omega_0, \omega_1) = H_1(\omega_0 - \pi, \omega_1)$.

The change-of-variables design method [2,18] requires the availability of a 2-D filter $C_0(\omega_0, \omega_1)$ with checker board geometry and the complementary filter $C_1(\omega_0, \omega_1)$. The required passbands and stopbands are shown in Figure 25.10. To achieve the desired filter geometries, the sampling matrix **M** with the corresponding FPD is used to apply the following change of variables in the frequency domain

$$\begin{bmatrix} \omega_0' \\ \omega_1' \end{bmatrix} = \frac{\mathbf{Q}^{\mathrm{T}}}{|\det(\mathbf{Q})|} \begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}. \tag{25.27}$$

For the case of diamond (and fan) filters, the quincunx matrix **Q** provides the appropriate mapping, which simplifies to $\omega_0' = (\omega_0 + \omega_1)/2$ and $\omega_1' = (\omega_1 - \omega_0)/2$. In the z-domain, we can express the change of variables as*

$$z_0 \rightarrow z_0^{\frac{1}{2}} z_1^{\frac{1}{2}} \quad \text{and} \quad z_1 \rightarrow z_0^{-\frac{1}{2}} z_1^{\frac{1}{2}}. \tag{25.28}$$

We should note that this scheme is general and works for other sampling matrices. The effect of the change of variables is to scale, rotate, and skew the passband regions of $C_0(\omega_0, \omega_1)$ and $C_1(\omega_0, \omega_1)$ to obtain the desired geometries.

To design an FFB with an efficient implementation, select a pair of 1-D prototypes that satisfy the QMF relationship

$$H_0(z) = H_1(-z). \tag{25.29}$$

It is well known that the polyphase representation for these filters is

$$\begin{aligned} H_0(z) &= E_0(z^2) + z^{-1}E_1(z^2), \\ H_1(z) &= E_0(z^2) - z^{-1}E_1(z^2). \end{aligned} \tag{25.30}$$

* In this chapter z_0 and z_1 are used to denote the 2-D z-transform complex variables.

It is easy to see that the checker board filters from Figure 25.10 can be designed from the 1-D prototypes using separable filtering operations. In the z-domain this is expressed as

$$\begin{aligned} C_0(z_0, z_1) &= H_0(z_0)H_0(z_1) + H_0(-z_0)H_0(-z_1) = 2[E_0(z_0^2, z_1^2) + z_0^{-1}z_1^{-1}E_1(z_0^2, z_1^2)], \\ C_1(z_0, z_1) &= H_0(z_0)H_1(z_1) + H_0(-z_0)H_1(-z_1) = 2[E_0(z_0^2, z_1^2) - z_0^{-1}z_1^{-1}E_1(z_0^2, z_1^2)]. \end{aligned} \tag{25.31}$$

The 2-D polyphase filters are defined as

$$\begin{aligned} E_0(z_0, z_1) &= E_0(z_0)E_0(z_1), \\ E_1(z_0, z_1) &= E_1(z_0)E_1(z_1). \end{aligned} \tag{25.32}$$

Finally, the diamond filters are obtained by applying the change of variables in Equation 25.28 to $C_0(z_0, z_1)$ and $C_1(z_0, z_1)$. This step produces the geometric transformation on the checkerboard support to obtain the desired diamond support filters. In the frequency domain, we have

$$\begin{aligned} H_0(\boldsymbol{\omega}) &= E_0(\mathbf{Q}^{\mathrm{T}}\boldsymbol{\omega}) + e^{-j\boldsymbol{\omega}^{\mathrm{T}}\mathbf{k}_1}E_1(\mathbf{Q}^{\mathrm{T}}\boldsymbol{\omega}), \\ H_1(\boldsymbol{\omega}) &= E_0(\mathbf{Q}^{\mathrm{T}}\boldsymbol{\omega}) - e^{-j\boldsymbol{\omega}^{\mathrm{T}}\mathbf{k}_1}E_1(\mathbf{Q}^{\mathrm{T}}\boldsymbol{\omega}), \end{aligned} \tag{25.33}$$

where $\mathbf{Q}$ is the quincunx matrix and $k_1 = [1\ 0]^{\mathrm{T}}$. These expressions are similar to those of Equation 25.30 for the 1-D case, implying that the filters are implementable in the polyphase domain. Moreover, the 2-D polyphase components are separable, providing a 2-D structure with low computational complexity. The resulting two-channel analysis structure is shown in Figure 25.11. The corresponding synthesis stage is analogous.

For the 1-D prototype $H_0(z)$, one can use the Johnston QMFs [21], which are aliasing-free but not PR. Another choice is to use the IIR linear phase filters proposed by Smith and Eddins in Ref. [22]. These filters have the form

$$H_0(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_2z^{-3} + b_1z^{-4} + b_0z^{-5}}{a_0 + a_1z^{-2} + a_0z^{-4}} \tag{25.34}$$

and can achieve exact reconstruction. The IIR filters are noncausal, but can be used for the case of finite length signals (like images), using forward and backward difference equations. Moreover, the implementation is very efficient, requiring only 5.1 multiplies and 5.6 adds per output sample, while achieving similar passband characteristics to that of the 32 tap Johnston QMF.

To construct fan filters, we apply the change of variables $z_0 \rightarrow -z_0$ in $H_0(z_0, z_1)$ and $H_1(z_0, z_1)$ to induce a shift by π along the ω_0-axis. Thus "off-the-shelf" QMFs can be used to generate high-quality

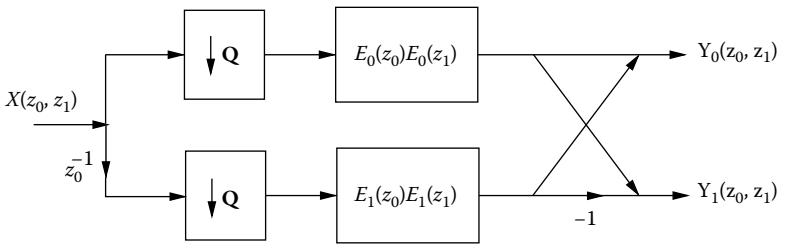


FIGURE 25.11 Efficient polyphase domain implementation of diamond filter bank. The filtering operations are separable.

fan filters. Similarly these QMFs can be applied in successive stages as discussed previously to produce directional filters. To illustrate this point, four directional filters are shown in Figure 25.12. These filters were designed using the Johnston 32D QMF, and are the constituent filters of the eight-band BDFB. Such filter banks can be used to decompose images, an example of which is shown in Figure 25.13.

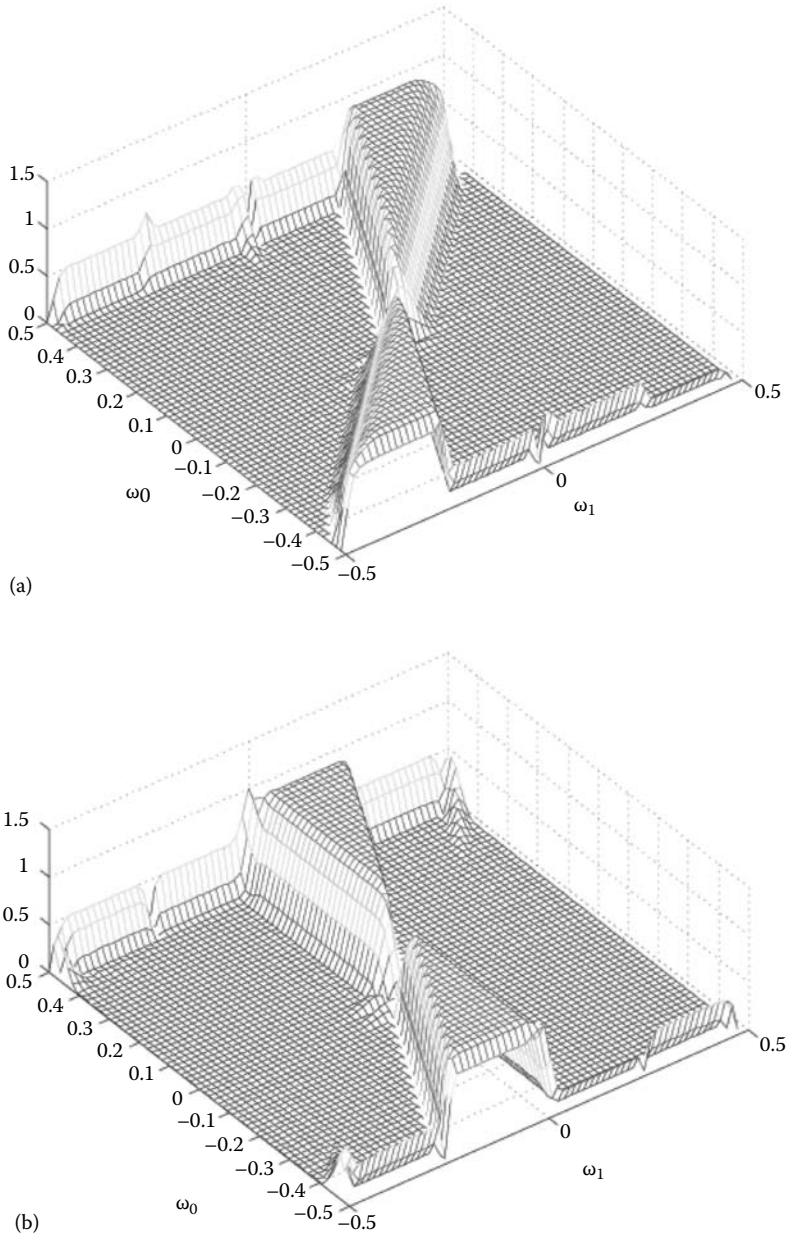


FIGURE 25.12 Examples of magnitude responses of the actual wedge filters constructed with the Johnston 32D FIR filters.

(*continued*)

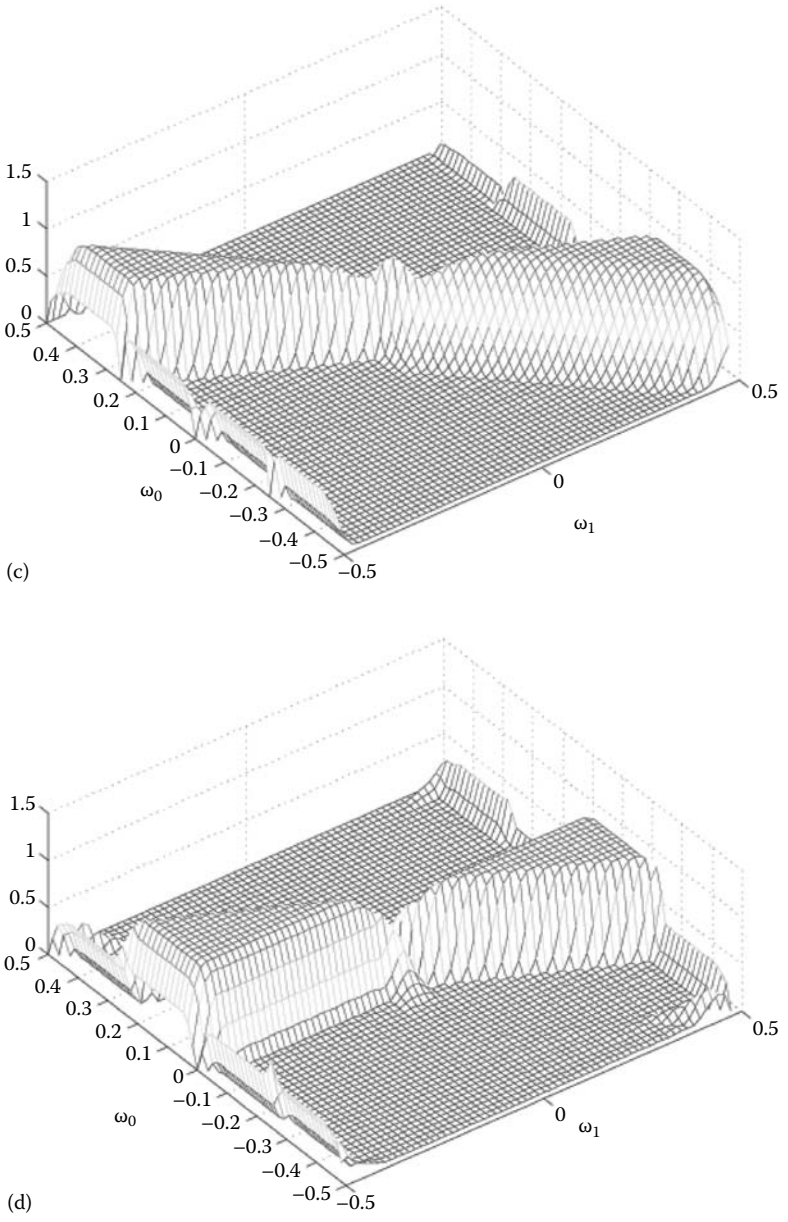


FIGURE 25.12 (continued)

25.5.2 BDFB Design Using Ladder Structures

Ladder structures (also known as lifting schemes [23]) are flexible structures for implementing filter banks and discrete wavelet transforms. The construction of wavelets is achieved by cascading simple filtering elements successively in ladder steps. The PR property is imposed structurally without the explicit need to meet the conditions used in the early PR filter bank design methodology. The construction of biorthogonal systems using ladder networks has been reported in various articles [23–25].

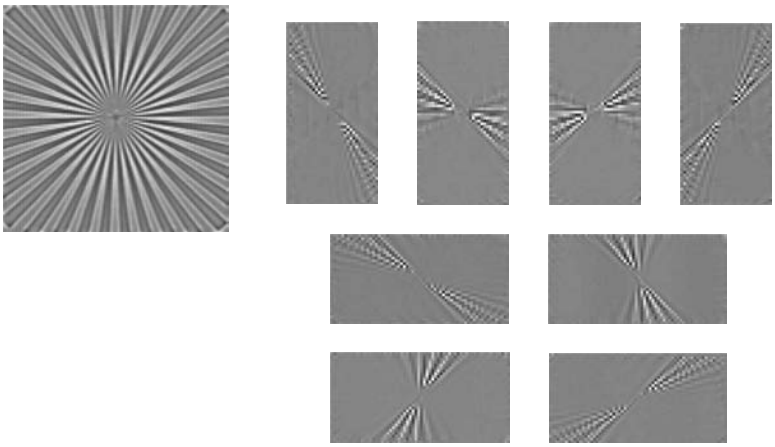


FIGURE 25.13 Example of an eight-band BDFB using a test image with localized directional structure.

Additionally, ladder structures allow straightforward implementation of nonlinear filter banks like integer-to-integer wavelet transforms [26], and adaptation to irregular sampling grids [27].

In this section, ladder structures are used as the framework for the design of digital fan-shaped filters. These structures have been explored for 2-D and multidimensional filter bank implementations [16,17,24,25,28], where the design method involves transforming a 1-D filter to a 2-D filter with a simple mapping. This 1-D to 2-D mapping was introduced by Kim and Ansari [61] and Phoong et al. [16] in the context of a two-stage ladder structure and was extended to a three-stage structure by Ansari et al. [17]. The three-stage design was shown to improve the control of filter frequency responses.

To begin, consider the 1-D case. The 1-D system is designed by constructing the analysis polyphase matrix as the product of ladder steps given by

$$\mathbf{E}(z) = \begin{bmatrix} 1 & 0 \\ -p_2\beta_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & zp_1\beta_1(z) \\ 0 & 1/(1+p) \end{bmatrix} \begin{bmatrix} p_0 & 0 \\ -p\beta_0(z) & 1 \end{bmatrix}, \tag{25.35}$$

which generates the analysis filters

$$H_0(z) = p_0 + p_1 z\beta_1(z^2)A(-z) \quad \text{and} \quad H_1(z) = A(-z) - p_2 H_0(z)\beta_2(z^2), \tag{25.36}$$

where

$$A(z) = \frac{(1 + z\beta_0(z^2))}{1+p} \tag{25.37}$$

and the constants p_0, p_1, p_2 are given by $p_0 = p_1 = (1+p)/2$ and $p_2 = (1-p)/(1+p)$.

This 1-D filter bank structure is illustrated graphically in Figure 25.14. The inherent structure associated with ladder filters implies that the filter bank is biorthogonal and consequently the relationship between the analysis and synthesis filters is given by

$$G_0(z) = -z^{-1}H_1(-z), \quad G_1(z) = z^{-1}H_0(-z), \tag{25.38}$$

where

$H_0(z)$ and $H_1(z)$ are the analysis filters
$G_0(z)$ and $G_1(z)$ are the synthesis filters

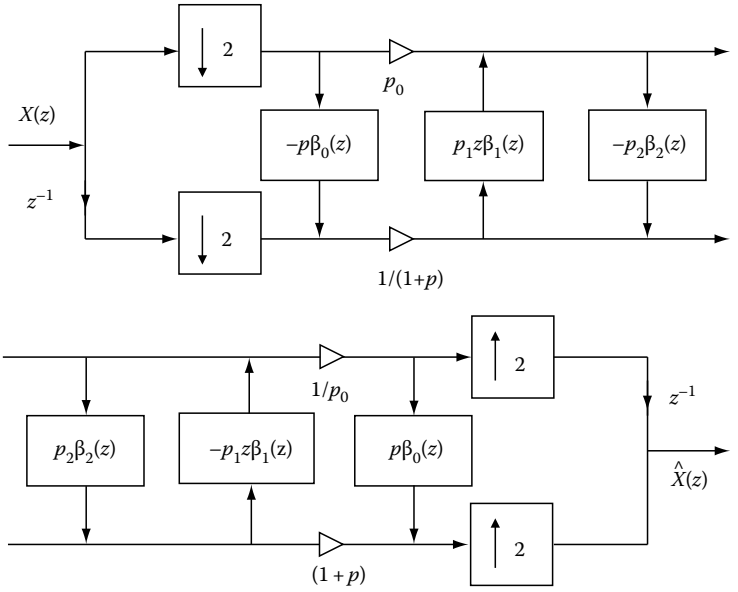


FIGURE 25.14 Analysis–synthesis ladder structure with three ladder steps, as proposed in Ref. [17].

It is easy to show that $H_0(z)$ and $G_0(z)$ are halfband filters. In order to obtain ideal lowpass filters, the following condition for the $\beta_i(z)$ functions must be met

$$\beta_i(e^{j2\omega}) = \begin{cases} e^{j(-2N+1)\omega} & \text{for} \quad 0 \le \omega \le \pi/2, \\ -e^{j(-2N+1)\omega} & \text{for} \quad \pi/2 < \omega \le \pi, \end{cases} \tag{25.39}$$

which implies $\beta_i(e^{j\omega})$ has allpass behavior. An FIR solution that approximates Equation 25.39 is obtained by designing an even length, linear phase function with a magnitude response optimized to approximate unity. This is a very simple requirement that can be satisfied with widely available filter design algorithms. It is possible to design the $\beta_i(z)$ filters separately or to choose the same function by making $\beta(z) = \beta_1(z) = \beta_2(z) = \beta_3(z)$, which significantly simplifies the design procedure. We note that $H_0(z)$ and $H_1(z)$ inherit the linear phase property of $\beta(z)$. Analysis filters obtained by approximating $\beta_i(z)$ using the Parks–McClellan algorithm with $L = 8$ are shown in Figure 25.16a. Additionally, these design constraints can be met by IIR designs, however here we focus on the FIR case [16].

Maximally flat 1-D ladder filters can be obtained using the closed-form Lagrange formula

$$v_k = \frac{(-1)^{N-k-1} \prod_{i=0}^{2N} (N + 1/2 - i)}{2(N-k)!(N-1+k)!(2k-1)}, \tag{25.40}$$

where

$N = L/2$ is the half length of $\beta(z)$

$\beta_{N-k} = \beta_{N+k-1} = v_k$ for $k = 1, 2, \ldots, N$

The maximally flat design method is relevant for the generation of regular biorthogonal wavelets.

For our purposes, the most attractive feature of ladder structures is the implementation of 2-D FFBs using a simple 1-D to 2-D change of variables. This transformation reported in Ref. [16] is applied to the entries of $\mathbf{E}(z)$ expressed in terms of lifting steps. First, the 1-D transfer function $\beta(z)$ is replaced with the

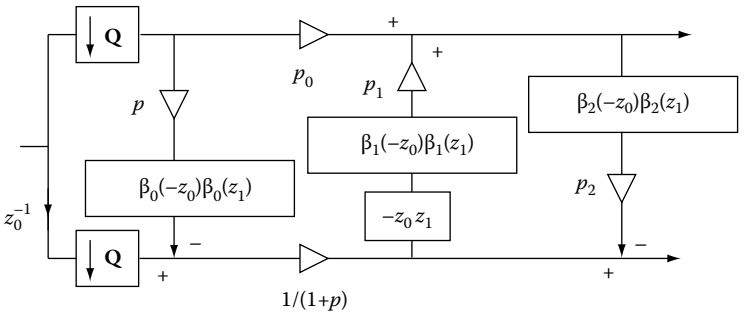


FIGURE 25.15 Ladder structure for the implementation of a 2-D two-channel biorthogonal analysis filter bank.

separable 2-D transfer function $\beta(z_0)\beta(z_1)$. Then, the 1-D delays z^{-1} are replaced with the 2-D delays $z_0^{-1}z_1^{-1}$. The resulting 2-D filters $H_0(z_0, z_1)$ and $H_1(z_0, z_1)$ have diamond-shaped support. In order to obtain the fan-shaped filters $F_0(z_0, z_1)$ and $F_1(z_0, z_1)$ the diamond filters are modulated by π letting $z_0 \rightarrow -z_0$. Following this procedure, the three-stage ladder structure from Figure 25.14 is transformed to the 2-D structure in Figure 25.15. The 2-D fan filter responses $|F_0(z_0, z_1)|$ and $|F_1(z_0, z_1)|$ obtained with this transformation are presented in Figure 25.16 using the same $\beta(z)$ function for all ladder stages.

An important property of this FFB implementation is its low computational complexity. The filtering operations can be implemented in the polyphase domain and only involve separable operations. Moreover since $\beta(z)$ is even length with linear phase, the number of multiplies per output is half the length of $\beta(z)$.

As in Section 25.5.1, the tree-structured BDFB from Figure 25.7 can be realized as a three-stage ladder FFB. The biorthogonal property is preserved by the 1-D to 2-D mapping and the resulting BDFBs have the attractive properties that (1) PR is achieved for FIR systems, (2) the filtering is separable in the polyphase domain and hence is efficient, (3) the 2-D filter quality can be controlled by a single function, (4) any 1-D filter design technique can be employed to generate $\beta(z)$, and (5) any set of

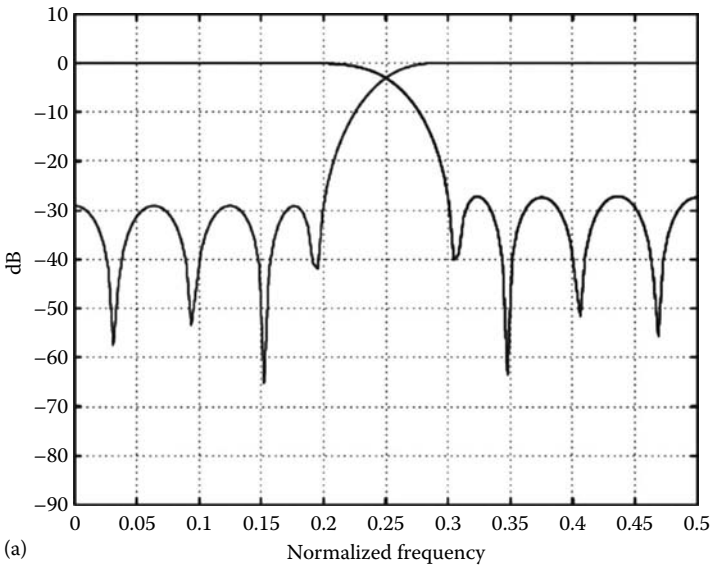


FIGURE 25.16 Magnitude response of the analysis fan filters obtained with a three-stage ladder structure. (a) Lowpass and highpass 1-D filter prototypes.

(*continued*)

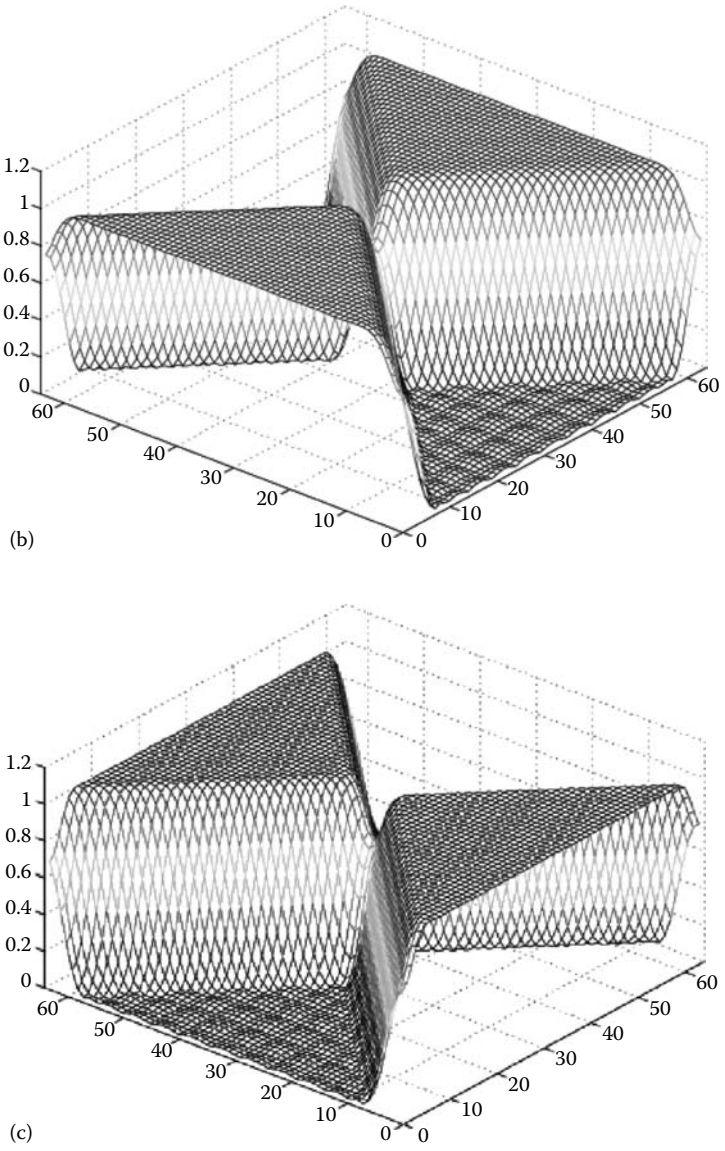


FIGURE 25.16 (continued) (b) and (c) Corresponding 2-D fan shaped filters.

biorthogonal wavelet filters can be used to implement the BDFB by factoring them into ladder steps and applying the 1-D to 2-D mapping.

25.6 Undecimated Directional Filter Bank

In many applications like pattern recognition and image enhancement, shift invariance is considered an important property. To achieve shift invariance the subbands cannot be maximally decimated. In the case of the 2-D discrete wavelet transform (DWT), shift invariance is achieved by removing the downsampling operations from the filter bank structure. This is the so-called undecimated DWT. It can be shown that PR can still be preserved under conditions of partial decimation and in the complete

absence of decimation [29]. While the undecimated representation is shift invariant, the coefficients are not decorrelated anymore. Thus, there is a trade-off.

For a number of applications it is desirable to have an undecimated version of the BDFB. A shift-invariant BDFB can be obtained from Figure 25.8 by removing the downsampling matrices D_ℓ and using the filters $G_\ell(\omega)$ as defined by Equation 25.24. This undecimated BDFB has a high computational cost since the filters are nonseparable and, in order to achieve good frequency selectivity, filters generally need to have order of 120×120 or higher. Although frequency domain implementations are possible, the boundary effects are more difficult to control.

Fortunately, there are ways to address efficiency. The BDFB tree structure from Figure 25.7 can be modified to obtain an UDFB. The derivation of the UDFB is explained in detail in Refs. [20,30]. The UDFB is obtained from the BDFB through the use of multirate identities and results in a modified tree structure. To illustrate this, the UDFB tree structure for the eight-band case is shown in Figure 25.19. In this case two building blocks are needed, the undecimated FFB (UFFB) and an undecimated checkerboard filter bank (UCFB). The UCFB is a two-channel structure with the complementary passbands and stopbands as shown in Figure 25.17.

For additional efficiency the UFFB can be implemented in a ladder structures as shown in Figure 25.18 where

$$\mathbf{Q} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}.$$

The UCFB is obtained from the UFFB structure by simply removing the resampling operations in Figure 25.18. This relationship has been described in Ref. [18] where fan filters are derived from checkerboard

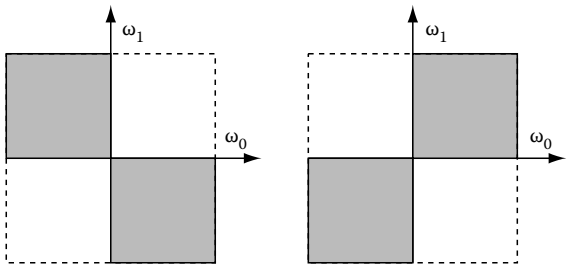


FIGURE 25.17 Ideal magnitude responses for the two-channel UCFB.

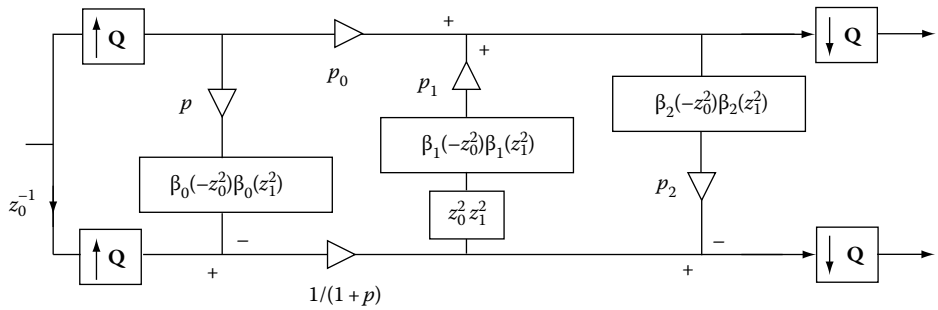


FIGURE 25.18 Ladder structure implementation for the UFFB.

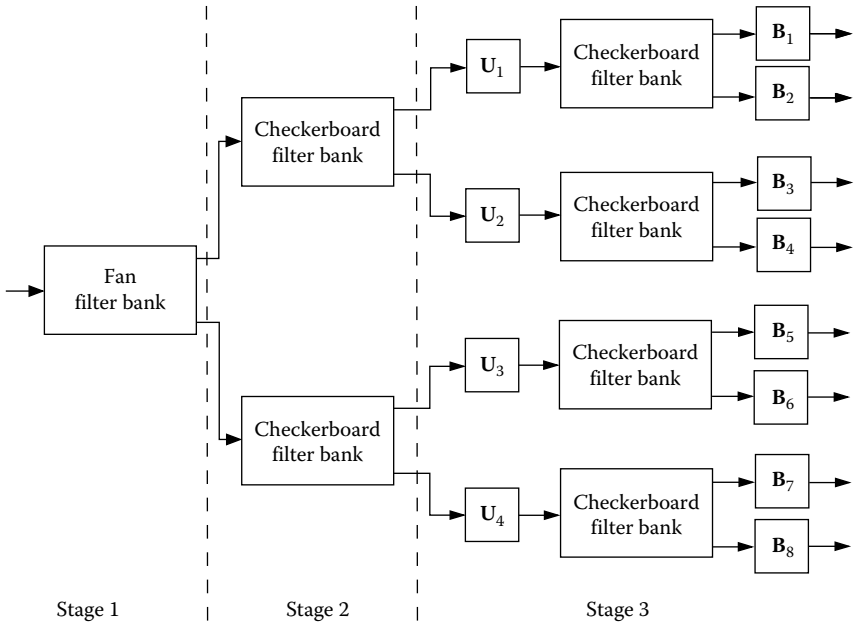


FIGURE 25.19 Implementation of an eight-band UDFB using a tree structure with UFFBs and UCFBs.

filters by a change-of-variables transformation. Finally, the filtering operations in Figure 25.18 remain separable and the $\beta_i(z)$ functions are changed to $\beta_i(z^2)$, which implies that each function has been upsampled by a factor of 2. As a result, low computational complexity of the BDFB is retained in the undecimated case.

The overall UDFB tree structure is shown in Figure 25.19. Similar to the maximally decimated case, it is possible to generate a 2^N-band UDFB by cascading UFFB and UCFB structures. The eight-band UDFB is implemented with a UFFB in the first stage and UCFBs in the remaining stages. UFFBs could be used in the third stage, but UCFBs are more efficient. As before, the third stage requires the use of unimodular resampling matrices $\mathbf{U}_i$. A possible set of matrices is given by

$$\mathbf{U}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{U}_3 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{U}_4 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

Finally, the matrices $\mathbf{B}_i$ are used in order to reestablish a rectangular sampling geometry. In this case $\mathbf{B}_i = \mathbf{U}_i^{-1}$. An example of an eight-band UDFB decomposition is shown in Figure 25.20. Note that all the subbands have the same size and that the decomposition is shift invariant.

25.7 Bamberger Pyramids

Other image decompositions like the 2-D DWT, the steerable pyramid [31], the complex-valued wavelet transform [32], and 2-D Gabor representations [33,34] separate information across different resolutions as well as directions. This multiresolution (MR) process could be an implicit part of the decomposition (e.g., the DWT), or could be implemented as a separate structure (e.g., the steerable pyramid [31]). In the latter case, we say that the decomposition is polar-separable, implying that a radial frequency decomposition (i.e., a pyramid) is performed independently of the angular decomposition.

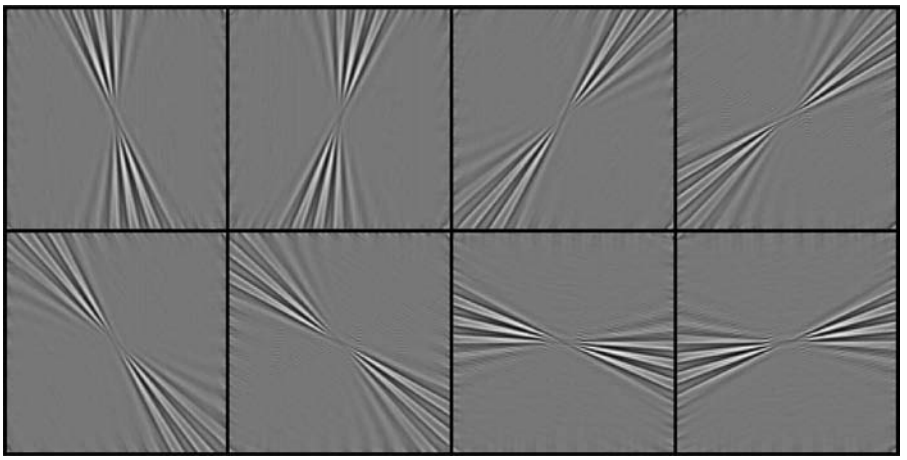


FIGURE 25.20 Example of an eight-band UDFB decomposition.

Given that many problems of interest in image processing and image analysis require MR directional analysis, extending the theory of the BDFB to polar-separable representations is desirable. As it turns out polar-separable versions of the BDFB and UDFB can be constructed and are straightforward to derive.

Different types of radial pyramids can be considered [30], but here we focus on a couple of examples associated with the BDFB. The simplest Bamberger pyramid consists of a "lowpass–highpass" decomposition. The image is first filtered with a lowpass filter $L_{\omega_c}(z_0, z_1)$ with cutoff frequency ω_c. A high frequency component is obtained by subtracting the filtered version from the original image. The resulting high frequency channel is decomposed into directional subbands using the BDFB or the UDFB, depending on the application.

Another straightforward Bamberger pyramid can be formed by combining a *J*-level Laplacian pyramid with the BDFB [4,30]. The analysis structure is presented in Figure 25.21. At the high- and mid-frequency levels the subbands can be processed with the BDFB. If required, the UDFB can be used in

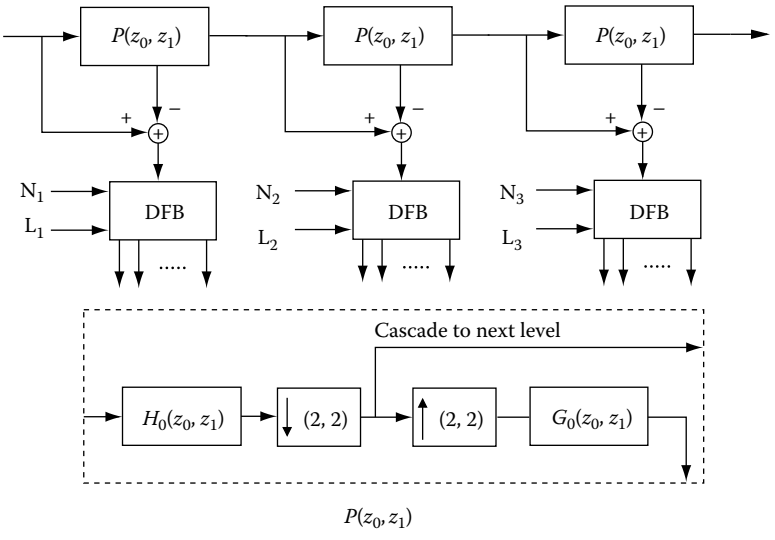


FIGURE 25.21 Bamberger pyramid using the Laplacian pyramid structure combined with the BDFB.

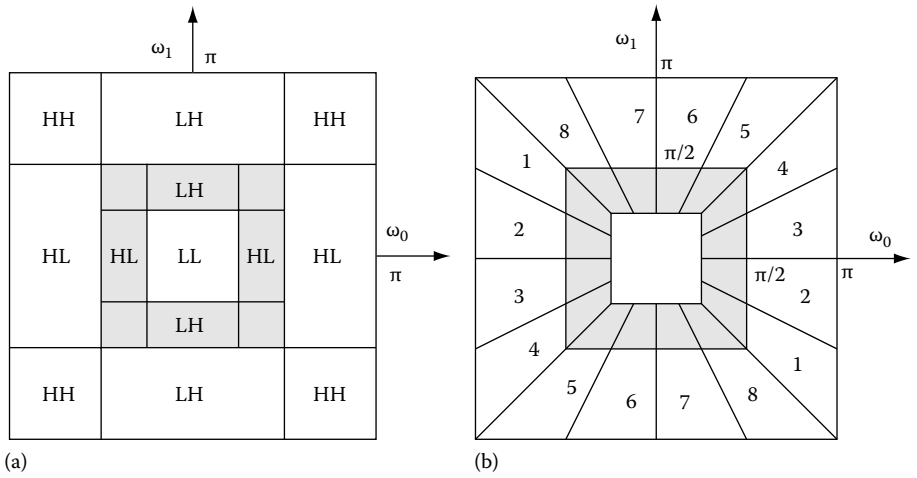


FIGURE 25.22 Comparison of the frequency plane partitioning of the 2-D separable DWT (a) and the proposed directional pyramids (b).

place of the BDFB. More generally the directional decomposition can be designed independently at each resolution. For instance, the number of subbands and the order of the $\beta_i(z)$ filters can be chosen independently. Since the polar components of Bamberger pyramids are invertible, it is easy to see that the overall system is PR.

For the maximally decimated case, the Laplacian-BDFB structure increases the data by approximately a factor of $\frac{4}{3}$. For the cases where the UDFB is used, the increase in data is significantly larger. For the case where all radial bands are decomposed into an N-band UDFB, the data increase is given by a factor of $\frac{4N}{3}$. Where shift invariance is needed at all resolutions and orientations, we can remove all downsampling operations from the Laplacian pyramid and modify the lowpass kernels resulting in $H_0(z_0^{2^j}, z_1^{2^j})$ and $G_0(z_0^{2^j}, z_1^{2^j})$ at each resolution level, where $j = 0, 1, \ldots, J-1$. When combined with the BDFB, the resulting Bamberger pyramid is overcomplete by a factor of J. If the UDFB is used, the data increases by a factor of $N(J-1)+1$.

A comparison of the frequency plane partitioning obtained with the Bamberger pyramids described above and the traditional separable 2-D DWT (and its undecimated version) is shown in Figure 25.22. The 2-D DWT has limited angular sensitivity (mixed-diagonal, horizontal, and vertical directions), while Bamberger pyramids can have 2^N directional subbands.

25.8 Applications

The BDFB has proven to be a useful tool for different problems in image processing and analysis. In this section, we present a summary of different applications reported in the literature. Detailed development and discussion is out of the scope of this work and can be obtained from the references.

25.8.1 Texture Analysis and Segmentation

Texture is an important characteristic present in many images that often contains useful information, which can be exploited in analysis applications. Multichannel methods are among the best in exploiting texture information and include Gabor decompositions, DCTs, DSTs, wavelets, wavelet packets, and dual-tree complex wavelets [35–37]. Decompositions based on the BDFB provide some of the best results for texture classification and segmentation reported in the literature [5,19,30,38,39].

Texture information is distributed across different scales and orientations according to the structural and statistical properties of the image and thus useful texture components can often be separated with a multichannel transform. A simple way to obtain a texture characterization is to measure the subband energies and form a feature vector **f**. Such feature vectors can be used in combination with a classifier or clustering algorithm for recognition or segmentation.

In texture classification the objective is to assign an unknown texture sample to a particular class within a set of known texture classes. The Bayes minimum distance classifier has been used extensively in texture classification [29,35,38,40,41]. This classifier is a supervised scheme that requires the estimation of the mean feature vector and feature covariance matrix for each class. The BDFB was evaluated for texture classification using the Brodatz data set [42] and the methodology described in Ref. [35]. BDFB-based classifiers provide some of the best results for this data set, achieving 99.62% correct classification using 10 features [30,38]. The BDFB results are comparable to the tree-structured wavelet transform and discrete sine transform reported as the best performers in Ref. [35].

In practice, classification systems have to deal with geometrical distortions like rotation. For this problem, rotation-invariant features can be attractive. Multichannel decompositions that split the frequency plane into directionally selective channels can achieve rotation invariance using a DFT-encoding step [39,43,44]. In this approach, a feature vector **f** is formed from the set of directional subbands at each resolution. Then, a DFT is computed for the features at each resolution separately. The resulting DFT coefficients are grouped in a vector **F**. A rigid texture rotation is encoded in **f** as a circular shift which in **F** is encoded as a complex exponential factor. A rotation-invariant feature set is obtained by taking the magnitude of **F** and discarding half of its components.

For instance, in Ref. [38] a Bamberger pyramid is used as the front end of a rotation-invariant classification system. The pyramid consists of an undecimated J-level Laplacian pyramid and an N-band BDFB which achieve the frequency plane partitioning shown in Figure 25.22b. A rotation-invariant feature set is obtained after applying DFT encoding to each pyramid resolution separately. Using the Bayes distance classifier, the system was tested with the data set introduced by Haley and Manjunath [41], which consists of 13 texture classes, each scanned at rotations of $0°, 30°, 60°, 90°, 120°$, and $150°$. The BDFB-based system obtained correct classification rates of 96.96% with 15 features. These results compare favorably with those reported by Haley and Manjunath [41] using a class of 2-D analytic Gabor wavelets; in their system they achieve 96.8% correct classification using a feature vector with 204 components.

Multichannel schemes have also been applied to texture segmentation. In this case a feature vector is used for each pixel in order to capture the local rather than global texture cues. A generic segmentation system is presented in Figure 25.23. The nonlinear operations are used as limiters that control the presence of outliers. For each spatial location (n, m) the local energy is estimated across all subbands. A weighted sum of squared coefficients is typically used in this step, usually composed of a squaring operation followed by a Gaussian smoothing operator. The local energy estimates are then grouped to form feature vectors. In the final step, a classifier is used to produce a segmentation map that assigns a label to each (n, m) location. Bamberger pyramids are well suited for this purpose [19,30]. A segmentation example is presented in Figure 25.24 that corresponds to an undecimated Bamberger pyramid with a redundancy factor of JN where $J = 4$ and $N = 8$. The learning vector quantization (LVQ) algorithm from Kohonen [45] was selected as the classifier following the work in Ref. [36]. The segmentation error

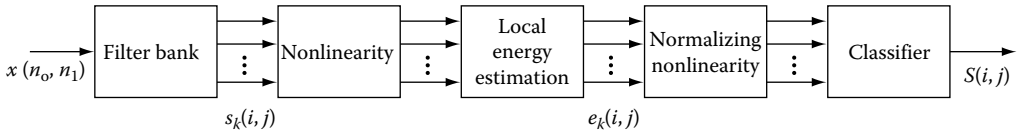


FIGURE 25.23 Classical segmentation system based on multichannel filtering.

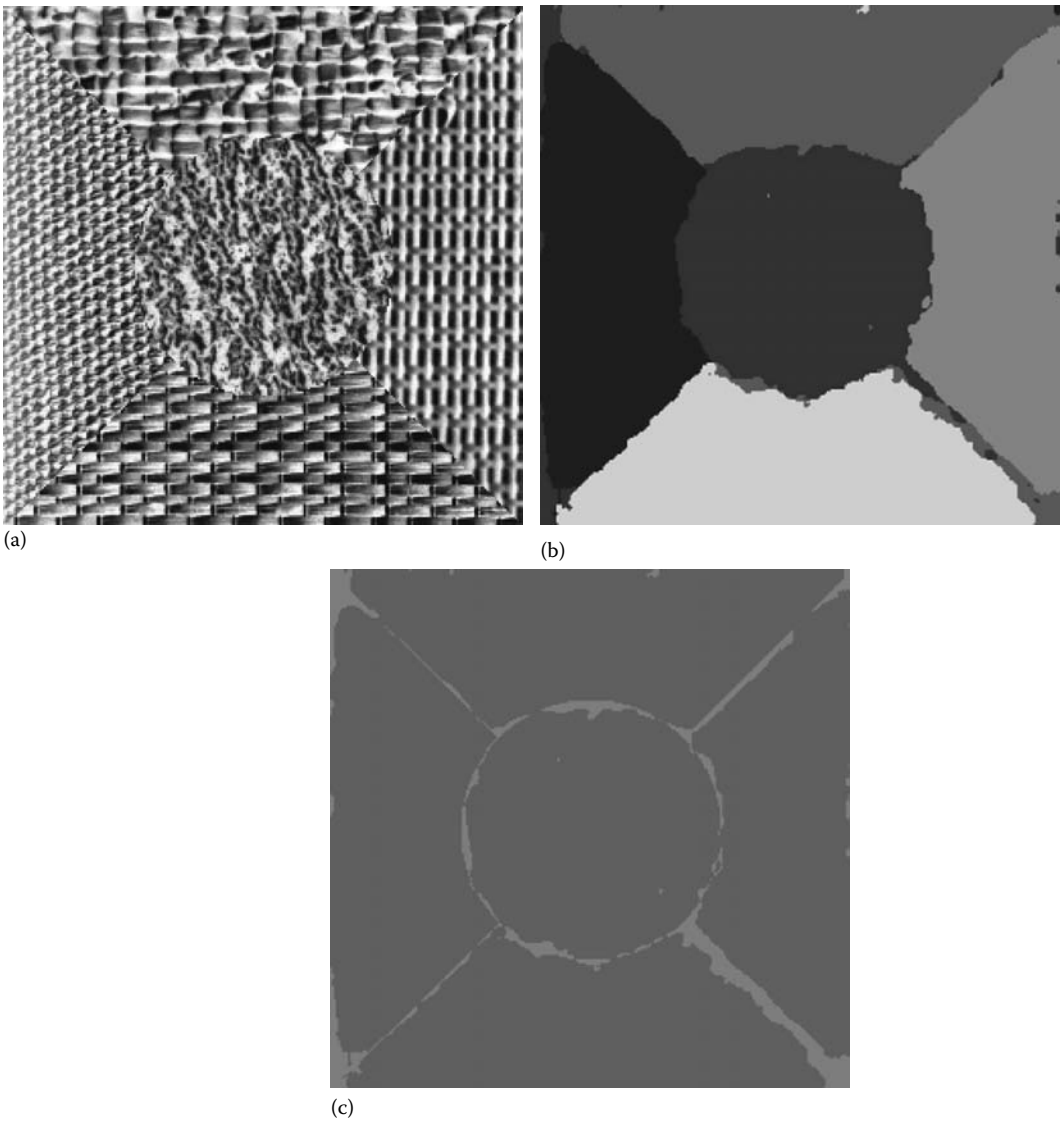


FIGURE 25.24 Example of texture segmentation using Bamberger pyramids. (a) Five texture mixture. (b) Segmentation map identifying the five texture classes. (c) Segmentation error, localized mainly along the texture boundaries.

for this example is 4.67%. A comparative study [30] showed that on average Bamberger pyramids provide the best performance compared to other multichannel decompositions like the DCT, DWT, undecimated uniform filter banks [36], and the DT-CWT [37].

25.8.2 Image Denoising Using Bamberger Pyramids

Denoising is a popular application in which wavelets have been employed. The goal in denoising is to enhance the quality of an image that has been contaminated by additive noise, often assumed to be Gaussian white noise. The basic wavelet denoising procedure consists of first decomposing the noisy image into subbands, after which each subband is processed by a subband-specific nonlinearity. In the last step, the processed subbands are recombined in a synthesis transform to reconstruct a noise

suppressed version of the noisy original. Since the forward and inverse transforms amount to an identity system, it is evident that the noise suppression can be directly attributed to the nonlinear operations performed on the subbands.

Typically, the nonlinearity is a shrinking or coring operation which takes a subband coefficient and modifies its magnitude. Small coefficients tend to be suppressed and large coefficients maintain their values. A commonly used operator for images is soft thresholding, whereby a subband coefficient $x(n_0, n_1)$ is modified to $\hat{x}(n_0, n_1) = \text{sgn}(x(n_0, n_1))(|x(n_0, n_1)| - T)$ when the coefficient magnitude is greater than T, and the coefficient is set to zero otherwise. The value of the threshold is set explicitly within each subband or adapted individually to each coefficient based on some criterion, such as energy or statistical characteristics [46,47].

This approach was first explored with maximally decimated filter banks and later with undecimated transforms. It turns out that better results are generally achieved with undecimated (shift-invariant) decompositions.

Since the Bamberger pyramid, whose decomposition is shown in Figure 25.22, provides both radial and angular subband resolution, one might imagine that it would perform well in a denoising application [47,48]. In fact such is the case as shown by the comparisons in Figure 25.25. Bamberger pyramids can provide better directional selectivity across resolutions along with shift invariance when the UDFB is used. Shown in Figure 25.25c is the denoising result for an undecimated Bamberger pyramid with frequency plane partitioning similar to the Steerable pyramid [31]. The midband pyramid levels in this particular system are decomposed with an eight-band UDFB.

For threshold selection, the spatially adaptive wavelet thresholding (SAWT) algorithm [46] was used, where a threshold is computed for each subband coefficient using local statistics under a Bayesian framework.



(a) (b) (c)

FIGURE 25.25 Denoising results using Lenna (a) Image with additive white Gaussian noise with $\sigma = 22.5$. (b) Denoised image using the UDWT. (c) Denoised image using Bamberger pyramids.

Also considered is a similar system using the undecimated DWT (UDWT) instead of the UDFB. The denoised image from the DWT system is shown in Figure 25.25b. Both the DWT and DFB systems have good performance as can be seen in contrast to the original image with visible additive noise shown in Figure 25.25a. Both UDWTs and DFBs are good choices for denoising applications.

25.8.3 Fingerprint Enhancement and Recognition

The analysis of fingerprint images can be a challenging problem. The obvious goal is to identify a fingerprint image as belonging to a particular individual from among a huge set of candidates. DFBs can play a role in the recognition process. Park et al. [49] proposed a new image-based fingerprint matching method that is robust to diverse rotations and translations of an input fingerprint. This scheme does not require minutiae extraction, as is the typical approach. Rather, the scheme of Park represents the fingerprint in terms of directional energies. The area within a certain radius around a reference point is used as a region of interest (ROI) for feature extraction. Fingerprint features are then extracted from the ROI using the BDFB. More specifically, the ROI for each subband is divided into blocks and directional energy values are calculated for each block. Only the blocks with dominant energy are retained, while the rest of the directional energies are set to zero, which effectively treats them as noise. As part of the matching process, rotational and translational alignments between the input and template are performed through a normalized Euclidean distance. Experimental results reported by Park et al. [49] demonstrate that the proposed DFB method has comparable verification accuracy to the other leading techniques while having faster processing speed and greater robustness to positional variation.

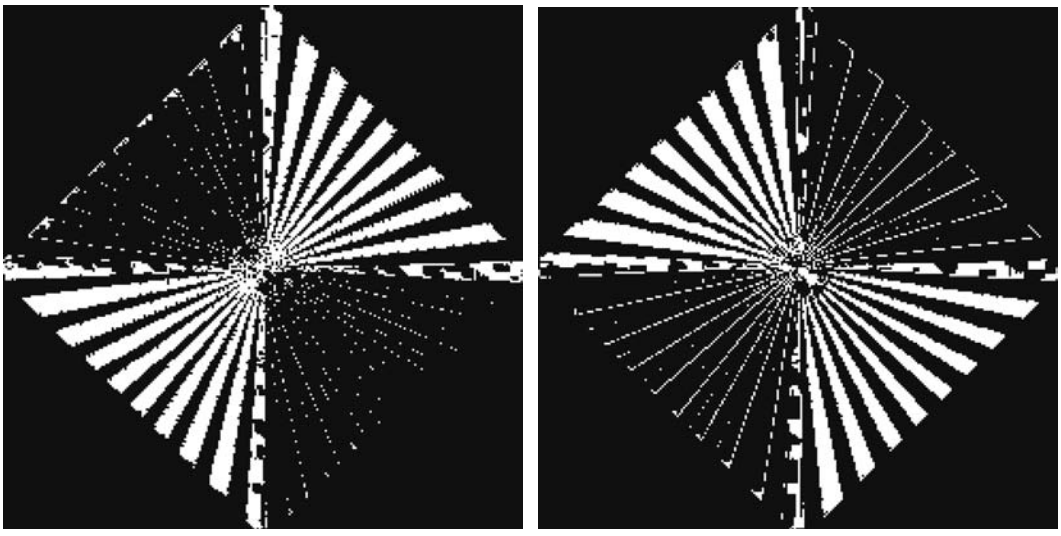
25.8.4 Iris Recognition

Another biometric identification system that has received attention in recent years uses iris patterns. The physiological characteristics of the iris provides a biometric difficult to modify or reproduce by synthetic methods. The work pioneered by Daugman [50] generates an iris code based on localized energy measurements of the iris texture using 2-D multiscale directional Gabor filters. As is well known the implementation of Gabor filter banks is computationally expensive and the frequency selectivity is limited. The BDFB has been explored as an alternative to Gabor filter banks. Helen Sulochana and Selvan [51] reported on a system that produces a feature vector from the BDFB subbands by dividing each subband into 9×9 blocks and calculating the energy from each block. The energies are then thresholded to form a binary iris signature. The BDFB-based system has similar performance to the leading systems, but is less complex and faster.

25.8.5 Finite-Field DFBs

BDFBs can also be applied to binary images using binary arithmetic. An important property of the ladder structure is that it allows for a straightforward implementation of nonlinear filter banks with PR. Since ladder steps are added in the analysis and subtracted in lock step in the synthesis, exact reconstruction is preserved regardless of the kind of ladder step operation being performed. Consequently nonlinear operations can be accommodated, such as quantization, rounding, rank order filtering, and so on. For instance, true integer-to-integer wavelet transforms are possible by adding rounding operations after each filtering step.

Since FFBs can be realized with ladder structures, nonlinear BDFBs can be constructed. A particular case of interest is processing bilevel or binary images with binary arithmetic. Randolph and Smith [52] reported on a binary BDFB that produces directional subbands that are also binary. To achieve binary valued subbands, a threshold is applied after filtering by $\beta(Z_0)\beta(Z_1)$ in the two-stage ladder FFB. The threshold maps negative values to zero and positive values to one. All remaining operations in the ladder



(a) (b)

FIGURE 25.26 Output of a two-channel binary BDFB. The subbands are binary valued.

structure are implemented using modulo-two arithmetic. An example of the output for a two-channel binary BDFB is shown in Figure 25.26 for a binary version of the image in Figure 25.13.

The binary BDFB has been explored in the context of printed character enhancement and character recognition. Specifically, it has been applied to rotation and scale-invariant character recognition [53] and enhancement of low-resolution documents [54,55].

25.8.6 Velocity Selective Filter Banks

Motion analysis is a critical element in video compression, object tracking, computer vision, and situation analysis. Popular approaches for analyzing motion have included optical flow and block matching techniques. An alternative, explored more recently, has been the use of velocity tuning filters where motion is determined by looking at the spatiotemporal distribution of energy along planes in the 3-D frequency domain. As objects move, they trace a trajectory in time captured by their spatial displacement from frame to frame. In Ref. [56], object trajectories have been extracted using a 3-D continuous wavelet transform (CWT). This transform can be tuned to find motion using spatial translation, temporal translation, scale, velocity magnitude, and velocity orientation. Such flexibility comes with a high computational cost. However, velocity selective filter banks (VSFB) have been reported in Refs. [11,57] that provide an attractive alternative to the oversampled CWT. The implementation boils down to a 3-D generalization of the BDFB which requires the use of full-rate or undecimated subbands. The VSFB is spatiotemporally separable meaning that the BDFB is computed for each individual frame and then subbands along a specific orientation across different frames are grouped and processed temporally using the BDFB. The VSFB produces 3-D wedges as depicted in Figure 25.27a. This wedge captures motion for objects moving along the directions over the fixed velocities captured by the temporal aperture of the wedge. The CWT is capable of additionally separating information across scales as shown in Figure 25.27b. Compared to the CWT, the VSFB is constrained to the speed and position resolutions determined by the number of directional subbands. On the other hand, its computational complexity is an order of magnitude lower than the CWT.

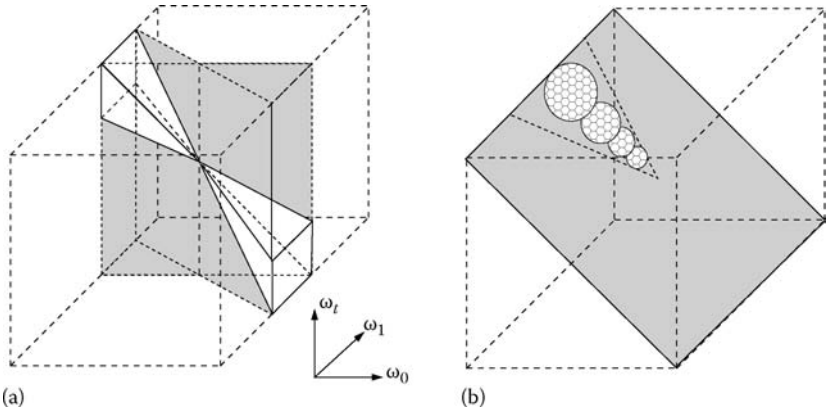


FIGURE 25.27 Comparison of the 3-D frequency supports generated by velocity tuning filters (a) VSFB (b) CWT.

25.9 Closing Remarks

DFBs have many properties that are both interesting and important. They can provide decompositions with a wide variety of angular resolutions, the filters can be designed to have good passband characteristics, and they can be designed to have exact reconstruction. Furthermore, DFBs can accommodate a range of decimation factors for the subbands, from maximally decimated to undecimated. This flexibility can be important in applications where either a compact representation is desirable, like image compression, or applications where shift invariance is deemed important, as is the case with image denoising.

This chapter is not intended as a comprehensive review of directional transforms, but rather an overview of the basic multidimensional theory, the DFB tree structure, the design and efficient implementation of the filter bank, and some applications that have been considered. The interested reader seeking a more comprehensive discussion of this material is directed to the references provided at the end of this chapter.

References

1. Roberto H. Bamberger and Mark J. T. Smith, Filter banks for directional filtering, in *Proc. 1989 IASTED Int. Conf. Adaptive Knowledge-Based Control Signal Process.*, Honolulu, Hawaii, August 1989, pp. 105–109.
2. Roberto H. Bamberger, The directional filterbank: A multirate filter bank for the directional decomposition of images, PhD thesis, Georgia Institute of Technology, November 1990.
3. Roberto H. Bamberger and Mark J. T. Smith, A filter bank for the directional decomposition of images, *IEEE Trans. Signal Process.*, 40(4), 882–893, April 1992.
4. M. N. Do and M. Vetterli, Pyramidal directional filter banks and curvelets, in *Proc. IEEE Int. Conf. Image Process.*, Thessaloniki, Greece, October 2001, pp. 158–161.
5. K. O. Cheng, N. F. Law, and W. C. Siu, Multiscale directional filter bank with applications to structured and random texture retrieval, *Pattern Recognit.*, 40(4), 1182–1194, April 2007.
6. Y. Lu and M. N. Do, Multidimensional directional filter banks and surfacelets, *IEEE Trans. Image. Process.*, 16, 918–931, April 2007.
7. R. Eslami and H. Radha, A new family of nonredundant transforms using hybrid wavelets and directional filter banks, *IEEE Trans. Image Process.*, 16, 1152–1167, April 2007.
8. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, NJ, 1933.

9. E. Viscito and J. P. Allebach, The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices, *IEEE Trans. Circuits Syst.*, 38(1), 29–41, January 1991.
10. Sang-Il Park, Mark J. T. Smith, and Russell M. Mersereau, A new directional filter bank for image analysis and classification, in *Proc. 1999 IEEE Int. Conf. Acoust., Speech Signal Process*., October 1999, vol. 2, pp. 1286–1290, Chicago, IL.
11. Sang-Il Park, New directional filter banks and their application in image processing, PhD thesis, Georgia Institute of Technology, Atlanta, GA, November 1999.
12. I. Shah and A. Kalker, Theory and design of multidimensional QMF sub-band filters from 1-d filters using transforms, in *Proc. 4th Int. Conf. Image Process Appl.*, April 1992, pp. 474–477.
13. T. Chen and P. P. Vaidyanathan, Multidimensional multirate filters and filter banks derived from one dimensional filters, *IEEE Trans. Signal Process.*, 41(5), 1749–1765, May 1993.
14. D. Tay and N. Kingsbury, Flexible design of multidimensional perfect reconstruction FIR 2-band filters using transformations of variables, *IEEE Trans. Image Process*., 2, 466–480, October 1993.
15. J. Kovacevic and M. Vetterli, Nonseparable two- and three-dimensional wavelets, *IEEE Trans. Signal Process*., 43(5), 1269–1273, May 1995.
16. See-May Phoong, Chai W. Kim, P. P. Vidyanathan, and Rashid Ansari, A new class of two-channel biorthogonal filter banks and wavelet bases, *IEEE Trans. Signal Process*., 43(3), 649–665, March 1995.
17. Rashid Ansari, Chai W. Kim, and M. Dedovic, Structure and design of two-channel filter banks derived from a triplet of halfband filters, *IEEE Trans. Circuits Syst.-II: Anal. Dig. Signal Process.*, 46(12), 1487–1496, December 1999.
18. R. Ansari, Efficient IIR and FIR fan filters, *IEEE Trans. Circuits Syst.*, CAS-34(8), 941–945, August 1987.
19. J. G. Rosiles and M. J. T. Smith, Texture segmentation using a biorthogonal directional decomposition, in *Systematics, Cybernetics and Informatics 2000*, July 2000, Orlando, FL.
20. J. G. Rosiles and M. J. T. Smith, A low complexity undecimated directional image decomposition, in *ICIP*, 2003, vol. 1, pp. 1049–1052.
21. J. D. Johnston, A filter family designed for use in quadrature mirror filter banks, in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*., April 1980, 291–294.
22. Mark J. T. Smith and Steven L. Eddins, Analysis/synthesis techniques for subband image coding, *IEEE Trans. Acoust. Speech Signal Process*., 38(8), 1446–1456, August 1990.
23. I. Daubechies and W. Sweldens, Factoring wavelet transforms into lifting steps, *J. Fourier Anal. Appl.*, 4(3), 245–267, 1998.
24. T. A. C. M. Kalker and I. A. Shah, Ladder structures for multidimensional linear phase perfect reconstruction filter banks and wavelets, in *SPIE Visual Communications and Image Processing'92*, 1992, vol. 1818, pp. 2–20.
25. L. Tolhuizen, H. Hollmann, and T. A. C. M. Kalker, On the realizability of biorthogonal, m-Dimensional two-band filter banks, *IEEE Trans. Signal Process*., 43(3), 640–648, March 1995.
26. A. R. Calderbank, Ingrid Daubechies, Wim Sweldens, and Boon-Lock Yeo, Lossless image compression using integer to integer wavelet transforms, in *Proc. IEEE Int. Conf. Image Process*., Washington, DC, October 1997, vol. 1, pp. 596–599.
27. V. Delouille, M. Jansen, and R. von Sachs, Second generation wavelet methods for denoising of irregularly spaced data in two dimensions, *Signal Process.*, 86(7), 1435–1450, 2006.
28. D. B. H. Tay and M. Palaniswami, A novel approach to the design of the class of triplet halfband filter banks, *IEEE Trans. Circ. Sys.-II*, 51(7), 378–383, July 2004.
29. M. Unser, Texture classification and segmentation using wavelet frames, *IEEE Trans. Image Process*., 4(11), 1549–1560, November 1995.
30. Jose Gerardo Gonzalez Rosiles, Image and texture analysis using biorthogonal angular filter banks, PhD thesis, Georgia Institute of Technology, July 2004.
31. E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, Shiftable multi-scale transforms, *IEEE Trans. Inf. Theory Special Issue Wavelets*, 38(2), 587–607, March 1992.

32. Nick G. Kingsbury, The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement, in *Proc. Euro. Signal Process. Conf., EUSIPCO 98*, September 1998, vol. 1, pp. 319–322, Rhodes.
33. Alan C. Bovik, M. Clark, and W. S. Geisler, Multichannel texture analysis using localized spatial filters, *IEEE Trans. PAMI*, 12, 55–73, 1990.
34. A. K. Jain and F. Farrokhnia, Unsupervised texture segmentation using gabor filters., *Pattern Recognit.*, 24(12), 1167–1186, 1991.
35. Tianhorng Chang and C.-C. Jay Kuo, Texture analysis and classification with tree-structured wavelet transform, *IEEE Trans. Image Process*. 2(4), 429–441, 1993.
36. T. Randen and J. H. Husøy, Filtering for texture classification: A comparative study, *IEEE Trans. PAMI*, 21(4), Apr. 1999.
37. Peter De Rivaz, Complex wavelet based image analysis and synthesis, PhD thesis, University of Cambridge, Cambridge, U.K., 2000.
38. J. G. Rosiles and M. J. T. Smith, Texture classification with a biorthogonal directional filter bank, in *ICASSP*, 2001, 3, 1549–1552.
39. J. G. Rosiles, M. J. T. Smith, and R. M. Mersereau, Rotation invariant texture classification using bamberger pyramids, in *IEEE Int. Conf. Multimedia Exp.*, July 2005.
40. M. Unser, Local linear transforms for texture measurements, *Signal Process.*, 11(1), 61–79, July 1986.
41. G. M. Haley and B. S. Manjunath, Rotation-invariant texture classification using a complete space-frequency model, *IEEE Trans. Image Process*., 2(8), 255–269, February 1999.
42. P. Brodatz, *Textures, A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.
43. H. Greenspan, S. Belongie, R. Goodman, and P. Perona, Rotation invariant texture recognition using a steerable pyramid, in *IEEE Proc. Int. Conf. Image Process.*, Jerusalem, Israel, October 1994.
44. P. R. Hill, D. R. Bull, and C. N. Canagarajah, Rotationally invariant texture features using the dual-tree complex wavelet transform, in *IEEE Proc. Int. Conf. Image Process.*, September, Vancouver, Canada, 2000.
45. T. Kohonen, The self-organizing map, *Proc. IEEE*, vol. 78, pp. 1464–1480, September 1990.
46. S. Grace Chang, Bin Yu, and Martin Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising, *IEEE Trans. Image Process*., 9(9), 1522–1531, September 2000.
47. J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, Image denoising using gaussian scale mixtures in the wavelet domain, *IEEE Trans. Image Process*., 12(11), 1338–1351, November 2003.
48. A. Achim, P. Tsakalides, and A. Bezerianos, Sar image denoising via bayesian wavelet shrinkage based on heavy-tailed modeling, *IEEE Trans. Geo. Rem. Sen.*, 41(8), 1773–1784, August 2003.
49. C.-H. Park, J.-J. Lee, M. J. T. Smith, S.-I. Park, and K.-H. Park, Directional filter bank-based fingerprint feature extraction and matching, *IEEE Trans. Circuits Syst. Video Tech.*, 14(1), 74–85, January 2004.
50. J. Daugman, High confidence visual recognition of persons by a test of statistical independence., *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(11), 1148–1161, November 1993.
51. C. Helen Sulochana and S. Selvan, Iris feature extraction based on directional image representation, *ICGST Int. J. Graph., Vis. Image Process*., 06, pp. 55–62, 2007.
52. T. R. Randolph and M. J. T. Smith, A directional representation for binary images, in *Ninth DSP Workshop (DSP 2000)*, October 2000.
53. T. Randolph, Image compression and classification using nonlinear filter banks, PhD thesis, Georgia Institute of Technology, May 2001.
54. T. R. Randolph and M. J. T. Smith, Enhancement of fax documents using a binary angular representation, in *Proc. 2001 Int. Symp. Intell. Multimedia, Video Speech Process.*, May 2001, pp. 125–128.
55. T. R. Randolph and M. J. T. Smith, Fingerprint image enhancement using a binary angular representation, in *ICASSP*, May 2001.

56. S.-I. Park, M. J. T. Smith, and R. M. Mersereau, A new motion parameter estimation algorithm based on the continuous wavelet, *IEEE Trans. Image Process*., 9(5), 873–888, May 2000.
57. F. A. Mujica, R. Murenzi, M. J. T. Smith, and S-I Park, Motion estimation using the spatio-temporal continuous wavelet transform: new results and alternative implementations, in *Proc. IEEE Int. Conf. Image Process*., September 2000, vol. 2, pp. 550–553.
58. R. Ansari and C.-L. Lau, Two-dimensional IIR filters for exact reconstruction in tree-structured subband decomposition, *Elec. Letters*, 23(12), 633–634, June 1987.
59. R. Ansari and C. Guillemot, Exact reconstruction filter banks using diamond FIR filters, in *Proc. Bilcon 1990*, Elsevier Press, Amsterdam, The Netherlands, pp. 1412–1424, July 1990.
60. R. Ansari, A. E. Cetin, and S. H. Lee, Subband coding of images using nonrectangular filter banks, in *Proc. SPIE Conf. Appl. Digital Image Process*, August 1988, vol. 974, pp. 315–323.
61. C. W. Kim and R. Ansari, Subband decomposition procedure for Quincunx sampling grids, in *Proc. SPIE Conf. Visual Comm. Image Process*., pp. 112–123, November 1991.

26

Nonlinear Filtering Using Statistical Signal Models

Kenneth E. Barner
University of Delaware

Tuncer C. Aysal
Cornell University

Gonzalo R. Arce
University of Delaware

26.1 Introduction **26**-1
26.2 *M*-Estimation **26**-4
Generalized Gaussian Density and Maximum Likelihood Estimation • Gaussian Statistics: Linear Filtering • Laplacian Statistics: Median Filtering
26.3 *LM*-Estimation **26**-11
Generalized Cauchy Density and Maximum Likelihood Estimation • Running Myriad Smoothers: Myriad Filtering • Running Meridian Smoothers: Meridian Filtering • Introduction of Real-Valued Weights and Optimization
26.4 Applications of Robust Nonlinear Estimators/Filters **26**-27
Baseband Communications • Power Line Communications • Highpass Filtering of a Multitone Signal
26.5 Concluding Remarks and Future Directions **26**-31
References **26**-32

26.1 Introduction

Linear methods that satisfied the principle of superposition dominate current signal processing theory and practice. Linear signal processing is founded in the rich theory of linear systems, and in many applications linear signal processing methods prove to be optimal. Moreover, linear methods are inherently simple to implement, with their low computational cost, perhaps the dominant reason for their widespread use in practice. While linear methods will continue to play a leading role in signal processing applications, nonlinear methods are emerging as viable alternative solutions.

The rapid emergence of nonlinear signal processing algorithms is motivated by the growth of increasingly challenging applications, for instance in the areas of multimedia processing and communications. Such applications require the use of increasingly sophisticated signal processing algorithms. The growth of challenging applications is coupled with continual gains in digital signal processing hardware, in terms of speed, size, and cost. These gains enable the practical deployment of more sophisticated and computationally intensive algorithms. Thus, nonlinear algorithms and filtering methods are being developed and employed to address an increasing share of theoretical problems and practical applications.

A disadvantage of nonlinear approaches is that, unlike their linear counterparts, nonlinear methods lack a unified and universal set of tools for analysis and design. The lack of unifying theory has led to the

development of hundreds of nonlinear signal processing algorithms. These algorithms range from theoretically derived broad filter classes, such as polynomial and rank–order based methods [1–9], to boutique methods tailored to specific applications. Thus the dynamic growth of nonlinear methods and lack of unifying theory makes covering the entirety of such operators in a single chapter impossible. Still, large classes of nonlinear filtering algorithms can be derived and studied through fundamentals that are well founded.

The fundamental approach adopted in this chapter is that realized through the coupling of statistical signal modeling with optimal estimation-based filter development. This general approach leads to a number of well-established filtering families, with the specific filtering scheme realized depending on the estimation methodology adopted and the particular signal model deployed. Particularly amenable to filter development is the maximum likelihood estimation (M-estimation) approach. Originally developed in the theory of robust statistics [10], M-estimation provides a framework for the development of statistical process location estimators, which, when employed with sliding observation windows, naturally extend to statistical filtering algorithms.

The characteristics of a derived family of filtering operators depend not only on the estimation methodology upon which the family is founded, but also on the statistical model employed to characterize a sequence of observations. The most commonly employed statistical models are those based on the Gaussian distribution. Utilization of the Gaussian distribution is well founded in many cases, for instance due to the central limit theorem, and leads to computationally simple linear operations that are optimal for the assumed environment. There are many applications, however, in which the underlying processes are decidedly non-Gaussian. Included in this broad array of applications are important problems in wireless communications, teletrafic, networking, hydrology, geology, economics, and imaging [11–15]. The element common to these applications, and numerous others, is that the underlying processes tend to produce more large magnitude observations, often referred to as outliers or impulses, than is predicted by Gaussian models. The outlier magnitude and frequency of occurrence predicted by a model is governed by the decay rate of the distribution tail. Thus, many natural sequences of interest are governed by distributions that have heavier tails (e.g., lower tail decay rates) than that exhibited by the Gaussian distribution. Modeling such sequences as Gaussian processes leads not only to a poor statistical fit, but also to the utilization of linear operators that suffer serious degradation in the presence of outliers.

Couplings, an estimation (filtering) methodology with a statistical model appropriate for the observed sequence, significantly improves performance. This is particularly true in heavy tailed environments. As an illustrative example, consider the restoration of an image corrupted by (heavy tailed) salt and pepper noise. Typical sources of salt and pepper include flecks of dust on the lens or inside the camera, or, in digital cameras, faulty CCD elements. Figure 26.1 shows a sample corrupted image, the results of two-dimensional linear and nonlinear filtering, and the true underlying (desired) image. It is clear that the linear filter, unable to exploit the characteristics of the corrupting noise, provides an unacceptable result. On the other hand, the nonlinear filter, utilizing the statistics of the image, provides a very good result. The nonlinear filtering utilized in this example is the median, which is derived to be optimal for certain heavy tailed processes. This appropriate statistical modeling results in performance that is far superior to linear processing, which is inherently based on the processing of light tailed samples.

To formally address the processing of heavy tailed sequences, this chapter first considers sequences of samples drawn from the generalizes Gaussian distribution (GGD). This family generalizes the Gaussian distribution by incorporating a parameter that controls the rate of exponential tail decay. Setting this parameter to 2 yields the standard Gaussian distribution, while for values less than two the GGD tails decay slower than in the standard Gaussian case, resulting in heavier tailed distributions. Of particular interest is the first order exponential decay case, which yields the double exponential, or Laplacian, distribution. The Gaussian and Laplacian GGD special cases warrant particular attention due to their theoretical underpinnings, widespread use, and resulting classes of operators when deployed in an M-estimation framework. Specifically, it is shown here that M-estimation of Gaussian distributed



FIGURE 26.1 The original figure depicted in (a) is corrupted with transmission noise, the result of which is given in (b). The received image is then processed with (c) linear (mean) and (d) nonlinear (median) filters.

observations samples leads to traditional linear filtering, while the same framework leads to median filtering for Laplacian distributed samples. Thus as linear filters are optimal for Gaussian processes, the median filter and its weighted generalizations are optimal for Laplacian processes. Median type filters are more robust than linear filters and operate more efficiently in impulsive environments, characteristics that result directly from the heavy tailed characteristic of the Laplacian distribution.

Although the Laplacian distribution has a lower rate of tail decay than the Gaussian distribution, extremely impulsive processes are not well modeled as Laplacian. The GGD family, in fact, is limited in its ability to appropriately model extremely impulsive sequences due to the constraint that, while incorporating freedom in the detail decay rate, the tail decay rate is, nonetheless, restricted to be exponential. Appropriate modeling of such sequences is of critical importance, as a wide variety of extremely impulsive processes are observed in practice. Many such sequences arise as the superposition of numerous independent effects. Examples of which include radar clutter, formed as the sum of many signal reflections from irregular surfaces, the received sum of multiuser transmitted signals observed at a detector in a communications problem, the many impulses caused by the contact of rotating machinery parts in electromechanical systems, and atmospheric noise resulting from the superposition of lightning-based electrical discharges around the globe.

The superposition nature of many observed heavy tailed processes has led to the utilization of α-Stable distributions as signal models for such processes [16–19]. Indeed, the family of α-Stable distributions can be justified by the Generalized Central Limit Theorem [19–25]. Moreover, the distribution family (other than the Gaussian limiting case) possesses algebraic tales, making α-Stable modeling of heavy tailed processes more accurate than exponential tailed GGD family modeling. Although an accurate model for many heavy tailed processes, the utility of the α-Stable family is limited by the fact that only a single heavy tailed distribution in the family possesses a closed form, namely the Cauchy distribution. Thus heavy-tail focused theories and methods derived from the α-Stable family are limited, and based on a single distribution.

To overcome the drawbacks associated with GGD and α-Stable based approaches, we present methods derived from a robust extension to *M*-estimation referred to as *LM*-estimation. The *LM*-estimation formulation yields operators that are significantly more robust than traditional (GGD) *M*-estimation based methods. Moreover, *LM*-estimation is statistically related to, and derives its optimality from, the generalized Cauchy density (GCD). Utilization of GCD derived methods is particularly advantageous in that (1) the GCD is a family of distributions possessing algebraic tail decay rates and that (2) they have closed form expressions. Thus like the α-Stable family, the GCD is an appropriate model for extremely impulsive phenomena. But in contrast to the α-Stable family (and like the GGD), the GCD is a broad family of distributions that can be represented in closed form, thereby presenting a framework from which estimation and filtering procedures can be derived. Thus the GCD combines the advantages of the α-Stable and GGD families (accurate heavy tailed modeling and closed form expressions), while eliminating their respective disadvantages.

Much like the Gaussian and Laplacian distributions represent special cases of importance within the GGD, the GCD possesses several special cases that are worthy of thorough investigation. Specifically, we cover in depth the Cauchy and Meridian distribution special cases of the GCD. These distributions are coupled with the *LM*-estimation framework and shown to yield the Myriad and Meridian filtering operations [26–29]. These filtering classes are proven and shown to be significantly more robust than traditional linear, and even median, filtering. Additionally, they contain a free parameter that controls the level of robustness. This degree of freedom allows for a wide array of filtering characteristics including limiting cases that converge to traditional filtering algorithms. Specifically, the (least robust) limiting case of the Myriad filter is the traditional linear filter, while the median filter is the (least robust) limiting case of the Meridian filter. This illustrates the broad range of filtering characteristics exhibited by GCD-based methods and justifies their in-depth coverage within this chapter.

The remainder of the chapter is organized as follows. Section 26.2 introduces *M*-estimation and couples this approach with the GGD signal modeling. Particular attention is given to the Gaussian and Laplacian distribution special cases and the resulting linear and median filtering operations. The *LM*-estimation robust extension to *M*-estimation is covered in Section 26.3. The link between the GCD family and *LM*-estimation is covered along with in-depth coverage of the Myriad and Meridian special cases. Coverage includes an analysis of the filtering objective functions, limiting special cases, evaluations of properties and characteristics, as well as the presentation of optimization procedures. Applications and numerical examples illustrating and contrasting the capabilities of covered filtering methods are presented in Section 26.4. Specifically considered applications are basedband communications, recently emerging powerline communications, and highpass filtering. Finally, conclusions are drawn and future research directions noted in Section 26.5.

26.2 *M*-Estimation

To address the filtering problem, we begin by formally developing the *M*-estimation framework and the commonly employed GGD statistical model. This development allows the problem to be rigorously addressed, but also presents an intuitive approach from which filtering algorithms can be derived and understood. It is shown that combining the GGD statistical model and a special case of *M*-estimation referred to as maximum likelihood estimation (ML), results in simple norms that, among other

applications, can be used to define filtering structures. The norms are distribution specific, and we consider in depth the Gaussian and Laplacian special cases of the GGD showing that the resulting norms are the commonly utilized L_2 and L_1 metrics, respectively. Moreover, the Gaussian distribution and L_2 metric directly lead to the class of linear filters, while the Laplacian and L_1 metric correspond to the family of median filters. The difference in norm and filtering characteristics is directly dependent on the distribution tail decay rates, with those derived from the heavier detailed Laplacian distribution being more robust than those derived from light tailed Gaussian distribution.

26.2.1 Generalized Gaussian Density and Maximum Likelihood Estimation

Perhaps the most fundamental form of estimation is the problem of location estimation. While fundamentally simple, location estimation is easily extended to the filtering problem through utilization of a sliding observation window. To develop location estimation based filtering operations, we first consider the slightly more general problem of ML estimation, which was developed as a special case of *M*-estimation within the theory of robust statistics [10,30].

To establish the estimation and filtering operators, consider a set of observations (input samples), $\{x(i) : i = 1, 2, \ldots, N\}$, formed from a signal $s(i; \theta)$, with and underlying parameter of interest θ, corrupted by additive noise, i.e.,

$$x(i) = s(i; \theta) + n(i), \tag{26.1}$$

where $n(i)$ represents the additive noise samples that are distributed as $n(i) \sim f_n(\cdot)$. The assumed model is quite general, as is the *M*-estimation formulation for the underlying parameter of interest, which is stated in the following definition.

Definition 26.1: Given the set of observations $\{x(i; \theta): i = 1, 2, \ldots, N\}$, the *M*-estimate of θ is given by

$$\hat{\theta}_M = \arg\min_{\theta} \sum_{i=1}^{N} \rho(x(i) - s(i; \theta)), \tag{26.2}$$

where $\rho(\cdot)$ is defined as the cost function of the *M*-estimate.

Note that the cost function can take on many forms, and can be tailored to a particular signal model or application. One particularly simple single model is to assume that the observation samples are statistically independent. Coupling this assumption with a cost function tied directly to the statistics of the observed samples, namely $\rho(u) = -\log\{f_n(u)\}$, yields the class of ML estimators. Maximum likelihood estimators have received broad attention, and have been applied across a large array of sample distributions and to a vast number of applications.

The effectiveness of ML estimation depends on the statistical model employed in the cost function, and how well this model represents the observation sequence. The most commonly employed statistical model is the Gaussian distribution, which in its generalized density form is expressed as

$$f_{\mathrm{GGD}}(u) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left\{-\left(\frac{|u|}{\alpha}\right)^{\beta}\right\}, \tag{26.3}$$

where $\Gamma(\cdot)$ is the gamma function. The parameter α determines the width of the density peak (standard deviation), while β controls the tail decay rate. Generally, α is referred to as the scale parameter while β is

called the shape parameter. The GGD can be used to model a broad range of noise processes. Moreover, taking the $n(i)$ terms in Equation 26.1 to be independent, identically distributed (i.i.d.) GGD samples yields a particularly simple form for the ML estimate of θ. That is, Equation 26.2 reduces to a compact, intuitive expression, which is given in the following theorem.

THEOREM 26.1

Consider the set of observations $\{x(i) : i = 1, 2, \ldots, N\}$ *corrupted by i.i.d. GGD distributed noise. The ML estimate of* θ *is the solution to the following minimization problem*:

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} |x(i) - s(i;\theta)|^{\beta}. \tag{26.4}$$

This result is obtained by letting $\rho(u) = -\log\{f_{\mathrm{GGD}}(u)\}$ and substituting the result into the M-estimation expression, Equation 26.2. To appreciate this result, it is instructive to consider the range of distributions within the GGD. An intuitive appreciation of the distribution family can be obtained through the examination of two special cases, namely the Gaussian and Laplacian distributions. The Gaussian distribution is realized for $\beta = 2$. For all $\beta < 2$ cases, the resulting distributions are heavier detailed than the Gaussian distribution. This is illustrated in Figure 26.2, which shows that density function for several values of β including the Laplacian distribution special case ($\beta = 1$). The figure clearly illustrates the relationship between β and the distribution tail decay rate—decreasing β increases the tail heaviness and vice versa.

A review of the ML estimate in the GGD case, Equation 26.4, makes clear that the distribution tail decay rate directly affects the estimate. Namely, the tail decay rate defines a norm under which the estimate is formulated. Thus, the GGD corresponds directly to norms of the form $\rho(u) = |u|^{\beta}$, where

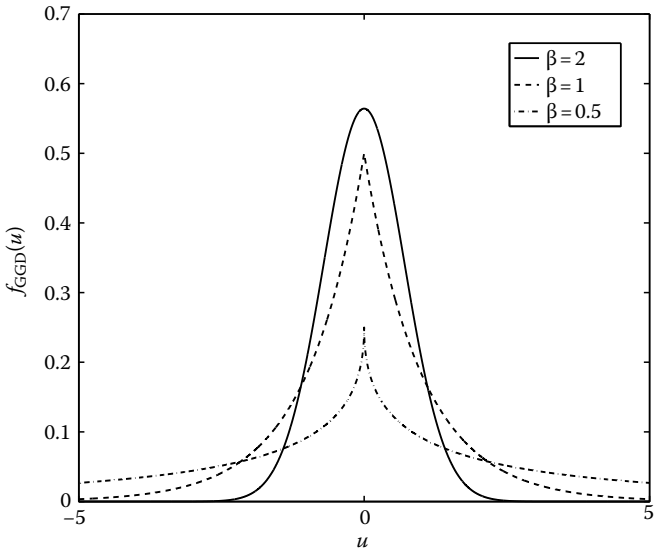


FIGURE 26.2 Generalized Gaussian density function for $\alpha = 1$ and $\beta \in \{2,1,0.5\}$. Note that the $\beta = 2$ and $\beta = 1$ cases correspond to the Gaussian and Laplacian distributions, respectively, and that the rate of the tail decay is proportional to β.

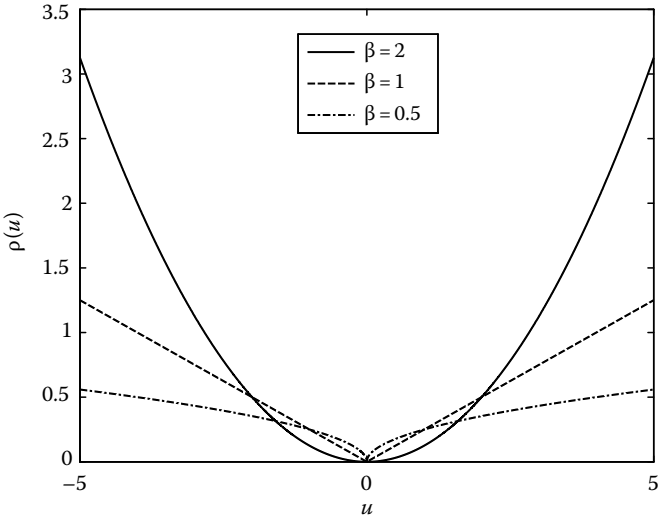


FIGURE 26.3 GCD derived $\rho(u) = |u|^{\beta}$ norms for $\beta \in \{2,1,0.5\}$. Note that the $\beta = 2,1$, and 0.5 cases correspond to the squared, absolute deviation, and fractional lower order moment formulations, respectively, and that the norm robustness is inversely proportional to β.

$\beta \in [0, 2]$. The special cases $\beta = 2$, $\beta = 1$, and $\beta < 1$ thus reduce to least-squares, least-absolute deviations, and fractional lower order moment formulations. These norms are widely applied in a range of applications including curve fitting, segmentation, filtering, and the vast majority of optimization problems. The behavior of the L_β norms, as they are sometimes referred to, is illustrated in Figure 26.3. Of note is the influence outliers have on the norms. In the $\beta = 2$ (Gaussian/L_2) case, the effect of outliers is squared, while outliers have a linear effect in the $\beta = 1$ (Laplacian/L_1) case and even less influence in the fractional lower order moment case. That is to say that as β decreases, the error norms become more robust, behavior directly linked to the prevalence of outliers in GCD distributions with small β values.

Having established the GGD family of distributions, ML estimation for the additive noise model, and the norms that arise from the GGD–ML coupling, these tools can now be applied to the development of filtering algorithms for specific distribution cases.

26.2.2 Gaussian Statistics: Linear Filtering

To directly address the filtering problem, the observed signal model in Equation 26.1 is simplified to be a direct location function of θ. This yields observation samples of the form

$$x(i) = \theta + n(i) \tag{26.5}$$

and reduces the *M*-estimation expression to simply

$$\hat{\theta}_M = \arg\min_{\theta} \sum_{i=1}^{N} \rho(x(i) - \theta). \tag{26.6}$$

This simplified observation model and estimation structure yields standard operations when combined with commonly utilized distribution models. Consider first the case in which the $n(i)$ observation noise samples are Gaussian distributed,

$$f(u) = \frac{1}{2\sigma^2} \exp\left\{ -\left(\frac{|u|^2}{2\sigma^2} \right) \right\}. \tag{26.7}$$

The following theorem addresses this case and shows that the resulting filter is a simple linear operator.

THEOREM 26.2

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Gaussian distribution with location* θ *and variance* σ^2*. The ML estimate of location is given by*

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} (x(i) - \theta)^2 \right] = \frac{1}{N} \sum_{i=1}^{N} x(i) = \text{mean}\{x(i) : i = 1, 2, \ldots, N\}. \tag{26.8}$$

This result follows from steps similar to those utilized in Section 26.2.1. The expression in Equation 26.8 shows clearly that the optimization criteria in this case reduces to the L_2 norm. Moreover, the resulting expression can be interpreted as a mean filtering structure, $y = 1/N \sum_{i=1}^{N} x(i)$, where y denotes the filter output. Windowing of the observation sequence can be used to form sliding observation sets, yielding an indexed output $y(i)$ and making the filtering operation explicit.

Although the results of the previous theorem can be interpreted as yielding a filtering operation, the realized operation is somewhat limited in that it does not utilize weights to control the characteristics of the filter. This is a direct result of the somewhat restrictive i.i.d. assumption imposed in the theorem. Fortunately, the identically distributed constraint can be relaxed. This results in a more general signal model and yields a more traditional linear filtering structure that incorporates sample weighting.

THEOREM 26.3

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Gaussian distribution with common location* θ *and (possibly) different variances* $\sigma^2(i)$*. The ML estimate of location is given by*

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} \frac{1}{\sigma^2(i)} (x(i) - \theta)^2 \right] = \frac{\sum_{i=1}^{N} h(i)x(i)}{\sum_{i=1}^{N} h(i)} \stackrel{\Delta}{=} \text{mean}\{h(i) \cdot x(i) : i = 1, 2, \ldots, N\}, \tag{26.9}$$

where $h_i = 1/\sigma^2(i) > 0$.

This is simply a linear filtering structure, $y = \sum_{i=1}^{N} \tilde{h}(i)x(i)$, where the $\tilde{h}(i) = h(i)/\sum_{i=1}^{N} h(i)$ terms are the normalized filter weights. As derived in this development, the weights are inversely proportional to individual sample variances. This is an intuitive result, as samples with high variability will be given small weight and have minimal influence on the result. The positivity constraint, however, restricts the resulting operators to the class of smoothers. In practice, this constraint is relaxed enabling the resulting class of linear finite impulse response (FIR) filters to employ both positive and negative weights that provide a wide array of spectral characteristics.

26.2.3 Laplacian Statistics: Median Filtering

The ML-based filter development can be extended to any distribution within the GGD, or, in fact, any valid distribution. Although not all distributions yield compact filtering expressions, specific special cases do correspond to simple, effective filtering structures. To derive such a structure that is more robust to

sample outliers than the Gaussian distribution optimal linear filter, consider the heavier tailed Laplacian distribution ($\beta = 1$) special case of the GGD,

$$f(u) = \frac{1}{2\sigma} \exp\left\{ -\left(\frac{|u|}{\sigma} \right) \right\}. \tag{26.10}$$

The following theorem shows that the median filter is the optimal operator for Laplacian distributed samples.

THEOREM 26.4

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Laplacian distribution with common location* θ *and variance* σ^2*. The ML estimate of location is given by*

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} |x(i) - \theta| \right] = \text{median}\{x(i) : i = 1, 2, \ldots, N\}. \tag{26.11}$$

The arguments utilized previously, with the appropriate distribution substitution, prove the result. The expression in Equation 26.11 shows that, in this case, the optimization criteria reduces to the more robust L_1 norm. Moreover, the resulting expression is simply a median filter structure, $y = \text{median}\{x(i) : i = 1, 2, \ldots, N\}$, where y denotes the filter output. This operation is clearly nonlinear as the output is formed by sorting the observation samples and taking the middle, or median, value as the output.*

Similarly to the mean filtering case, the median filtering operation can be generalized to admit weights. The theoretical motivation for this generalization is, like in the previous case, the relaxation of the identically distributed constraint placed on the observation samples in the above theorem.

THEOREM 26.5

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Laplacian distribution with common location* θ *and (possibly) different variances* $\sigma^2(i)$*. The ML estimate of location is given by*

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} \frac{1}{\sigma^2(i)} |x(i) - \theta| \right] = \text{median}\{h(i) \diamond x(i) : i = 1, 2, \ldots, N\}. \tag{26.12}$$

where $h_i = 1/\sigma^2(i) > 0$ *and* $\diamond$ *is the replication operator defined as* $h(i) \diamond x(i) = \overbrace{x(i), x(i), \ldots, x(i)}^{h(i)\text{times}}$.

The weighting operation in this case is achieved through repetition, rather than the scaling employed in the linear filter. But like the linear case, sample weights are inversely proportional to the sample variances, indicating again that samples with large variability contribute less to the determination of the output than well behaved (smaller variance) samples. This magnitude relationship between a sample's weight and its influence holds even for the relaxed case of positive and negative weights. This relaxation on the weights employs sign coupling and enables a broader range of filtering characteristics to be realized by weighted median (WM) filters [31]:

$$y = \text{median}\{|h(i)| \diamond \text{sgn}(h(i))x(i) : i = 1, 2, \ldots, N\}, \tag{26.13}$$

* For cases in which the number of observation samples is an even number, the median value is set as the average of the two central samples in the ordered set.

where

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{if } x < 0. \end{cases} \tag{26.14}$$

Considerable analysis is available in the literature on the detail preservation and outliers rejection characteristics of WM filters [8,31–36].

To contrast the performance of linear and WM filters, consider the simple problem of running (constant) location estimation from noisy measurements. Figure 26.4 shows such an example for two noise processes, Laplacian and α-Stable distributed samples. The Laplacian distribution is within the

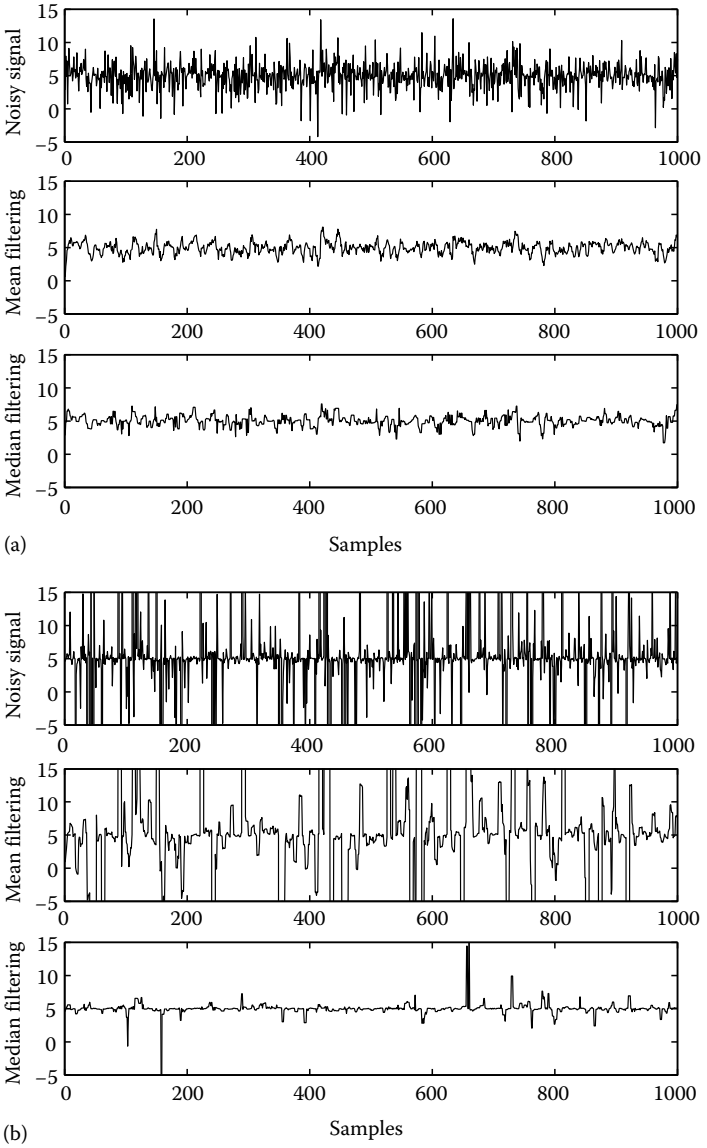


FIGURE 26.4 Constant signal corrupted with (a) Laplacian and (b) α-Stable noise ($\alpha = 0.5$) processed with mean and median filters.

GGD family, and as such the linear and WM filters perform well in this case, with the Laplacian noise optimal WM returning the best performance. The α-Stable distribution, however, has significantly heavier tails and both linear and WM filters breakdown in this environment. This indicates that GGD-based methods are not well suited to extremely impulsive environments, and that more sophisticated methods for addressing samples characterized by very heavy (algebraic) tailed distributions must be developed and employed.

26.3 *LM*-Estimation

Many contemporary applications contain samples with very heavy tailed statistics including the aforementioned powerline communications, economic forecasting, network traffic processing, and biological signal processing problems [15,37–44]. The GGD family of distributions, while representing a broad class of statistics with varying tail parameters, is, nevertheless, restricted to distributions with an exponential rate of tail decay. Distributions with exponential rates of tail decay are generally considered light tailed, and do not accurately model the prevalence or magnitude of outliers in true heavy-tailed processes. Such processes are often modeled utilizing the α-Stable family of distributions [16–19]. While α-Stable distributions do possess tails with algebraic decay rates, and are thus appropriate models for impulsive sequences, the distribution lacks a full-family closed form expression and it is therefore not easily coupled with estimation techniques such as ML.

To overcome the drawbacks of GGD and α-Stable-based techniques, we derive a generalization of the *M*-estimation framework that exhibits a spectrum of optimality characteristics including greater robustness. This generalization is referred to as *LM*-estimation, the general form of which is given in the following definition.

Definition 26.2: Given the set of independent observations $\{x(i) : i = 1, 2, \ldots, N\}$ formed as $x(i) = s(i;\theta) + n(i)$, the L*M*-estimate of θ is defined as

$$\hat{\theta}_{LM} = \arg\min_{\theta} \sum_{i=1}^{N} \log\{\delta + \rho(x(i) - s(i;\theta))\}, \tag{26.15}$$

where $\delta > 0$ and $\rho(\cdot)$ are the robustness parameter and cost function, respectively.

In the following, we show that *LM*-estimation is statistically related to the GCD. The GCD family consists of algebraic detailed distributions with closed form expressions, and is therefore an appropriate model for heavy tailed sequences and a family from which estimation and filtering techniques can be derived. We consider GCD and *LM*-estimation based filters, focusing on the Cauchy and Meridian distribution special cases and their resulting filter structures. Properties of the filters are detailed along with optimization procedures. While the GGD-based results are well established and reported in numerous works, the *LM*-estimation and GCD material presented represents the newest developments in this area, and as such proofs for many results are included.

26.3.1 Generalized Cauchy Density and Maximum Likelihood Estimation

As the previous section covering GGD-based methods shows, the robustness of error norms, estimation techniques, and filtering algorithms derived from a density is directly related to the density tail decay rate. The robustness of *LM*-estimation derives from its statistical relation to the GCD. The GCD function is defined by

$$f_{\mathrm{GCD}}(u) = \frac{\lambda\Gamma(2/\lambda)}{2[\Gamma(1/\lambda)]^2} \frac{\nu}{(\nu^{\lambda} + |u|^{\lambda})^{2/\lambda}}. \tag{26.16}$$

As in the GGD case, ν is referred to as the scale parameter while λ is called the shape parameter.

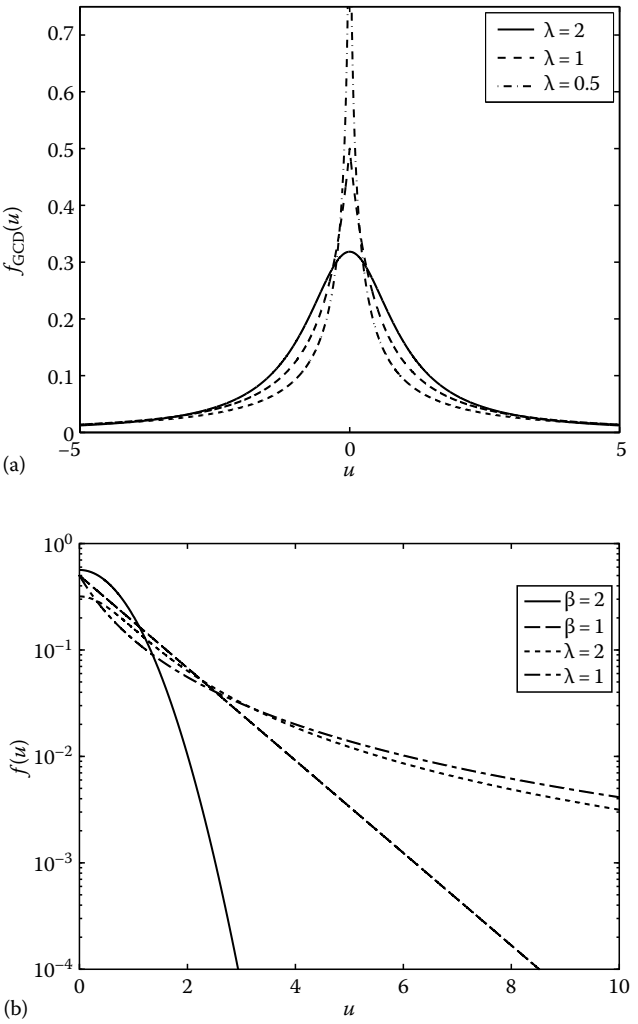


FIGURE 26.5 (a) GCD function for $\nu = 1$ and $\lambda \in \{2,1,0,5\}$; and (b) enlarged tail function for select GGD and GCD distributions.

The Generalized Cauchy distribution was first proposed by Rider in 1957 [45], and "rediscovered" under a different parametrization by Miller and Thomas in 1972 [14]. Distributions within the GCD family have algebraically decaying tails and are thus appropriate models of impulsive sequences. Indeed, the GCD is used in several studies of impulsive radio noise [14,46,47]. Of note within the GCD family are two special cases, namely the Cauchy and Meridian distributions that are realized for $\lambda = 2$ and $\lambda = 1$, respectively [8,28,29]. The GCD is depicted in Figure 26.5a for the $\lambda \in \{2,1,0.5\}$ cases, all with $\nu = 1$. The slow rate of GCD tail decay, which is inversely proportional to λ, is clearly seen in the figure. To make plain the difference in GGD and GCD tail decay rates, Figure 26.5b plots enlargements of tail sections from distributions in each family.

The utility of a heavy-tailed distribution family defined by closed forum expressions is that it can readily be applied in ML estimation, as detailed for the GCD in the following theorem.

THEOREM 26.6

Consider the set of observations $\{x(i) : i = 1, 2, \ldots, N\}$ *corrupted by i.i.d. GCD distributed noise. The ML estimate of* θ *is the solution to the following minimization problem:*

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} \log\{\nu^{\lambda} + |x(i) - s(i;\theta)|^{\lambda}\}. \tag{26.17}$$

Proof: The ML estimate is defined by

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{N} f_x(x(i)). \tag{26.18}$$

Substituting the GCD expression and denoting $C(\lambda) = \lambda\Gamma(2/\lambda)/(2[\Gamma(1/\lambda)]^2)$ yields

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{N} C(\lambda) \frac{\nu}{(\nu^{\lambda} + |x(i) - s(i;\theta)|^{\lambda})^{2/\lambda}}. \tag{26.19}$$

Taking the natural $\log\{\cdot\}$ and noting that $C(\lambda)$ and ν are constant with respect to the maximization of θ gives

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} -\frac{2}{\lambda} \log\{\nu^{\lambda} + |x(i) - s(i;\theta)|^{\lambda}\}. \tag{26.20}$$

Finally, noting that maximizing $g(u)$ is equivalent of minimizing $-g(u)$, and that $2/\lambda$ is constant with respect to θ, gives the desired result.

This theorem shows that the ML estimate for GCD distributed samples is simply a special case of LM estimation. Comparing Equations 26.15 and 26.17 shows the equivalence holds for $\rho(\cdot) = |\cdot|^{\lambda}$ and $\delta = \nu^{\lambda}$. Thus LM estimation is a more general framework, but one that derives its optimality from GCD–ML estimation. Moreover, *M*-estimation is a (least robust) limiting case of *LM*-estimation, as is shown in the following proposition. This theorem utilizes the somewhat simplified case of location estimation (i.e., $s(i;\theta) = \theta$), which we again consider from this point forward as it is most directly related to filter development.

PROPOSITION 26.1

The LM-estimator reduces to an M-estimator as δ *tends to infinity, i.e.*,

$$\lim_{\delta\to\infty} \hat{\theta}_{LM} = \arg\min_{\theta} \sum_{i=1}^{N} \rho(x(i) - \theta) \tag{26.21}$$

with cost function $\rho(\cdot)$.

Proof: Utilizing the properties of arg min and log functions, we have the following equalities

$$\lim_{\delta\to\infty} \hat{\theta}_{LM} = \lim_{\delta\to\infty} \arg\min_{\theta} \sum_{i=1}^{N} \log\{\delta + \rho(x(i) - \theta)\} \tag{26.22}$$

$$= \lim_{\delta\to\infty} \arg\min_{\theta} \sum_{i=1}^{N} \log\left\{1 + \frac{\rho(x(i)-\theta)}{\delta}\right\} + \log\{\delta\} \tag{26.23}$$

$$= \lim_{\delta\to\infty} \arg\min_{\theta} \sum_{i=1}^{N} \log\left\{1 + \frac{\rho(x(i)-\theta)}{\delta}\right\} \tag{26.24}$$

$$= \lim_{\delta\to\infty} \arg\min_{\theta} \sum_{i=1}^{N} \delta \log\left\{1 + \frac{\rho(x(i)-\theta)}{\delta}\right\} \tag{26.25}$$

$$= \lim_{\delta\to\infty} \arg\min_{\theta} \sum_{i=1}^{N} \log\left\{1 + \frac{\rho(x(i)-\theta)}{\delta}\right\}^{\delta}. \tag{26.26}$$

Applying the fact that

$$\lim_{\delta\to\infty} \log\left\{1 + \frac{u}{\delta}\right\}^{\delta} = u \tag{26.27}$$

yields the desired result.

The fact that *M* estimation is a limiting case of *LM* estimation indicates that the latter is a more general family of estimators and that methods derived under this framework are, consequently, more general and subsume those emanating from *M* estimation. Thus, *LM* estimation based methods are inherently more efficient than (or at least equally efficient to) *M* estimation based methods. It should also be recalled that in the *LM* estimation definition, Equation 26.15, δ is referred to as the robustness parameter. Thus the above shows equality between *LM* and *M* estimation at the robustness limit. It is easy to see that this is the least robust limit of *LM* estimation. To make this plain, consider the error norm defined by the GCD in case of Equation 26.15, which can be expressed in simplified form as

$$\rho(u) = \log\left\{1 + \frac{|u|^{\lambda}}{\delta}\right\}^{\delta}. \tag{26.28}$$

Figure 26.6 plots this norm, showing the effects of varying δ and λ. The impact of δ on robustness is clear, justifying the naming of this parameter and indicating that the $\delta \to \infty$ case is indeed the least robust case. Also evident is the greater robustness of GCD norms over their GGD counterparts, with equality occurring at the limit point (up to a scaling factor). Having established *LM* estimation and the related GCD norms, these tools are now employed to develop filtering algorithms that arise from the consideration of specific distribution cases, namely the Cauchy and Meridian distributions.

26.3.2 Running Myriad Smoothers: Myriad Filtering

Consider first the Cauchy distribution special case ($\lambda = 2$). The following theorem sets the sample myriad as the optimal estimate of location for samples obeying this distribution.

THEOREM 26.7

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Cauchy distribution with common location* θ *and scale* γ*. The ML estimate of location, or sample myriad, is given by*

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} \log\{\gamma^2 + (x(i)-\theta)^2\}\right] = \text{myriad}\{x(i) : i = 1, 2, \ldots, N; \gamma\}, \tag{26.29}$$

where γ *is the linearity parameter.*

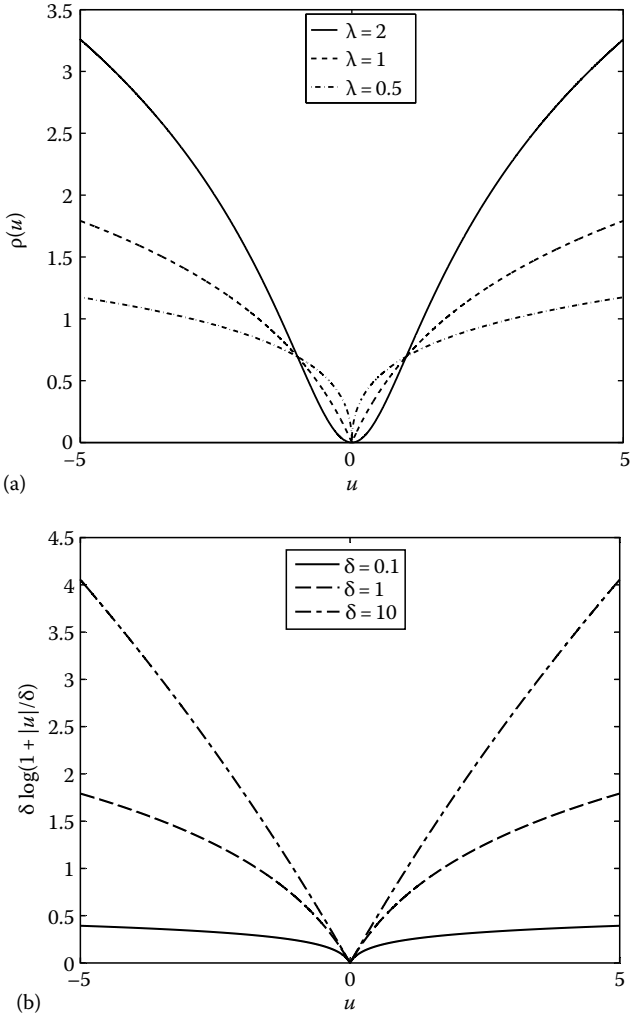


FIGURE 26.6 The error norms defined by the GCD function for (a) $\lambda \in \{0.5,1,2\}$ with $\delta = 1$ and (b) $\lambda = 1$ and $\delta \in \{0.1,1,10\}$.

Note the slight change in notation, γ replacing ν, which we employ to be consistent with the literature [26–28]. Also, the reference to γ as the linearity parameter is made clear in subsequent properties. An appreciation of the myriad operator is obtained through an investigation of the cost function defining the operator. Thus, let $Q(\theta)$ denote the objective function minimized in Equation 26.29, i.e.,

$$Q(\theta) \stackrel{\Delta}{=} \sum_{i=1}^{N} \log\{\gamma^2 + (x(i) - \theta)^2\}. \tag{26.30}$$

The following proposition brings together a few key properties of the myriad cost function. The properties are illustrated by Figure 26.7, which shows the form of a typical objective function.

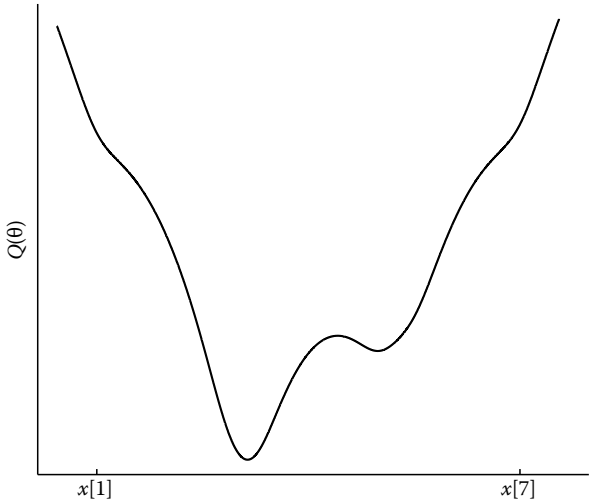


FIGURE 26.7 Typical sketch of the objective function minimized by the myriad operator (for $\gamma = 0.5$ and randomly generated $N = 7$ samples).

PROPOSITION 26.2

Let x[*i*] *signify the order statistics (samples sorted in increasing order of amplitude) of the input samples* $\{x(i) : i = 1, 2, \ldots, N\}$, *with the smallest* $x[1]$ *and the largest* $x[N]$. *The following statements hold*:

(1) *Objective function* $Q(\theta)$ *has a finite number* (*at most N*) *of local extrema*.
(2) *Myriad is one of the local minima of* $Q(\theta)$: $Q'(\theta) = 0$.
(3) $Q'(\theta) > 0$ [$Q(\theta)$ *strictly increasing*] *for* $\theta > x[N], Q'(\theta) < 0$ *and* [$Q(\theta)$ *strictly decreasing*] *for* $\theta < x[1]$.
(4) *All the local extrema of* $Q(\theta)$ *lie within the range* [$x[1]$, $x[N]$] *of the input samples*.
(5) *Myriad is in the range of input samples*: $x[1] \leq \hat{\theta} \leq x[N]$.

Note that, unlike the mean or median, the definition of the myriad involves the free-tunable parameter γ. Importantly, Proposition 26.1 shows that in the limit of this parameter (or equivalently δ) *LM*-estimation converges to *M*-estimation. A particularly interesting realization of this general result holds for the myriad case. Namely, the myriad converges to the mean in the limiting case, as formally defined in the following.

COROLLARY 26.1

Given a set of samples $\{x(i): i = 1, 2, \ldots, N\}$, *the sample myriad* $\hat{\theta}$ *converges to the sample mean as* γ *tends to infinity. That is*,

$$\lim_{\gamma \to \infty} \hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} x(i). \tag{26.31}$$

The fact that an infinite value of γ converts the nonlinear myriad operation to the linear sample average illustrates why γ is aptly named the linearity parameter: the larger the value of γ, the closer the behavior

of the myriad is to the (linear) mean estimator. As γ is decreased, the myriad becomes more robust. In the limit, when γ tends to zero, the estimator treats every observation as a possible outlier, assigning more credibility to the most repeated observation values. This "mode-type" characteristic is reflected in the name mode–myriad given this limiting case.

COROLLARY 26.2

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$, *the sample myriad* $\hat{\theta}$ *converges to a mode–type estimator as* $\gamma \to 0$. *That is,*

$$\lim_{\gamma \to 0} \hat{\theta} = \arg \min_{x(j) \in \mathcal{M}} \left[\prod_{i=1,\, x(i) \neq x(j)}^{N} |x(i) - x(j)| \right], \tag{26.32}$$

where $\mathcal{M}$ *is the set of most repeated values.*

The linearity parameter also allows the meridian filter to address three special cases of the α-Stable distribution family. Those three cases are (1) $\alpha = 1$, which yields the Cauchy distribution for which the myriad is optimal, (2) $\alpha = 2$, which yields the Gaussian distribution under which optimal filtering is realized by letting $\gamma \to \infty$, and (3) $\alpha \to 0$, in which case the distribution is extremely impulsive and $\gamma = 0$ yields the optimal results. These three optimality points have been complemented with a simple empirical formula relating γ to the characteristic exponent (α) and dispersion parameter (κ) of an α-Stable asymmetric distribution,

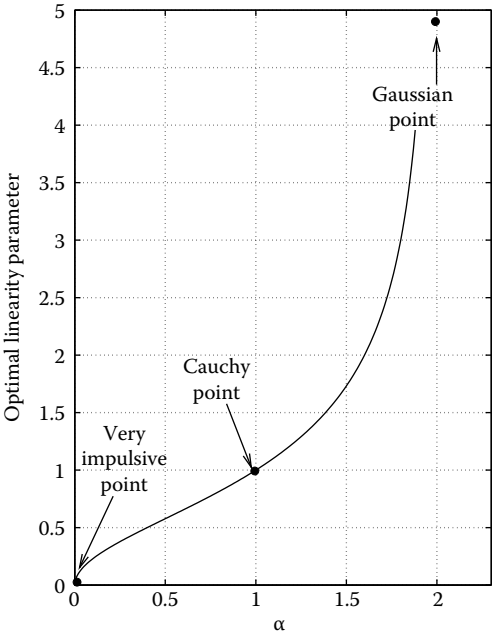


FIGURE 26.8 Empirical α–γ curve for α-Stable distributions. The curve values at $\alpha = 0$,1, and 2 constitute the optimality points of the α-Stable triplet. (From Gonzalez, J. G. and Arce, G. R., *IEEE Trans. Signal Process.*, 49, 438, 2001. With permission.)

$$\gamma(\alpha) = \sqrt{\frac{\alpha}{2 - \alpha}} \kappa^{1/\alpha}, \tag{26.33}$$

which is plotted in Figure 26.8.

Having established the myriad filter, the role of the linearity parameter, and two limiting cases (sample mean and mode–myriad), we present three myriad filter properties that are of importance in signal processing, namely, no under-shoot/overshoot, shift and sign invariance, and unbiasedness. To simplify the notation in the properties, the myriad output is written compactly as $\hat{\theta}(\mathbf{x}) = \text{myriad}\{x(i) : i = 1, 2, \ldots, N; \gamma\}$, where $\mathbf{x} \stackrel{\Delta}{=} [x(1), x(2), \ldots, x(N)]^T$.

PROPERTY 26.1 (No Undershoot/Overshoot)

The output of the myriad estimator operating on samples $\{x(i) : i = 1, 2, \ldots, N\}$ is always bounded by

$$x[1] \leq \hat{\theta}(\mathbf{x}) \leq x[N]. \tag{26.34}$$

PROPERTY 26.2 (Shift and Sign Invariance)

Consider the observation set $\{x(i) : i = 1, 2, \ldots, N\}$ and let $z(i) = x(i) + b$. Then,

(1) $\hat{\theta}(\mathbf{z}) = \hat{\theta}(\mathbf{x}) + b$;
(2) $\hat{\theta}(\mathbf{z}) = -\hat{\theta}(-\mathbf{z})$.

PROPERTY 26.3 (Unbiasedness)

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$ that are independent and symmetrically distributed around a symmetry center c, $\hat{\theta}(\mathbf{x})$ is also symmetrically distributed around c. In particular, if $\mathbb{E}\{\hat{\theta}(\mathbf{x})\}$ exists, then $\mathbb{E}\{\hat{\theta}(\mathbf{x})\} = c$.

The shift invariance and unbiasedness of the myriad filter show that the operator can be applied without concerns to an overall change in location (e.g., a shift in the mean luminance of an image) and that the operator preserves the mean. The undershoot/overshoot property shows that the myriad can be applied without concerns of introducing amplifying artifacts, such as ringing effects.

While the myriad filter offers increased robustness over GGD-based methods and possesses several properties of importance, the equal treatment of samples within the observation set limits the filter, as defined above, to a single operation. Weighting of samples provides a much broader range of filtering operations. This is theoretically justified in a fashion analogous to that utilized in the generalizations of the mean and median operators. Specifically, the following theorem shows that considering samples with a common location but varying scale factors leads to the weighted myriad operator.

THEOREM 26.8

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Cauchy distribution with common location* θ *and (possibly) varying scale factors* $s(i) = \gamma/\sqrt{w(i)}$ [8,28]. *The ML estimate of location, or weighted myriad, is given by,*

$$\hat{\theta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \log\{\gamma^2 + w(i)(x(i) - \beta)^2\} \right] = \text{myriad}\{w(i) \circ x(i) : i = 1, 2, \ldots, N; \gamma\}, \qquad (26.35)$$

where $\circ$ *denotes the weighting operation in the minimization problem.*

As in the weighted linear and median filter cases, the weight applied to a sample in the myriad filter is inversely proportional to the sample's variability. This not only minimizes the effect of unreliable samples (those with large scale), but can also be used to exploit correlations between samples. This is the case, for instance, when a sliding window is employed and higher weights are assigned to samples at spatial locations that are most highly correlated with the desired output, e.g., samples further away in time and less correlated are given smaller weight. Weighting thus allows myriad filters to take on a wide array of characteristics and to be tuned to the needs of specific applications. Moreover, the properties detailed above for the unweighted myriad filter also hold in the weighted case. The properties are not restated for the weighted case since they are identical. We do, however, formally state corollaries governing the behavior of the weighted myriad operator at the limiting cases of the linearity parameter.

COROLLARY 26.3

Given a set of samples $\{x(i): i = 1, 2, \ldots, N\}$ *and corresponding (positive) weights* $\{w(i): i = 1, 2, \ldots, N\}$, *the weighted myriad* $\hat{\theta}$ *converges to a normalized linear estimate as* γ *tends to infinity. That is,*

$$\lim_{\gamma \to \infty} \hat{\theta} = \frac{\sum_{i=1}^{N} w(i)x(i)}{\sum_{i=1}^{N} w(i)}. \tag{26.36}$$

COROLLARY 26.4

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$ *and corresponding (positive) weights* $\{w(i): i = 1, 2, \ldots, N\}$, *the weighted myriad* $\hat{\theta}$ *converges to a weighted mode-type estimate as* γ *tends to* 0. *That is,*

$$\lim_{\gamma \to 0} \hat{\theta} = \arg \min_{x(j) \in \mathcal{M}} \left(\frac{1}{w(j)} \right)^{r/2} \left[\prod_{i=1, x(i) \neq x(j)}^{N} |x(i) - x(j)| \right], \tag{26.37}$$

where $\mathcal{M}$ *is the set of most repeated values* and *r is the number of times a member of* $\mathcal{M}$ *is repeated in the sample.*

The linearity parameter again controls the behavior of the weighted myriad, ranging between a normalized linear operator, in the least robust limit, and a weighted mode operator, in the most robust limit. Thus weighted myriad filters are more general than linear filters, including the latter as a special case. But unlike linear filters, or even median-based filters, the output of the weighted myriad is not available in explicit form. Computation of the output is therefore nontrivial, requiring minimization of the weighted myriad objective function, $Q(\theta)$. Fortunately, $Q(\theta)$ has a number of characteristics that can be exploited to construct fast iterative methods for computing the objective function minimum.

Recall that the weighted myriad is given by

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \; Q(\theta) \\ &= \arg \min_{\theta} \sum_{i=1}^{N} \log \left[1 + \left(\frac{x(i) - \theta}{s(i)} \right)^2 \right], \end{aligned} \tag{26.38}$$

where the change in notation, $s(i) = \gamma / \sqrt{w(i)}$, introduced in Theorem 26.8 is employed. As the output is a local minima of $Q(\theta)$, these points can be identified by determining $Q'(\hat{\theta})$, which, after some manipulations, can be written as

$$Q'(\theta) = 2 \sum_{i=1}^{N} \frac{\left(\frac{\theta - x(i)}{s(i)^2} \right)}{1 + \left(\frac{x(i) - \theta}{s(i)} \right)^2}. \tag{26.39}$$

Defining $\psi(\upsilon) \stackrel{\Delta}{=} \frac{2\upsilon}{1+\upsilon^2}$, the following equation is obtained for the local extrema of $Q(\theta)$:

$$Q'(\theta) = -\sum_{i=1}^{N} \frac{1}{s(i)} \cdot \psi \left(\frac{x(i) - \theta}{s(i)} \right) = 0. \tag{26.40}$$

By introducing the positive functions

$$h(i;\theta)\stackrel{\Delta}{=}\frac{1}{s(i)^2}\cdot\varphi\left(\frac{x(i)-\theta}{s(i)}\right)>0, \tag{26.41}$$

for $i=1,2,\ldots,N$, where $\varphi(v)\stackrel{\Delta}{=}\frac{\psi(v)}{v}=\frac{2}{1+v^2}$, the local extrema of $Q(\theta)$ in Equation 26.40 can be formulated as

$$Q'(\theta)=-\sum_{i=1}^{N}h(i;\theta)\cdot(x(i)-\theta)=0. \tag{26.42}$$

This formulation implies that the *sum of weighted deviations* of the samples is zero, with the (positive) weights themselves being functions of θ. This property, in turn, leads to a simple fixed point iterative procedure for computing the myriad weights.

To develop the fixed point algorithm, note that Equation 26.42 can be written as

$$\theta=\frac{\sum_{i=1}^{N}h(i;\theta)\cdot x(i)}{\sum_{i=1}^{N}h(i;\theta)}, \tag{26.43}$$

which is a weighted mean of the input samples, $x(i)$. Since the weights $h(i;\theta)$ are always positive, the right-hand side of Equation 26.43 is in $[x[1],x[N]]$, confirming that all the local extrema lie within the range of the input samples. By defining the mapping

$$T(\theta)\stackrel{\Delta}{=}\frac{\sum_{i=1}^{N}h(i;\theta)\cdot x(i)}{\sum_{i=1}^{N}h(i;\theta)}, \tag{26.44}$$

the local extrema of $Q(\theta)$, or the roots of $Q'(\theta)$, are seen to be the *fixed points* of $T(\cdot)$:

$$\theta^\star=T(\theta^\star). \tag{26.45}$$

The following fixed point iteration results in an efficient algorithm to compute these fixed points:

$$\theta_{m+1}\stackrel{\Delta}{=}T(\theta_m)=\frac{\sum_{i=1}^{N}h(i;\theta_m)\cdot x(i)}{\sum_{i=1}^{N}h(i;\theta_m)}. \tag{26.46}$$

In the classical literature, this is also called the *method of successive approximation* for the solution of the equation $\theta=T(\theta)$. This method is proven to converge to a fixed point of $T(\cdot)$, indicating that

$$\lim_{m\to\infty}\theta_m=\theta^\star=T(\theta^\star). \tag{26.47}$$

The speed of convergence depends on the initial value θ_0. A simple approach to initializing the algorithm is to set $\hat{\theta}_0$ equal to the input sample $x(i)$ that yields the smallest cost $P(x(i))$, where $\log(P(\theta))\stackrel{\Delta}{=}Q(\theta)$, i.e., $P(\theta)=\prod_{i=1}^{N}\left(\gamma^2+w(i)(x(i)-\theta)^2\right)$. The fixed point weighted myriad optimization can now be summarized as follows.

Fixed point weighted myriad search

Step 1: Select the initial point $\hat{\theta}_0$ among the values of the input samples:

$$\hat{\theta}_0 = \arg\min_{x(i)} P(x(i)).$$

Step 2: Using $\hat{\theta}_0$ as the initial value, perform L iterations of the fixed point recursion $\theta_{m+1} = T(\theta_m)$, the full expression of which is given in Equation 26.46. The final value of these iterations is then chosen as the weighted myriad output, $\hat{\theta} = T^{(L)}(\hat{\theta}_0)$.

This algorithm is compactly written as

$$\hat{\theta} = T^{(L)}\left(\arg\min_{x(i)} P(x(i))\right). \tag{26.48}$$

Note that for the special case $L = 0$, no fixed point iterations are performed and the above algorithm computes the selection weighted myriad.

26.3.3 Running Meridian Smoothers: Meridian Filtering

Having established the myriad as the optimal filtering operation that arises from the location estimation problem in Cauchy distributed samples, we consider a second special case of the GCD, namely the Meridian distribution. This special case ($\lambda = 1$) has even heavier tails than the Cauchy distribution, and is an appropriate model for the most impulsive sequences seen in practice. As the Cauchy and Meridian distributions fall within the GCD, the development for the Meridian is similar to that for the Cauchy and so many proofs with similar steps are omitted.

THEOREM 26.9

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Myriad distribution with common location* θ *and scale* δ. *The ML estimate of location, or sample meridian, is given by*

$$\hat{\theta} = \arg\min_{\theta}\left[\sum_{i=1}^{N} \log\{\delta + |x(i) - \theta|\}\right] = \text{meridian}\{x(i) : i = 1, 2, \ldots, N; \delta\}, \tag{26.49}$$

where δ *is referred to as the medianity parameter.*

As in all previous cases, the performance of the meridian filter is directly related to the defining objective function. Properties describing the objective function are given in the following proposition. The properties described therein are illustrated in Figure 26.9, which plots an example of the objective function, $Q(\theta)$, that results from a set of typical observation samples in the $N = 7$ case.

PROPOSITION 26.3

Let

$$Q(\theta) \stackrel{\Delta}{=} \sum_{i=1}^{N} \log\{\delta + |x(i) - \theta|\}. \tag{26.50}$$

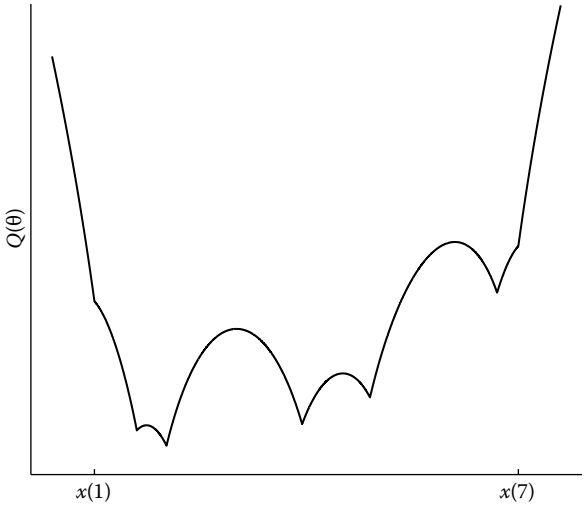


FIGURE 26.9 Typical plot of the objective function minimized by the meridian operator (for $\delta = 0.5$ and randomly generated $N = 7$ samples). (From Aysal, T. C. and Barner, K. E., *IEEE Trans. Signal Process.*, 55, 3949, 2007. With permission.)

The following statements hold:

(1) $Q'(\theta) > 0$ ($Q(\theta)$ *is strictly increasing*) *for* $\theta > x[N]$, *and* $Q'(\theta) < 0$ ($Q(\theta)$ *is strictly decreasing*) *for* $\theta < x[1]$.
(2) *All local extrema of* $Q(\theta)$ *lie within the range of input samples*, $[x[1], x[N]]$.
(3) *Objective function* $Q(\theta)$ *has a finite number of local minima* (*at most equal to the number of input samples*, N).
(4) *Meridian* $\hat{\theta}$ *is one of the local minima of* $Q(\theta)$, *i.e., one of the input samples.*

The meridian estimator output is hence the input sample that yields the smallest $Q(\theta)$ function value. The selective nature of the meridian estimator, shared with the median estimator, facilitates the filter output computation which is formulated as

$$\hat{\theta} = \arg\min_{\theta \in \mathbf{x}} \sum_{i=1}^{N} \log\{\delta + |x(i) - \theta|\}. \tag{26.51}$$

The behavior of the meridian operator is markedly dependent on the value of δ, which is referred to as the *medianity* parameter. As the name suggests, we show in the following that the sample meridian is equivalent to the sample median for large values of δ, whereas the estimator acquires the form of the sample mode for small δ.

COROLLARY 26.5

Given a set of samples $\{x(i): i = 1, 2, \ldots, N\}$, *the sample meridian* $\hat{\theta}$ *converges to the sample median as* $\delta \to \infty$. *That is*,

$$\lim_{\delta \to \infty} \hat{\theta} = \lim_{\delta \to \infty} \text{meridian}\{x(i) : i = 1, 2, \ldots, N; \delta\} = \text{median}\{x(i): i = 1, 2, \ldots, N\}. \tag{26.52}$$

Thus the family of meridian estimators subsumes the sample median as a limiting case. This simple fact makes the meridian filter class inherently more efficient than (or at least equally efficient to) median filters over all noise distribution including the Laplacian. The opposite limiting case, which results in the mode–meridian operator, is addressed in the following.

COROLLARY 26.6

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$, *the sample meridian* $\hat{\theta}$ *converges to a mode–type estimator as* $\delta \to 0$. *That is,*

$$\lim_{\delta \to 0} \hat{\theta} = \arg\min_{x(j) \in \mathcal{M}} \left[\prod_{i=1,\, x(i) \neq x(j)}^{N} |x(i) - x(j)| \right], \tag{26.53}$$

where $\mathcal{M}$ *is the set of most repeated values.*

Thus the myriad and meridian operators converge to a common mode operator as their linearity/medianity parameters go to 0.

Since all the operators covered in this chapter belong to the class of *M-estimators* [10], many robust statistics tools are available for evaluating their robustness [10,48]. *M*-estimators are formulated by a set of implicit functions, where $\rho(\cdot)$ is an arbitrary objective function. Assuming that $\psi(x) = \partial(\rho(x))/\partial x$ exists, the *M-estimator* is obtained by solving

$$\sum_{i=1}^{N} \psi(x(i) - \theta) = 0, \tag{26.54}$$

where $\psi(\cdot)$ is proportional to the so-called influence function. The influence function of an estimator is important in that it determines the effect of contamination on the estimator.

To further characterize *M*-estimates, it is useful to list the desirable features of a robust influence function [10,48]:

- B-robustness. An estimator is *B-robust* if the supremum of the absolute value of the influence function is finite.
- Rejection Point. The *rejection point* defined as the distance from the center of the influence function to the point where the influence function becomes negligible, should be finite. The rejection point measures whether the estimator rejects outliers and, if so, at what distance.

The influence functions for the sample mean, median, myriad, and meridian can be shown to be

$$\psi(x) = 2x, \tag{26.55}$$

$$\psi(x) = \text{sgn}(x), \tag{26.56}$$

$$\psi(x) = \frac{2x}{\gamma^2 + x^2}, \tag{26.57}$$

$$\psi(x) = \frac{\text{sgn}(x)}{\delta + |x|}, \tag{26.58}$$

respectively. Figure 26.10 plots each of these for comparison. Since the influence function of the mean is unbounded, a gross error in the observations leads to severe distortion in the estimate. The mean is

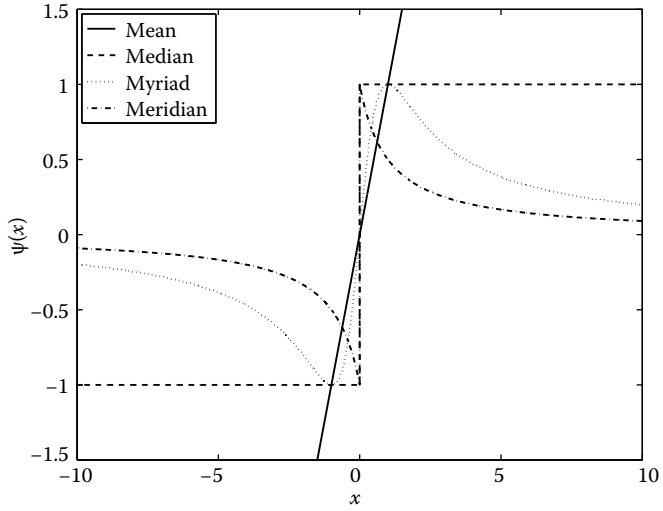


FIGURE 26.10 Influence functions for (solid:) the sample mean, (dashed:) the sample median, (dotted:) the sample myriad, and (dash–dotted:) the sample meridian. (From Aysal, T. C. and Barner, K. E., *IEEE Trans. Signal Process.*, 55, 3949, 2007. With permission.)

clearly not B-robust and its rejection point is infinite. On the other hand, a similar gross error has a limited effect on the median estimate.

While the median is B-robust, its rejection point, like the mean, is not finite. Thus the median estimate is always affected by outliers. The myriad estimate is clearly B-robust and the effect of the errors decreases as the error increases. The meridian estimate is also B-robust, and, in addition, its rejection point is smaller than that of myriad. This indicates that the operators can be ordered from least to most robust as: linear, median, myriad, and meridian.

In addition to desirable influence function characteristics, the meridian possesses the following properties important in signal processing applications.

PROPERTY 26.4

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$, let $\delta < \infty$. Then the meridian output is such that

$$\lim_{x(N)\to\pm\infty} \hat{\theta}(x(1), x(2), \ldots, x(N)) = \hat{\theta}(x(1), x(2), \ldots, x(N-1)). \tag{26.59}$$

According to Property 26.4, large errors are efficiently eliminated by a meridian estimator with a finite medianity parameter. Note that this is not the case for the median estimate, as large positive or negative values can always shift the output. In robust statistics, *M*-estimators satisfying the outlier rejection property are called redescending. It can be proven that a necessary and sufficient condition for an *M*-estimator to be redescending is that $\lim_{x\to\pm\infty} \partial/\partial x(\rho(x)) = 0$. Note that this condition holds for meridian and myriad operators, whereas it does not for the mean and median. The meridian also possesses the same no undershoot/overshoot, shift and sign invariance, and unbiasedness properties as the myriad.

PROPERTY 26.5 (No Undershoot/Overshoot)

The output of the meridian estimator operating on samples $\{x(i) : i = 1, 2, \ldots, N\}$ is always bounded by

$$x[1] \leq \hat{\theta}(\mathbf{x}) \leq x[N]. \tag{26.60}$$

PROPERTY 26.6 (Shift and Sign Invariance)

Consider the observation set $\{x(i) : i = 1, 2, \ldots, N\}$ and let $z(i) = x(i) + b$. Then,

(1) $\hat{\theta}(\mathbf{z}) = \hat{\theta}(\mathbf{x}) + b$;
(2) $\hat{\theta}(\mathbf{z}) = -\hat{\theta}(-\mathbf{z})$.

PROPERTY 26.7 (Unbiasedness)

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$ that are independent and symmetrically distributed around a symmetry center c, $\hat{\theta}(\mathbf{x})$ is also symmetrically distributed around c. In particular, if $\mathbb{E}\{\hat{\theta}(\mathbf{x})\}$ exists, then $\mathbb{E}\{\hat{\theta}(\mathbf{x})\} = c$.

The meridian characteristics can be broadened through the introduction of weights. The weighted meridian possesses the same properties as the unweighted version and converges to the expected special cases in the limit of the medianity parameter. The weighted case is formally defined and the limiting cases stated, but the properties are omitted due to their direct similarity to the previous formulations.

THEOREM 26.10

Consider a set of N independent samples $\{x(i) : i = 1, 2, \ldots, N\}$ *each obeying the Meridian distribution with common location* θ *and (possibly) varying scale parameters* $v(i) = \delta/w(i)$. *The ML estimate of location, or weighted meridian, is given by*

$$\hat{\theta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \log\{\delta + w(i)|x(i) - \theta|\} \right] = \text{meridian}\{w(i) \star x(i) : i = 1, 2, \ldots, N\}, \tag{26.61}$$

where $\star$ *denotes the weighting operation in the minimization problem.*

COROLLARY 26.7

Given a set of samples $\{x(i) : i = 1, 2, \ldots, N\}$ *and corresponding (positive) weights* $\{w(i): i = 1, 2, \ldots, N\}$, *the weighted meridian* $\hat{\theta}$ *converges to the weighted median as* $\delta \to \infty$. That is,

$$\lim_{\delta \to \infty} \hat{\theta} = \lim_{\delta \to \infty} \text{meridian}\{w(i) \star x(i) : i = 1, 2, \ldots, N; \delta\} = \text{median}\{w(i) \diamond x(i) : i = 1, 2, \ldots, N\}. \tag{26.62}$$

COROLLARY 26.8

Given a set of samples $\{x(i): i = 1, 2, \ldots, N\}$ *and corresponding (positive) weights* $\{w(i): i = 1, 2, \ldots, N\}$, *the weighted meridian* $\hat{\theta}$ *converges to one of the most repeated values in the sample set as* $\delta \to 0$. *That is*,

$$\lim_{\delta \to 0} \hat{\theta} = \arg\min_{x_j \in \mathcal{M}} \left[\left(\frac{1}{w(j)} \right)^r \prod_{i=1,\, x(i) \neq x(j)}^{N} |x(i) - x(j)| \right], \tag{26.63}$$

where $\mathcal{M}$ *is the set of most repeated values* and r *is the number of times a member of* $\mathcal{M}$ *is repeated in the sample set.*

Table 26.1 summarizes the *M*-estimators and filters covered in this chapter. The GGD derived operators, for $\beta = 2$ and $\beta = 1$ are reported on table rows 1, 2 and 5, 6, while the GCD optimal operators, for $\lambda = 2$ and $\lambda = 1$, are reported on rows 3, 4 and 7, 8. This table brings together and contrast the objective functions for the GGD-based linear and median filters and the GCD-based myriad and meridian filters.

26.3.4 Introduction of Real-Valued Weights and Optimization

The coverage to this point has enforced the positivity constraint that arises naturally in each of the operators' estimation-based development. Only in the median case is the sign coupling approach utilized to allow the introduction of real-valued (positive and negative) weights. See Equations 26.13 and 26.14. For convenience, Equation 26.13 is repeated here:

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} |h(i)| \cdot |\mathrm{sgn}(h(i))x(i) - \theta| \right] = \mathrm{median}\{|h(i)| \ \diamond \ \mathrm{sgn}(h(i))x(i)|_{i=1}^{N}\}. \tag{26.64}$$

TABLE 26.1 *M*-Estimator and M-Smoother (Weighted *M*-Estimator) Objective Functions and Outputs for Various Filter Families

Smoother	Cost Function	Filter Output
Mean	$\sum_{i=1}^{N} (x(i) - \theta)^2$	$\mathrm{Mean}\{x(i): i = 1, 2, \ldots, N\}$
Median	$\sum_{i=1}^{N} \lvert x(i) - \theta \rvert$	$\mathrm{Median}\{x(i): i = 1, 2, \ldots, N\}$
Myriad	$\sum_{i=1}^{N} \log\{\gamma^2 + (x(i) - \theta)^2\}$	$\mathrm{Myriad}\{x(i): i = 1, 2, \ldots, N; \gamma\}$
Meridian	$\sum_{i=1}^{N} \log\{\delta + \lvert x(i) - \theta \rvert\}$	$\mathrm{Meridian}\{x(i): i = 1, 2, \ldots, N; \delta\}$
Weighted mean	$\sum_{i=1}^{N} w(i)(x(i) - \theta)^2$	$\mathrm{Mean}\{w(i) \cdot x(i): i = 1, 2, \ldots, N\}$
Weighted median	$\sum_{i=1}^{N} w(i)\lvert x(i) - \theta \rvert$	$\mathrm{Median}\{w(i) \diamondsuit x(i): i = 1, 2, \ldots, N\}$
Weighted myriad	$\sum_{i=1}^{N} \log\{\gamma^2 + w(i)(x(i) - \theta)^2\}$	$\mathrm{Myriad}\{w(i) \circ x(i): i = 1, 2, \ldots, N; \gamma\}$
Weighted meridian	$\sum_{i=1}^{N} \log\{\delta + w(i)\lvert x(i) - \theta \rvert\}$	$\mathrm{Meridian}\{w(i) \star x(i): i = 1, 2, \ldots, N; \delta\}$

Source: Aysal, T. C. and Barner, K. E., *IEEE Trans. Signal Process.*, 55, 3949, 2007. With permission.

The most general cases of the myriad and meridian are, by direct analogy, given by

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} \log\{\gamma^2 + |h(i)| \cdot (\mathrm{sgn}(h(i))x(i) - \theta)^2\} \right] = \mathrm{myriad}\{|h(i)| \circ \mathrm{sgn}(h(i))x(i)|_{i=1}^{N}\} \quad (26.65)$$

and

$$\hat{\theta} = \arg\min_{\theta} \left[\sum_{i=1}^{N} \log\{\delta + |h(i)| \cdot |\mathrm{sgn}(h(i))x(i) - \theta|\} \right] = \mathrm{meridian}\{|h(i)| \star \mathrm{sgn}(h(i))x(i)|_{i=1}^{N}\}, \quad (26.66)$$

respectively. Under this formulation, the operators are no longer restricted to be smoothers and can, therefore, be set to exhibit a wide range of frequency-selective filtering operations, including bandpass and highpass filtering.

Also of importance is that the filter weights can be optimized. This is most easily done through a training process, typically under the mean absolute error (MAE). For instance, using a steepest descent approach yields a relatively simple weight update for the myriad case,

$$h(i; n+1) = h(i; n) - \mu e(n) \left[\frac{\gamma^2 \mathrm{sgn}(h(i)) \left(\hat{\theta} - \mathrm{sgn}(h(i))x(i)\right)}{\left(\gamma^2 + |h(i)| \cdot \left(\hat{\theta} - \mathrm{sgn}(h(i))x(i)\right)^2\right)^2} \right], \quad (26.67)$$

where $e(n)$ is simply the error between the filter output and the desired signal and μ is the iteration step size.

It is important to note that the optimal filtering action is independent of the choice of γ; the filter only depends on the value of $w/(\gamma^2)$. Note that similar updates can be derived for the median and meridian filters—all are analagous to the LMS update for the linear filter.

For cases in which no training sequence is available, there are simple, suboptimal approaches to setting the operator parameters. Recall that the myriad has the linearization free parameter (γ), while the meridian has medianization free parameter (δ). These parameters control the robustness of the operators. Numerous optimization techniques exist for linear and median filters. Thus one approach to setting the filter parameters is to design weights for the simpler linear or median filter that address the problem at hand, then decrease the free parameter value to achieve the desired level of additional robustness. This is referred to as myriadization in the myriad case and meridianization in the meridian case. This approach allows the rich set of design tools in the more traditional filtering domains to be applied to the newer, more robust operators.

26.4 Applications of Robust Nonlinear Estimators/Filters

The covered filtering methods are evaluated in two communications problems, standard baseband communications and power line communications, as well as an illustrative frequency selective filtering problem. In each of these applications, an underlying desired signal is to be extracted. Complicating the extraction is additive noise. While many different noise distributions can be investigated and justified by various problem physics, we avoid direct use of either GGD or GCD densities. Employing, for instance, Gaussian noise will yield the obvious result of linear filtering providing the best performance, while the myriad filter is clearly optimal for applications in which the noise is Cauchy distributed. The presentation of such obvious results provides little information or insight into the characteristics of the individual

algorithms. Rather, results are presented for the commonly utilized α-Stable density family. This provides a fairer comparison and illustrates the performance of the filters operating on a widely used heavy-tailed distribution, but one that shares only tail order with the GCD.

Utilization of α-Stable distributed samples, like Gaussian distributed samples, is very appealing for the modeling of real-life phenomena because they constitute the only variables that can be described as the superposition of many small independent and identically distributed effects [15]. The class of symmetric α-Stable distributions is usually described by its characteristic function:

$$\phi(\omega) = \exp\left(-\kappa|\omega|^{\alpha}\right). \tag{26.68}$$

The parameter κ, which is commonly called the *dispersion*, is a positive constant related to the scale of the distribution and $\kappa^{1/\alpha}$ is the scale parameter of the distribution. In order for Equation 26.68 to define a characteristic function, the value of α must be restricted to the interval [0, 2]. Importantly, α determines the impulsiveness, or tail heaviness, of the distribution, where smaller values of α indicate increased levels of impulsiveness. The limit $\alpha = 2$ case corresponds to the zero-mean Gaussian distribution with variance 2κ. All other values of α correspond to impulsive distributions with infinite variance and algebraic tail behavior [15]. Indeed, α-Stable distributions are successfully utilized to model impulsive environments accross a wide array of applications [8,12,15,23,28,31,34,46,49].

26.4.1 Baseband Communications

Consider first the baseband communication model given in Figure 26.11 [50]. Suppose that A (real) is to be communicated over the channel. Denote $s(t)$ as the combined impulse response of the transmitter and channel, and take the pulse $As(t)$ to be corrupted by additive white noise. The received pulse is then given by

$$r(t) = As(t) + n(t), \tag{26.69}$$

which, after sampling at rate $1/T$, corresponds to the sequence

$$r(kT) = As(kT) + n(kT). \tag{26.70}$$

Taking the common case assumption that $s(kT) \neq 0$ only for k an integer, the communications goal is to estimate A using the samples $As(kT) + n(kT), k = 1, 2, \ldots, K$.

The formulated problem is well known to be optimally addressed through matched filtering. Moreover, each of the covered filters can be formulated in matched form. Thus to compare the performances of the matched linear, median, myriad, and meridian filters, 10,000 Gaussian distributed $\{A(i) : i = 1, 2, \ldots, 10{,}000\}$ parameters are generated, sent through the baseband communication channel, sampled with $K = 21$, and filtered with each of the matched filters to obtain the estimates $\{\hat{A}(i) : i = 1, 2, \ldots, 10{,}000\}$. The corrupting channel noise is α-Stable distributed with $\alpha = 0.4$. The

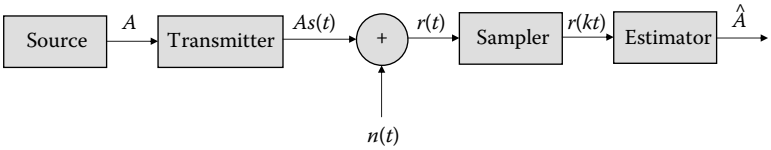


FIGURE 26.11 Baseband communication model: $s(t)$ denotes the combined impulse response of the transmitter and the channel, and $As(t)$ is corrupted by additive white noise $n(t)$.

TABLE 26.2 Matched Linear, Median, Myriad and Meridian Filters Output MAEs and MSEs

Performance Criteria	Matched Linear	Matched Median	Matched Myriad	Matched Meridian
MAE	5.9215×10^3	0.2080	0.1605	0.1121
MSE	3.2949×10^7	0.1687	0.0520	0.0380

Source: Aysal, T. C. and Barner, K. E., *IEEE Trans. Signal Process.*, 55, 3949, 2007. With permission.

pulse carrying the symbol is taken to be rectangular. The mean absolute and squared errors of the matched filter outputs are tabulated in Table 26.2. The tabulated results show that the linear filter completely breaks down in the presence of the heavy tailed corruption. The median is significantly more robust, but still suffers from its inability to completely discount outliers. The myriad and meridian operators, both derived for such algebraic tailed environments, provide significantly better results than the GGD-based methods. The noise parameter utilized in this example, $\alpha = 0.4$, results in extremely impulsive corruption, which is why the best performance is provided by the most robust operator covered, the matched meridian filter.

26.4.2 Power Line Communications

Consider next the problem of power line communications (PLCs). The use of existing power lines for transmitting data and voice has received considerable interest in recent years [51–54]. The advantages of PLCs are obvious due to the ubiquity of power lines and power outlets. The potential of power lines to deliver broadband services, such as fast Internet access, telephone, fax services, and home networking is emerging in new communications industry technology. However, there remain considerable challenges for PLCs, such as communications channels that are hampered by the presence of large amplitude noise superimposed on top of traditional white Gaussian noise. The overall interference is appropriately modeled as an algebraic tailed process, with α-Stable ($\alpha \approx 1$) often chosen as the parent distribution [51].

To compare the various filtering algorithms, consider a PLC problem in which there are a set of voltage levels, $\mathcal{V} = \{-2, -1, 0, 1, 2\}$ unknown at the receiver.* A signal randomly composed of these voltage levels, an example of which is shown in Figure 26.12a, is transmitted through a powerline. The observed signal is given in Figure 26.12b, where the noise is modeled as α-Stable distributed with $\alpha = 1.25$. Note that this results in corruption that is somewhat less impulsive than in the previously considered example. The observed powerline signal is processed with mean, median, myriad, and meridian filters, each utilizing window length $N = 9$, the results of which are given in Figures 26.12c through, respectively.

Inspection of the figures shows that the mean is vulnerable to impulsive noise even when α is relatively large. The median, myriad, and meridian, in contrast, all perform relatively well. The fact that the median performs well in this example indicates that the noise impulsivity is reasonably close to the GGD Laplacian distribution special case. The meridian is, perhaps, overly robust for the level of corruption observed in this example, but still performs nearly as well as the myriad and median. This example shows that the myriad and meridian give up little in terms of performance in relatively light tailed environments while, as the previous example shows, offering considerably better performance in the presence of truly heavy tailed observations.

26.4.3 Highpass Filtering of a Multitone Signal

As a final example, consider the problem of preserving a high-frequency tone while removing all low frequency terms. This requires highpass filtering, which employs both positive and negative weights within the filter window. Such a problem demonstrates not only the utilization of positive and negative

* In the case where the signal alphabet is known at the receiver, the linear, median, myriad, and meridian estimators are readily extended to detectors.

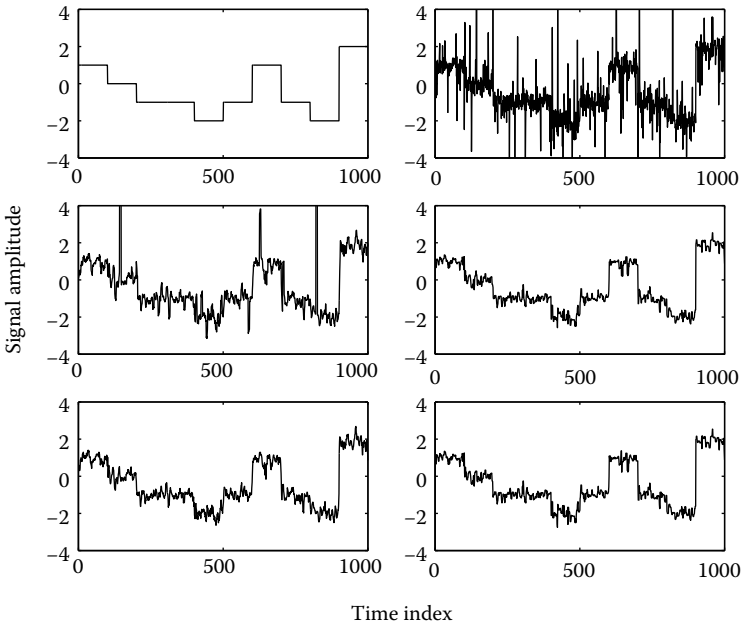


FIGURE 26.12 Power line communications signal enhancement: (a) transmitted signal, (b) observed signal corrupted by $\alpha = 1.25$ noise, output of the (c) mean [2.5388]{8.6216}, (d) median [2.4267]{7.4393}, (e) myriad [2.4385]{7.4993}, and (f) meridian filters [2.4256]{7.4412}, where $[\cdot]$ and $\{\cdot\}$ denotes the mean absolute and squared error for the corresponding filtering structure.

weights, but also the ability of robust nonlinear filters to perform frequency selectivity that is traditionally accomplished through linear filtering. Figure 26.13a depicts a two-tone signal with normalized frequencies 0.02 and 0.4 Hz. The signal is 1000 samples long, although a cropped version of the original signal $\{0,200\}$ is shown for presentation purposes. Figure 26.13b shows the multitone signal filtered by a 40-tap linear FIR filter designed by the MATLAB fir1 command with a normalized cutoff frequency of 0.3 Hz. The myriad and median filter weights are optimized in this case utilizing the adaptive methods detailed in the previous section.* Noting that the median filter is a limiting case of the meridian filter, the meridian filter weights are set equal to those of the median.

The clean multitone input signal is corrupted by additive α-Stable noise with $\alpha = 0.4$ to form a corrupted observation from which the single high frequency tone must be extracted, Figure 26.13c. The noisy multitone signal is processed by the linear, median, myriad, and meridian filters with the results given in Figures 26.13d through g, respectively. Similarly to the first example, the observation signal in this case contains very heavy tailed outliers. These outliers cause the linear filter to, once again, break down. Each of the nonlinear filters, in contrast, offers more robust processing, with the outputs more closely reflecting the range and frequency content of the desired signal. The increasing robustness of the operators is again apparent, with the myriad being more robust to outliers than the median and the meridian being the most robust. It should be noted that the nonlinear filters not only minimize the influence of outliers, but their flexible weighted structures are able to effectively pass desired frequency content while rejecting content outside the desired frequency band. The ability to simultaneously reject

* The clean multitone signal and the desired high-frequency signal are utilized as the input and desired signals, respectively. For more detail see Refs. [28,29,31].

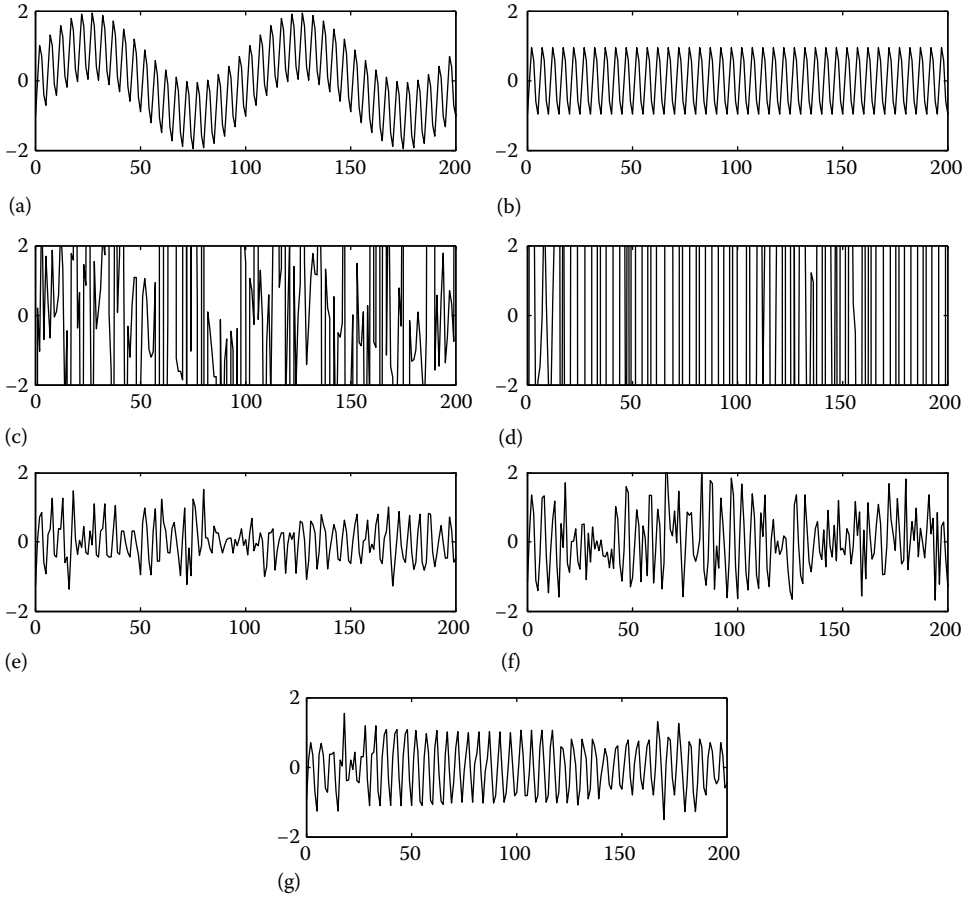


FIGURE 26.13 High frequency–selection filtering in impulsive noise: (a) clean two-tone input signal, (b) (desired) highpass component of the input signal (processed with lowpass FIR filter), (c) two-tone signal corrupted with stable noise ($\alpha = 0.4$), noisy two-tone input signal processed with (d) linear, (e) median, (f) myriad, and (g) meridian filters. (From Aysal, T. C. and Barner, K. E., *IEEE Trans. Signal Process.*, 55, 3949, 2007. With permission.)

outliers and perform frequency selectivity enables the nonlinear median, myriad, and meridian filters to be deployed in a wide range of applications, including those previously dominated by, but not necessarily well addressed by, linear filtering.

26.5 Concluding Remarks and Future Directions

The development and application of nonlinear filters is necessitated by the complexity and challenging environments of many contemporary problems. In this chapter we have taken a fundamental approach to filter development, considering operators from the first principles of location estimation. This development focused on two important broad classes of distributions, generalized Gaussian and generalized Cauchy distribution families. Within the GGD family, particular focus was placed on the Gaussian and Laplacian distribution special cases and their resulting linear and median filtering operations. It was shown that while the GGD family constitutes a broad array of distributions with varying tail parameters, distributions within the family are not capable of accurately modeling the most impulsive and environment seen in practice due to the constraint that GGD density tail decay rates be exponential. The consequence of this tail decay rate constraint is that the linear and median filters deriving their optimality from the

GGD are not effective in the processing of very impulsive sequences. The GCD family, in contrast, consists of distributions with algebraic tail decay rates that do accurately modeled these most demanding impulsive environments. Within this family we again focus on two special cases, the Cauchy and Meridian distributions and their resulting myriad and meridian filtering operations. As these filters are derived from heavy tailed distributions, they are well suited for applications dominated by very impulsive statistics.

Properties and optimization procedures are presented for each of the filters covered. In particular, we show that the operators can be ordered in terms of their robustness, from least to most robust, as linear, median, myriad, and meridian. Moreover, myriad filters contain linear filters as special cases, while meridian filters contain median operators as special cases. Thus the myriad and meridian operators are inherently more efficient than their traditional (subset) counterparts. Simulations presented in communications and frequency selective filtering applications show and contrast the performances of the filters in applications with varying levels of heavy tailed statistics. As expected, linear operators breakdown in these environments while the robust, nonlinear operators yield desirable results.

Although the presentation in this chapter ranges from theoretical development through properties, optimization, and applications, the coverage is, in fact, simply an overview of one segment within the broad array of nonlinear filtering algorithms. To probe further, the interested reader is referred to the cited articles, as well as numerous other works in this area. Additionally, there are many other areas of research in nonlinear methods that are under active investigation. Research areas of importance include (1) order-statistic based signal processing, (2) mathematical morphology, (3) higher order statistics and polynomial methods, (4) radial basis function and kernel methods, and (5) emerging nonlinear methods. Researchers and practitioners interested in the broader field of nonlinear signal processing are encouraged to see the many good books, monographs, and research papers covering these, and other, areas in nonlinear signal processing.

References

1. B. I. Justusson, Median filtering: statistical properties, *Two Dimensional Digital Signal Processing II*. New York: Springer Verlag, 1981.
2. I. Pitas and A. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Application*. New York: Kluwer Academic, 1990.
3. K. E. Barner and G. R. Arce, Eds., *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*. Boca Raton, FL: CRC Press, 2004.
4. M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.
5. C. L. Nikias and A. P. Petropulu, *Higher Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
6. V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York: John Wiley & Sons, Inc., 2000.
7. M. B. Priestrley, *Non-linear and Non-stationary Time Series Analysis*. New York: Academic, 1988.
8. G. R. Arce, *Nonlinear Signal Processing: A Statistical Approach*. New York: John Wiley & Sons, Inc., 2005.
9. H. A. David and H. N. Nagaraja, *Order Statistics*. New York: John Wiley & Sons, Inc., 2003.
10. P. Huber, *Robust Statistics*. New York: John Wiley & Sons, Inc., 1981.
11. H. Hall, A new model for impulsive phenomena: application to atmospheric noise communications channels, Stanford University Electronics Laboratories Technical Report, no. 3412–8 and 7050–7, 1966, sU-SEL-66-052.
12. J. Ilow, Signal proceesing in alpha–stable noise environments: Noise modeling, detection and estimation, PhD dissertation, University of Toronto, Toronto, ON, 1995.
13. B. Mandelbrot, Long-run linearity, locally Gaussian processes, H-spectra, and infinite variances, *International Economic Review*, 10, 82–111, 1969.

14. J. H. Miller and J. B. Thomas, Detectors for discrete-time signals in non-Gaussian noise, *IEEE Transactions on Information Theory*, 18, 241–250, Mar. 1972.
15. C. L. Nikias and M. Shao, *Signal Processing with Alpha–Stable Distributions and Applications*. New York: Wiley, 1995.
16. G. A. Tsihrintzis and C. L. Nikias, Fast estimation of the parameters of alpha-stable impulsive interference, *IEEE Transactions on Signal Processing*, 44(6), 1492–1503, Jun. 1996.
17. A. G. Dimakis and P. Maragos, Phase-modulated resonances modeled as self-similar processes with application to turbulent sounds, *IEEE Transactions on Signal Processing*, 53(11), 4261–4272, Nov. 2005.
18. G. A. Tsihrintzis and C. L. Nikias, Data-adaptive algorithms for signal detection in sub-Gaussian impulsive interference, *IEEE Transactions on Signal Processing*, 45(7), 1873–1878, Jul. 1997.
19. J. P. Nolan, *Stable Distributions: Models for Heavy Tailed Data*. Boston, MA: Birkhuser, 2005.
20. G. A. Tsihrintzis and C. L. Nikias, Incoherent receivers in alpha-stable impulsive noise, *IEEE Transactions on Signal Processing*, 43(9), 2225–2229, Sep. 1995.
21. J. Ilow and D. Hatzinakos, Analytic alpha-stable noise modeling in a Poisson field of interferers or scatterers, *IEEE Transactions on Signal Processing*, 46(6), 1601–1611, Jun. 1998.
22. R. F. Brcich, D. Iskander, and A. Zoubir, The stability test for symmetric alpha-stable distributions, *IEEE Transactions on Signal Processing*, 53(3), 997–986, Mar. 2005.
23. E. Masry, Alpha-stable signals and adaptive filtering, *IEEE Transactions on Signal Processing*, 48(11), 3011–3016, 2000.
24. R. Durrett, *Probability: Theory and Examples*, 2nd edn., Belmont, CA: Duxbury Press, 1996.
25. V. M. Zolotarev, *One Dimensional Stable Distributions*. Translations of Mathematical Monographs 65, American Mathematical, Providence, Rhode Island, 1986.
26. S. Kalluri and G. R. Arce, Adaptive weighted myriad filter algorithms for robust signal processing in α–Stable environments, *IEEE Transactions on Signal Processing*, 46(2), 322–334, Feb. 1998.
27. J. G. Gonzalez and G. R. Arce, Optimality of the myriad filter in practical impulsive-noise environments, *IEEE Transactions on Signal Processing*, 49(2), 438–441, Feb. 2001.
28. S. Kalluri and G. R. Arce, Robust frequency–selective filtering using weighted myriad filters admitting real-valued weights, *IEEE Transactions on Signal Processing*, 49(11), 2721–2733, Nov. 2001.
29. T. C. Aysal and K. E. Barner, Meridian filtering for robust signal processing, *IEEE Transactions on Signal Processing*, 55(8), 3949–3962, Aug. 2007.
30. S. A. Kassam and H. V. Poor, Robust techniques for signal processing, *Proceedings of IEEE*, 73, 433–481, Mar. 1985.
31. G. R. Arce, A general weighted median filter structure admitting negative weights, *IEEE Transactions on Signal Processing*, 46, 3195–3205, Dec. 1998.
32. R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, Optimal weighted median filters under structural constraints, *IEEE Transactions on Signal Processing*, 43(3), 591–604, 1995.
33. L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, Weighted median filters: A tutorial, *IEEE Transactions on Circuits and Systems*, 41, 157–192, May 1996.
34. K. E. Barner and T. C. Aysal, Polynomial weighted median filtering, *IEEE Transactions on Signal Processing*, 54(2), 636–650, Feb. 2006.
35. T. C. Aysal, Filtering and estimation theory: First-order, polynomial and decentralized signal processing, PhD dissertation, University of Delaware, DE, 2007.
36. T. C. Aysal and K. E. Barner, Hybrid polynomial filters for gaussian and non-gaussian noise environments, *IEEE Transactions on Signal Processing*, 54(12), 4644–4661, Dec. 2006.
37. R. Adler, R. Feldman, and E. M. S. Taqqu, *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy-Tailed Distributions*. Boston, MA: Birkhauser, 1997.
38. E. J. Wegman, S. G. Schwartz, and J. Thomas, *Topics in Non–Gaussian Signal Processing*. New York: Academic Press, 1989.

39. G. R. Wilson and D. R. Powell, Experimental and modelled density estimates of underwater acoustic returns, In *Statistical Signal Processing*, E. J. Wegman and G. Smith, Eds., Marcel Decker, New York, pp. 223–239, 1984.
40. A. Briassouli and M. G. Strintzis, Locally optimum nonlinearities for DCT watermark detection, *IEEE Transactions on Image Processing*, 13(12), 1604–1617, Dec. 2004.
41. M. N. Do and M. Vetterli, Wavelet-based texture retrieval using generalized Gaussian density and kullback-leibler distance, *IEEE Transactions on Image Processing*, 11(2), 146–158, Feb. 2002.
42. J. M. Leski, Robust weighted averaging [of biomedical signals], *IEEE Transactions on Biomedical Engineering*, 49(8), 796–804, Aug. 2002.
43. P. Sun, Q. H. Wu, A. M. Weindling, A. Finkelstein, and K. Ibrahim, An improved morphological approach to background normalization of ECG signals, *IEEE Transactions on Biomedical Engineering*, 50(1), 117–121, Jan. 2003.
44. J. M. Leski and A. Gacek, Computationally effective algorithm for robust weighted averaging, *IEEE Transactions on Biomedical Engineering*, 51(7), 1280–1284, Jul. 2004.
45. P. R. Rider, Generalized cauchy distributions, *Annals of the Institute of Statistical Mathematics*, 9, 215–223, 1957.
46. S. A. Kassam, *Signal Detection in Non-Gaussian Noise*. New York: Springer-Verlag, 1985.
47. J. G. Gonzalez, Robust techniques for wireless communications in non-Gaussian environments, PhD dissertation, University of Delaware, New York, DE, 1997.
48. F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley, 1986.
49. T. C. Aysal and K. E. Barner, Second-order heavy-tailed distributions and tail analysis, *IEEE Transactions on Signal Processing*, 54(7), 2827–2835, Jul. 2006.
50. J. Astola and Y. Neuvo, Matched median fitlering, *IEEE Transactions on Communications*, 40(4), 722–729, Apr. 1992.
51. Y. H. Ma, P. L. So, and E. Gunawan, Performance analysis of OFDM systems for broadband power line communications under impulsive noise and multipath effects, *IEEE Transactions on Power Delivery*, 20(2), 674–682, Apr. 2005.
52. O. G. Hooijen, A channel model for the residential power circuit used as a digital communications medium, *IEEE Transactions on Electromagnetic Compatibility*, 40(4), 331–336, Nov. 1998.
53. O. G. Hooijen, On the channal capacity of the residential power circuit used as a digital communications medium, *IEEE Communcations Letter*, 2(10), 267–268, Oct. 1998.
54. M. Zimmerman and K. Dostert, Analysis and modeling of impulsive noise in broadband power line communications, *IEEE Transactions on Electromagnetic Compatibility*, 44(1), 249–258, Feb. 2002.

27

Nonlinear Filtering for Image Denoising*

27.1 Filtering for Image Denoising **27**-2
Nonlinear Filtering in the Transform Domain • Transform: Separable or Nonseparable? • Transform: Fixed or Adaptive? • Chapter Organization

27.2 Polar Cosine Transform **27**-5
Continuous Transform • Discrete Transform • Radon-Based Digital Implementation • Butterfly-Based Digital Implementation • Nonlinear Approximation • Polar Cosine Packets • Best Basis Selection • Multiscale Polar Cosine Transform • Relationship with Other Transforms • Image Denoising in the PCT Domain • Denoising Results

27.3 Planelets **27**-23
Introduction • Continuous Planelet Transform • Discrete Planelet Transform • Denoising Algorithm • Planar Feature Extraction

27.4 3-D Wavelet Packets **27**-26
Adaptive 3-D WP Transform • Restoration Algorithm

27.5 Discussion and Conclusion **27**-28

References **27**-31

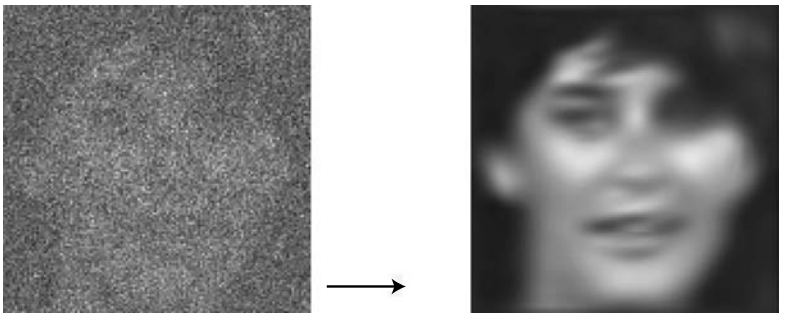
Nasir M. Rajpoot
University of Warwick

Zhen Yao
University of Warwick

Roland G. Wilson
University of Warwick

Denoising is one of the fundamental problems in digital image processing and can be defined as the removal of noise from observed image data, for example in broadcast and surveillance as well as in medical imaging applications. Often performed as a preprocessing step prior to any analysis or modeling of the images, denoising improves the image quality thus reducing the effects of noise on the output of any subsequent operations. Let $f(\mathbf{r})$ denote some original image data corrupted by additive white Gaussian noise $\nu(\mathbf{r})$, resulting in observed image data $g(\mathbf{r})$, where $\mathbf{r}$ denotes the n-dimensional coordinates of the image data, and $n \in \{2, 3, 4\}$. For $n = 2, \mathbf{r}$ usually represents the horizontal and vertical coordinates (often denoted by x and y), whereas for $n = 4$ for instance in case of real-time capture of three-dimensional (3-D) images (such as fMRI image data), $\mathbf{r}$ may be composed of the three spatial coordinates (x, y, and z) and a time coordinate. Corruption of the original image data can, therefore, be expressed in the following form,

* Parts of this chapter have appeared in the Proceedings of Baiona SPC'2003, ICIP'2004, and WIAMIS'2004, and ICIP'2005. Much of the material in Section 27.2 appears in Yao's PhD thesis, submitted to the University of Warwick in October 2007, and is in preparation for submission to a journal.



$$g(\mathbf{r}) = f(\mathbf{r}) + \nu(\mathbf{r}) \tag{27.1}$$

where $\nu = \mathcal{N}(0, \sigma^2)$ denotes uncorrelated, zero-mean, Gaussian noise with a standard deviation of σ. The addition of white Gaussian noise often takes place at the imaging side, for instance due to the finite exposure time of the imaging devices, thermal noise in the CCD arrays, quantization noise, or a combination of some or all of these. The problem of image denoising can be stated as follows.

PROBLEM STATEMENT:

Given the observed image data $g(\mathbf{r})$ and our knowledge about the noise process $\nu(\mathbf{r})$ being the white Gaussian noise with the level of noise being unknown, find an approximation $\tilde{f}(\mathbf{r})$ of the original image data such that the following criterion of total distortion

$$\sum_{\mathbf{r}\in\Omega} \|f(\mathbf{r}) - \tilde{f}(\mathbf{r})\|^2 \tag{27.2}$$

is minimized, where $\Omega \subset \mathcal{R}^n$ is the set of all coordinates for the image data.

It is evident from the above problem statement that denoising is an *inverse problem* as well as an ill-posed one, since the objective here is to find a mapping $\Phi: g(\mathbf{r}) \mapsto \tilde{f}(\mathbf{r})$ which minimizes the sum of squared errors between $f(\mathbf{r})$ and $\tilde{f}(\mathbf{r})$, without having any prior knowledge of $f(\mathbf{r})$.

27.1 Filtering for Image Denoising

Denoising methods proposed in the literature can be categorized into two main classes: spatial (or spatiotemporal) and transform domain. Spatial or spatiotemporal denoising methods modify the intensity of the observed image data elements by applying a *linear* filter (e.g., averaging or Gaussian smoothing) or *nonlinear* (median filtering, edge-preserving filter such as Kuwahara filter or anisotropic diffusion) filtering operation on intensity values of the observed image data. Denoising methods that operate in the transform domain aim to amend the transform coefficients in such a way that prominent features of the original image are retained or highlighted and noise-related coefficients or noisy components of the transform coefficients are suppressed. The modification of transform coefficients can also be done in a linear, for example, multiplication or convolution, or nonlinear, thresholding or manipulation of coefficients using a nonlinear function such as $\tanh(\cdot)$, fashion.

Linear filters in both spatial and transform domains usually face the dilemma that although they can be computationally inexpensive, they often result in smoothing (also known as *blurring*) of important features, such as edges in images.

Nonlinear filters, on the other hand, aim at preserving image features that are crucial to the fidelity of reconstructed image data. Furthermore, nonlinear filtering in the transform domain offers the added advantage that it can faithfully reconstruct as well as highlight prominent image features, provided an appropriate transform for representing the image data is employed. Assuming the representation is sparse, filtering can be performed by simply thresholding the transform coefficients. The overall mapping function $\Phi(g)$ can then be defined as follows:

$$\Phi(g) = \tilde{f} = \Gamma^{-1}(\mathcal{T}\{\Gamma(g)\}), \tag{27.3}$$

where

Γ and Γ^{-1}, respectively, denote the forward and inverse transforms
$\mathcal{T}\{\cdot\}$ denotes the thresholding operation

The denoising function Φ is a nonlinear filter, even when a unitary transform Γ is employed, due to the presence of the thresholding operator $\mathcal{T}$, which is a highly nonlinear operator.

27.1.1 Nonlinear Filtering in the Transform Domain

The use of multiresolution image representations in the restoration of noisy images dates back to the late 1980s [1], in which a statistical estimator based on a simple quadtree image model was used in smoothing images corrupted by additive Gaussian noise. For image denoising, the transform Γ in Equation 27.3 ought to be chosen such that it is an appropriate representation of the most prominent spatial or spatiotemporal features present in $g(\mathbf{r})$ with the help of basis functions with suitable shape and localization. An additional advantage of the transform-based approach to filtering is that it allows one to extract important features for analysis purposes; for example, curvilinear features in still images, moving edges in video sequences, tissue interfaces in medical images, or fault surfaces in seismic images.

But what criteria should one use to choose an appropriate transform? Sparsity of a representation is widely believed to be one of the criteria used to judge its effectiveness for a particular type of feature. A representation can be regarded as being *efficient* if its basis functions resemble closely the type of feature being sought in the image data, thus resulting in its sparse representation. It is believed that during the millenia of evolution, the external visual stimuli of complex natural scenarios have influenced the human visual system (HVS), so that cells in the primary visual cortex can respond to certain important spatiotemporal features. Hubel and Wiesel [2] first showed that biological visual systems in mammals actually analyze images along dimensions such as orientation, scale, and frequency. More recently, it was shown by Olshausen and Field [3,4] that the sparse components generated by image-dependent linear transformation known as independent component analysis (ICA) resemble simple-cell receptive fields. This matches the hypothesis that the HVS captures essential information with a minimum number of excited neurons, which can be understood as a biological form of sparse representation.

27.1.2 Transform: Separable or Nonseparable?

The ordinary separable wavelet transform offers most of the desiderata that a suitable representation mimicking the HVS is required to satisfy multiresolution, sparse, localized. However, the fact that it is computed separably, i.e., on rows and columns in 2-D, for instance, implies that its basis functions are simply tensor products of 1-D wavelet basis functions along the rows and columns, resulting in capturing edges along only three orientations in 2-D: horizontal, vertical, and diagonal. Since the separable 2-D wavelet basis functions can only capture features of three possible orientations, the ordinary

wavelet transform results in a nonsparse representation of edges in other directions. This limited *orientation selectivity* can be a severe limitation in applications where detection, coding, or denoising where reconstruction of strong directional features is important. In higher dimensions also, separability seriously limits the ability of wavelets to efficiently represent higher dimensional features (such as lines in images or planes in 3-D image volumes). To overcome this limitation, several nonseparable geometric wavelets (mostly in 2-D) have been proposed in the last decade or so. These are commonly referred to as the *X-lets* in the literature, such as ridgelets [5], curvelets [6,7], wedgelets [8], beamlets [9], contourlets [10,11], brushlets [12], and arclets [13].

27.1.3 Transform: Fixed or Adaptive?

Most of the nonseparable *X-lets* mentioned above employ fixed basis functions. Some researchers believe that since the type of features that would excite the retinal neurons is fixed, it also suggests that the HVS is more likely a representation with a fixed basis rather than an adaptive one like the ICA. However, in some cases, image-dependent adaptive bases have been shown to be superior in terms of their ability to represent prominent image features, particularly when the representation is not *overcomplete*.* They offer the advantage of compactly representing the image features using basis functions that are adapted to the contents of the image data. In summary, while the fixed transform idea is appealing in terms of its computational complexity and in its comparison to the HVS, the adaptive transform may be better suited in some cases due to its adaptability but may be prohibitively expensive computation-wise.

27.1.4 Chapter Organization

In this chapter, we describe three wavelet-like representations employed to filter noisy 2-D and 3-D image data:

- Polar cosine transform (PCT) is a nonseparable 2-D transform designed to capture two of the most important features in 2-D images: curvilinear edges and texture. The development of PCT is motivated by the fact that natural images contain not only piecewise smooth regions separated by edges, but also regions with strong textural characteristics. Two constructions of PCT, one using Radon transform and another using DCT and DST in a butterfly fashion, are described in Section 27.2. The former PCT construction can also be tailored to be image-dependent such that it can be adapted to both directional linear features as well as directional textures in a unified manner according to the "*image* = *edge* + *texture*" model.
- Planelets are nonseparable 3-D basis functions, resembling planar wave functions and having compact support in space–time as well as spatiotemporal frequency. Locally planar structures can be found in many 3-D applications, including video sequences, where they represent sweeping luminance edges, and medical volume data, where they represent the interfaces between different tissue types, for example. Planelets can be regarded as an extension of the complex wavelet bases [14] which have been increasing in popularity in recent years. Because transformation to the new basis is efficient computationally, the overall denoising algorithm is highly efficient. The representation is translation invariant, offers good directional selectivity, and can be computed efficiently. The success of the approach is demonstrated using noisy video sequence and a noisy human knee MR image volume.

* An overcomplete representation is one that employs more basis functions than are required for an exact reconstruction of the original image data.

- Adaptive 3-D wavelet packet (A3-DWP) representation, which is separable but uses basis functions that are adapted to the contents of a given 3-D image data, a video sequence, or an image volume. The A3-DWP representation is optimal in terms of compactly representing local, in both space and time, spatiotemporal frequencies in the given image data. In order to reduce the effects of Gibbs phenomenon in the restored image data, translation dependence is removed by averaging the restored instances of the shifted data in all three directions.

27.2 Polar Cosine Transform

It is widely recognized that natural images consist of piecewise smooth regions separated by edges as well as textured regions. While various different models for curved edges exist, including some of the *X-lets* mentioned in Section 27.1, texture representation has faced difficulties due to lack of a single widely accepted mathematical definition of texture patterns that can be found in images. There are, however, certain assumptions that can be made for most natural textures:

- Oriented. Many spatial textures have a prominent direction, for example wood patterns and seismograms, or a combination of several such components.
- Periodic. The texture tends to repeat certain basic elements at a given frequency and tends to repeat itself with a degree of affine invariance.
- Localized. In natural images, the textures are usually confined within a particular region. In other words, they are spatially localized.

Over the years, several wave packet bases such as local cosines [15,16], wavelet packets [17,18], and brushlets [19] have been proposed as suitable representations for oscillatory textures. However, they either lack orientation selectivity, or their spatial-frequency localization is not satisfactory. Moreover, due to the Uncertainty Principle, a sparse representation for both singularities and periodic oscillations in a fixed basis is a fundamental dilemma [20]. The resort is to find an image-dependent *adaptive* basis which can accommodate both directional linear features as well as directional textures in a unified manner according to the "*image* = *edge* + *texture*" model.

27.2.1 Continuous Transform

A prototypical oriented texture pattern can be seen as a higher dimensional function which has a waveform constant along a fixed direction. Such functions are usually referred to as "planar waves" or "ridge functions" in approximation related literature [21]. Formally, a ridge function is a multivariate function of the form $\psi(\vec{\xi} \cdot \vec{\theta})$ where $\psi(\cdot)$ is a univariate function usually referred to as the *ridge profile*, with spatial coordinate vector $\vec{\xi} \in \mathbb{R}^d$ and $\vec{\theta} \in \mathbf{S}^{d-1}$. This means $\vec{\theta}$ is on the unit sphere $\mathbf{S}^{d-1}$ in dimension $d > 1$ and indicates the orientation.

In order to model oriented texture, a real-to-complex ridge profile $\psi: \mathbb{R} \to \mathbb{C}$ can be defined as

$$\psi_\omega(\xi) = e^{i\omega\xi}. \tag{27.4}$$

which is basically the 1-D Fourier basis function, and the corresponding multivariate ridge function by definition is

$$\psi_\omega(\vec{\xi} \cdot \vec{\theta}) = e^{i\omega(\vec{\xi}\cdot\vec{\theta})}. \tag{27.5}$$

The ridge function gives a complex trigonometric oscillatory constant along the direction $\vec{\theta}$. Taking the inner product of it with a function $f(\vec{\xi})$ gives

$$\hat{f}(\omega, \vec{\theta}) = \langle f(\vec{\xi}),\ \psi_\omega(\vec{\xi} \cdot \vec{\theta}) \rangle = \int f(\vec{\xi}) e^{-i\omega(\vec{\xi}\cdot\vec{\theta})} d\vec{\xi}. \tag{27.6}$$

This is exactly the *d*-dimensional Fourier transform expressed in polar coordinate form, or just simply *polar Fourier transform*. The conversion to Cartesian coordinates can be done by separating the radial frequency into a frequency vector:

$$\vec{\omega} = \omega\vec{\theta}. \tag{27.7}$$

With a slight rearrangement of Equation 27.6, the equivalence becomes obvious:

$$\hat{f}(\omega, \vec{\theta}) = \int f(\vec{\xi}) e^{-i\vec{\xi}\cdot(\omega\vec{\theta})}\ d\vec{\xi} = \int f(\vec{\xi}) e^{-i\vec{\xi}\cdot\vec{\omega}}\ d\vec{\xi} = \hat{f}(\vec{\omega}). \tag{27.8}$$

Meanwhile, the definition for the continuous Radon transform in *d*-dimensions is

$$\mathcal{R}f(t, \vec{\theta}) = \int f(\vec{\xi}) \delta(\vec{\xi} \cdot \vec{\theta} - t) d\vec{\xi}. \tag{27.9}$$

The relationship between the Radon transform and the polar Fourier transform is stated as the Fourier Slice Theorem [22]:

THEOREM 27.1 (Fourier Slice Theorem)

The 1-D Fourier transform with respect to t of the projection $\mathcal{R}f(t, \vec{\theta})$ *is equal to a central slice, at a given orientation* $\vec{\theta}$*, of the higher dimensional Fourier transform of the function* $f(\vec{\xi})$*, that is,*

$$\widehat{\mathcal{R}f(t,\ \vec{\theta})} = \hat{f}(\omega,\ \vec{\theta}). \tag{27.10}$$

If some boundary condition is to be posed at each polar orientation, instead of being defined on $\mathbb{R}$, the Radon slice $\mathcal{R}_{\vec{\theta}} f(t)$ is assumed to be an even function $\mathcal{R}_{\vec{\theta}} f(t) = \mathcal{R}_{\vec{\theta}} f(-t)$, then its Fourier transform can be written as follows:

$$\widehat{\mathcal{R}f(t, \vec{\theta})} = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} \mathcal{R}f(t, \vec{\theta}) \cos(\omega t) dt. \tag{27.11}$$

It is noted that only the real cosine part remains in the above equation due to the even boundary assumption on the slices. This is equivalent to a cosine tranform with ridge-type basis functions defined as

$$\psi_\omega^{\mathcal{C}}(\vec{\xi} \cdot \vec{\theta}) = \sqrt{\frac{2}{\pi}} \cos(\omega\vec{\xi} \cdot \vec{\theta}). \tag{27.12}$$

In the same way by assuming $\mathcal{R}_\theta f(t) = -\mathcal{R}_\theta f(-t)$, the sine counterpart is

$$\psi_\omega^{\mathcal{S}}(\vec{\xi}\cdot\vec{\theta}) = \sqrt{\frac{2}{\pi}}\sin(\omega\vec{\xi}\cdot\vec{\theta}). \tag{27.13}$$

Correspondingly, the continuous polar cosine and sine transform are denoted, respectively, as

$$\mathcal{C}f(\omega,\vec{\theta}) = \langle f(\vec{\xi}),\ \psi_\omega^{\mathcal{C}}(\vec{\xi}\cdot\vec{\theta})\rangle = \sqrt{\frac{2}{\pi}}\int_0^{+\infty} f(\vec{\xi})\cos(\omega\vec{\xi}\cdot\vec{\theta})\mathrm{d}\vec{\xi} \tag{27.14}$$

and

$$\mathcal{S}f(\omega,\vec{\theta}) = \langle f(\vec{\xi}),\ \psi_\omega^{\mathcal{S}}(\vec{\xi}\cdot\vec{\theta})\rangle = \sqrt{\frac{2}{\pi}}\int_0^{+\infty} f(\vec{\xi})\sin(\omega\vec{\xi}\cdot\vec{\theta})\mathrm{d}\vec{\xi}. \tag{27.15}$$

These two transforms, along with the polar Fourier transform, can be regarded as a family of *Polar Trigonometric Transforms*. In the context of image processing, the PCT is the transform of interest in this work, due to its better approximation convergence properties attributed to the even boundary extension [23]. Since it is essentially the Fourier transform, the completeness of the transform makes the transform operator $\mathcal{C}$ unitary which means $\mathcal{C}^{-1} = \mathcal{C}^\star$. Since the transform is real, the operator is self-conjugate, making the inverse transform exactly the same as the forward transform:

$$\mathcal{C}^{-1} = \mathcal{C}. \tag{27.16}$$

The same also holds for the continuous polar sine transform and its inverse.

27.2.2 Discrete Transform

The continuous PCT essentially replaces the polar Fourier transform's complex basis with a real cosine sinusoid as the ridge function, by assuming the projected Radon slice is even. In a discrete case, the Fourier-related transforms that operate on a function over a finite domain can be thought of as implicitly defining an extension of that function outside the domain. The discrete Fourier transform (DFT) implies a periodic extension of the original function. A discrete cosine transform (DCT), like the continuous cosine transform, implies an even extension of the original function.

However, when the transform is to operate on finite, discrete sequences, two issues arise that are not relevant in case of the continuous cosine transform. First, one has to specify whether the function is even or odd at both the left and right boundaries of the domain. Second, one has to specify around what point the function is even or odd. These issues result in several different versions of DCTs, a full list of which along with corresponding DSTs is given in Table 27.1. For a discrete polar cosine transform, any of the DCTs can be adopted to form the basis ridge function. In particular, the so-called DCT-II is widely used in many applications such as the famed JPEG compression [24] due to its even boundary extensions on both ends, which gives better approximation convergence. The 1-D DCT-II function is defined by the ridge profile:

$$\psi_k[n] = \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]. \tag{27.17}$$

where

N is the possible number of frequencies as well as the length of the Radon projection slice
$k = 0, 1, \ldots, N-1$ is the frequency index

TABLE 27.1 List of DCTs and DSTs

DCT-I	$\frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right]$
DCT-II	$\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}(n+\frac{1}{2})k\right]$
DCT-III	$\frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n\left(k+\frac{1}{2}\right)\right]$
DCT-IV	$\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]$
DST-I	$\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N+1}(n+1)(k+1)\right]$
DST-II	$\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)(k+1)\right]$
DST-III	$\sum_{n=0}^{N-2} x_n \sin\left[\frac{\pi}{N}(n+1)\left(k+\frac{1}{2}\right)\right]$
DST-IV	$\sum_{n=0}^{N-1} x_n \sin\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]$

In the context of image processing, the discussion of discretization of the PCT will be further restricted to the 2-D case, which means the unit orientation vector becomes

$$\vec{\theta} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}. \tag{27.18}$$

With $\vec{\xi} = [\xi_1, \xi_2]$, the discrete cosine ridge function is

$$\psi_k[\xi_1 \cos\theta + \xi_2 \sin\theta] = \cos\left[\frac{\pi}{N}\left(\xi_1 \cos\theta + \xi_2 \sin\theta + \frac{1}{2}\right)k\right]. \tag{27.19}$$

Then the discrete polar cosine transform on an $M \times M$ 2-D image $f[\vec{\xi}]$ can be defined as

$$\begin{aligned} \mathcal{C}f[k,\theta] &= \left\langle f[\vec{\xi}],\ \psi_k[\xi_1 \cos\theta + \xi_2 \sin\theta] \right\rangle \\ &= \sum_{\xi_1=0}^{M-1} \sum_{\xi_2=0}^{M-1} f\left[\xi_1, \xi_2\right] \cos\left[\frac{\pi}{N}\left(\xi_1 \cos\theta + \xi_2 \sin\theta + \frac{1}{2}\right)k\right]. \end{aligned} \tag{27.20}$$

The inverse transform for DCT-II is the DCT-III transform. The corresponding cosine ridge function for the inverse transform is

$$\psi_k^{-1}[\xi_1 \cos\theta + \xi_2 \sin\theta] = \lambda_k \cos\left[\frac{\pi}{N}\left(\xi_1 \cos\theta + \xi_2 \sin\theta + \frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right], \tag{27.21}$$

where

$$\lambda_k = \begin{cases} 1/2 & \text{if } k = 0, \\ 1 & \text{if } k \neq 0, \end{cases}$$

and the discrete inverse polar cosine transform operator $\mathcal{C}^{-1}$ is given by

$$\begin{aligned} \mathcal{C}^{-1}f[k,\theta] &= \langle f[\vec{\xi}],\ \psi_k[\xi_1 \cos\theta + \xi_2 \sin\theta] \rangle \\ &= \sum_{\xi_1=0}^{M-1} \sum_{\xi_2=0}^{M-1} f[\xi_1,\ \xi_2] \lambda_k \cos\left[\frac{\pi}{N}\left(\xi_1 \cos\theta + \xi_2 \sin\theta + \frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]. \end{aligned} \tag{27.22}$$

27.2.3 Radon-Based Digital Implementation

Equations 27.10 and 27.11 suggest that the discrete PCT can be implemented by a Radon transform, that is, taking the 1-D DCT on Radon slices:

$$\mathcal{C}f[k,\ \theta] = \sum_{t=0}^{N-1} \mathcal{R}f[t, \theta] \cos\left[\frac{\pi}{N}\left(t + \frac{1}{2}\right)k\right]. \tag{27.23}$$

As the Fourier Slice Theorem shows, the Radon transform can be implemented by taking the central slice in the Fourier spectrum and then performing a 1-D inverse Fourier transform on it. Applying DCT on the Radon slices, we can obtain the discrete PCT.

This gives a means for a digital implementation of the discrete polar cosine transform. For the Radon transform, various ways to implement it in a discrete fashion have been attempted, among which the Fast Slant Stack [25] was chosen, which is based on a pseudo-polar Fast Fourier Transform (FFT) [26]. According to Ref. [25], the transform is computationally efficient, algebraically exact, geometrically faithful, and its inversion is numerically stable.

With the implementation described above, the PCT basis vectors for an 8×8 image block can be seen in Figure 27.1. The basis vectors are indexed by frequency k and orientation θ. As a result of the Cartesian-to-polar conversion in the frequency domain, the low-frequency basis vectors are clearly being over-sampled, resulting in a redundant frame where the number of coefficients is four times the original data size.

27.2.4 Butterfly-Based Digital Implementation

The above digital implementation of the PCT based on the Radon transform is akin to the digital ridgelet transform [27]. Such a Radon-based approach is flexible in constructing various directional ridge-type transforms, but it suffers from several drawbacks of the underlying Radon transform. First is the high computational requirement both for the forward and inverse transforms. The forward transform includes a pseudo-polar Fourier transform, 1-D inverse Fourier transforms on polar slices, and forward cosine

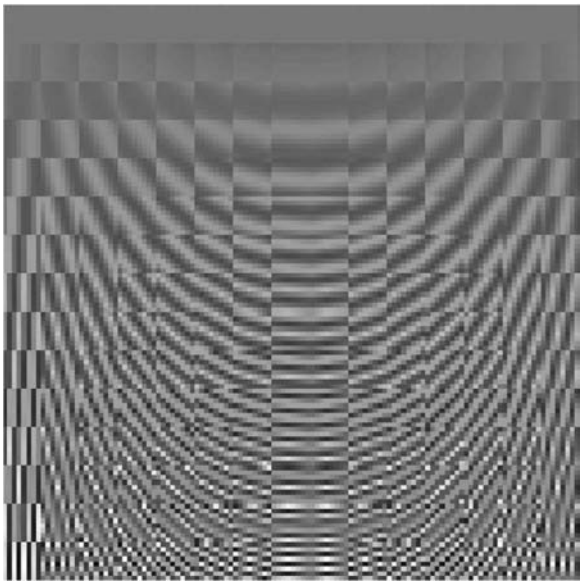


FIGURE 27.1 8×8 discrete polar cosine basis vectors.

transforms. The inverse Fast Slant Stack uses an iterative algorithm in the reconstruction to minimize numerical instability. Secondly, the redundancy factor could be an issue in some applications. While it seems implausible to construct a digital ridgelet transform without invoking the Radon transform, it does not necessarily hold true for the PCT. A possible construction of such directional real trigonometric transform in the context of modulated lapped transforms was discussed in Ref. [28]. The same construction is adopted here in implementing a digital PCT.

It can be observed from Table 27.1 that the DCT-IV and DST-IV share the same function parameters. In 2-D, Cartesian separable basis functions can be formed by the tensor product of 1-D bases:

$$\psi^{\mathcal{C}}_{k_1,k_2}[\xi_1,\,\xi_2] = \cos\left[\frac{\pi}{N}\left(\xi_1+\frac{1}{2}\right)\left(k_1+\frac{1}{2}\right)\right]\cos\left[\frac{\pi}{N}\left(\xi_2+\frac{1}{2}\right)\left(k_2+\frac{1}{2}\right)\right]. \tag{27.24}$$

$$\psi^{\mathcal{S}}_{k_1,k_2}[\xi_1,\,\xi_2] = \sin\left[\frac{\pi}{N}\left(\xi_1+\frac{1}{2}\right)\left(k_1+\frac{1}{2}\right)\right]\sin\left[\frac{\pi}{N}\left(\xi_2+\frac{1}{2}\right)\left(k_2+\frac{1}{2}\right)\right]. \tag{27.25}$$

By setting $A = \frac{\pi}{N}(\xi_1+\frac{1}{2})(k_1+\frac{1}{2})$ and $B = \frac{\pi}{N}(\xi_2+\frac{1}{2})(k_2+\frac{1}{2})$, the expression is simplified as follows:

$$\begin{aligned}\psi^{\mathcal{C}}_{k_1,k_2}[\xi_1,\,\xi_2] - \psi^{\mathcal{S}}_{k_1,k_2}[\xi_1,\,\xi_2] &= \cos A\cos B - \sin A\sin B\\ &= \cos\left[A+B\right]\\ &= \cos\left[\frac{\pi}{N}\left(\left(\xi_1+\frac{1}{2}\right)\left(k_1+\frac{1}{2}\right)+\left(\xi_2+\frac{1}{2}\right)\left(k_2+\frac{1}{2}\right)\right)\right]\\ &= \cos\left[\frac{\pi}{N}\left(\xi_1k_1+\xi_2k_2+\frac{1}{2}(\xi_1+\xi_2+k_1+k_2+1)\right)\right].\end{aligned} \tag{27.26}$$

Therefore the difference between the 2-D basis functions of DCT-IV and DST-IV at given frequencies can be reduced into a cosine component. Furthermore, since $k_1 = k\cos\theta$, $k_2 = k\sin\theta$, where $k = \sqrt{k_1^2+k_2^2}$ is the polar frequency index and $\theta = \arctan(k_2/k_1)$, Equation 27.26 can be rewritten in the polar cosine form:

$$\begin{aligned}\psi^{\chi}_{k,\theta}[\xi_1,\,\xi_2] &= \psi^{\mathcal{C}}_{k\cos\theta,k\sin\theta}[\xi_1,\,\xi_2] - \psi^{\mathcal{S}}_{k\cos\theta,k\sin\theta}[\xi_1,\,\xi_2]\\ &= \cos\left[\frac{\pi}{N}\left((\xi_1\cos\theta+\xi_2\sin\theta)k+\frac{1}{2}(\xi_1+\xi_2+k\cos\theta+k\sin\theta+1)\right)\right].\end{aligned} \tag{27.27}$$

However, it should be noted that compared with the PCT-II defined in Equation 27.20, although the basis vectors do not represent a strict form of ridge functions, they are directional and can be arranged in polar form. The above is set for the case when $k_1 = 0, \ldots, N-1$ and $k_2 = 0, \ldots, N-1$. For negative frequencies, it is clear that

$$\begin{aligned}\psi^{\chi}_{k_1,-k_2-1}[\xi_1,\xi_2] &= \psi^{\mathcal{C}}_{k_1-k_2-1}[\xi_1,\,\xi_2] - \psi^{\mathcal{S}}_{k_1,-k_2-1}[\xi_1,\,\xi_2]\\ &= \cos A\cos\left[-B\right] - \sin A\sin\left[-B\right]\\ &= \cos A\cos B + \sin A\sin B\\ &= \psi^{\mathcal{C}}_{k_1,k_2}[\xi_1,\,\xi_2] + \psi^{\mathcal{S}}_{k_1,k_2}[\xi_1,\xi_2].\end{aligned} \tag{27.28}$$

This suggests that for $-k_2-1 = -1, \ldots, -N$, the basis function is just the sum of DCT-IV and DST-IV. In the same way, it is not difficult to see that when $k_1 = -1, \ldots, -N$ and $k_2 = -N, \ldots, N-1$, which is the basis functions for the lower half-plane of the frequency spectrum:

$$\psi^{\chi}_{k_1,k_2}[\xi_1,\xi_2] = \psi^{\chi}_{-k_1-1,-k_2-1}[\xi_1,\,\xi_2]. \tag{27.29}$$

Thus the corresponding transform domain exhibits the same Hermite symmetry as the Fourier transform, with basis functions very close to the FFT's basis functions. As a result, the transform

$$\chi f[k_1, k_2] = \langle f[\xi_1, \xi_2],\ \psi^{\chi}_{k_1,k_2}[\xi_1, \xi_2] \rangle \tag{27.30}$$

is only twice redundant, by removing a half-plane of the spectrum and can be efficiently implemented by a butterfly computation. A schematic illustration of the transform can be seen in Figure 27.2. For the sake of convenience, this particular digital implementation will be referred to as PCT-X. A complete basis for size 8×8 is shown in Figure 27.3.

It is observed that the PCT-X is closely related to the Fourier transform itself, despite the new transform being real. The expansion of PCT-X, therefore, can be expected to have similarities with the Fourier magnitude spectrum.

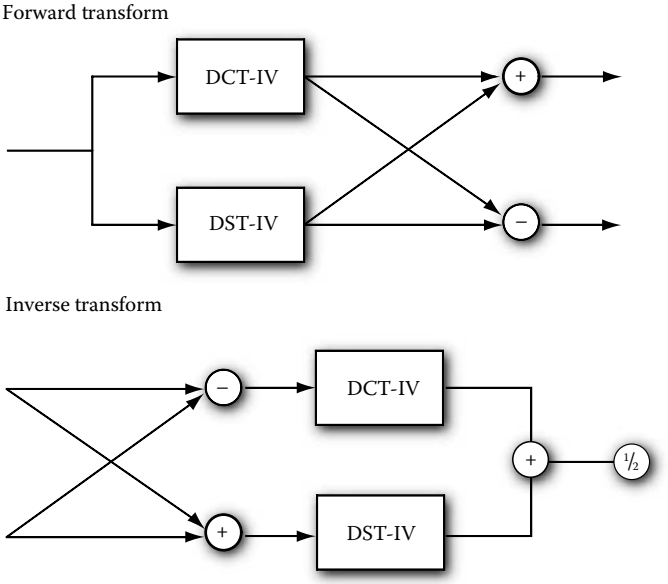


FIGURE 27.2 Forward PCT-X transform and the inverse transform.

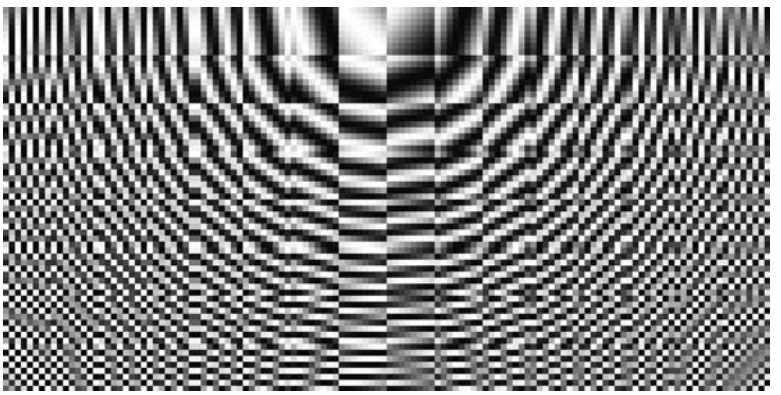


FIGURE 27.3 8×8 discrete polar cosine basis vectors.

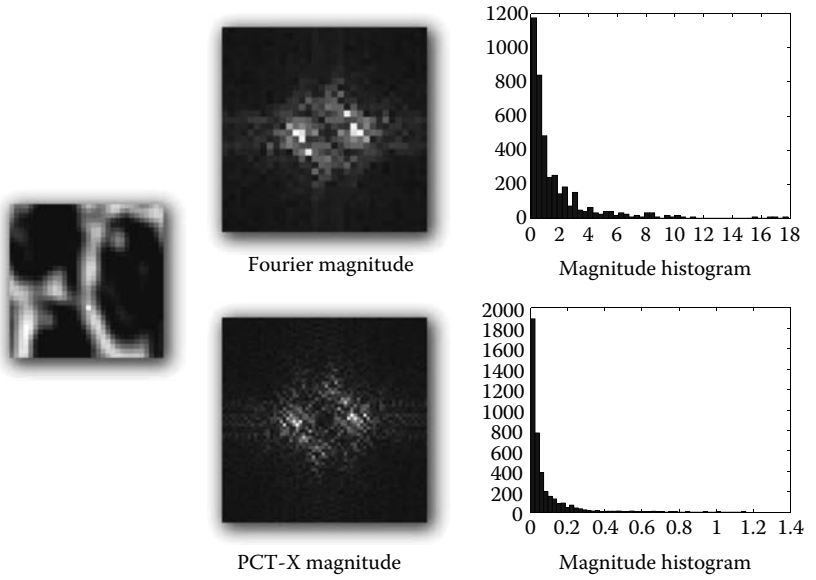


FIGURE 27.4 Spectrum comparison between the PCT-X and the Fourier transform.

In Figure 27.4, a comparison between the PCT-X spectrum and the Fourier spectrum is made. A reptile texutre patch is chosen as the test data and the two transforms yield very similar magnitude responses. However, the histograms suggest that the coefficients of the PCT-X expansion are sparser than those of the Fourier transform, attributed to the symmetric boundary extension of the underlying DCT-IV and DST-IV.

27.2.5 Nonlinear Approximation

As previously mentioned in Section 27.2, with the difficulty in mathematically formulating texture, it is hard to approach the nonlinear approximation problem quantitatively. The theoretical study of the transform in the context of computational harmonic analysis is beyond the scope of this thesis, where the main concern is on examining its effectiveness in providing sparsity in the transformed expansion. The approach taken here is more empirical, where the nonlinear approximation results are presented in the form of output from numerical experiments.

Two digital PCT implementations were selected for nonlinear approximation experiments: the Radon-based PCT with DCT-II as ridge profile (PCT-II) and the butterfly PCT-X implementation as shown in Figure 27.2. They are compared with three other transforms: the FFT; the DCT-II, which is a Cartesian separable real-to-real implementation of FFT; and the *fast curvelet transform* (FCT) which is implemented by wrapping of specially selected subbands of Fourier samples [29].

Two different image patches are tested for nonlinear approximation. The first one contains a portion of a fingerprint image, which consists of repeated curved ridges. The second one is a simple straight line. The purpose of including such a linear singularity is that although it is not ideal to be represented by an oscillatory basis, it would be still worth comparing with other trigonometric transforms. All the patches are sized 256×256, cropped from the original 512×512 images. They are prewhitened by taking the highpass subband Laplacian pyramid.

The nonlinear approximation results are represented in Figures 27.5 and 27.6 as PSNR curves. The PSNR values are plotted against the top percentage of coefficients, from 2% to 50%. The percentage

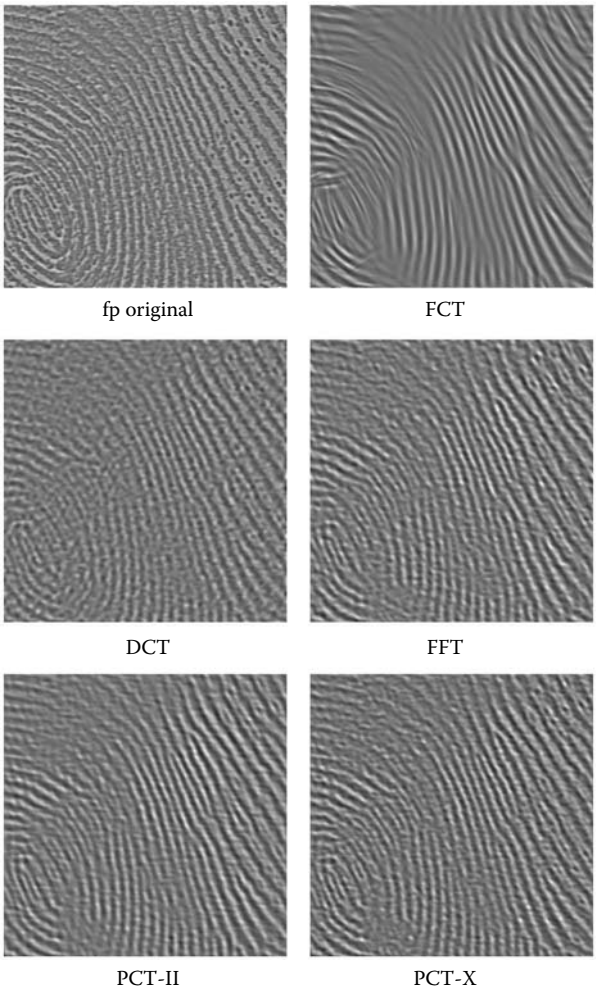


FIGURE 27.5 Nonlinear approximation PSNR curves for fp.

instead of the number of coefficients is used due to the fact that different transforms yield different number of coefficients. For the sake of convenience, a table of percentages and the corresponding retained numbers of coefficients for different transforms are listed in Table 27.2. It is observed that for the textured image **fp**, the PCTs outperform the other candidates consistently. With fewer coefficients, the PCT-X's PSNR is usually close to the PCT-II, occasionally outperforms it (on **fp**). This is due to the fact that PCT-II's reconstruction is less stable with fewer coefficients while PCT-X's inverse transform is exact. However, the goodness of DCT-II's even symmetric boundaries eventually yields better PSNR with relatively more coefficients involved. While low PSNR performances are to be expected from the trigonometric transforms, it is also reassuring that the two PCTs give better PSNR than the DCT and FFT.

Figure 27.7 shows the reconstructions from the top 2% coefficients of these four transforms on **fp**. The reconstruction from FCT looks artificial, due to its inability to capture directional harmonics of the image, an oscillatory pattern has to be described as several directional singularities. The DCT reconstruction, while being closer to the original, does not highlight the oriented patterns as well as

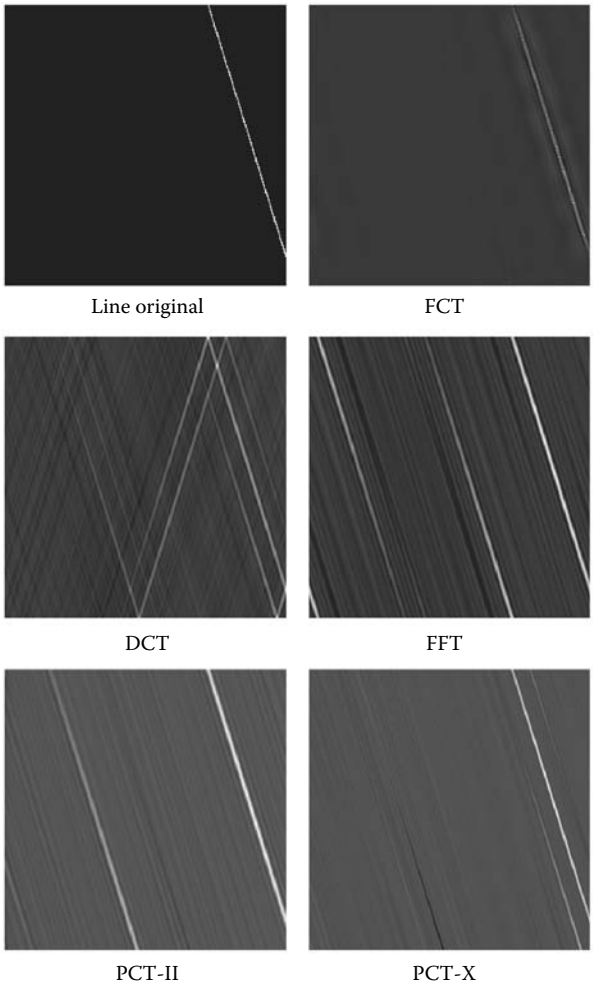


FIGURE 27.6 Nonlinear approximation PSNR curves for line.

TABLE 27.2 Percentage versus Number of Retained Coefficients for Different Transforms

Percentage (%)	FCT	DCT	FFT	PCT-II	PCT-X
0.1	185	66	66	262	132
2	3700	1311	1311	5243	2622
15	27748	9830	9830	39322	19660
100	184985	65536	65536	262144	131072

the PCTs. The results from FFT and PCTs are very similar, while the PCTs give slightly better reconstructions with the directional regularities on the fingerprint ridges, particularly by PCT-X.

Another example is the line in Figure 27.8. The percentage of coefficients is kept as little as 0.1%. It is not surprising that the FCT restores the line faithfully even at such a low ratio. The reconstructions from the four Fourier-type bases (DCT, FFT, and the two PCTs) all exhibit the ''ghosting'' artifacts sometimes referred to as the Gibbs phenomenon, due to suppressing too many high frequency

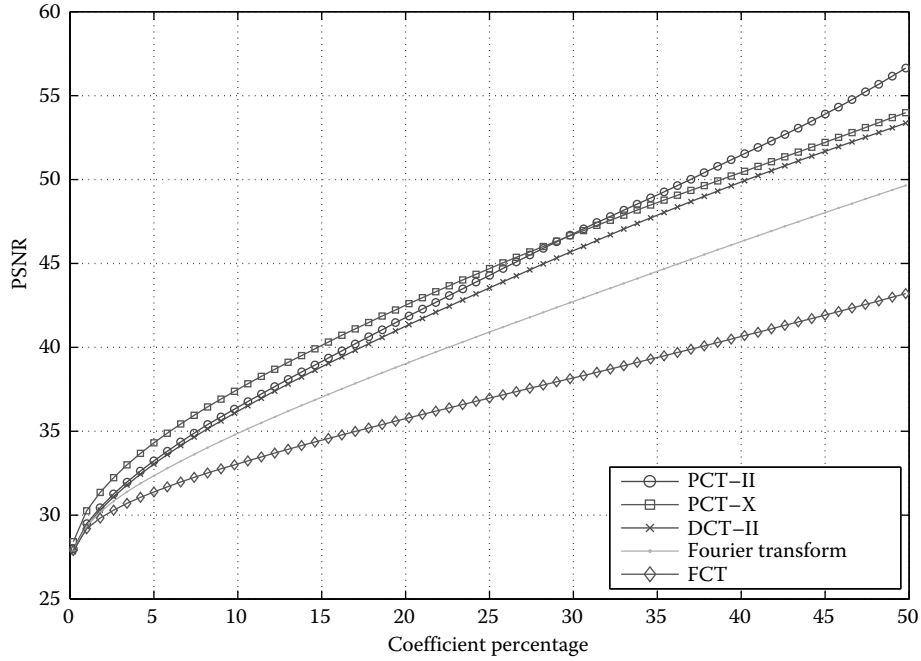


FIGURE 27.7 Illustrative results for nonlinear approximation on fp.

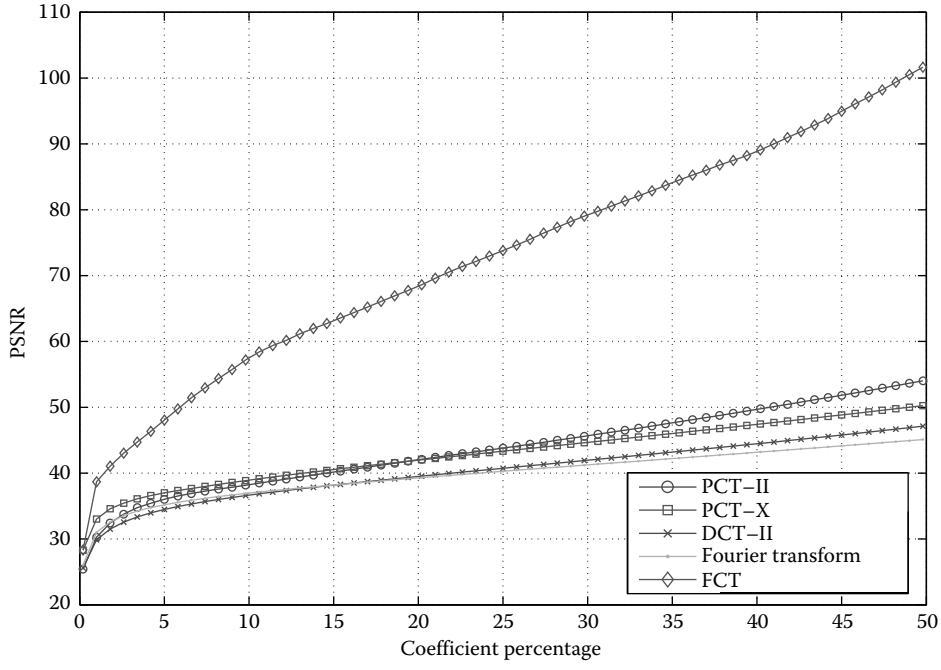


FIGURE 27.8 Illustrative results for nonlinear approximation on line.

coefficients. The DCT shows the ghosting artifact on two different orientations, the FFT has two ghost lines appearing on the reconstruction. The PCT-II, on the other hand, due to the boundary extension being assumed on the Radon slices, shows only one significant ghost line. The ghosting artifact is further reduced in PCT-X's reconstruction.

27.2.6 Polar Cosine Packets

As previously discussed, the Radon transform can reduce the PCT into a matter of performing the cosine transform on Radon slices. This also makes it possible to analyze directional features other than sinusoidal ridges. This is achieved by taking the transform in separate spatial windows. Local trigonometric bases proposed by Coifman and Meyer [30] and by Malvar [31] use smooth window functions to split the signal and to fold overlapping parts back into the original pieces so that the orthogonality is preserved. This treatment can avoid the discontinuity artifacts caused by a rectangular window, while introducing no redundancy.

The local cosine basis is composed of basis functions of the form

$$\varphi_{p,k}(t) = w_p(t)\sqrt{\frac{2}{|I_p|}}\cos\left[\pi\left(k+\frac{1}{2}\right)\frac{t-c_p}{|I_p|}\right]. \tag{27.31}$$

The Cosine-VI basis is used and modulated by a window function $w_p(t)$ which lies on an interval $[a_{p-1}, a_p]$ with $c_p = (a_p + a_{p-1})/2$, and $|I_p| = c_{p+1} - c_p$ being the length of the window, with the overlapping part included. With a careful choice of the window (see Ref. [15]), the set $\{a_p\} \subset \mathbb{R}$ forms a partition of unity and the local cosine basis associated to such partition is said to form a library of orthonormal bases usually referred to as the cosine packets.

The polar cosine packets can be implemented by placing the 1-D cosine packets on the Radon slices; the operator is denoted as

$$\mathcal{P}f[p,k,\theta] = \langle \mathcal{R}f[t,\theta], \varphi_{p,k}[t]\rangle. \tag{27.32}$$

Unfortunately, due to its implementation, it is not possible to construct similar polar cosine packets with the PCT-X.

27.2.7 Best Basis Selection

The arbitrary choice over the library of local trigonometric bases over a compact interval U is of an extremely large range. To seek a feasible "best basis," the library of cosine packets is reduced to only taking dyadically partitioned decomposition of U only. This organization is depicted schematically in Figure 27.9. Then I_{00} is a cosine basis on the entire U and $I_{p,s}$ will correspond to the local cosine basis over interval p of the 2^s intervals at level s of the tree. The best basis can be found by induction on s.

Let $B_{p,s}$ denote the cosine basis vectors corresponding to interval $I_{p,s}$, and $A_{p,s}$ be the best basis. For $s = 0$, the best basis $A_{p,0} = B_{p,0}$, otherwise

$$A_{p,s+1} = \begin{cases} B_{p,s+1} & \text{if } \mathcal{M}(B_{p,s+1}x) < \mathcal{M}(A_{2p,s}x) + \mathcal{M}(A_{2p+1,s}x), \\ A_{2p,s} \oplus A_{2p+1,s} & \text{otherwise.} \end{cases} \tag{27.33}$$

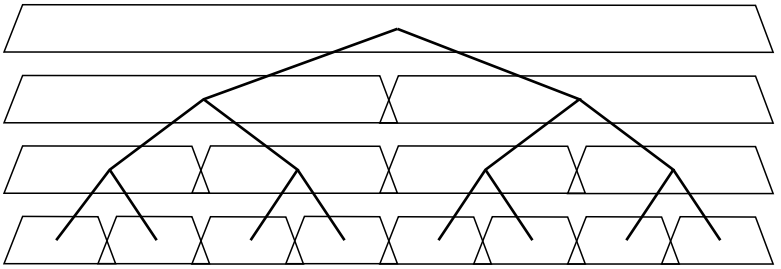


FIGURE 27.9 Binary tree of the localization of partitioned intervals.

where

$\oplus$ denotes a concatenation operation

$\mathcal{M}(\cdot)$ is a certain cost function

The resulting best basis is optimal relative to the cost function $\mathcal{M}$, which means different choices of the cost function reflecting certain criterion can result in different selections of the best basis. Such an approach is called ''entropy-based best basis selection.'' For a complete treatment on the subject, see Ref. [32].

The best basis can be sought with the local cosine basis on the Radon slices. The resulting transform is called the polar cosine packet transform (PCPT), while the choice of the cost function $\mathcal{M}$ depends on the application.

27.2.8 Multiscale Polar Cosine Transform

It was recognized, as early as in 1978 [33], that, for an image representation to be useful, the transform should be well spatially localized. More importantly, the transform should be multiscale, in order to capture patterns of interest at different scales. While the PCT is able to provide a sparse expansion for directional patterns with good frequency resolution, putting it into a multiresolution, spatially localized manner is required. The result is a new wavelet-type transform called the multiscale polar cosine transform (MPCT), which is discussed in the following sections.

27.2.8.1 Construction of MPCT

A prototypical multiscale polar cosine function has the form

$$\Psi_{k,\vec{\theta},s,\vec{\eta}}(\vec{\xi}) = w\left(\frac{\vec{\xi}-\vec{\eta}}{s}\right)\psi_k\left(\frac{(\vec{\xi}-\vec{\eta})\cdot\vec{\theta}}{s}\right). \tag{27.34}$$

where

$k, \vec{\theta}, \vec{\eta}$, and s denote the frequency, orientation, location, and scale parameters of the function, respectively

$w(\cdot)$ is the smooth window function chosen along with the sampling interval to ensure invertibility of the discrete form of the transform

Effectively the transform can be viewed as a stack of windowed polar cosine transforms at a range of scales, with different block sizes for windowing. The coarser the scale s is, the larger the window becomes. The digital implementation of such a multiscale lapped transform requires two operators: a

level operator which decomposes the signal into different scales, and a local operator which handles the decomposition locally within a block—in this case could be either the PCT operator $\mathcal{C}$ or the Polar cosine packet operator $\mathcal{P}$, which gives us the multiscale polar cosine packet transform (MPCPT).

A desirable level operator should have the following two properties:

1. Operator should be able to separate the signal into different frequency subbands.
2. Decomposed subbands should be isotropic, which then can be exploited well by the high-frequency resolution and the directional selectivity of the PCT.

A reasonable choice of the level operator is the Laplacian pyramid [34]. As in Chapter 3, for a particular level x_s of subband, it is computed as

$$x_s(\vec{\eta}) = (\mathbf{I} - \mathbf{G}_{s,s+1}\mathbf{G}_{s+1,s})x'_s. \tag{27.35}$$

where

$\mathbf{I}$ is the identity operator

x'_s is the Gaussian lowpass pyramid representation of $x(\vec{\eta})$

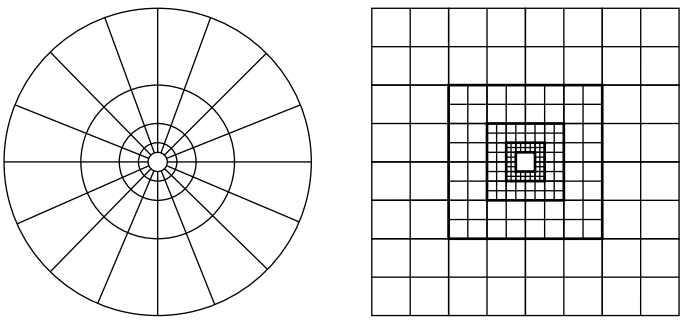
$$x'_s(\vec{\eta}) = \sum_{l=0}^{s-1} \mathbf{G}_{l+1,l}x(\vec{\eta}). \tag{27.36}$$

$\mathbf{G}_{s,s+1}, \mathbf{G}_{s+1,s}$ are the raising and lowering operators associated with transitions between levels in the Gaussian pyramid, as defined previously in Chapter 3

Due to the dyadic decimation, the pyramid is known to have some 33% extra redundancy in 2-D, which is acceptable for general use in many image processing tasks. This also allows us to use a constant window size over different scales, which is equivalent to a dyadic increment of the window size.

27.2.8.2 Basis Functions and Frequency Tiling

The basis functions of the MPCT are 2-D cosine ridges with certain frequency, orientation confined to certain spatial location and scale. Several MPCT basis functions are shown in both spatial and frequency domains in Figure 27.10. These are three members of the MPCT basis at increasing scales on a 256×256 grid. Every function oscillates coherently in a preferred direction and frequency and confined by a



Continuous case Discrete case

FIGURE 27.10 Basis functions at different scales and their frequency responses.

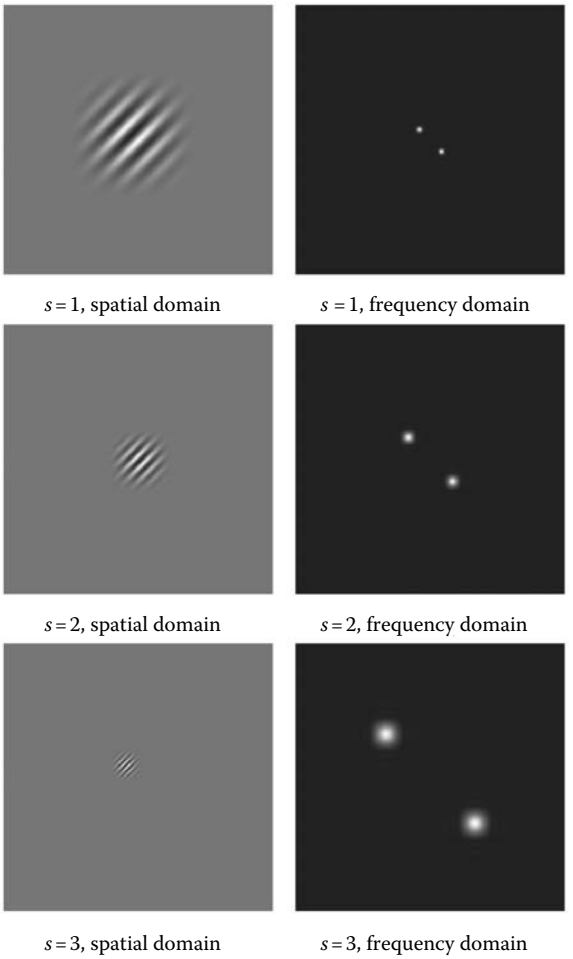


FIGURE 27.11 Frequency tilings of the MPCT.

cosine-squared window. In the frequency domain, the Fourier transform of these functions is well-localized as well, exhibiting two symmetric blobs at an orientation perpendicular to that of the spatial functions. The size of the frequency blob increases with the scale, while the spatial window size decreases dyadically.

Figure 27.11 schematically demonstrates how the frequency plane is partitioned by the MPCT decomposition, both in continuous and discrete cases. It is clear that the polar orientation resolutions at different scales are the same and each of the subbands from the level operator is partitioned into equal number of boxes.

27.2.9 Relationship with Other Transforms

The construction of the MPCT presented here follows a long legacy in both harmonic analysis and image processing, and is directly connected with many other transforms proposed in the literature.

27.2.9.1 Multiresolution Fourier Transform

The multiresolution Fourier transform (MFT) is a windowed Fourier transform at different resolutions and the digital implementation is usually done by a similar lapped block transform like the MPCT with a

local FFT operator. Due to the close relationship between PCT and FFT, the MPCT can also be seen as a real version of the MFT, and the assumption of polar symmetric boundary extension increases the sparsity of the transformed coefficients. Unlike the conventional 2-D DCT, which is constructed by tensor product of the 1-D transform, the local operator PCT of MPCT is a true polar transform, combining the merits of both FFT and DCT into one setting.

27.2.9.2 Ridgelets and Curvelets

The first digital curvelet transform, the so-called curvelet-99 implementation proposed in Ref. [7] also adopts a multiscale lapped block transform approach. The level operator used in that specific transform was an undecimated *á trous* wavelet transform. While the redundancy is useful in the denoising task, substituting it with a less redundant operator like the Laplacian pyramid would still retain the curvelet notion. The local operator is the digital ridgelet transform which is a 1-D wavelet transform on Fast Slant Stack Radon slices. Therefore, the essential difference between the curvelet-99 transform and the MPCT transform is just the 1-D transform performed on Radon slices: one being the wavelet and another being the cosine transform.

In Ref. [35], implementations of possible digital ridgelet packets were discussed. One of these uses a basis on the Radon domain from a wavelet packet or cosine packet dictionary. This coincides with the polar cosine packets discussed before, which can be used to deal with features like both ridgelet and polar cosine basis functions.

It is clear that the Fourier Slice Theorem is the fundamental tool in implementing the Radon transform, and it links the curvelets, polar cosine, and Fourier transforms. The relationship between these multiscale lapped transforms can be illustrated schematically as in Figure 27.12.

27.2.9.3 Brushlets and Wave Atoms

The brushlets, proposed by Meyer and Coifman [19], partition the Fourier frequency plane by local cosine windowing, achieving basis functions of directional oscillating patterns localized in orientation, frequency, location, and scale. However, due to the nature of frequency implementation and the choice of the window, the basis functions are not well localized in space, with significant spreads. The recently introduced 2-D wave atoms frame [36] as implemented by using the Villemoes wavelet packets [37] in the frequency domain. The resulting basis functions are much like brushlets, but with only two bumps in the spatial domain.

The MPCT's basis functions are in essence same as the brushlets and wave atoms, but the transform is implemented by spatial windowing instead of frequency windowing. The MPCT and these two transforms are spatial-frequency duals of the same idea, like wavelet packets and the local trigonometric bases.

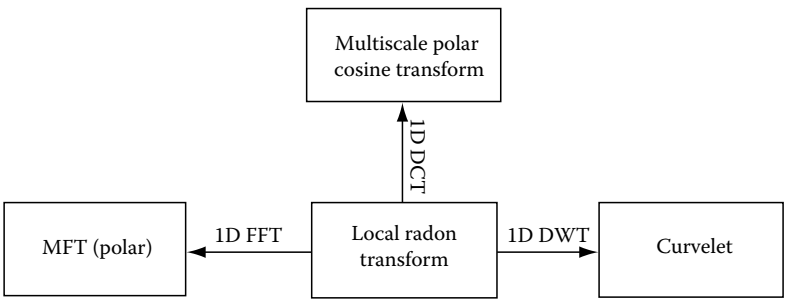


FIGURE 27.12 Relationships between MFT, local radon, curvelet, and MPCT.

27.2.10 Image Denoising in the PCT Domain

In order to demonstrate the effectiveness of the MPCT transform, the transform is applied to the task of noise removal in 2-D by thresholding the transformed coefficients.

The MPCT implementation used in the denoising has the following configuration:

1. Level of decomposition, or the total number of scales is set to $J = 5$.
2. Local window is 16×16, modulated with a squared cosine.
3. Windows are overlapped by 50%.
4. Threshold for each scale is computed as

$$\mathcal{T}_s = a \frac{\sqrt{2 \log M^2} \sigma}{1.23^{J-s}} \tag{27.37}$$

with a set to $a = 0.08$ for the MPCT and $a = 0.062$ for the MPCPT. In the above equation, σ is the standard deviation of the noise. The best basis is computed according to the cost function [38]:

$$\mathcal{M}(f, \mathcal{T}_s) = \sum_{i=1}^{N} \Phi\left(|\langle f, \psi_k[i] \rangle|^2\right), \tag{27.38}$$

where

$$\Phi(u) = \begin{cases} u - \sigma^2 & \text{if } u \leq \mathcal{T}_s^2, \\ \sigma^2 & \text{if } u > \mathcal{T}_s^2. \end{cases} \tag{27.39}$$

27.2.11 Denoising Results

The potential of nonlinear filtering in the PCT domain is demonstrated here for two standard images: barbara which contains some directional and nondirectional periodic textures and lena which can be regarded as one of the ''curvelet-friendly'' images, since it mainly consists of linear singularities at different scales. These images are given in Figure 27.13.

It can be seen from the denoising results in Figures 27.14 and 27.15 that the wave atoms approach produces unpleasant artifacts on lena, due to the fact that its basis functions are not well localized



Barbara Lena

FIGURE 27.13 Two standard images for denoising experiments.

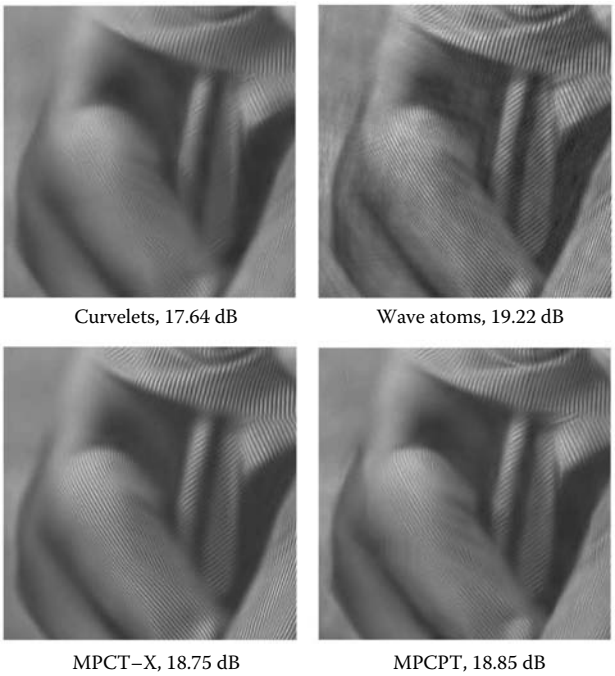


FIGURE 27.14 Comparative denoising results on detailed 10 dB barbara.

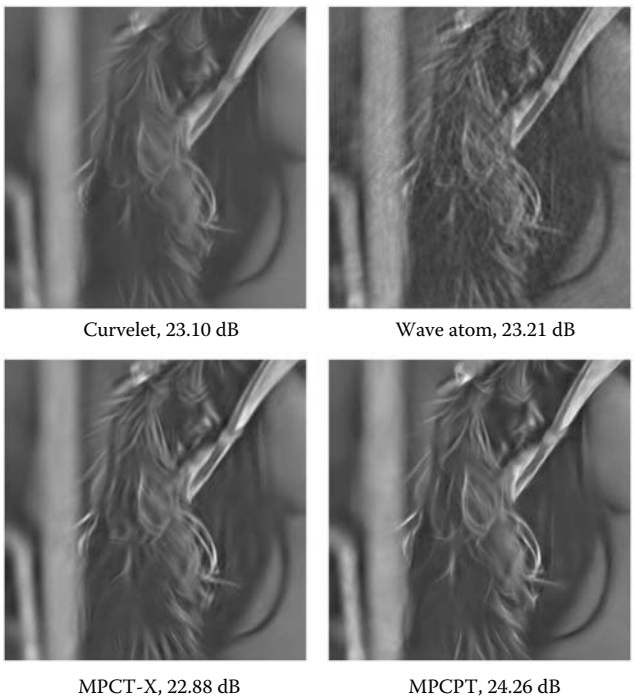


FIGURE 27.15 Comparative denoising results on detailed 15 dB lena.

in space and due to its inability to model edges well. The curvelet transform, MPCT, and MPCPT are all constructed by block-based local transforms on top of the Laplacian pyramid. The MPCT gives consistently better numerical results than curvelets, as well as well-matched visual results. Although the underlying local operator has an oscillatory pattern instead of anisotropic ridge forms, the overlapping blocks effectively compensate for the drawback in edge areas by confining the oscillations locally. In textural regions, the MPCT shows significant gains over the curvelets. The MPCT packet denoising, as expected, combines the merits of both curvelet transform and the MPCTs into one setting.

The difference in the numerical results between the MPCT-II and MPCT-X is compatible with the nonlinear approximation results in Section 27.2.5. The MPCT-X generally performs better with heavier noise, and MPCT-II can outperform MPCT-X by a significant margin when there is less noise. However, visually, the two implementation do not differ significantly. Overall, the MPCPT can be regarded as the winner for its better visual quality as well as numerical error measures.

27.3 Planelets

We now turn our attention to planelets, a special-purpose nonseparable representation for 3-D image data.

27.3.1 Introduction

As mentioned earlier in this chapter, locally planar structures, such as moving edges or interfaces between volumes, convey most of the information in 3-D image data. Preservation of such features requires a basis in which they are sparsely represented. An obvious choice might, therefore, seem to be the Fourier basis, since a planar surface in a volume corresponds to a line in the Fourier domain. Of course, this misses the key epithet: *local*; all image data show only local planarity, with curvature a significant feature at larger scales.

Planelets are designed specifically to efficiently represent planar singularities in 3-D image data. Extraction of such planar features may be useful in various applications, such as video denoising, video coding, geometry estimation [39], and tracking of objects in video sequences. Planelet representation offers translation invariance, good orientation-selectivity, localization in both space and time, invertibility, and efficient computation. The planelet basis has a combination of scaling, locality, and directional characteristics which are well matched to the locally planar surfaces of interest in applications.

Planelets can be regarded as a modification of the complex wavelet bases proposed in Refs. [40,41]. The computational complexity of a planelet transform is $O(n)$, where n is the number of points in analysis window. The computational complexity of planelet denoising is essentially equivalent to a windowed Fourier transform (WFT) and it does not require any motion estimation. In its current form, the representation provides a nonorthogonal basis and is redundant by less than 14%.

27.3.2 Continuous Planelet Transform

A prototypical planelet basis function in 1-D is of the following form:

$$f_{\xi,\omega,a}(x) = win\left(\frac{x-\xi}{a}\right)\exp\left[-j\frac{\omega(x-\xi)}{a}\right], \tag{27.40}$$

where ξ, ω, and a are, respectively, the location, frequency, and scale parameters of the function. The function $win(\cdot)$ is a window function, chosen alongwith the sampling interval to ensure invertibility

of the discrete form of the transform. In 2-D, the planelet basis can be regarded as a modification of the complex wavelet bases proposed in Refs. [40,41], which show both translation invariance and directional selectivity, and may be used as an alternative to the ridgelet representation. In 3-D, the basis comprises of the set of Cartesian products over ξ, ω at each scale a. That the continuous transform defined by Wilson et al. [40] is invertible follows directly from the observation that it is simply the MFT [40].

27.3.3 Discrete Planelet Transform

The discrete form, however, is significantly different from that described in Ref. [40]. The discrete planelet transform (DPT) is a combination of two well-known image transforms: the Laplacian pyramid [42] and the WFT. In some ways, it is similar to the octave band Gabor representation proposed in Ref. [43], but avoids some of the more unpleasant numerical properties of the Gabor functions. The DPT of a video sequence x, in vector form, at scale m is given by

$$\hat{X}_m = \mathcal{F}_n(I - G_{m,m+1}G_{m+1,m})x_m, \tag{27.41}$$

where

$\hat{X}_m$ denotes the DPT at scale m
$\mathcal{F}_n$ is the WFT operator with window size $n \times n \times n$
I is the identity operator
x_m is the Gaussian pyramid representation of x at level m

$$x_m = \prod_{0}^{m-1} G_{l+1,l}x \tag{27.42}$$

$G_{m,m+1}, G_{m+1,m}$ are the raising and lowering operators associated with transitions between levels in the Gaussian pyramid

Invertibility follows directly from Equations 27.41 and 27.42.

THEOREM 27.2

The representation defined by Equation 27.41 is invertible.

Proof: First we note that the WFT operator $\mathcal{F}_n$ has an inverse, which can be denoted by $\mathcal{F}_n^{-1}$. Second, we know from Burt and Adelson that the Laplacian pyramid is invertible, since, trivially,

$$x_m = x_m - G_{m,m+1}x_{m+1} + G_{m,m+1}x_{m+1} \tag{27.43}$$

and the proof is completed by induction on m.

Importantly, although both the pyramid and WFT operators are Cartesian separable, the closeness of the Burt and Adelson filter to a Gaussian function gives the pyramid virtually isotropic behavior, which can be exploited well by the high-frequency resolution of a Fourier basis. The planelet basis functions resemble planar structures and have compact support in both space–time and spatiotemporal frequency.

27.3.4 Denoising Algorithm

The denoising algorithm works in three steps: forward DPT, adaptive thresholding, and inverse DPT. The coefficients of D forward DPT of the image volume are computed using the algorithm outlined in Section 27.3.3. Since the presence of additive Gaussian white noise means that almost all the DPT coefficients are affected by it, soft thresholding would reduce the contribution of noise to the restored image volume. Based on the assumption that the coefficients relatively small in magnitude at each resolution (i.e., below a certain threshold) are most probably due to the noise variation, coefficients with magnitude above a certain threshold are kept while the remaining ones are discarded. The inverse DPT, therefore, provides an estimation of the original uncorrupted image volume.

The choice of threshold is crucial to the performance of this type of transform domain denoising. Donoho and Johnstone [44] have shown that an adaptive threshold θ given by

$$\theta = \sigma\sqrt{2\log n} \tag{27.44}$$

is an asymptotically optimal choice for threshold value when denoising a 1-D noisy signal, where n denotes the number of samples in the signal and σ is standard deviation of the additive Gaussian white noise. Our experiments showed that using an adaptive threshold for coefficients at different resolutions gives better denoising results as compared to using same threshold value for transform coefficients at all resolutions. We use threshold value θ_i for coefficients at level i of the pyramid as given by

$$\theta_i = \mathcal{L}(\sigma)\sqrt{2\log n_i}, \tag{27.45}$$

where

n_i denotes the number of pixels at level i of the pyramid
$\mathcal{L}(\sigma)$ is a suitably chosen function of σ

The following expression for $\mathcal{L}(\sigma)$ was empirically chosen for our experiments:

$$\mathcal{L}(\sigma) = a\log_{10}\sigma + b, \tag{27.46}$$

where $a, b \in \Re$ and $b = 2a$.

27.3.5 Planar Feature Extraction

Planelets provide an ideal tool for representing local planes in a video sequence (or an image volume, in general) due to their ability to localize planar surfaces which correspond to lines in the Fourier domain. The presence of planar surface in a local analysis window can be inferred by computing the eigenvalues of the local inertia tensor in the window and analysing them. The parameters for orientation of the local planar surface and translation from center of the window can also be estimated by analyzing the most significant coefficients in the locality. Consider a video sequence synthesized by moving the center of a circle on a sinusoidal wave in the time direction. Nonlinear approximations of this sequence using only 0.07% of the wavelet and planelet coefficients are shown in Figures 27.16a and b, respectively. It is clear from this example that the planelet approximation of a video sequence containing locally planar surfaces can result in a smaller approximation error as compared to that using wavelets. Planelets, therefore, can also be used for a piecewise planar

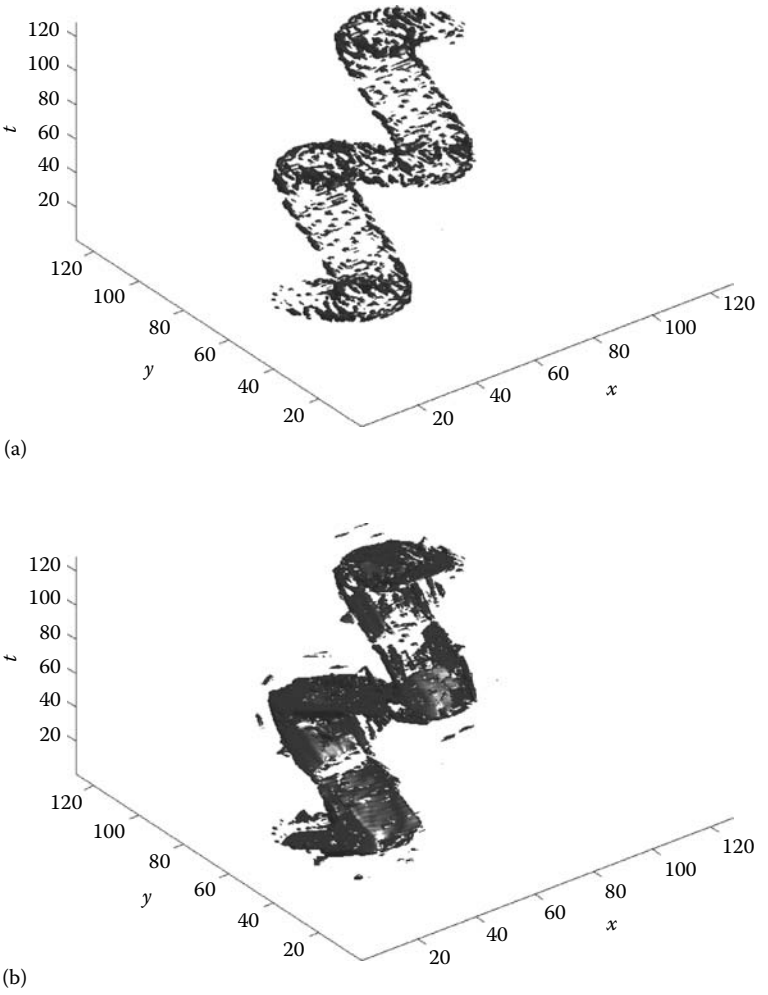


FIGURE 27.16 Nonlinear approximation of a video sequence containing a moving circle using (a) wavelets and (b) planelets.

approximation of a video sequence. Moreover, the approximation can be made to be adaptive to the local scale of the planar surfaces.

27.4 3-D Wavelet Packets

The case for thresholding in spatiotemporal wavelet domain is supported by the fact that certain errors in motion estimation can be overcome by including the temporal direction in the realm of wavelet domain. Recent attempts to solve the video restoration problem have included combined spatial and temporal wavelet denoising [45], and the use of thresholding in nonseparable transform domains, such as oriented 3-D ridgelets [46] and 3-D complex wavelets [47].

Although wavelet shrinkage performs significantly better than most other commonly used denoising methods, visual quality of the restored video can sometimes suffer from *ringing* type of artifacts, valleys

around the edges, due to the Gibbs phenomenon. The shift-variant nature of the wavelet transform worsens the effect of the Gibbs phenomenon, resulting in unpleasant artifacts. Translation invariant (TI) wavelet denoising of Coifman and Donoho [48] was developed to counter such artifacts by averaging out the translation dependence. Another feature of wavelet denoising is that it imposes a fixed dyadic wavelet basis on all types of input signals. Not only can the use of dyadic wavelets result in a blurred reconstruction, it can also limit the analysis of a locally occurring phenomenon in the spatiotemporal frequency domain. The solution to this problem lies in the use of basis functions which are well localized in spatiotemporal frequency as well as in space and time.

27.4.1 Adaptive 3-D WP Transform

The ability of wavelet packets to capture locally occurring frequency phenomena in a signal has led to their successful application to many problems including image coding [49,50]. The fundamental idea is to relax the restricted decomposition of only the lowpass subband and allow the exploration of all frequency bands up to the maximum depth. The discrete wavelet packet transform (DWPT) of a 1-D signal x of length N can be computed as follows:

$$
\begin{aligned}
w_{2n,d,l} &= \sum_k g_{k-2l} w_{n,d-1,k} \quad && l = 0, 1, \ldots, N2^{-d} - 1, \\
w_{2n+1,d,l} &= \sum_k h_{k-2l} w_{n,d-1,k} \quad && l = 0, 1, \ldots, N2^{-d} - 1, \\
w_{0,0,l} &= x_l && l = 0, 1, \ldots, N - 1,
\end{aligned}
$$

where $d = 1, 2, \ldots, J-1$ is the scale index, with $J = \log_2 N$, n and l, respectively, denote the frequency and position indices, $\{h_n\}$ and $\{g_n\}$ correspond to the lowpass and highpass filters, respectively, for a two-channel filter bank and the transform is invertible if appropriate dual filters $\{\tilde{h}_n\}, \{\tilde{g}_n\}$ are used on the synthesis side. These equations can be used to compute full wavelet packet (FWP) tree of the signal decomposition. However, this implies that a large number of combinations of basis functions is now available to completely represent the signal. A tree-prunning approach such as Ref. [51] can be used to efficiently select the *best basis* with respect to a cost function.

The 3-D DWPT can be computed by applying above equations separably in all three directions to get the FWP decomposition up to the coarsest resolution of subbands. The best basis can be selected in $O(N \log N)$ time, where N denotes the number of samples (frame resolution times the number of frames) in the video sequence. Given the goal here is to capture the significant spatiotemporal frequency phenomena in a video sequence, we used the Coifman–Wickerhauser entropy [51] as a cost function to select the best basis.

27.4.2 Restoration Algorithm

The effect of the Gibbs phenomenon can be weakened by averaging the restored signal over a range of circular shifts [48]. For this reason, we apply soft threhsolding to the 3-D wavelet packet coefficients of the shifted (in all three directions) noisy video sequence. A modified BayesShrink [52] method is used to compute the optimal value of threshold adaptively for each subband. Threshold θ_b for a subband of length N in an L-level WP decomposition is given by

$$\theta_b = \sqrt{\log N/L}\left(\frac{\sigma^2}{\sqrt{\max\left(\sigma_b^2 - \sigma^2, 0\right)}}\right),$$

where
σ_b^2 is the subband variance
σ^2 is the noise variance

If σ^2 is not known, a robust median estimate for noise standard deviation $\hat{\sigma}$ is obtained as follows:

$$\hat{\sigma} = \mathcal{E}\{\hat{\Sigma}\}, \quad \hat{\sigma}_i = \frac{\text{Median}(|Y_i|)}{0.6745},$$

where $\hat{\sigma}_i \in \hat{\Sigma}$, $Y_i \in \{\mathcal{Y}\}$, set of all HHH bands in the decomposition tree, and the mean $\mathcal{E}$ is taken only on the smaller half of the sorted $\hat{\Sigma}$ excluding the smallest value.

27.5 Discussion and Conclusion

The denoising algorithms presented in Sections 27.3.4 and 27.4.2 were tested against a number of other algorithms for restoration of several standard video sequences, three of which are included here: *Miss America, Hall,* and *Football*, all at a resolution of 128^3. The video sequences were corrupted with additive white Gaussian noise, with the SNR of the noisy sequences being 0, 5, 10, and 15 dB. Table 27.3 gives denoising results in terms of SNR for these noisy sequences using the following algorithms: TI hard thresholding in 3-D wavelet domain (TIW3-D) 3-D wavelet packet (WP3-D) with BayesShrink [52], both non-TI and TI 3-D wavelet packet (TIWP3-D) with the modified form of

TABLE 27.3 SNR Results for Three Standard Video Sequences

		Denoising Algorithm (Transform + Thresholding)				
Video Sequence	Noise (dB)	TIW3D Hard	WP3D Bayes	WP3D Proposed	TIWP3D Proposed	Planelet SURE
	0	17.9	17.2	17.4	**18.9**	17.3
	5	19.5	19.0	19.3	**20.7**	19.6
Miss America	10	21.5	21.1	21.5	**23.0**	21.5
	15	23.9	23.1	23.8	**25.2**	23.5
	0	14.7	14.8	15.0	**16.7**	14.8
	5	16.6	17.2	17.3	**18.9**	17.2
Hall	10	19.0	19.5	19.8	**21.3**	19.5
	15	21.7	22.1	22.5	**24.1**	21.8
	0	11.9	11.9	12.0	**12.8**	12.1
	5	13.1	12.9	13.3	**14.3**	13.2
Football	10	15.0	13.9	15.2	**16.9**	14.7
	15	18.0	15.3	17.9	**20.0**	16.6

BayesShrink described in Section 27.4.2, and nonseparable planelet [53] domain thresholding using SUREShrink [54] method. Comparative SNR curves for individual frames for two of the test sequences are provided in Figure 27.17. For all our experiments, the proposed algorithm produces by far the best results in terms of both overall and individual SNR. Some of the frames of the test sequences restored by our algorithm and TIW3-D-Hard, a 3-D realization of the algorithm in Ref. [48], are shown in Figure 27.18. While TIW3-D restores clean and smooth version of the original frames, some of the details are restored by TIWP3-D.

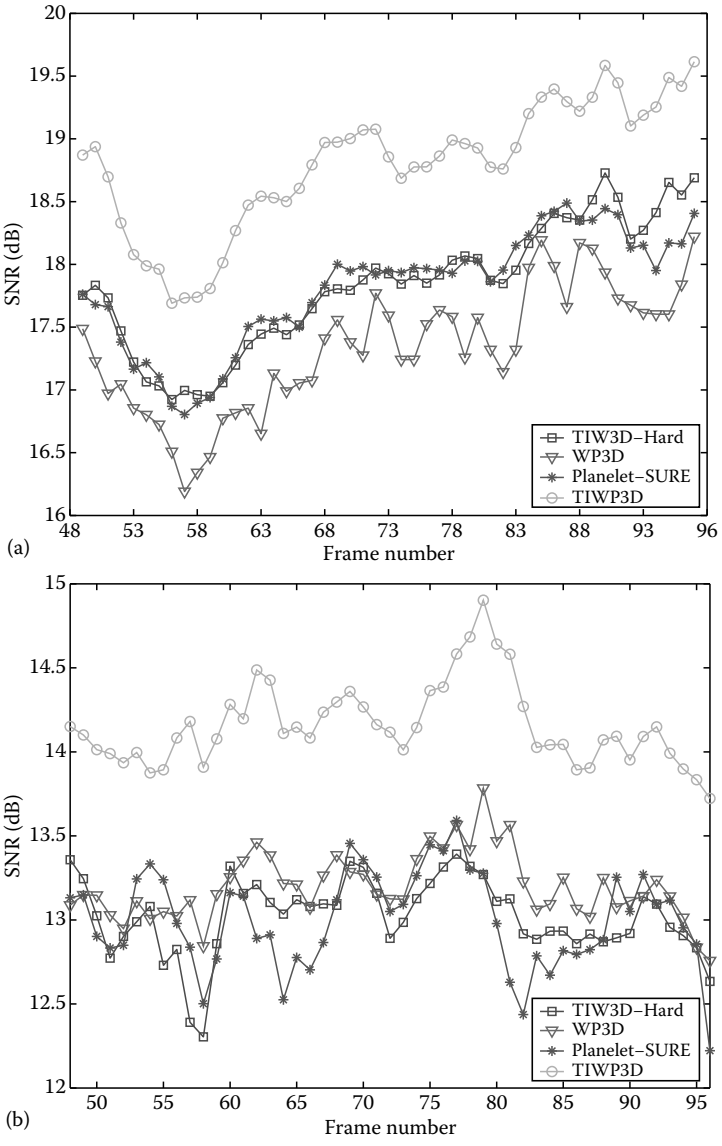


FIGURE 27.17 Frame-by-frame comparative results (a) *Miss America* and (b) *Football*.

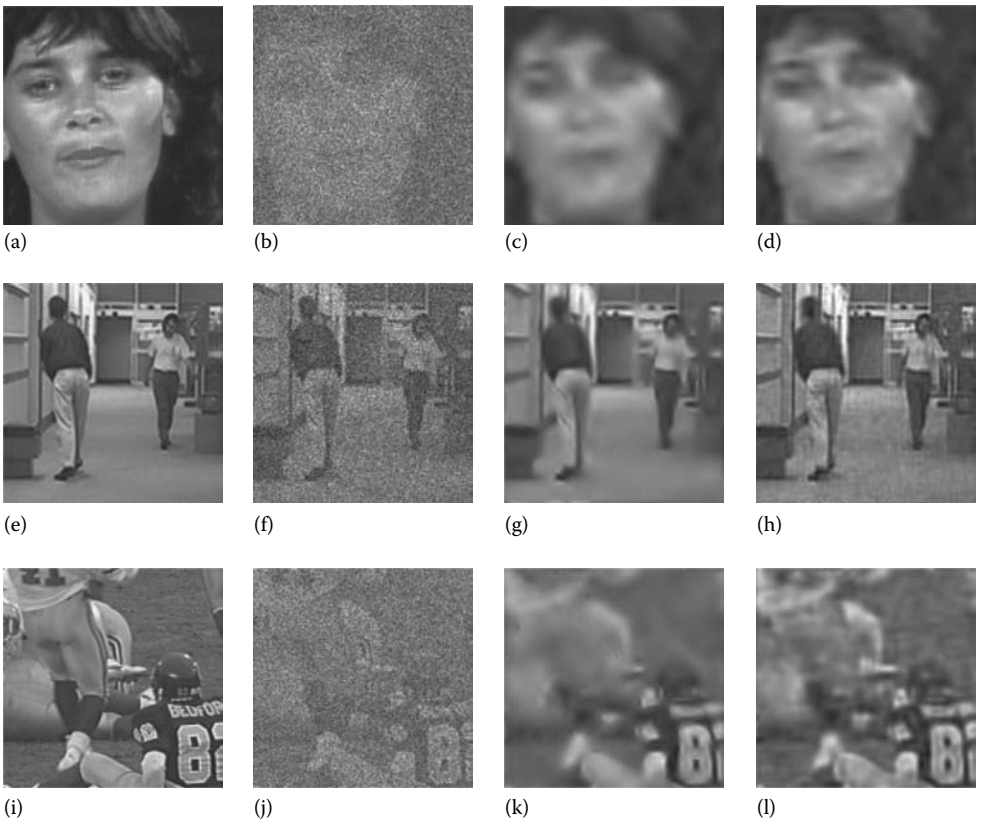


FIGURE 27.18 Denoising results for three standard video sequences. Frame# 90 of *Miss America*: (a) Original, (b) Noisy (SNR = 0 dB), (c) TIW3D-Hard (SNR = 17.9 dB), and (d) TIWP3D (SNR = 18.9 dB); Frame# 106 of *Hall*: (e) Original, (f) Noisy (SNR = 10 dB), (g) TIW3D-Hard (SNR = 19.0 dB), and (h) TIWP3D (SNR = 21.3 dB); Frame# 60 of *Football*: (i) Original, (j) Noisy (SNR = 5 dB), (k) TIW3D-Hard (SNR = 13.1 dB), and (l) TIWP3D (SNR = 14.3 dB).

For comparison purposes, computational complexity for each of the algorithms considered is also provided in Table 27.4. It is clear from this table that the planelet algorithm of Ref. [53] is the least computationally expensive, whereas the TI implementations of 3-D wavelet and 3-D WP are towards the more expensive side with TIWP3-D being the most expensive due to the additional one-off cost of best basis selection.

TABLE 27.4 Computational Complexity of the Tested Algorithms

Algorithm	Complexity
TIW3D Hard	$O(N + l^3 N)$
WP3D Bayes	$O(N \log N)$
WP3D Proposed	$O(N \log N)$
TIWP3D Proposed	$O(N \log N + l^3 N)$
Planelet SURE	$O(n)$

Note: N and n, respectively, denote sequence size and the planelet window size, and l denotes length of the wavelet filter.

In conclusion, planelets are an efficient representation tool for 3-D functions with planar singularities. Such singularities are commonly found in video sequences in the form of moving luminance edges. A piecewise planar approximation of a video sequence can be obtained by using a very small fraction of transform coefficients in the planelet domain. The ability of planelets to extract planar features from a video sequence makes them an attractive tool for analysis in various applications. It is worth noting that while being the least expensive, the planelet method [53] produces SNR results which are still comparable to those of TIW3-D-Hard. These results also suggest that the localization of spatiotemporal frequency, achieved by TIWP3-D, is a desirable feature of the domain in which video sequences are represented.

References

1. S. Clippingdale and R. Wilson. Least squares image restoration based on a multiresolution model. In *Proceedings ICASSP-89,* Glasgow, U.K., 1989.
2. D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.
3. B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properities by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.
4. B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.
5. E. J. Candés. Ridgelets: Theory and applications. PhD thesis, Department of Statistics, Stanford, University, Stanford, CA, 1998.
6. E. J. Candés and D. L. Donoho. Curvelets—A suprisingly effective non-adaptive representation for objects with edges. In C. Rabut, A. Cohen, and L. L. Schumaker, editors, *Curves and Surfaces*, pp. 105–120. Vanderbilt University Press, Nashville, TN, 2000.
7. J. Starck, E. J. Candés, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, June 2002.
8. D. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics*, 27:353–382, 1999.
9. D. L. Donoho and X. Huo. Beamlets and multiscale image analysis, *Multiscale and Multiresolution Methods: Theory and Applications,* 20:149–196, 2002.
10. M. N. Do and M. Vetterli. Contourlet. In G. V. Welland, editor, *Beyond Wavelets*. Academic Press, New York, 2003.
11. M. N. Do and M. Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Transactions Image Processing,* 14(12):2091–2106, December 2005.
12. F. G. Meyer and R. R. Coifman. Brushlets: A tool for directional image analysis and image compression. *Applied and Computational Harmonic Analysis,* 4(2):147–187, 1997.
13. P. Pongpiyapaiboon. Development of efficient algorithms for geometrical representation based on arclet decomposition. Master's thesis, Technische Universität München, Germany, 2005.
14. N. G. Kingsbury. Image processing with complex wavelets. *Philosophical Transactions of the Royal Society London*, A(357):2543–2560, September 1999.
15. P. Auscher, G. Weiss, and M. V. Wickerhauser. Local sine and cosine bases of Coifman and Meyer and the construction of smooth wavelets. In C. K. Chui, editor, *Wavelets: A Tutorial in Theory and Applications,* pp. 237–256. Academic Press, San Diego, 1992.
16. G. Aharoni, R. Coifman A. Averbuch, and M. Israeli. Local cosine transform—A method for the reduction of the blocking effect in JPEG. *Journal of Mathematical Imaging and Vision*, 3:7–38, 1993.
17. R. R. Coifman and Y. Meyer. Orthonormal wave packet bases. Technical report, Department of Mathematics, Yale University, New Haven, 1990.
18. K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate distortion sense. *IEEE Transactions on Image Processing*, 2(2):160–175, April 1993.

19. F. G. Meyer and R. R. Coifman. Brushlets: Steerable wavelet packets. In J. Stoeckler and G. V. Welland, editors, *Beyond Wavelets*, pp. 1–25. Academic Press Inc., New York, 2001.
20. R. Wilson and G. H. Granlund. The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(11):758–767, 1984.
21. A. Pinkus. Approximating by ridge functions. In A. Le Mehaute, C. Rabut, and L. L. Schumaker, editors, *Surface Filling and Multiresolution Methods*, pp. 279–292, Vanderbilt University Press, Nashville, TN, 1997.
22. R. N. Bracewell. Numerical transforms. *Science*, 248:697–704, 1990.
23. N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 100(23):90–93, January 1974.
24. G. K. Wallace. The JPEG still picture compression standard. *Consumer, Electronics, IEEE Transactions*, 38(1):30–44, April 1992.
25. A. Averbuch, R. Coifman, D. L. Donoho, and M. Israeli. Fast slant stack: A notion of Radon transform for data in a Cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible. To appear in *SIAM Scientific Computing*.
26. A. Averbuch, R. Coifman, D. Donoho, M. Israeli, and J. Walden. The pseudopolar FFT and its applications. Technical report, University of Yale, New Haven, CI 1999. YaleU/DCS/RR-1178.
27. E. J. Candés and D. L. Donoho. Ridgelets: A key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London A*, 357(1760): 2495–2509, 1999.
28. T. Aach and D. Kunz. A lapped directional transform for spectral image analysis and its application to restoration and enhancement. *Signal Processing*, 80:2347–2364, 2000.
29. E. J. Candés, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. Technical report, California Institute of Technology Pasadena, CA, July 2006.
30. R.R. Coifman and Y. Meyer. Remarques sur l'analyse de Fourier à fenêtre. *Comptes Rendus de I'Academie des Sciences, Série 1, Mathematique*, 312(3):259–261, 1991.
31. H. S. Malvar. Lapped transforms for efficient transform/subband coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 38:969–978, 1990.
32. R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.
33. G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8:155–173, 1978.
34. P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.
35. A. G. Flesia, H. Hel-Or, A. Averbuch, E. J. Candés, R. R. Coifman, and D. L. Donoho. Digital implementation of ridgelet packets. In G. Welland, editor, *Beyond Wavelets*, pp. 31–60. Academic Press, New York, September 2003.
36. L. Demanet and L. Ying. Wave atoms and sparsity of oscillatory patterns. *Applied Computational Harmonic Analysis*, 23(3):368–387, 2007.
37. L. Villemoes. Wavelet packets with uniform time–frequency localization. *Comptes-Rendus Mathematique*, 335(10):793–796, 2002.
38. H. Krim, D. Tucker, S. Mallat, and D. Donoho. On denoising and best signal representation. *IEEE Transactions on Information Theory*, 45(7):2225–2238, November 1999.
39. A. Bhalerao and R. Wilson. A Fourier approach to 3-D local feature estimation from volume data. In *Proceedings British Machine Vision Conference*, Manchester, U.K., 2001.
40. R. G. Wilson, A. D. Calway, and E. R. S. Pearson. A generalized wavelet transform for Fourier analysis: The multiresolution Fourier transform and its applications to image and audio signal analysis. *IEEE Transancation on Information Theory*, 38(2):674–690, March 1992.
41. N. Kingsbury. Image processing with complex wavelets. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 357(1760): 2543–2560, 1999.

42. P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.
43. M. Porat and Y. Y. Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Transactions on PAMI*, 10:452–468, 1988.
44. D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 31:425–455, 1994.
45. A. Pizurica, V. Zlokolika, and W. Philips. Combined wavelet domain and temporal video denoising. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* July 2003.
46. P. Carre, D. Helbert, and E. Andres. 3-D fast ridgelet transform. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, September 2003.
47. I. W. Selesnick and K. Y. Li. Video denoising using 2-D and 3-D dual-tree complex wavelet transforms. In *Proceedings SPIE Wavelets X*, August 2003.
48. R. R. Coifman and D. L. Donoho. Translation-invariant denoising. In *Wavelets and Statistics. Lecture Notes in Statistics*, 1995.
49. F. G. Meyer, A. Z. Averbuch, and J.-O. Strömberg. Fast adaptive wavelet packet image compression. *IEEE Transactions on Image Processing*, 9:792–800, May 2000.
50. N. M. Rajpoot, R. G. Wilson, F. G. Meyer, and R. R. Coifman. Adaptive wavelet packet basis selection for zerotree image coding. *IEEE Transactions on Image Processing*, 12(12):1460–1472, December 2003.
51. R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, March 1992.
52. G. Chang, B. Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, September 2000.
53. N. M. Rajpoot, R. G. Wilson, and Z. Yao. Planelets: A new analysis tool for planar feature extraction. In *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS),* April 2004.
54. M. Jansen. *Noise Reduction by Wavelet Thresholding*. Springer-Verlag, New York, 2001.

28

Video Demosaicking Filters

28.1 Introduction **28**-1
28.2 Imaging Model **28**-2
28.3 Demosaicking Methods **28**-4
Single-Channel Interpolation • Constant-Hue-Based Interpolation • Edge-Directed Interpolation • Using Gradients as Correction Terms • Frequency-Domain Approach • Homogeneity-Directed Interpolation • Projections onto Convex Sets Approach • Spectral Response Modeling
28.4 Demosaicking of Video and Super-Resolution Reconstruction **28**-16
28.5 Related Research Problems **28**-18
28.6 Evaluation of Demosaicking Algorithms **28**-19
28.7 Conclusions and Future Directions **28**-20
Acknowledgments **28**-20
References **28**-20

Bahadir K. Gunturk
Louisiana State University

Yucel Altunbasak
Georgia Institute of Technology

28.1 Introduction

Consumer-level digital cameras were introduced in mid-1990s; in about a decade, the digital camera market has grown rapidly to exceed film camera sales. Today, there are point-and-shoot cameras with more than 8 million pixels; professional digital single lens reflex (SLR) cameras with more than 12 million pixels are available; resolution, light sensitivity, and dynamic range of the sensors have been improved significantly. Image quality of digital cameras has become comparable to that of film cameras.

During an image capture process, a digital camera performs a significant amount of processing to produce a viewable image. This processing includes auto focus, white balance adjustment, color interpolation, color correction, compression, and more. A very important part of the imaging pipeline is color filter array interpolation or demosaicking.

To produce a high-quality color image, there should be at least three color samples at each pixel location. One approach is to use beam-splitters along the optical path to project the image onto three separate sensors as illustrated in Figure 28.1. Using a color filter in front of each sensor, three full-channel color images are obtained. This is a costly approach as it requires three sensors and these sensors should be aligned precisely. A more convenient approach is to put a color filter array (CFA) in front of the sensor to capture one color component at a pixel and then interpolate the missing two color components. Because of the mosaic pattern of the CFA, this interpolation process is known as demosaicking.

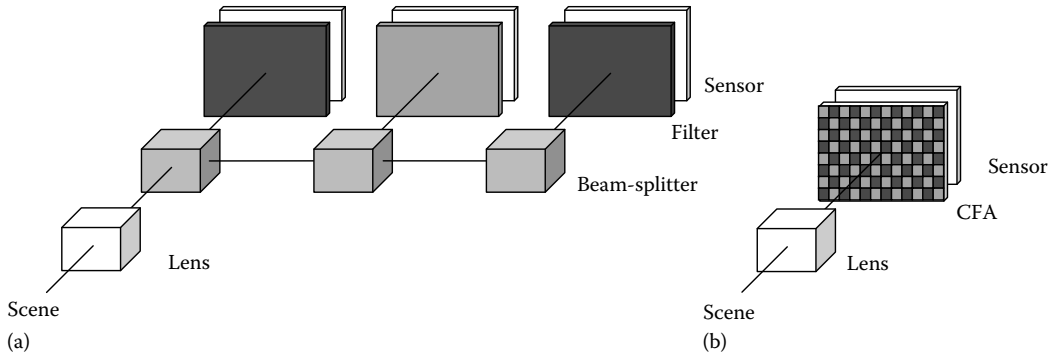


FIGURE 28.1 Illustration of optical paths for multichip and single-chip digital cameras.

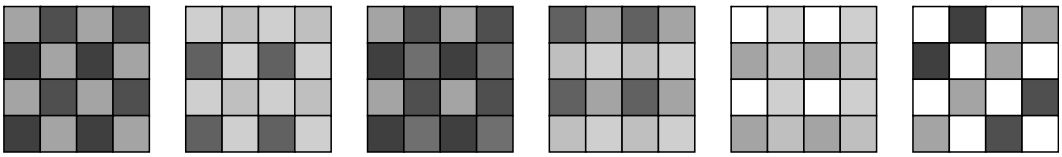


FIGURE 28.2 Several CFA designs are illustrated. From left to right: (a) This is the most commonly used CFA pattern: the Bayer CFA pattern. It consists of red, green, and blue samples. It leads to very good color reproduction performance. (b) This is the Bayer pattern with subtractive primaries: yellow, magenta, and cyan. The color filters have high transmittance values; therefore, good performance in low-light conditions is expected. (c) This pattern uses red, green, blue, and emerald. It is recently used in some Sony cameras. (d) This is a pattern commonly used in video cameras; it consists of yellow, magenta, cyan, and green. (e) This pattern consists of yellow, cyan, and green filters, and unfiltered pixels. The unfiltered pixels improve light sensitivity. (f) This is a pattern that is introduced very recently by Kodak. It has red, green, blue, and unfiltered pixels.

A variety of patterns exist for the color filter array. Some of these patterns are illustrated in Figure 28.2. Among these, the most common array is the Bayer color filter array. The Bayer array measures the green image on a quincunx grid and the red and blue images on rectangular grids. The green image is measured at a higher sampling rate because the peak sensitivity of the human visual system lies in the medium wavelengths, corresponding to the green portion of the spectrum (see Figure 28.3). Although this chapter discusses the demosaicking problem with reference to the Bayer CFA, the discussions and algorithms can in general be extended to other patterns.

The simplest solution to the demosaicking problem is to apply a standard image interpolation technique to each channel separately. However, this neglects the correlation among color channels and results in visible artifacts. For example, in Figure 28.4, Bayer sampling is applied on a full-color image and later bicubic interpolation is applied on each channel. The resulting image suffers from color artifacts. This result motivates the need to find a specialized algorithm for the demosaicking problem. There have been many algorithms published on this topic; this chapter surveys the main approaches.

28.2 Imaging Model

Most demosaicking algorithms model the imaging process as subsampling from a full-color image to a mosaicked data. This is a sufficient model when the goal is only to estimate the missing color samples. (When the goal is to obtain a higher resolution image, then the modulation transfer function of the camera should also be taken into account.)

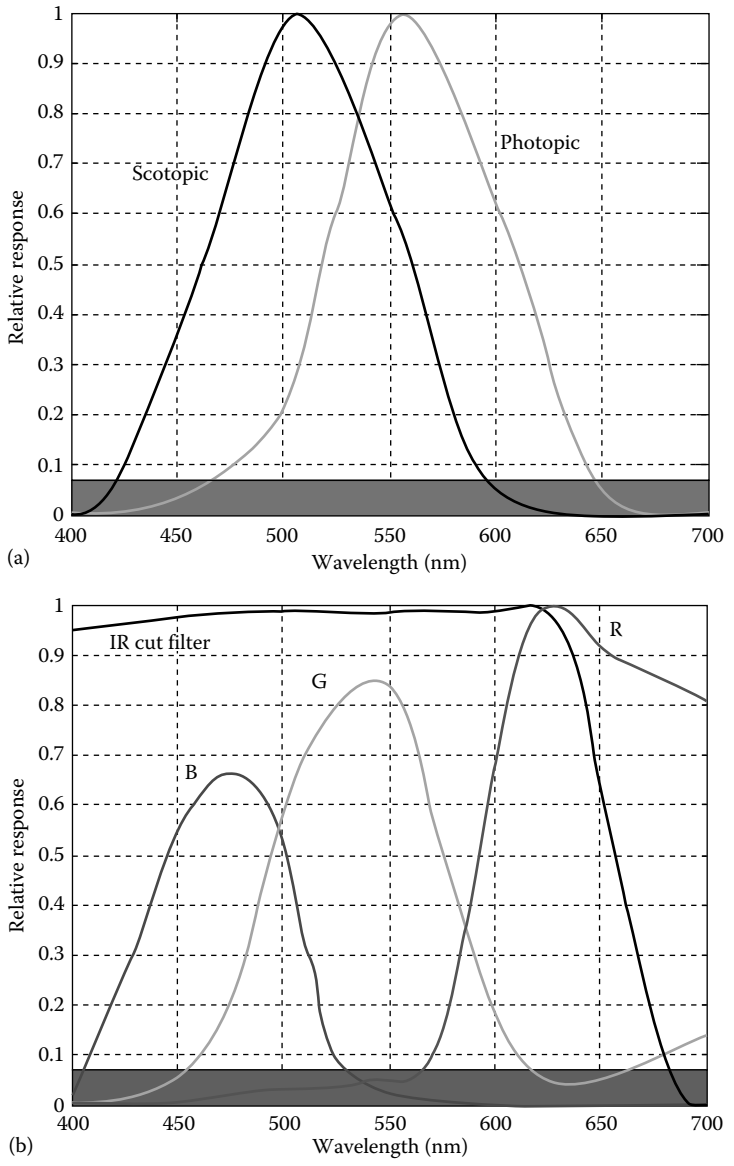


FIGURE 28.3 (a) Luminous efficiency of human visual system. Photopic response is the luminance response of the cone receptors. Scotopic response is the luminance response of the rod receptors, that is, response in low-light conditions. (b) Typical color filter responses in a digital camera.

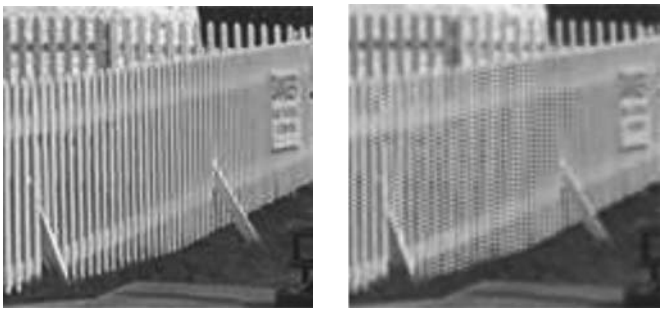
According to this model, a full-color channel S, where $S = R$ for red, $S = G$ for green, and $S = B$ for blue, is converted to a mosaicked observation z according to a CFA sampling pattern:

$$z = \sum_{S=R,G,B} z_S = \sum_{S=R,G,B} M_S S, \tag{28.1}$$

where

z_R, z_G, z_B are the subsampled color channels

mask M_S takes a color sample at a pixel according to the CFA pattern



(a) (b)

FIGURE 28.4 Bicubic interpolation used for color filter array interpolation results in numerous artifacts. (a) Original image and (b) bicubic interpolation.

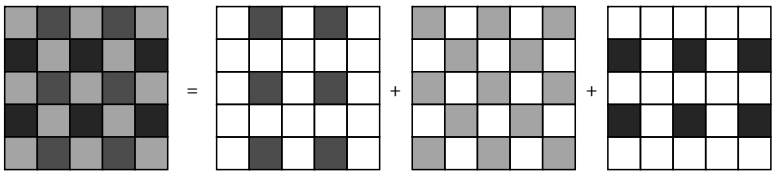


FIGURE 28.5 The mosaicked image z and the sampled components z_R, z_G, and z_B for the Bayer pattern.

For example, at a red pixel location, $[M_R, M_G, M_B]$ is [1, 0, 0].

In Ref. [1], the masks are explicitly written for a Bayer CFA in terms of cosine functions:

$$\begin{aligned} z_R(i,j) &= M_R(i,j)R(i,j) = \frac{1}{4}(1-\cos\pi i)(1+\cos\pi j)R(i,j), \\ z_G(i,j) &= M_G(i,j)G(i,j) = \frac{1}{2}(1+\cos\pi i\cos\pi j)G(i,j), \\ z_B(i,j) &= M_B(i,j)B(i,j) = \frac{1}{4}(1+\cos\pi i)(1-\cos\pi j)B(i,j), \end{aligned} \tag{28.2}$$

where (i, j) indicate the pixel coordinates, starting with $(0, 0)$. Figure 28.5 illustrates the CFA image z and the sampled components z_R, z_G, and z_B.

28.3 Demosaicking Methods

28.3.1 Single-Channel Interpolation

Single-channel interpolation methods treat each channel separately without utilizing any interchannel correlation. Standard image interpolation techniques, such as bilinear interpolation, bicubic interpolation, spline interpolation, and adaptive methods (e.g., edge-directed interpolation) are applied to each color channel individually. These methods, in general, do not perform as well as the methods that use interchannel correlation.

Among these methods, bilinear interpolation is commonly used as a part of other demosaicking methods. Figure 28.6 provides the linear filters used to perform bilinear interpolation.

$$\begin{bmatrix} 0 & 1/4 & 0 \\ -1/4 & 1 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix} \quad \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

(a) (b)

FIGURE 28.6 Filters for bilinear interpolation. (a) Filter applied on z_G to obtain the green channel and (b) filter applied on z_R/z_B to obtain the red/blue channels.

28.3.2 Constant-Hue-Based Interpolation

One commonly used assumption in demosaicking is that the hue (color ratios or differences) within an object in an image is constant. Although this is a simplification of image formation, it is a reasonable assumption within small neighborhoods of an image. This perfect interchannel correlation assumption is formulated such that the color ratios or differences within small neighborhoods are constant. This prevents abrupt changes in color intensities, and has been extensively used for the interpolation of the chrominance (red and blue) channels [2–6]. This approach is called the constant-hue-based interpolation approach.

As a first step, these algorithms interpolate the luminance (green) channel, which is typically done using bilinear or edge-directed interpolation. The chrominance (red and blue) channels are then estimated from the interpolated ''red hue'' (red-to-green ratio) and ''blue hue'' (blue-to-green ratio). To be more explicit, the interpolated ''red hue'' and ''blue hue'' values are multiplied by the green value to determine the missing red and blue values at a particular pixel location. The hues can be interpolated with any method (bilinear, bicubic, edge-directed, etc.).

As mentioned, instead of interpolating the color ratios, it is also possible to interpolate the color differences or the logarithm of the color ratios. This is illustrated in Figure 28.7.

It is also possible to update all color channels iteratively. That is, the green channel is interpolated first. The red/blue channels are interpolated using constant-hue-based interpolation. The green channel is then updated using the interpolated red/blue channels; and so on [7].

The constant difference idea is sometimes combined with median filtering and used as a postprocessing step to reduce color artifacts [5,8,9]. For example, in Ref. [9], the interpolated color channels are updated as follows:

$$\begin{aligned} G'(i,j) &= \frac{\left(R(i,j) - \underset{(i,j)}{\text{median}}(R-G)\right) + \left(B(i,j) - \underset{(i,j)}{\text{median}}(B-G)\right)}{2}, \\ R'(i,j) &= G'(i,j) + \underset{(i,j)}{\text{median}}(R-G), \\ B'(i,j) &= G'(i,j) + \underset{(i,j)}{\text{median}}(B-G), \end{aligned} \tag{28.3}$$

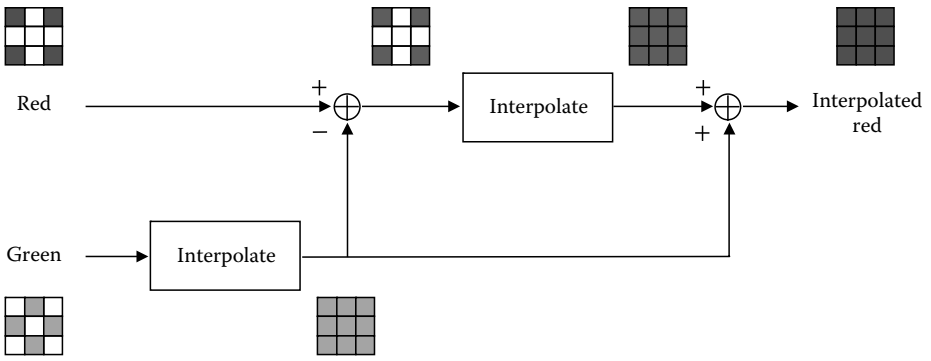


FIGURE 28.7 Constant-hue-based interpolation is illustrated for the interpolation of red channel. (From Gunturk, B. K., et al., *IEEE Signal Process. Mag.*, 22, 44, 2005. With permission.)

where median$_{(i,j)}(\cdot)$ returns the median within a small neighborhood of (i,j). In Ref. [8], red and blue channels are updated first as given in the above equation, followed by the green channel update. This procedure is repeated several times.

28.3.3 Edge-Directed Interpolation

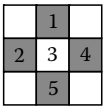
Although nonadaptive algorithms can provide satisfactory results in smooth regions of an image, they usually fail in textured regions and edges. Edge-directed interpolation is an adaptive approach, where edge detection is performed for each pixel in question, and interpolation is done along the edges rather than across them.

In the demosaicking problem, edge-directed interpolation is first applied to the green channel, which is sampled more densely and therefore is less likely to be aliased. Red and blue channel interpolations follow, based on the edge directions found for the green channel. A simple way of performing edge detection is to compare the absolute difference among the neighboring pixels [10]. Referring to Figure 28.8, horizontal and vertical gradients at a missing green location can be calculated from the horizontally and vertically adjacent green pixels. If the horizontal gradient is larger than the vertical gradient, suggesting a possible edge in the horizontal direction, interpolation is performed along the vertical direction. If the vertical gradient is larger than the horizontal gradient, interpolation is performed only in the horizontal direction. When the horizontal and vertical gradients are equal, the green value is obtained by averaging its four neighbors. It is also possible to compare the gradients against a predetermined threshold value [10].

The edge-directed interpolation approach in Ref. [10] can be modified by using larger regions (around the pixel in question) with more complex predictors and by exploiting the texture similarity in different color channels. In Ref. [4], the red and blue channels (in the 5×5 neighborhood of the missing pixel) are used instead of the green channel to determine the gradients. In order to determine the horizontal and vertical gradients at a blue (red) sample, second-order derivatives of blue (red) values are computed in the corresponding direction. This algorithm is illustrated in Figure 28.9.

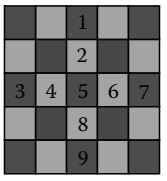
Once the missing samples of the green channel are computed, the red and blue channels are interpolated. A typical approach for the red/blue interpolation is constant-hue-based interpolation, which was explained in Section 28.3.2.

In the algorithms explained so far, the edge direction is determined first, and then the missing sample is estimated by interpolating along the edge. This is a ''hard'' decision process. Instead, the likelihood of an edge in a certain direction can be found, and the interpolation can be done based on the edge likelihoods. Such



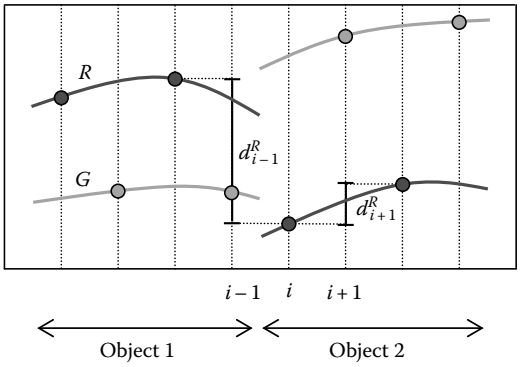
ΔH = |G2 – G4| //horizontal gradient
ΔV = |G1–G5| //vertical gradient
if ΔH > ΔV,
    G3 = (G1 + G5)/2
else if ΔH < V,
    G3 = (G2 + G4)/2
else
    G3 = (G1 + G5 + G2 + G4)/4

FIGURE 28.8 Edge-directed interpolation in Ref. [10] is illustrated. *G*1, *G*2, *G*4, and *G*5 are measured green values; *G*3 is the estimated green value at pixel 3. (From Gunturk, B. K., et al., *IEEE Signal Process. Mag.*, 22, 44, 2005. With permission.)



ΔH = |(R3 + R7)/2 – R5| //horizontal gradient
ΔV = |(R1 + R9)/2 – R5| //vertical gradient
if ΔH > ΔV,
    G5 = (G2 + G8)/2
else if ΔH < V,
    G5 = (G4 + G6)/2
else
    G5 = (G2 + G8 + G4 + G6)/4

FIGURE 28.9 Edge-directed interpolation in Ref. [4] is illustrated for estimating the green (*G*) value at pixel 5. The red (*R*) values are used to determine the edge direction. When the missing green pixel is at a blue pixel, the blue values are used to determine the edge direction. (From Gunturk, B. K., et al., *IEEE Signal Process. Mag.*, 22, 44, 2005. With permission.)



Define:

$$d_i^S = S(i+1) - S(i-1)$$

$$e_i^S = \frac{1}{\sqrt{1 + \left(d_i^S\right)^2}}$$

- Interpolate the green at the missing locations:

$$G(i) = \frac{e_{i-1}^R G(i-1) + e_{i+1}^R G(i+1)}{e_{i-1}^R + e_{i+1}^R}$$

- Repeat for three times:
  - Interpolate the red using the ratio rule:

$$R(i) = G(i)\,\frac{e_{i-1}^G \dfrac{R(i-1)}{G(i-1)} + e_{i+1}^G \dfrac{R(i+1)}{G(i+1)}}{e_{i-1}^G + e_{i+1}^G}$$

  - Correct the green to fit the ratio rule:

$$G(i) = R(i)\,\frac{e_{i-1}^R \dfrac{G(i-1)}{R(i-1)} + e_{i+1}^R \dfrac{G(i+1)}{R(i+1)}}{e_{i-1}^R + e_{i+1}^R}$$

FIGURE 28.10 Reference [6] is illustrated for a one-dimensional signal. S is a generic symbol for red (R) and green (G). d_i^S is the gradient for channel S at location i; and e_i^S is the corresponding edge indicator.

an algorithm is presented in Ref. [6]. The algorithm defines edge indicators in several directions as measures of edge likelihood in those directions, and determines a missing pixel intensity as a weighted sum of its neighbors. If the likelihood of an edge crossing in a particular direction is high, the edge indicator returns a small value, which results in less contribution from the neighboring pixel of that direction. The algorithm for one-dimensional signals is illustrated in Figure 28.10. The green channel is interpolated first; the red and blue channels are interpolated from the red/green and blue/green ratios. The color channels are then updated iteratively to obey the color-ratio rule. The extension to two-dimensional images is given in Ref. [6].

A similar algorithm is proposed in Ref. [9], where edge indicators are determined in a 7×7 window for the green and a 5×5 window for the red/blue channels. In this case, the edge indicator function is based on the L_1 norm (absolute difference) as opposed to the L_2 norm of Ref. [6]. In Ref. [11], gradients are filtered adaptively using local means and variances before deciding edge directions. Based on the edge directions, interpolation is done horizontally, vertically, or bidirectionally. This edge direction selection procedure is illustrated in Figure 28.11.

28.3.4 Using Gradients as Correction Terms

Linear interpolation methods have much less computational complexity compared to nonlinear methods, but do not perform as well. When interchannel correlation is included in the linear interpolation, much better performance can be achieved. Recently, the authors of Ref. [12] demonstrated that gradients in one



Calculate horizontal gradients

$$\Delta H(i) = 2\big(G(i+1) - G(i-1)\big) + \big(R(i+2) - R(i-2)\big)$$

Calculate mean and variance of horizontal gradients

$$\mu_i = \frac{1}{3}\sum_{j=i-1}^{i+1} \Delta H(j)$$

$$\sigma_i^2 = \frac{1}{3}\sum_{j=i-1}^{i+1} \big(\Delta H(j) - \mu_j\big)^2$$

Filter the horizontal gradients

$$\Delta H^*(i) = \mu_{i-1} + \frac{\sigma_{i-1}^2}{\sigma_{i-1}^2 + \sigma_{i+1}^2}\big(\mu_{i+1} - \mu_{i-1}\big)$$

Repeat the same procedure to obtain vertical gradients $\Delta V^*(i)$

If $|\Delta H^*(i)| < \alpha |\Delta V^*(i)|$, then interpolate horizontally

Else if $|\Delta V^*(i)| < \alpha |\Delta H^*(i)|$, then interpolate vertically

Else interpolate bidirectionally

FIGURE 28.11 Reference [11] is illustrated. Gradients are calculated, and then adaptively filtered. The interpolation direction at a pixel is selected based on the relative magnitude of filtered gradients. α is a number in the range [0–1].

channel can improve the interpolation performance in another one; and it can be put in a linear interpolation framework: Suppose, the green value at a red location will be estimated by adding the gradient of the red channel to an initial estimate:

$$\hat{G}(i,j) = \hat{G}_{\text{bilinear}}(i,j) + \alpha \Delta_R(i,j), \tag{28.4}$$

where

$\hat{G}_{\text{bilinear}}(i,j)$ is the initial estimate obtained through bilinear interpolation

α is a scale factor that controls the amount of correction

$\Delta_R(i,j)$ is the gradient of the red channel defined as follows:

$$\Delta_R(i,j) = R(i,j) - \frac{1}{4}\sum_{(m,n)=\{(0,-2),\,(0,2),\,(-2,0),\,(2,0)\}} R(i+m, j+n). \tag{28.5}$$

Similarly, the red pixel values at green and blue locations can be estimated using green and blue gradients:

$$\hat{R}(i,j) = \hat{R}_{\text{bilinear}}(i,j) + \beta \Delta_G(i,j) \tag{28.6}$$

and

$$\hat{R}(i,j) = \hat{R}_{\text{bilinear}}(i,j) + \gamma \Delta_B(i,j). \tag{28.7}$$

Specific green and blue gradient definitions are given in Ref. [12]. (Similar equations can be written for the green pixels at blue locations and for the blue pixels at green and red locations.) To determine appropriate values for the gain parameters α, β, γ, Ref. [12] uses training images to find the least squares estimates for the gain parameters. These gain parameters are then approximated by integer multiples of

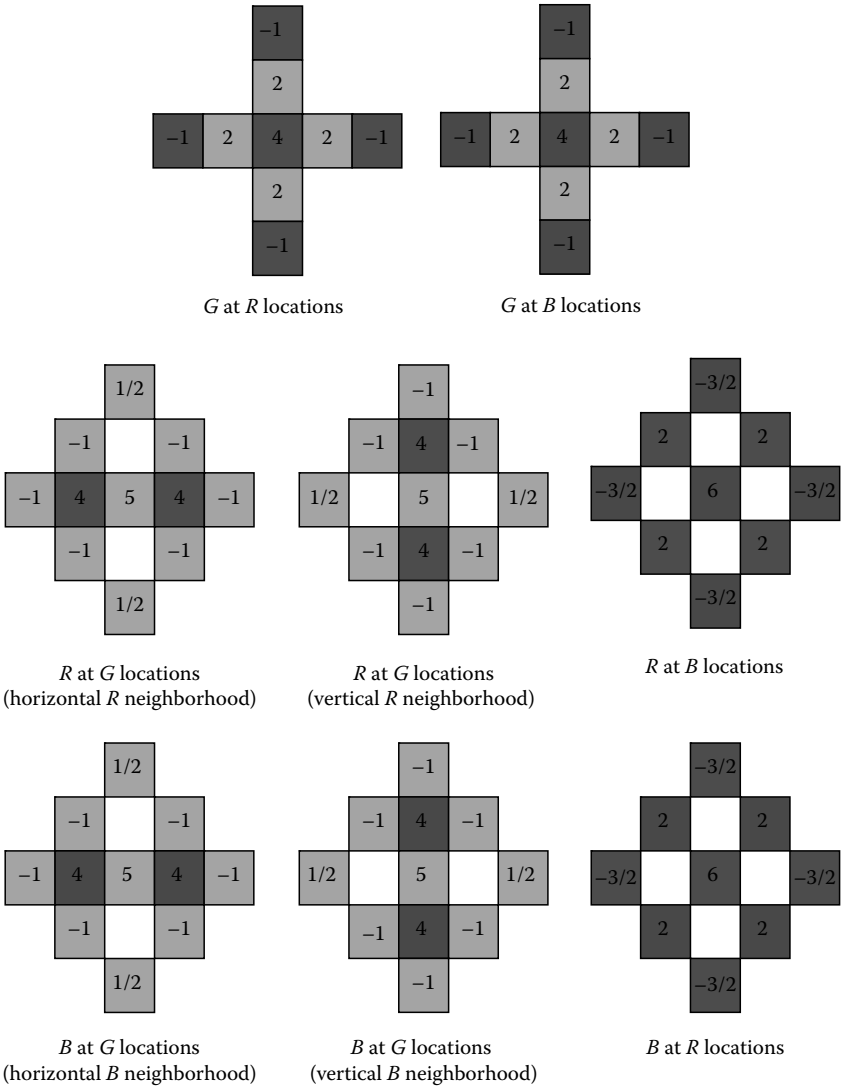


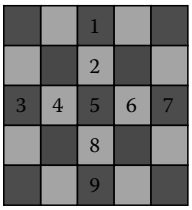
FIGURE 28.12 Filters (From Malvar, H. S., et al., *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Montreal, Canada, 3, 485–488, 2004. With permission.)

small powers of 1/2. The final results are $\alpha = 1/2$, $\beta = 5/8$, and $\gamma = 3/4$. The equivalent linear FIR filter coefficients for each interpolation case are shown in Figure 28.12.

It is possible to incorporate edge-directed interpolation idea to this approach. Reference [13] presents such an algorithm. The gradient terms are added in either horizontal or vertical directions. The direction is chosen adaptively based on edge direction estimates. (Historically earlier than Ref. [12], Ref. [13] does not use training to obtain the optimal coefficients.) Figure 28.13 illustrates this algorithm.

28.3.5 Frequency-Domain Approach

There are two observations that are important for the demosaicking problem. The first is that for natural images, there is a high correlation among the red, green, and blue channels. All three channels are very



$\Delta H = |G4 - G6| + |-R3 + 2 \times R5 - R7|$ //horizontal gradient
$\Delta V = |G2 - G8| + |-R1 + 2 \times R5 - R9|$ //vertical gradient
if $\Delta H > \Delta V$,
    $G5 = (G2 + G8)/2 + (-R1 + 2 \times R5 - R9)/4$
else if $\Delta H < V$,
    $G5 = (G4 + G6)/2 + (-R3 + 2 \times R5 - R7)/4$
else
    $G5 = (G2 + G8 + G4 + G6)/4 + (-R1 - R3 - R7 - R9 + 4 \times R5)/8$

FIGURE 28.13 Graphical illustration. (From Gunturk, B. K., et al., *IEEE Signal Process. Mag.*, 22, 44, 2005. With permission.)

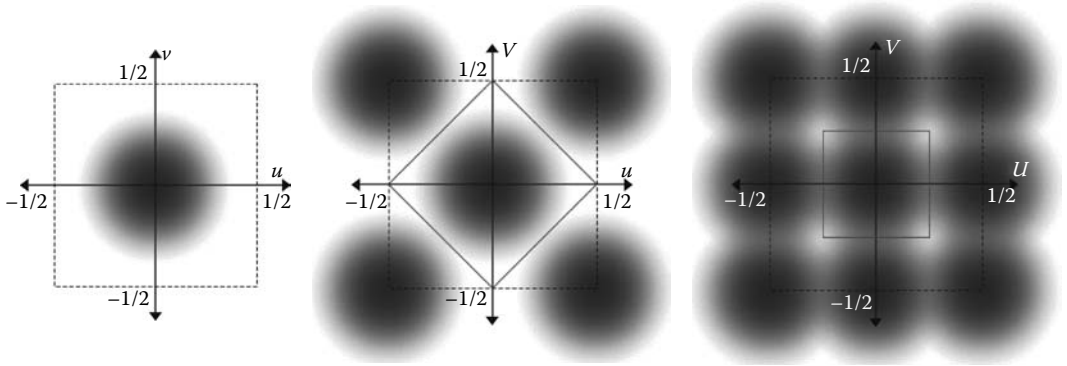


FIGURE 28.14 Frequency domain analysis of CFA sampling. (a) Suppose this is the frequency spectrum of the red, green, and blue channels; (b) frequency spectrum of the sampled green channel; and (c) frequency spectrum of the red/blue channels. Note that while there is no aliasing for the green channel, red and blue channels are aliased. The green channel can be fully recovered with a low-pass filter whose pass-band is outlined in the middle figure. For the red/blue channels, such a low-pass filtering operation cannot eliminate the aliasing.

likely to have the same texture and edge locations. (Because of the similar edge content, we expect this interchannel correlation to be even higher when it is measured between the high-frequency components of the channels [14].) The second observation is that digital cameras often use a CFA in which the luminance (green) channel is sampled at a higher rate than the chrominance (red and blue) channels. Therefore, the green channel is less likely to be aliased, and details are preserved better in the green channel than in the red and blue channels. This is illustrated in Figure 28.14.

28.3.5.1 Alias Canceling Interpolation

In Ref. [15], it is assumed that the high-frequency contents of green and red/blue channels are identical; therefore, high-frequency content of the green image is used to remove aliasing in the red and blue images. First, the red and blue images are interpolated with a rectangular low-pass filter according to the rectangular sampling grid. This fills in the missing values in the grid, but allows aliasing distortions into the red and blue output images. These output images are also the missing high-frequency components needed to produce a sharp image. However, because the green image is sampled at a higher rate, the high-frequency information can be taken from the green image to improve an initial interpolation of the red and blue images. A horizontal high-pass filter and a vertical high-pass filter are applied to the green image. This provides the high-frequency information that the low sampling rate of the red and blue images cannot preserve. Aliasing occurs when high-frequency components are shifted into the low-frequency portion of the spectrum, so if the outputs of the high-pass filters are modulated into the low-frequency regions, an estimate of the aliasing in the red and blue images can be found. This estimate is used to reduce the aliasing in the red and blue images, as illustrated in Figure 28.15. This method relies

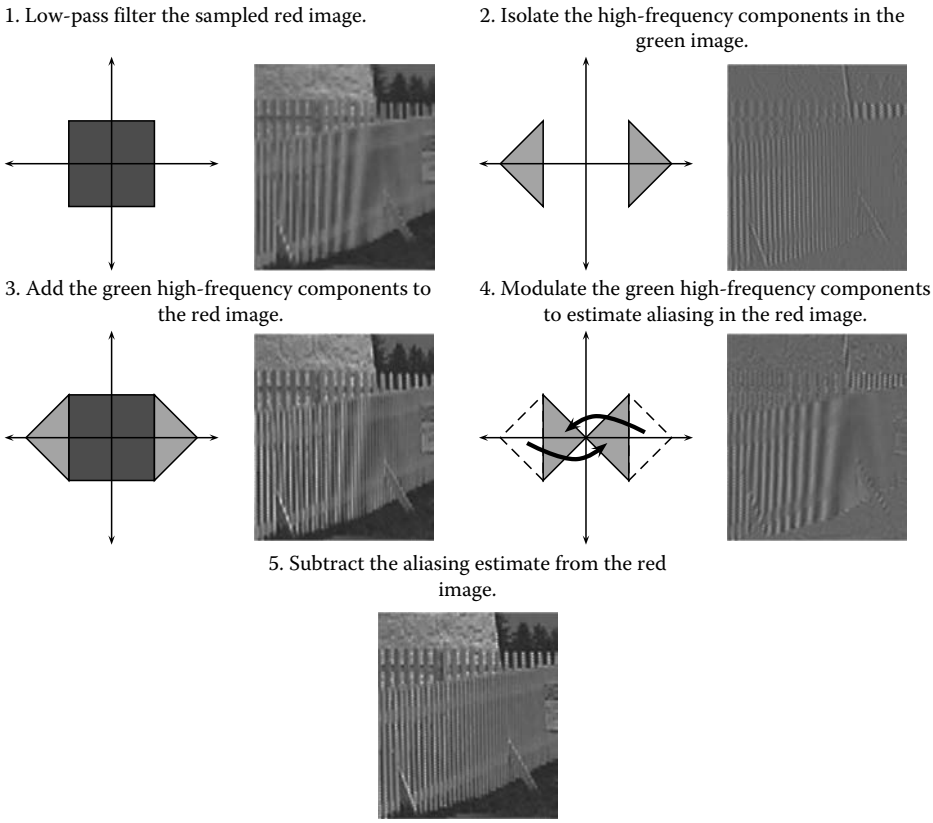


FIGURE 28.15 An illustration of Ref. [15] by John Glotzbach. High-frequency information from the green image is modulated and used to cancel aliasing in the red image. (From Gunturk, B. K., et al., *IEEE Signal Process. Mag.*, 22, 44, 2005. With permission.)

on the assumption that the high-frequency information in the red, green, and blue images is identical. If this assumption does not hold, the addition of the green information into the red and blue images can add unwanted distortions. This method also makes the assumption that the input image is band-limited within the diamond-shaped Nyquist region of the green quincunx sampling grid (the region outlined in Figure 28.14). When this assumption fails, the aliasing artifacts are enhanced instead of reduced because the green image also contains aliasing. This system is composed entirely of linear filters, making it efficient to implement.

28.3.5.2 Frequency-Domain Filtering

In Ref. [1], the CFA-sampled image is reorganized into newly defined luminance and chrominance components. Analyzing the CFA sampling in Fourier domain, filters that recover these components are designed. The derivation is as follows. Using Equation 28.2, the mosaicked image can be written as

$$\begin{aligned} z(i,j) &= z_R(i,j) + z_G(i,j) + z_B(i,j) \\ &= \tfrac{1}{4}(R(i,j) + 2G(i,j) + B(i,j)) \\ &\quad + \tfrac{1}{4}(B(i,j) - R(i,j))(\cos \pi i - \cos \pi j) \\ &\quad + \tfrac{1}{4}(-R(i,j) + 2G(i,j) - B(i,j)) \cos \pi i \cos \pi j. \end{aligned} \tag{28.8}$$

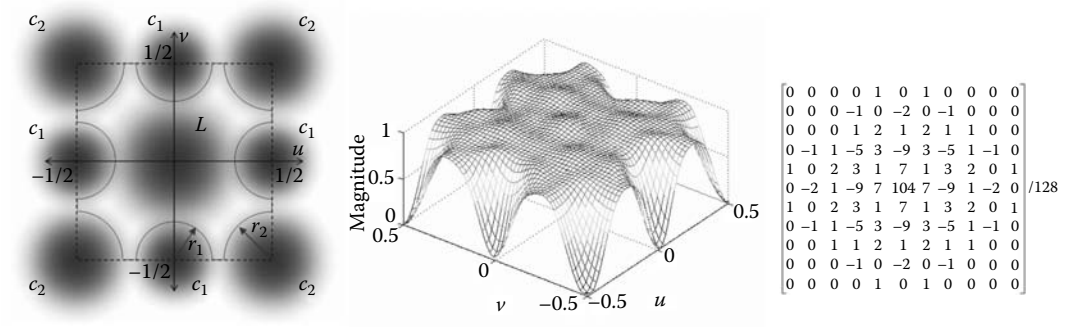


FIGURE 28.16 Filter design in Ref. [1]. Left: Luminance and chrominance components when the Fourier transform of the mosaicked image is taken. The red line outlines the passband of the filter. The radii r_1 and r_2 are design parameters, whose values are determined empirically. Middle and Right: Fourier domain and spatial domain representations of the filter used in Ref. [1].

In this equation, $L(i,j) = (1/4)[R(i,j) + 2G(i,j) + B(i,j)]$ is defined as the luminance component, and $C_1(i,j) = (1/4)[B(i,j) - R(i,j)]$ and $C_2(i,j) = (1/4)[-R(i,j) + 2G(i,j) - B(i,j)]$ as the chrominance components. With these definitions, Equation 28.8 becomes

$$z(i,j) = L(i,j) + C_1(i,j)\ (\cos \pi i - \cos \pi j) + C_2(i,j) \cos \pi : \cos \pi j, \tag{28.9}$$

where

$$\begin{bmatrix} L \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ -1/4 & 0 & 1/4 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{28.10}$$

When the Fourier transform of Equation 28.9 is taken, the luminance component stays in the baseband, the C_1 component will be modulated at frequencies (0.5, 0) and (0, 0.5), and the C_2 component will be modulated at frequency (0.5, 0.5). This is illustrated in Figure 28.16. By designing appropriate filters, the luminance and chrominance components can be recovered. Using the inverse of the matrix given in Equation 28.10, the red/green/blue values are obtained. Figure 28.16 also shows the filter to recover the luminance component.

An extension of this approach is given in Ref. [16], where it is noticed that the C_1 component suffers from spectral overlap, and the overlap often occurs in only one of the (horizontal/vertical) directions. Therefore, a weighted sum of the horizontally and vertically filtered C_1 components is taken, where the weight is less for the one with least crosstalk.

Reference [17] gives another extension of Ref. [1]. This time adaptive filtering is applied on the luminance component. The luminance values at green locations are estimated using a filter similar to the one in Ref. [1]; while the values at red/blue locations are estimated as a weighted sum of neighboring luminance values, where the weights are selected according to the horizontal and vertical gradients (edge indicators).

28.3.6 Homogeneity-Directed Interpolation

Instead of choosing the interpolation direction based on edge indicators, Ref. [8] uses local homogeneity as an indicator. The homogeneity-directed interpolation imposes the similarity of the luminance and chrominance values within small neighborhoods, and it leads to very good perceptual results. The

underlying idea is to interpolate color channel horizontally and vertically, and to pick up either the horizontally interpolated pixel values or the vertically interpolated pixel values at every pixel location based on the homogeneity.

Reference [8] defines the homogeneity as follows. Suppose that $R(i,j)$, $G(i,j)$, $B(i,j)$ are the values in the RGB space, and $L(i,j)$, $a(i,j)$, $b(i,j)$ are the corresponding luminance and chrominance values in the CIELab space. Three neighbor sets of (i,j) are defined. The first one is the set of pixel locations that are close in space:

$$N_D(i,j) = \left\{(m,n) \,|\, \left[(m-i)^2 + (n-j)^2\right]^{1/2} \leq \varepsilon_D\right\}. \tag{28.11}$$

The other two neighbor sets are the sets of pixel locations with similar luminance and chrominance values:

$$N_L(i,j) = \{(m,n) \mid |L(m,n) - L(i,j)| \leq \varepsilon_L\} \tag{28.12}$$

and

$$N_C(i,j) = \left\{(m,n) \mid \left([a(m,n) - a(i,j)]^2 + [b(m,n) - b(i,j)]^2\right)^{1/2} \leq \varepsilon_C\right\}. \tag{28.13}$$

Then the homogeneity is defined as

$$H(i,j) = \frac{\text{size}[N_D(i,j) \cap N_L(i,j) \cap N_C(i,j)]}{\text{size}[N_D(i,j)]}. \tag{28.14}$$

Referring to Figure 28.17, the algorithm works as follows. The RGB data is first interpolated horizontally and vertically. (The green channel is interpolated using red and blue data as correction terms, as in Ref. [13]. The red and blue channels are interpolated from the interpolated red–green difference and blue–green difference as shown in Figure 28.7.) The interpolated images are then converted to the CIELab space. The homogeneity maps for horizontally and vertically interpolated images are found. The homogeneity maps are smoothed with a 3×3 averaging filter. At each pixel, either the horizontally or the vertically interpolated color values are taken depending on which has the largest homogeneity.

In Ref. [8], the neighborhood parameter ε_D is kept constant, while ε_L and ε_C are determined adaptively at each pixel such that they reflect typical variations among pixels of the same object. This

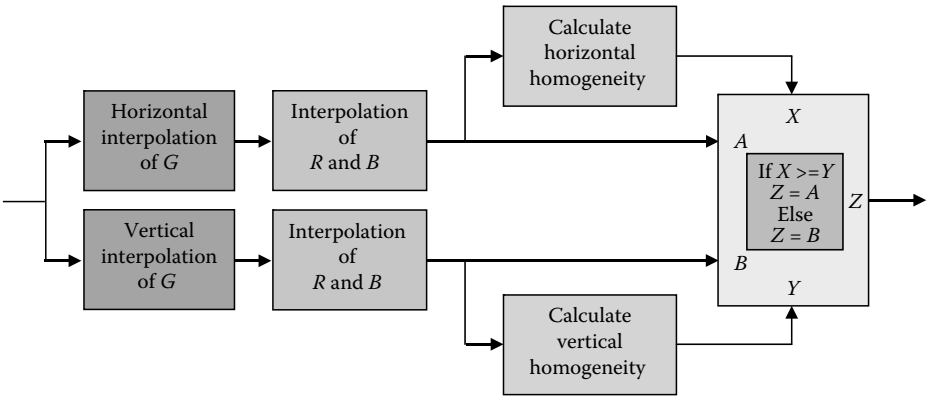


FIGURE 28.17 Block diagram of the homogeneity-directed interpolation (From Hirakawa, K. and Parks, T. W., *IEEE Trans. Image Process.*, 14, 360, 2005. With permission.)

is achieved by analyzing the four nearest neighbors of the pixel in question. ε_L at location (i,j) is calculated as follows:

$$\begin{aligned}\varepsilon_{LH}(i,j) &= \max\{|L(i-1,j) - L(i,j)|,\ |L(i+1,j) - L(i,j)|\},\\ \varepsilon_{LV}(i,j) &= \max\{|L(i,j-1) - L(i,j)|,\ |L(i,j+1) - L(i,j)|\},\\ \varepsilon_{L}(i,j) &= \min\{\varepsilon_{LH}(i,j), \varepsilon_{LV}(i,j)\},\end{aligned} \tag{28.15}$$

where the first two equations give the maximum variations in horizontal and vertical directions, and the last equation picks up the minimum of these, by which the maximum variation within the object is determined. ε_C is determined similarly.

28.3.7 Projections onto Convex Sets Approach

The algorithm presented in Ref. [15] proposes to decompose the green channel into its frequency components and then add the high-frequency components of the green channel to the low-pass filtered red and blue channels. This is based on the observation that the high-frequency components of the red, blue, and green channels are similar and the fact that the green channel is less likely to be aliased. One problem with this approach is that the high-frequency components of the red, green, and blue channels may not be identical. Therefore, replacement of the high-frequency components of the red and blue channels with those of the green channel may not work well. Reference [14] proposes an algorithm that ensures data consistency at the cost of higher computational complexity. The algorithm defines two constraint sets, one ensuring that the restored images are consistent with the measured data and the other imposing similar high-frequency components in the color channels, and reconstructs the color channels using the projections onto convex sets (POCS) technique.

28.3.7.1 Constraint Sets

The first constraint set guarantees that the restored color channels are consistent with (are identical to) the color samples captured by the digital camera.

The second constraint set is a result of the high interchannel correlation. Reference [14] shows that color channels have very similar detail (high-frequency) subbands. This information would not be enough to define constraint sets if all channels lost the same amount of information in sampling. However, the red and blue channels lose more information (details) than the green channel when captured with a color filter array. Therefore, it is reasonable to define constraint sets on the red and blue channels that force their high-frequency components to be similar to the high-frequency components of the green channel. The similarity is imposed such that the detail coefficients are within a fixed proximity.

28.3.7.2 Alternating Projections Algorithm

The block diagram of the POCS algorithm is given in Figure 28.18. The projection onto the observation constraint set inserts the observed data into their corresponding locations in the current image. This is illustrated in Figure 28.19. The projection operation onto the detail constraint set is illustrated in

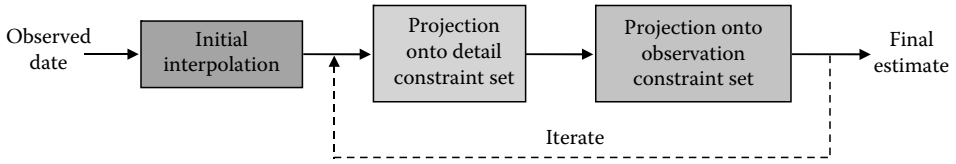


FIGURE 28.18 Block diagram of the algorithm given in Ref. [14]. Red/green/blue channels are interpolated first. The red/blue channels are then updated by iteratively projecting on the detail and observation constraint sets.

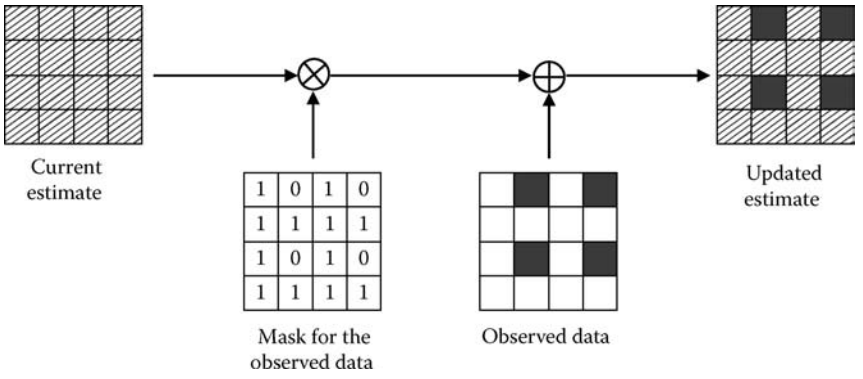


FIGURE 28.19 Projection onto observation constraint set (From Gunturk, B. K., Altunbasak, Y. and Mersereau, R. M., *IEEE Trans. Image Process*., 11, 997, 2002. With permission.)

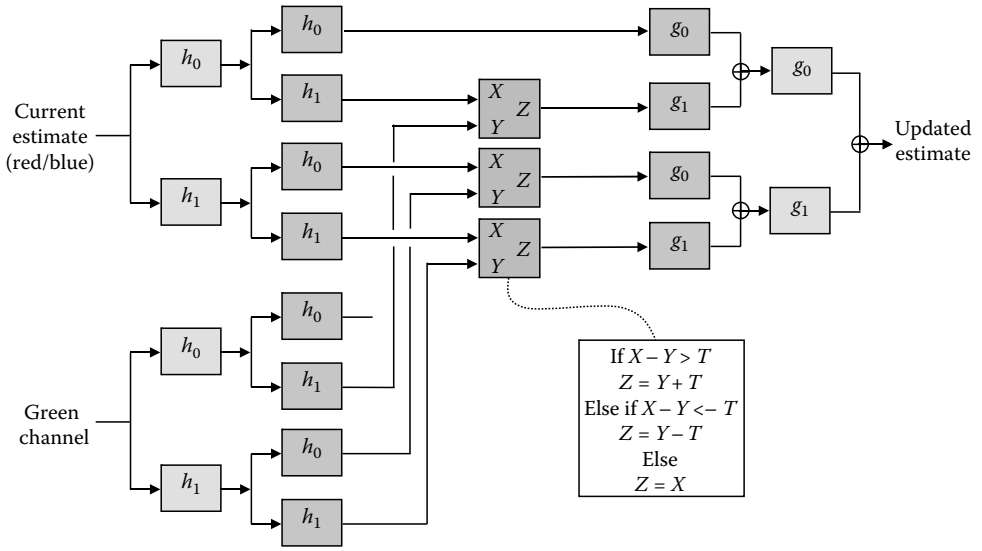


FIGURE 28.20 Projection onto detail constraint set [14]. h_0 and h_1 are the low-pass and high-pass analysis filters. g_0 and g_1 are the corresponding synthesis filters.

Figure 28.20. By alternately applying these two projections onto the initial red and blue channel estimates, these channels are enhanced.

28.3.8 Spectral Response Modeling

The last group of methods models the image formation process and formulates the demosaicking problem based on this model. To understand this approach, we first need to understand the imaging process. The image acquisition process is usually modeled as a linear process between the light radiance arriving at the camera and the pixel intensities produced by the sensors. An image sensor has a specific spectral response $L_S(\lambda)$, which is a function of the spectral wavelength λ, and a spatial response $h_S(x, y)$, which results from optical blur and the spatial integration at each sensor site. (Typical spectral sensor sensitivities are illustrated in Figure 28.3. The λ space is typically modeled as a 7–11 dimensional space.) The imaging process can be formulated as

$$S(x,y) = \iiint L_S(\lambda) h_S(x-u, y-v) r(u,v,\lambda) \mathrm{d}u \; \mathrm{d}v \; \mathrm{d}\lambda + N_S(x,y), \tag{28.16}$$

where

$S(x, y)$ is the pixel value at spatial location (x, y)
$r(x, y, \lambda)$ is the incident radiance
$N_S(x, y)$ is the additive noise that is a result of thermal/quantum effects and quantization

There are couple of assumptions in this formulation: (1) the input–output relation is assumed to be linear; (2) the spatial blur $h_S(x, y)$ is assumed to be space-invariant and independent of wavelength; and (3) only the additive noise is considered. These assumptions are reasonable for practical purposes.

Since we are dealing with digital data, we need to have the discrete version of Equation 28.16

$$S(i,j) = \sum_{l} \sum_{m,n} L_S(l) h_S(i-m, j-n) r(m,n,l) + N_S(i,j). \tag{28.17}$$

The color filters sample the signal $S(i, j)$ to produce a mosaicked data $z(i, j)$ as given in Equation 28.21. Therefore, the observation model is a linear system, which can be written in the compact form

$$\mathbf{z} = \mathbf{H}\mathbf{r} + \mathbf{N}, \tag{28.18}$$

where

r, **z**, and **N** are the stacked forms of $r(m, n, l)$, $z(i, j)$, and CFA-sampled $N_S(i, j)$, respectively
H is the matrix that includes the combined effects of optical blur, sensor blur, spectral response, and CFA sampling

In Refs. [18–20] the minimum mean square error (MMSE) solution of Equation 28.18 is given:

$$\mathbf{r}^{\mathrm{MMSE}} = E\left[\mathbf{r}\mathbf{z}^{\mathrm{T}}\right]\left(E\left[\mathbf{z}\mathbf{z}^{\mathrm{T}}\right]\right)^{-1}\mathbf{z}, \tag{28.19}$$

where $E[\cdot]$ is the expectation operation. In Ref. [20], the point spread function (PSF) is taken as an impulse function; and **r** is represented as a weighted sum of spectral basis functions to reduce the dimensionality of the problem. (Later, Ref. [21] extended Ref. [20] to include the PSF in the reconstruction.) In Ref. [19], adaptive reconstruction and ways to reduce computational complexity are discussed. Reference [18] constructs a FIR filter based on a wide sense stationary assumption.

28.4 Demosaicking of Video and Super-Resolution Reconstruction

When there are multiple images, it is possible to estimate the missing samples than in the case of a single image. Even if a color sample does not exist in an image as a result of Bayer sampling, that particular sample could have been captured in another frame (due to motion). By warping all captured samples onto the common frame to be demosaicked, a better estimate of a missing sample can be obtained.

Figure 28.21 illustrates this multiframe interpolation idea. In the figure, red channels of three Bayer-sampled images are shown. The grid locations with ''triangles,'' ''circles,'' and ''squares'' show the red samples in these images. We would like to estimate the missing samples in the middle image. The other two images are warped onto the middle image. The estimation problem now becomes an interpolation problem from a set of nonuniformly sampled data. Taking the weighted of these samples, where the weights are inversely proportional with the distance to the pixel in question, the missing sample can be estimated.

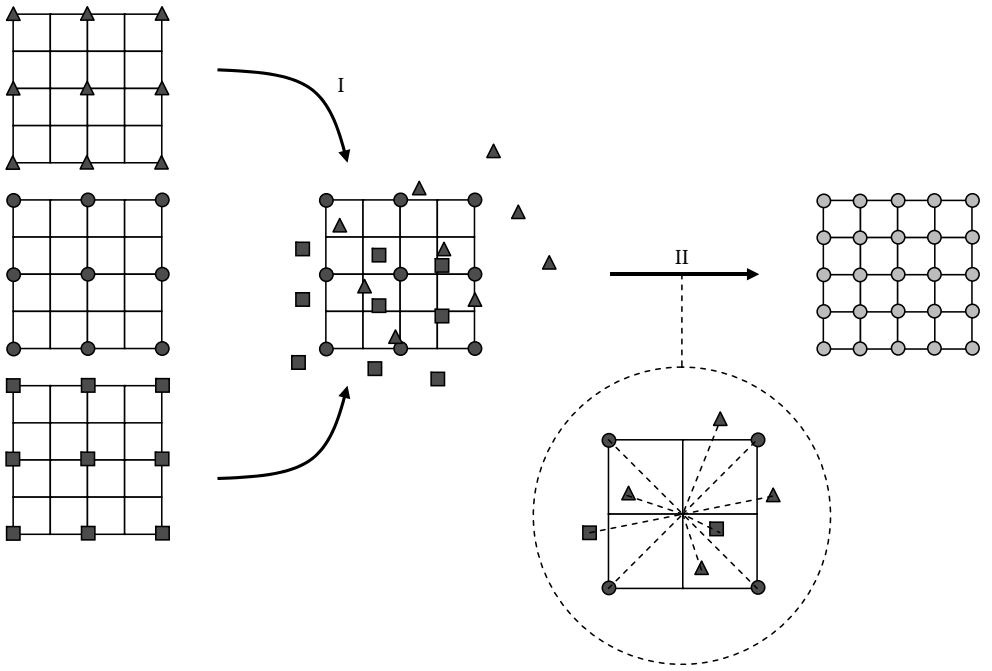


FIGURE 28.21 I: Input images are warped onto reference frame. II: Weighted average of samples are taken to find values on the reference sampling grid.

This idea can be combined with the other demosaicking ideas, such as constant-hue-based interpolation, to formulate a solution. When these multiple frame are available, we can now talk about a spatiotemporal neighborhood instead of a spatial neighborhood of a pixel.

It is also possible to obtain subpixel resolution by combining multiple images. This is known as super-resolution reconstruction, and it was recently applied to CFA sampled color images. The imaging model for super-resolution reconstruction starts with a high-resolution image: Let x_S be a color channel of a high-resolution image, where a channel can be red (x_R), green (x_G), or blue (x_B). The ith observation, $S^{(i)}$, is obtained from this high-resolution image through spatial warping, blurring, and downsampling operations:

$$S^{(i)} = DCW^{(i)}x_S, \quad \text{for } S = R, G, B, \quad \text{and} \quad i = 1, 2, \ldots, K, \tag{28.20}$$

where

K is the number of input images
$W^{(i)}$ is the warping operation (to account for the relative motion between observations)
C is the convolution operation (to account for the point spread function of the camera)
D is the downsampling operation (to account for the spatial sampling of the sensor)

The full-color image $(R^{(i)}, G^{(i)}, B^{(i)})$ is then converted to a mosaicked observation $z^{(i)}$ according to a CFA sampling pattern:

$$z^{(i)} = \sum_{S=R,\,G,\,B} M_S y_S^{(i)} \tag{28.21}$$

as we explained earlier. Then, the super-resolution problem becomes estimation of the high-resolution image $x_S, S = R, G, B$, from low-resolution mosaicked data $z^{(i)}$, $i = 1, 2, \ldots, K$. A typical flowchart of super-resolution reconstruction is illustrated in Figure 28.22.

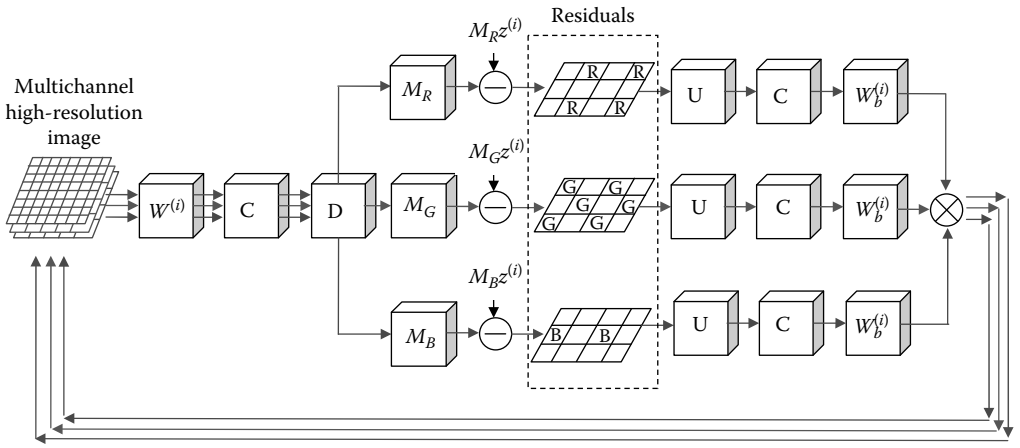


FIGURE 28.22 Typical super-resolution reconstruction algorithm is illustrated. The algorithm starts with a high-resolution image estimate. Simulated observations are obtained by forward imaging operations, including the CFA sampling. The residuals are computed on Bayer pattern samples for each channel, and then back-projected. The notation in the figure is as follows. $W^{(i)}$: Spatial warping onto ith observation, C: Convolution with the PSF, D: Downsampling by the resolution enhancement factor, U: Upsampling by zero insertion, $W_b^{(i)}$: Back-warping to the reference grid.

In Ref. [22], data fidelity and regularization terms are combined to produce high-resolution images. The data fidelity term is based on a cost function that consists of the the sum of residual differences between actual observations and high-resolution image projected onto observations (simulated observations). Regularization functions are added to this cost function to eliminate color artifacts and preserve edge structures. These additional constraints are defined as luminance, chrominance, and orientation regularization [22]. A similar algorithm is presented in Ref. [23]. Reference [24] extends the POCS algorithm of Ref. [14] to multiple frames, where observation constraint set is obtained through bilateral filtering of samples.

28.5 Related Research Problems

In this chapter, we have covered some of the basic demosaicking approaches. There are various others, such as the Bayesian estimation based [25,26] and the neural network based [27]. There are also research problems that are related to the CFA sampling, some of which are listed below:

Denoising: Denoising becomes critical in a digital camera pipeline, especially for images captured in low-light conditions and with high ISO speed. Although standard denoising algorithms can be combined with standard demosaicking algorithms, denoising and demosaicking can be done jointly. Reference [28] proposes such a joint technique based on the total least squares denoising method. The technique reports superior image quality compared to sequential applications of standard denoising and demosaicking algorithms.

Compression: Compression is the last process in a digital camera pipeline. Typically, demosaicking is done on raw CFA data to obtain full-color images, which are later compressed with a compression algorithm, such as the JPEG for still images or the MPEG for videos. The image/video compression algorithms generally decompose images into luminance and chrominance channels and then down-samples chrominance channels to achieve higher compression rates. Because of this downsampling, it seems redundant to do demosaicking before compression. Compression could be done on CFA data; and demosaicking could be added to the end of the decoding processing. Recent studies [29,30] show that, with this alternative processing chain, higher image quality can be achieved at low compression rates. Also, the processing cost is reduced at the camera side.

Camera identification and forgery detection: Due to the demosaicking process, pixels in a digital image are correlated; since different cameras use different demosaicking algorithms, the correlation among the pixels can be used to identify the camera or to detect image forgeries, such as cropping and pasting a region from one image to another [31,32].

Optimal spectral sensitivity functions: Selection of the spectral sensitivity functions is an important part of the digital camera design. The Bayer RGB CFA is known for its superior color reproduction; while the CMYG CFA has better signal-to-noise ratio performance in low-light conditions. Methods for designing optimal spectral sensitivities for color reproduction have been studied earlier [33]. Recently, importance of spectral sensitivities on demosaicking [34] and design of optimal sensitivities for both color and spatial reproduction [35] are discussed.

28.6 Evaluation of Demosaicking Algorithms

Since raw CFA data is not available for most digital cameras, demosaicking methods have been compared based on simulations: Full color images are sampled according to a CFA pattern; the original and the restored images are later compared quantitatively. (This neglects many of the processes in the camera pipeline, most importantly, optical low-pass filtering. Therefore, simulation results may be misleading;

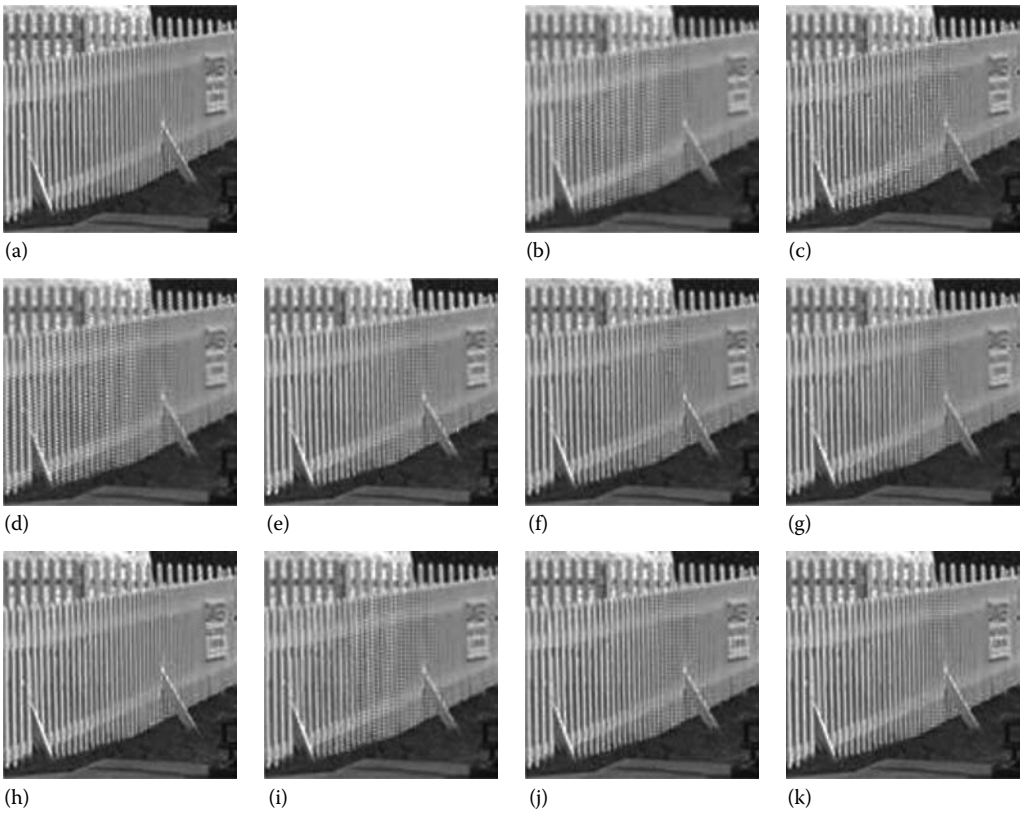


FIGURE 28.23 Result images for example lighthouse image. (a) Original image, (b) bilinear interpolation, (c) edge-directed interpolation in Ref. [10], (d) constant hue-based interpolation in Ref. [3], (e) weighted sum in Ref. [6], (f) second-order gradients as correction terms in Ref. [13], (g) Bayesian approach in Ref. [25], (h) homogeneity-directed in Ref. [8], (i) pattern matching (Chang) in Ref. [39], (j) Alias cancellation in Ref. [15], and (k) POCS in Ref. [14].

the demosaicking algorithms should be tested on raw data; different cameras may have different demosaicking algorithms working best for them.) Two commonly used quantitative measures for evaluating demosaicking algorithms are mean square error and peak signal-to-noise ratio. Euclidean distances in the perceptually uniform CIELab and CIELuv spaces and the s-CIELab [36] are better measures considering the human visual perception.

For the demosaicking algorithms, the Kodak color image database [37] has become a standard test image set. Many algorithms evaluate their performances using these images, enabling comparison among different algorithms. Figure 28.23 provides a visual comparison of several demosaicking algorithms available in 2005 [38]. Recent papers present improvement over these methods.

In addition to the restoration performance, computational complexity is also an important factor for demosaicking algorithms due to the limited resources of a digital camera. The trade-off between the image quality and computational time should be considered in designing the camera pipeline.

28.7 Conclusions and Future Directions

As the sensor technology and processing power of digital cameras advance, the image quality of digital still and video cameras will continue to improve. Eventually, the image quality of consumer level digital cameras will catch and exceed the quality of film cameras. Digital video cameras will be able to produce images with the quality of digital still cameras.

Although the manufacturers are able to fit more and more pixels in a fixed size chip, higher pixel count does always not correspond to higher image quality. As the dimensions of the photosensitive region of a pixel decrease, the dynamic range and noise performance of the sensor decreases. The solution is therefore to produce larger size sensor chips and to increase the fill ratio (photosensitive region area/pixel area).

There is also research to develop alternative sensor technologies. For example, a recent technology exploits the absorption characteristics of silicon to eliminate CFA and increase resolution. The blue portion of the light is absorbed at the surface of a silicon, while the red portion penetrates deeper. By putting detectors at various depths, color information can be extracted without the need of a CFA. However, for such a sensor, the color components are convoluted, the color reproduction and noise performance could be problematic. Another approach is to place micrograting above pixels. These micrograting diffract light according their spectral content, and detectors capture different spectral information.

Although these technologies look promising, single-chip image capture, and therefore, demosaicking seem to remain essential for some time. Soon HDTVs will become widespread, and higher image quality from digital video cameras will be expected. Some of the demosaicking algorithms that we explained in this article cannot be included in a camera due to the limited resources. However, the computational power of cameras will increase, and advanced image processing algorithms will be utilized in digital cameras. The research on the inter-channel correlation can also help in modeling and restoration of multi-frame and hyper-spectral data.

Acknowledgments

This chapter is an extension of Ref. [38]. We would like to acknowledge the vital contributions of John Glotzbach, Ronald W. Schafer, and Russell M. Mersereau. This work is supported in part by the National Science Foundation under Grant 0528785.

References

1. D. Alleysson, S. Susstrunk, and J. Herault, Linear demosaicing inspired by the human visual system, *IEEE Trans. Image Process.*, 14(4), 439–449, April 2005.
2. D. R. Cok, Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal, U.S. Patent 4,642,678, February 1986.

3. J. E. Adams, Interactions between color plane interpolation and other image processing functions in electronic photography, *Proc. SPIE Technology for Electronic Imaging Systems*, 2416, 144–151, February 1995.
4. C. A. Laroche and M. A. Prescott, Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients, U.S. Patent 5,373,322, December 1994.
5. W. T. Freeman, Method and apparatus for reconstructing missing color samples, U.S. Patent 4,774,565, 1988.
6. R. Kimmel, Demosaicing: Image reconstruction from ccd samples, *IEEE Trans. Image Process.*, (9), 1221–1228, September 1999.
7. X. Li, Demosaicing by successive approximation, *IEEE Trans. Image Process.*, 14(3), 370–379, March 2005.
8. K. Hirakawa and T. W. Parks, Adaptive homogeneity-directed demosaicing algorithm, *IEEE Trans. Image Process.*, 14(3), 360–369, March 2005.
9. W.-M. Lu and Y.-P. Tan, Color filter array demosaicking: New method and performance measures, *IEEE Trans. Image Process.*, 12(10), 1194–1210, October 2003.
10. R. H. Hibbard, Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients, U.S. Patent 5,382,976, January 1995.
11. C.-Y. Tsai and K.-Tai Song, Heterogeneity-projection hard-decision color interpolation using spectral–spatial correlation, *IEEE Trans. Image Process.*, 16(1), 78–91, January 2007.
12. H. S. Malvar, L.-W. He, and R. Cutler, High-quality linear interpolation for demosaicing of color images, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Montreal, Canada, 2004, vol. 3, pp. 485–488.
13. J. E. Adams and J. F. Hamilton, Design of practical color filter array interpolation algorithms for digital cameras, *Proc. SPIE Real Time Imaging II*, 3028, 117–125, February 1997.
14. B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, Color plane interpolation using alternating projections, *IEEE Trans. Image Process.*, 11(9), 997–1013, September 2002.
15. J. W. Glotzbach, R. W. Schafer, and K. Illgner, A method of color filter array interpolation with alias cancellation properties, in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, 2001, vol. 1, pp. 141–144.
16. E. Dubois, Frequency-domain methods for demosaicking of bayer-sampled color images, *IEEE Trans. Signal Process. Lett.*, 12(12), 847–850, December 2005.
17. N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, Adaptive filtering for color filter array demosaicking, *IEEE Trans. Image Process.*, 16(10), 2515–2525, October 2007.
18. D. Taubman, Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior, in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, 2000, vol. 3, pp. 801–804.
19. H. J. Trussell and R. E. Hartwig, Mathematics for demosaicking, *IEEE Trans. Image Process.*, 3(11), 485–492, April 2002.
20. D. H. Brainard, Bayesian method for reconstructing color images from trichromatic samples, *Proc. IS & T 47th Annual Meeting*, Rochester, New York, pp. 375–380, 1994.
21. P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, Perceptual assessment of demosaicing algorithm performance, *Proc. IEEE*, 90(1), 123–132, January 2002.
22. S. Farsiu, M. Elad, and P. Milanfar, Multiframe demosaicing and super-resolution of color images, *IEEE Trans. Image Process.*, 15(1), 141–159, January 2006.
23. T. Gotoh and M. Okutomi, Direct super-resolution and registration using raw CFA images, in *Proc. IEEE Computer Vision and Pattern Recognition*, Washington, D.C., July 2004, vol. 2, pp. 600–607.
24. M. Gevrekci, B. K. Gunturk, and Y. Altubasak, POCS-based restoration of bayer-sampled image sequences, in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Honolulu, HI, April 2007, vol. 1, pp. 753–756.
25. J. Mukherjee, R. Parthasarathi, and S. Goyal, Markov random field processing for color demosaicing, *Pattern Recognit. Lett.*, 22(3–4), 339–351, March 2001.

26. Y. Hel-Or and D. Keren, Image demosaicing method utilizing directional smoothing, U.S. Patent 6,404,918, July 2002.
27. J. Go, K. Sohn, and C. Lee, Interpolation using neural networks for digital still cameras, *IEEE Trans. Consumer Elec.*, 4(3), 610–616, August 2000.
28. K. Hirakawa and T. W. Parks, Joint demosaicing and denoising, *IEEE Trans. Image Process.*, 15(8), 2146–2157, August 2006.
29. N.-X. Lian, L. Chang, V. Zagorodnov, and Y.-P. Tan, Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling, *IEEE Trans. Image Process.*, 15(11), 3261–3278, November 2006.
30. C. C. Koh, New efficient methods of image compression in digital cameras with color filter array, *IEEE Trans. Consumer Elec.*, 49(4), 1448–1456, November 2003.
31. S. Bayram, H. Sencar, N. Memon, and I. Avcibas, Source camera identification based on CFA interpolation, in *Proc. IEEE Int. Conf. Image Processing*, Genoa, Italy, September 2005, vol. 3, pp. 69–72.
32. Y. Long and Y. Huang, Image based source camera identification using demosaicking, in *Proc. IEEE 8th Workshop on Multimedia Signal Processing*, Victoria, Canada, October 2006, pp. 419–424.
33. Mathematical methods for the design of color scanning filters, *IEEE Trans. Image Process.*, 6(2), 312–320, February 1997.
34. D. Alleysson, S. Susstrunk, and J. Marguier, Influence of spectral sensitivity functions on demosaicing, in *Proc. 11th IS&T Color Imaging Conference*, Scottsdale, AZ, November 2003, pp. 351–357.
35. M. Parmar and S. J. Reeves, Selection of optimal spectral sensitivity functions for color filter arrays, in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, GA, October 2006, pp. 1005–1008.
36. X. Zhang and B. A. Wandell, A spatial extension of cielab for digital color image reproduction, *J. Soc. Inf. Display*, 5(1), 61–63, March 1997.
37. Eastman Kodak Company, Kodak Photo CD PCD0992. Available online at http://r0k.us/graphics/kodak/
38. B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, Demosaicking: Color filter array interpolation, *IEEE Signal Process. Mag.*, 22, 44–54, 2005.
39. E. Chang, S. Cheung, and D. Y. Pan, Color filter array recovery using a threshold-based variable number of gradients, *Proc. SPIE Image Sensor Design and Characterization*, 3650, 36–43, January 1999.

Index

A

Active filters
  cascaded realization
    CAD methods, 4-24
    operational amplifier, 4-26
    polynomial coefficient, 4-23
    Q-factor, 4-26–4-28
    stages, 4-25–4-26
  cascade realization, 1-23–1-24
  first-order *RC* network, 1-25
  inductorless filter, 1-23
  Thévenin equivalent circuit, 1-24
  transfer function, 1-24–1-25
Active *RC* prototype, *see* Tow–Thomas biquad
Adaptive 3-D wavelet packet transform, 27-27
Åkerberg–Mossberg biquad, 13-24–13-25
Alias-free 2-D decimation
  antialiasing filter, 25-5
  frequency cell, 25-6
  periodicity matrix, 25-6–25-7
  quincunx downsampling, 25-6
Allpass filters, 1-19
  block diagram, 19-10
  cascade filter, 19-9–19-10
  magnitude response, 19-8
  phase response, 19-9
  pole-zero reciprocal pair, 19-8–19-9
All-pole filter, *see* Chebyshev filters
Analog filter characteristics
  allpass filters, 1-19
  amplitude and delay distortion, 1-16–1-17
  applications, 1-1–1-2
  components, classification, 1-2
  design process
    approximation step, 1-21–1-22
    implementation, 1-23
    realization step and imperfections, 1-22
  discrete- and continuous-time signal, 1-2
  ideal and practical filters, 1-15–1-16
  Laplace transform, 1-3–1-4
  minimum-phase filters
    amplitude response, 1-18
    phase response, 1-18–1-19
    zero-pole plots, 1-17–1-18
  nonminimum-phase transfer functions, 1-20–1-21
  realization
    active filters, 1-23–1-25
    biquads, 1-25–1-26
    passive *RLC* filters, 1-23
  section types
    allpass section, 1-28–1-29
    bandpass section, 1-27–1-28
    high-pass section, 1-27
    low-pass section, 1-26
    notch section, 1-28
  time-domain response
    impulse and step responses, 1-8
    inversion formula, 1-6–1-7
    overshoot, delay time, and rise time, 1-8–1-10
    partial-fraction, 1-7
  transfer function
    convolution integral, 1-4–1-5
    definition, 1-4
    methods, network analysis, 1-5–1-6
    stable and causal network, 1-6
Analog prototype filters
  bandpass filter
    bandpass-to-low-pass conversion, 19-30
    frequency response, 19-31
    LPF to BPF transformation, 19-29–19-30
    magnitude response specification, 19-28–19-29
    specification, 19-30–19-31
    transfer function, 19-31
  bandstop filter, 19-32
  design steps, 19-20
  high-pass filter
    analog and digital frequency, 19-27
    frequency response, 19-28–19-29
    inverse Chebyshev filter transfer function, 19-28
    LPF to HPF transformation, 19-27
    specification, 19-27–19-28
  low-pass filter
    bilinear transformation, 19-26
    impulse invariant method, 19-24–19-25
  transformation
    bilinear transformation, 19-23

impulse invariance method, 19-23–19-24
types, 19-20–19-22
Antisymmetrical impulse response, odd filter length
error function, 18-32
frequency response, 18-31–18-32
problem formulation, 18-33
Remez algorithm, 18-34
Antoniou CGIC biquad, *see* 2-OA CGIC biquad
Application-specific integrated circuit (ASIC)
programmable logic implementation, 21-11
special-purpose implementation, 21-8, 21-10
Approximation
approximation problem, 2-1–2-2
bandpass filter specifications, 2-7
Bessel–Thompson LPP approximation
gain and group delay characteristics, 2-22–2-23
loss function, 2-19
MFD and MFM approximation, 2-21
rational function, 2-20
recursion formula, 2-22
third-order Bessel filter, 2-22–2-23
Butterworth LPP approximation
definition, 2-8
Feldtkeller's equation, 2-9
gain and characteristic function, 2-8–2-9
$G(s)G(-s)$ poles, 2-10–2-11
loss function, 2-8–2-9
maximally flat magnitude (MFM) approximation, 2-9–2-10
second- and third-order Butterworth transfer function, 2-11
Chebyshev LPP approximation
complex number, 2-14
elliptic approximation, 2-23–2-24
equiripple behavior, 2-13
frequency response, 2-14
$G(s)G(-s)$ poles, 2-15–2-17
minimax error criterion, 2-12
pole location, 2-17–2-18
polynomial behavior, 2-12–2-13
ripple parameter, 2-14, 2-17
elliptic approximation
Chebyshev rational function, 2-26–2-27
elliptic and ordinary sine function, 2-28–2-29
elliptic rational functions, 2-31
filter rank ordering, 2-32
gain and characteristic functions, 2-24
gain plot, 2-31
Jacobian elliptic functions, 2-27
minimax error, 2-30–2-31
phase compensation, 2-32
rational approximation, 2-25
running integral, 2-27–2-28
sine transformation, 2-29–2-30
Weierstrass theorem, 2-23, 2-25–2-26
filter types, 2-3–2-5

B

Bamberger pyramids
2-D separable DWT, 25-24
image denoising, 25-26–25-28
Laplacian pyramid structure, 25-23–25-24
lowpass–highpass decomposition, 25-23
multiresolution (MR) process, 25-22
polar-separable representation, 25-22–25-23
texture analysis and segmentation, 25-25–25-26
Bandpass filters (BPF)
analog prototype filters
bandpass-to-low-pass conversion, 19-30
frequency response, 19-31
LPF to BPF transformation, 19-29–19-30
magnitude response specification, 19-28–19-29
specification, 19-30–19-31
transfer function, 19-31
approximation, 2-7
digital frequency transformation, 19-34–19-35
Sallen and Key filter, 11-8–11-9
Bandstop filters (BSF)
analog prototype filters, 19-32
digital frequency transformation, 19-35–19-36
Bartlett window
low-pass magnitude response, 18-10–18-11
spectrum, 18-10
Baseband communication, 26-28–26-29
Bayes minimum distance classifier, 25-25
Berka–Herpy biquad, 13-24–13-25
Bessel–Thompson low-pass prototype (LPP) approximation
gain and group delay characteristics, 2-22–2-23
loss function, 2-19
MFD and MFM approximation, 2-21
rational function, 2-20
recursion formula, 2-22
third-order Bessel filter, 2-22–2-23
Bilinear transformation
analog filter transformation, 23-2, 23-4
1-D analog transfer function, 23-18–23-19
1-D discrete transfer function, 23-21
nonessential singularity, 23-22
Biorthogonal filter bank, 24-16
Biquad multiloop feedback, 11-17
Biquadratic filter, 17-23–17-24
Blackman window
amplitude spectrum, 18-13
expression, 18-12
magnitude response, 18-13–18-14
Broadband matching network design
Butterworth networks
all-pass function, 10-15–10-16
Butterworth transducer power-gain, 10-15
equalizer back-end impedance, 10-16–10-17

final reactive element, 10-17
LC ladder network, 10-17–10-18
load impedance, 10-14
minimum-phase solution, 10-15
Chebyshev networks
all-pass function, 10-23
equalizer back-end impedance, 10-24
Laurent series expansion, 10-23
$RC\omega_c \sinh a<2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-25
$RC\omega_c \sinh a<2 \sin \gamma_1$ and $L_{b1}<L$, 10-25
$RC\omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1}<L$, 10-24–10-25
$RC\omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-24
transducer power-gain, 10-22
coefficient constraints
normalized reflection coefficient, 10-5–10-6
passive impedance, 10-7–10-8
zero of transmission, 10-2–10-4
design procedure, 10-8–10-9
network configuration, 10-1–10-2
transmission and reflection coefficients, 10-1–10-2
Brune section
positive-real impedance, 6-9–6-10
two-port network, 6-8
type-A and type-B section, 6-8–6-9
type-E section, 6-7–6-8
Butterworth filters
factorized and normalized Butterworth polynomials, 19-13
magnitude response square, 19-12–19-13
transfer function, 19-11–19-12
Butterworth low-pass prototype (LPP) approximation
definition, 2-8
Feldtkeller's equation, 2-9
gain and characteristic function, 2-8–2-9
$G(s)G(-s)$ poles, 2-10–2-11
loss function, 2-8–2-9
maximally flat magnitude (MFM) approximation, 2-9–2-10
second- and third-order butterworth transfer function, 2-11
Butterworth networks
broadband matching network design
all-pass function, 10-15–10-16
Butterworth transducer power-gain, 10-15
equalizer back-end impedance, 10-16–10-17
final reactive element, 10-17
LC ladder network, 10-17–10-18
load impedance, 10-14
minimum-phase solution, 10-15
resistively terminated network design
element values, 9-6–9-7
input impedance, 9-5
input reflection coefficient, 9-4–9-5
LC ladder, 9-6
minimum-phase solution, 9-5
recurrence formula, 9-6
transducer power gain, 9-4

C

Cauer canonical form
LCM one-port network, 7-4
RC one-port network, 7-9
Cauer filters, *see* Elliptic filters
CCCS, *see* Current-controlled current sources
CCVS, *see* Current-controlled voltage sources
Cell Broadband Engine Architecture, 21-8
CGIC, *see* Current generalized immittance convertors
Chebyshev filters
magnitude response square, 19-14–19-15
normalized Chebyshev polynomials, 19-15–19-16
polynomial coefficients, 19-14
transfer function, 19-13–19-14
Chebyshev low-pass prototype (LPP) approximation
complex number, 2-14
elliptic approximation, 2-23–2-24
equiripple behavior, 2-13
frequency response, 2-14
$G(s)G(-s)$ poles, 2-15–2-17
minimax error criterion, 2-12
pole location, 2-17–2-18
polynomial behavior, 2-12–2-13
ripple parameter, 2-14, 2-17
Chebyshev networks
broadband matching network design
all-pass function, 10-23
equalizer back-end impedance, 10-24
Laurent series expansion, 10-23
$RC\omega_c \sinh a<2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-25
$RC\omega_c \sinh a<2 \sin \gamma_1$ and $L_{b1}<L$, 10-25
$RC\omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1}<L$, 10-24–10-25
$RC\omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-24
transducer power-gain, 10-22
resistively terminated network design
element values, 9-10–9-11
input impedance, 9-10
input reflection coefficient, 9-9
minimum-phase solution, 9-9–9-10
transducer power-gain, 9-9
Circularly symmetric 2-D IIR filter
HP, BP, and BS filter realization configuration, 23-24–23-25
Nyquist point, 23-25
subfilter contour plot, 23-23–23-24
LP filter
1-D analog transfer function, 23-18–23-21
1-D discrete transfer function, 23-21–23-22
nonessential singularity elimination, 23-22–23-23

prescribed specification, 23-25–23-26
Coefficient sensitivity, 4-4–4-5
Color filter array (CFA)
Bayer pattern, 28-4
bicubic interpolation, 28-2, 28-4
designs, 28-2
mosaic pattern, 28-1
sampling
compression, 28-18
Fourier domain, 28-11
luminance channel, 28-10
optimal spectral sensitivity function, 28-19
super-resolution reconstruction, 28-17–28-18
Common-mode feedback (CMFB) circuit, 17-22–17-23
Conjugate quadrature filter bank (CQF), 24-16
Continuous-time integrated filters
g_m-C filters
cascade realization, 16-23–16-26
circuit symbols, 16-15
difference current, 16-16
first- and second-order sections, 16-22–16-23
gyrators, 16-20–16-22
integrators, 16-17–16-20
ladder simulation, 16-26–16-31
MOS device, 16-16
resistors, 16-15–16-16
small-signal equivalent circuit, 16-14–16-15
transconductance designs, 16-16–16-17
MOSFET-C method
balanced linear MOSFET conductance, 16-3–16-4
cascade realization, 16-9–16-11
equivalent resistor circuit, 16-4
first- and second-order sections, 16-7–16-9
integrator, 16-5–16-7
ladder simulation, 16-11–16-14
nonlinear drain current, 16-3
tuning process, 16-31–16-32
Continuous-time signal, 1-2
Cosine modulated filter bank, 24-43
Current-controlled current sources (CCCS), 4-18
Current-controlled voltage sources (CCVS), 4-18
Current generalized immittance converter (CGIC) biquads
Antoniou CGIC (2-OA CGIC biquad)
composite GIC biquad, 14-18, 14-19
configuration, 14-2–14-3
design and tuning procedure, 14-6–14-7
3G and 4G ports, 14-2
realization, 14-4–14-5
second-order transfer function, 14-3–14-4
sensitivity analysis, 14-4, 14-6
stability properties, 14-4
universal biquad, 14-11, 14-13–14-14
3-OA CGIC biquad
design and tuning procedure, 14-15–14-16
sixth-order elliptic BP filter design, 14-16–14-17
structure, 14-11, 14-14–14-15
Curvelet-99 transform, 27-20

D

Darlington type-D section
cascade connection, 6-12–6-13
impedance matrix, 6-13–6-14
positive-real impedance, 6-11–6-12
transmission matrix, 6-12–6-13
Daubechies wavelets
Bernstein polynomial, 24-18
conjugate quadrature filter bank (CQF), 24-16
Daubechies 8-tap symlet filters, 24-18–24-19
impulse and frequency responses, 24-16–24-18
$P_t(Z)$ factorization, 24-16
Device under test (DUT), 17-31
DFT, *see* Discrete Fourier transform
Diamond filters, 22-22–22-23, 25-8, 25-14
Difference equation, 20-15–20-16
Digital differentiators, 18-34–18-35
Directional filter banks (DFBs)
Bamberger directional filter bank (BDFB)
analysis filter, 25-10
analysis tree structure, 25-11
downsampling matrix, 25-11–25-12
eight-band BDFB analysis, 25-9–25-10
frequency band partition, 25-9
M-channel parallel structure, 25-10–25-11
wedge-shaped passband, 25-10–25-11
Bamberger pyramids
2-D separable DWT, 25-24
Laplacian pyramid structure, 25-23–25-24
lowpass–highpass decomposition, 25-23
multiresolution (MR) process, 25-22
polar-separable representation, 25-22–25-23
2-D maximally decimated filter bank, 25-7–25-8
2-D multirate system
alias-free 2-D decimation, 25-5–25-7
continuous time signal, 25-2
2-D decimation and interpolation, 25-4–25-5
2-D unimodular matrix, 25-7
2-D upsampler and interpolator, 25-7
fundamental parallepiped, 25-4
quincunx lattice, 25-3–25-4
2-D two-channel fan filter bank
BDFB design, ladder structure, 25-16–25-20
FFB design, 1-D quadrature mirror filter (QMF), 25-12–25-16
fingerprint enhancement and recognition, 25-28
finite-field DFB, 25-28–25-29
image denoising
additive white Gaussian noise and UDWT, 25-27–25-28

Bamberger pyramid, 25-27
Gaussian white noise, 25-26
wavelet denoising procedure, 25-26–25-27
iris recognition, 25-28
texture analysis and segmentation
Bamberger pyramid, 25-25–25-26
classification system, 25-25
multichannel method, 25-24–25-25
undecimated directional filter bank (UDFB)
decomposition, 25-22–25-23
implementation, 25-21–25-22
shift invariance, 25-20–25-21
UCFB and UFFB, 25-21–25-22
undecimated discrete wavelet transform (UDWT), 25-20
unimodular resampling matrix, 25-22
velocity selective filter bank (VSFB), 25-29–25-30
Discrete cosine transform (DCT), 24-43
Discrete Fourier transform (DFT), 22-10, 24-43
Discrete-space Fourier transform, 22-7
Discrete-time signal, 1-2
Discrete wavelet packet transform (DWPT), 27-27
Discrete wavelet transform (DWT), 24-8–24-9
1-D multirate filter banks
2-band filter bank
analysis, 24-3
downsampling and upsampling, 24-4–24-5
output expression, 24-3
reconstruction operation, 24-3–24-4
binary filter trees
decomposition levels, 24-6–24-7
size and bandwidth, 24-7
transformation, 24-7–24-8
discrete wavelet transform (DWT), 24-2
FIR filters and wavelets
antialiasing condition, 24-12
biorthogonal filter bank, 24-16
Daubechies wavelets, 24-16–24-19
high-pass filter, 24-14
LeGall 3,5-tap filter, 24-14–24-15
linear interpolation, 24-14
linear phase and balanced frequency response, 24-19–24-21
low-pass product filter, 24-13
number of zeros, 24-13–24-14
perfect reconstruction condition, 24-12
smoother wavelet, 24-21–24-24
Hilbert pairs
à trous algorithm, 24-43–24-44
common-factor dual-tree filter design, 24-45
dual-tree wavelet transform, 24-44–24-45
metrics for shift dependence, 24-50–24-52
Q-shift dual-tree filter design, 24-46–24-50
shift dependence and shift invariance, 24-43
IIR filter
all-pass filter design, 24-25–24-26
causal and anticausal filtering, 24-24
transformation-based design, 24-26–24-28
types, 24-24
lifting scheme
construction, 24-34–24-36
implementation, 24-37–24-39
M-band filter bank
lifting, 24-42–24-43
type 1 polyphase decomposition, 24-41–24-42
type 2 polyphase decomposition, 24-42
types, 24-43
multirate filtering, 24-5–24-6
nonlinear filter bank, 24-40–24-41
polyphase representation
analysis and synthesis outputs, 24-30
decomposition, 24-29–24-30
even- and odd-indexed coefficients, 24-29
linear phase structure, 24-32–24-33
orthogonal lattice structure, 24-32
total system polyphase matrix, 24-31–24-32
signal sparsity, 24-2
wavelets and scaling functions
analysis side equation, 24-9
convergence condition, 24-11
equivalent transfer function, 24-10–24-11
Fourier domain, 24-10
impulse and frequency responses, 24-8–24-9
piecewise constant function, 24-11
synthesis side equation, 24-9–24-10
vanishing moments (VM), 24-11
Dual-amplifier twin-*T* biquads
band-rejection filter, 13-6–13-7
Cauer low-pass and high-pass filter, 13-8–13-9
high-pass filter, 13-7–13-8
low-pass filter, 13-7
twin-*T* resonator, 13-5–13-6
voltage transfer ratios, 13-7
3-D wavelet packets
adaptive transform, 27-27
restoration algorithm, 27-27–27-28
thresholding, 27-26
wavelet denoising, 27-26–27-27
DWPT, *see* Discrete wavelet packet transform

E

Elliptic approximation
Chebyshev rational function, 2-26–2-27
elliptic and ordinary sine function, 2-28–2-29
elliptic rational functions, 2-31

filter rank ordering, 2-32
gain and characteristic functions, 2-24
gain plot, 2-31
Jacobian elliptic functions, 2-27
minimax error, 2-30–2-31
phase compensation, 2-32
rational approximation, 2-25
running integral, 2-27–2-28
sine transformation, 2-29–2-30
Weierstrass theorem, 2-23, 2-25–2-26
Elliptic filters
elliptic integral, 19-18
equiripples, 19-17
magnitude response square, 19-17, 19-19
pole and coefficient calculation, 19-18–19-19
rational function, 19-18
Equiripple FIR filters, 18-15–18-16
alternation theorem, 18-19
amplitude response, 18-17–18-18
antisymmetrical impulse response and odd filter length
error function, 18-32
frequency response, 18-31–18-32
problem formulation, 18-33
Remez algorithm, 18-34
arbitrary amplitude response, 18-35
cubic-interpolation search method, 18-25–18-26
digital differentiators, 18-34–18-35
error function, 18-16–18-17
frequency response, 18-16
multiband filter, 18-35, 18-37
noniterative/iterative methods, 18-15–18-16
piecewise-constant amplitude-response specification, 18-30
Remez exchange algorithm
exhaustive step-by-step search, 18-21–18-23
extremal initialization, 18-20–18-21
iterative multivariable algorithm, 18-20
maxima of error function, 18-21
superfluous potential external rejection, 18-23–18-24
selective step-by-step search method, 18-24–18-25
transfer function, 18-16
weighted-Chebyshev method, 18-16

F

Fan filter banks (FFBs), 25-8
Fast Fourier transform (FFT)
approximate minimax FIR filter, 22-13–22-14
window-based 2-D FIR filter design, 22-10
FDNR, *see* Frequency-dependent negative resistors
Field programmable gate array (FPGA)
GNURadio, 21-8
programmable logic implementation
FIR filter, 21-12–21-13
IIR filter, 21-13–21-14
Filter noise analysis, 11-27–11-29
Fingerprint enhancement and recognition, 25-28
Finite-impulse response (FIR) filters
code segment
ADSP-21020 architecture, 21-4–21-5
Motorola 56001 architecture, 21-6–21-7
TMS320C30 architecture, 21-3
TMS320C50 architecture, 21-5
2-D filter implementation, 22-24
equiripple FIR filters
alternation theorem, 18-19
amplitude response, 18-17–18-18
antisymmetrical impulse response and odd filter length, 18-31–18-34
arbitrary amplitude response, 18-35
cubic-interpolation search method, 18-25–18-26
digital differentiators, 18-34–18-35
error function, 18-16–18-17
frequency response, 18-16
multiband filter, 18-35, 18-37
noniterative/iterative methods, 18-15–18-16
piecewise-constant amplitude-response specification, 18-30
Remez exchange algorithm, 18-20–18-24
selective step-by-step search method, 18-24–18-25
transfer function, 18-16
weighted-Chebyshev method, 18-16
filter banks, 22-25–22-26
frequency response
antisymmetric impulse response and $N = even$, 18-5
antisymmetric impulse response and $N = odd$, 18-4–18-5
masking approach, 18-38–18-45
symmetric impulse response and $N = even$, 18-4
symmetric impulse response and $N = odd$, 18-3–18-4
linear phase property
difference equation, 18-2
impulse response, 18-3
phase and group delay functions, 18-2
phase response, 18-2–18-3
transfer function, 18-1–18-2
L_p norm, 22-11-22-12
multistage frequency-response masking approach
complexity minimization, 18-48–18-49
implementation form, 18-47
synthesis, 18-47–18-48
three-stage filter, 18-47
transfer function, 18-46–18-47
narrowband filter design
amplitude response, 18-51–18-52
Jing–Fam approach, 18-49, 18-55, 18-57–18-58

multiplier, minimization, 18-52–18-53
order of $F(z)$ and $G(z)$, 18-52
passband and stopband ripples, 18-56
single-stage design, 18-57
subfilter design, 18-51
synthesis, 18-50–18-51, 18-57–18-58
transfer function, 18-49, 18-51, 18-55–18-56
narrow transition band, 18-38
nonlinear-phase low-pass filter, 22-4–22-5
programmable logic implementation, 21-12–21-13
roundoff noise, 20-6–20-7
separable filter design, 22-16–22-17
special-purpose implementation
ASIC comparison, 21-10–21-11
Motorola 56200 architecture, 21-9–21-10
parallelism and pipelining, 21-9
180 MHz sampling rate, 21-9–21-10
wavelets
antialiasing condition, 24-12
biorthogonal filter bank, 24-16
Daubechies wavelets, 24-16–24-19
high-pass filter, 24-14
LeGall 3,5-tap filter, 24-14–24-15
linear interpolation, 24-14
linear phase and balanced frequency response, 24-19–24-21
low-pass product filter, 24-13
number of zeros, 24-13–24-14
perfect reconstruction condition, 24-12
smoother wavelet, 24-21–24-24
wideband filter design, 18-53–18-54
window-based filter design, 22-6–22-10
windowing techniques
Bartlett window, 18-10
Blackman window, 18-12–18-14
filter coefficient, 18-6
Hamming window, 18-11–18-12
Hanning window, 18-10–18-12
ideal low-pass filter response, 18-7
Kaiser window, 18-13–18-15
rectangular window, 18-7–18-9
zero locations, 18-5–18-6
zero-phase filters, 22-4–22-5
Finite wordlength effects
coefficient quantization error
alternate realization structure, 20-16–20-17
clustered root, 20-16
definition, 20-15
grid, 20-15–20-16
realizable pole locations, 20-16–20-17
transfer function, 20-15
fixed-point quantization error
sum of products, 20-2–20-3
types, 20-3–20-4
floating-point quantization error, 20-4–20-5
limit cycles
definition, 20-12
filter realization structure, 20-14
implementation sequence, 20-13
second-order filter, 20-12–20-13
number representation, 20-1–20-2
overflow oscillations, 20-14–20-15
realization, 20-17–20-18
roundoff noise
FIR filter, 20-6–20-7
fixed-point IIR filter, 20-7–20-10
floating-point IIR filter, 20-10–20-12
zero-mean white noise filter, 20-5
First-order *RC* network, 1-25
Fixed-point processors, 21-2
FLF, *see* Follow-the-leader feedback
Floating-point processors, 21-2
Follow-the-leader feedback (FLF) topology
feedback resistors, 15-11
first-order lowpass functions, 15-8
GIC bandpass sections, 15-11–15-12
lowpass-to-bandpass transformation, 15-8
matrix expression, 15-9
primary resonator block (PRB) design process, 15-10–15-11
second-order bandpass biquads, 15-7
summer and feedback factor implementation, 15-6–15-7
Foster canonical form
LCM one-port network, 7-3
RC one-port network, 7-8–7-10
Fourier slice theorem, 27-6–27-7
Fourier transform
analog filter, 1-16–1-17
Bartlett window, 18-10
Blackman window, 18-13
Hamming window, 18-11–18-12
Hanning window, 18-10–18-11
Rectangular window, 18-7–18-8
FPGA, *see* Field programmable gate array
Frequency-dependent negative resistors (FDNR), 15-27
Frequency-domain analysis
graphical construction
amplitude and phase responses, 1-13
frequency response evaluation, 1-13–1-14
gain and phase shift, 1-13
transfer function, 1-12–1-13
loss function, 1-14–1-15
sinusoidal response, 1-11–1-12
Frequency-response masking approach
complementary periodic filter pair, 18-40
error function, 18-43
$F(L\omega)$ design, 18-43
filter synthesis problem, 18-44
low-pass filter design, 18-40–18-42

passband and stopband ripples, 18-42
prototype transfer function, 18-40
subfilter order, 18-44–18-45
transfer function, 18-38–18-39
transition bandwidth, 18-44
zero-phase frequency response, 18-39–18-40
Frequency sampling, 22-10–22-11
Frequency transformations
bandpass transformation
imaginary frequency axis, 3-8–3-9
narrow-band approximation, 3-8
network transformation, 3-9–3-10
passband and stopband specifications, 3-8–3-9
rational function, 3-7
band-reject transformation
characterization, 3-10–3-11
frequency responses, 3-11
LPP transfer function, 3-12
network transformation, 3-12–3-13
frequency scaling
LP network transformation, 3-3–3-4
LPP transfer function, 3-2–3-3
high-pass transformation, 3-5–3-6
impedance scaling, 3-4
low-pass prototype, 3-1–3-2
Friend biquad, 12-12–12-13
Function sensitivity, 4-3–4-4

G

Gain-bandwidth
noninverting and inverting amplifier, 11-23–11-25
Sallen–Key second-order low-pass filter, 11-26–11-27
transfer function, 11-26
Generalized impedance converter (GIC)
dual-amplifier biquads
all-pass building block, 13-14–13-15
bandpass building block, 13-12–13-13
Cauer high-pass building block, 13-17–13-18
Cauer low-pass building block, 13-15–13-16
high-pass building block, 13-13–13-14
low-pass building block, 13-10
notch filter building block, 13-14–13-15
resonator, 13-9
time constant and conductance ratio, 13-11
transfer function, 13-12
voltage transfer ratio, 13-9–13-10
three-amplifier biquads, 13-18–13-19
g_m-C filters
cascade realization
capacitor values, 16-25–16-26
filter parameters, 16-24
low-pass filter structure, 16-25–16-26
low-pass g_m-C section, 16-24–16-25
circuit symbols, 16-15
difference current, 16-16
gyrators
definition, 16-20
inductive admittance, 16-21
inverting and noninverting transconductor, 16-20–16-21
passive equivalent circuit, 16-21–16-22
integrators
floating/grounded capacitor, 16-17–16-18
g_m-C–op amp integrator, 16-19
nominal correction, phase errors, 16-19–16-20
parasitic capacitors, 16-18–16-19
quality factor, 16-18
transfer function, 16-17
ladder simulations
element replacement methods, 16-26–16-29
signal-flow graph methods, 16-29–16-31
MOS device, 16-16
resistors, 16-15–16-16
small-signal equivalent circuit, 16-14–16-15
transconductor, 16-14
GNURadio, 21-7–21-8
Gorski-Popiel's method
inductance networks, 15-25–15-26
LC ladder realization, 15-26
sixth-order *LC* bandpass filter, 15-26–15-27
Graphical construction, frequency-domain analysis
amplitude and phase responses, 1-13
frequency response evaluation, 1-13–1-14
gain and phase shift, 1-13
transfer function, 1-12–1-13
Group delay function, 2-20–2-21
Gyration resistance, 20

H

Haar filters
all-pass filter, 24-25–24-26
4-level tree, 24-8–24-9
Half-plane symmetric filter, 23-14
Hamming window
amplitude spectrum, 18-11–18-12
2-D FIR filter design, 22-8–22-9
magnitude response, 18-12–18-13
Hanning window
amplitude spectrum, 18-10–18-11
expression, 18-10
magnitude response, 18-10, 18-12
High-order filters
cascade realization
Åckerberg–Mossberg circuit, 15-5–15-6
biquadratic function, 15-3
intermediate gain-scaled transfer functions, 15-4
*n*th-order transfer function, 15-2–15-3

sixth-order transfer function, 15-5
ladder simulations
dynamic range maximization, 15-18–15-23
element substitution, 15-23–15-27
signal-flow graph methods, 15-13–15-18
low-gain active filter, 11-16–11-17
multiple-loop feedback realizations
feedback resistors, 15-11
first-order lowpass functions, 15-8
GIC bandpass sections, 15-11–15-12
leapfrog topology, 15-12
lowpass-to-bandpass transformation, 15-8
matrix expression, 15-9
primary resonator block (PRB) design process, 15-10–15-11
second-order bandpass biquads, 15-7
summer and feedback factor implementation, 15-6–15-7
High-pass filters (HPF)
analog prototype filters
analog and digital frequency, 19-27
frequency response, 19-28–19-29
inverse Chebyshev filter transfer function, 19-28
LPF to HPF transformation, 19-27
specification, 19-27–19-28
configuration, 23-24–23-25
derivation, 23-23–23-24
digital frequency transformation, 19-33–19-34
1-D multirate filter banks, 24-14
Nyquist point, 23-25
Sallen and Key filter, 11-9–11-10
transfer function, 23-23–23-24
Hilbert pairs
à trous algorithm, 24-43–24-44
common-factor dual-tree filter design, 24-45
dual-tree wavelet transform, 24-44–24-45
metrics, shift dependence
aliasing energy ratio, 24-51
analysis and reconstruction, 24-50–24-51
dual-tree complex wavelet transform (DT-CWT), 24-51–24-52
shift invariance and shift dependence, 24-50
Q-shift dual-tree filter design
impulse response, 24-46
iterative energy minimization method, 24-48–24-49
zero-forcing method, 24-47
shift dependence and shift invariance, 24-43
HPF, *see* High-pass filters
Hurwitz polynomials, 5-7–5-8, 9-10

I

Image denoising, 25-26–25-28
Impulse response, 1-8
Independent component analysis (ICA), 27-3
Infinite impulse response (IIR) filters
allpass filter
block diagram, 19-10
cascade filter, 19-9–19-10
magnitude response, 19-8
phase response, 19-9
pole-zero reciprocal pair, 19-8–19-9
analog low-pass filter (LPF)
Butterworth filter, 19-11–19-13
Chebyshev filter, 19-13–19-16
elliptic filter, 19-17–19-19
inverse Chebyshev filter, 19-15–19-17
specification, 19-11
analog prototype filters
bandpass filter, 19-28–19-31
bandstop filter, 19-32
design steps, 19-20
high-pass filter, 19-27–19-29
low-pass filter, 19-24–19-26
transformation, 19-20–19-24
causality, 19-2–19-3
code segment
Motorola 56001 architecture, 21-7
TMS320C30 architecture, 21-3–21-4
TMS320C50 architecture, 21-5–21-6
computer-aided design
least squares ($\mathcal{L}_2$-norm) criterion, 19-36–19-40
weighted Chebyshev ($\mathcal{L}\infty$-norm) criterion, 19-40
digital frequency transformation
bandpass filter, 19-34–19-35
bandstop filter, 19-35–19-36
design process, 19-32–19-33
high-pass filter, 19-33–19-34
low-pass filter, 19-33
digital LPF specification, 19-11–19-12
direct form realizations
cascade second-order filter, 19-6
coefficient quantization, 19-7–19-8
parallel second-order filter, 19-7
system function, 19-5–19-6
1-D multirate filter banks
all-pass filter design, 24-25–24-26
causal and anticausal filtering, 24-24
transformation-based design
magnitude response, 24-28
phase response, 24-27–24-28
prototype linear phase low-pass filter, 24-27
types, 24-24
fixed-point IIR filter
first-order filter, 20-7
limit cycle, 20-13
normalized fixed-point roundoff noise variance, 20-9
quantization, noise source, 20-8
roundoff noise-to-signal ratio, 20-8–20-9
second-order filter, 20-8–20-9

state-space filter realization, 20-10
transfer function, 20-7–20-8
floating-point IIR filter
first-order filter, 20-10–20-11
limit cycle, 20-13
normalized floating-point roundoff noise variance, 20-12
roundoff noise-to-signal ratio, 20-11–20-12
second-order filter, 20-11–20-12
zero-mean white noise, 20-10
frequency response
complex function, 19-3–19-4
group delay, 19-5
magnitude square, 19-4
phase response, 19-5
lattice digital filters
injected numerator structure, 19-41–19-42
injector gains, 19-44–19-45
reflection coefficient, 19-42–19-44
tap gains, 19-44
tapped numerator structure, 19-41–19-42
third-order one-multiplier injected numerator filter, 19-42–19-43
minimum phase filter, 19-8
poles and zeros, 19-3
programmable logic implementation, 21-13–21-14
special-purpose implementation, 21-11
stability
optimization techniques, 19-40–19-41
second-order system, 19-3–19-4
system function and impulse response, 19-1–19-2
Inverse Chebyshev filters, 19-15–19-17
Iris recognition, 25-28
Iterative filter design algorithm, 22-13–22-14

J

Jing–Fam approach, 18-49, 18-55, 18-57–18-58

K

Kaiser window
definition, 18-15
magnitude response, 18-14–18-15
prolate spheroidal wave function, 18-14
spectral properties, 18-15
stopband attenuation, 18-13–18-14
Kirchhoff current law equation, 5-1

L

Ladder simulations
dynamic range maximization
active circuit, 15-22–15-23
feed-in resistors, 15-23
fourth-order elliptic lowpass ladder filter, 15-20–15-21
passive series branch, 15-19–15-20
passive shunt branch, 15-20
signal-level scaling, 15-18–15-19
transmittance, 15-21
voltage and current maxima, 15-21
element replacement methods
circuit capacitor value, 16–28–16-29
component values, 16-27
dual-input OTA, 16-27–16-28
third-order elliptic low-pass ladder, 16–28–16-29
transconductor-C simulation, 16-26–16-27
transfer function, 16–28
element substitution
Antoniou's general impedance converter (GIC), 15-24
Gorski-Popiel's method, inductor replacement, 15-25–15-27
impedance converter, 15-23–15-24
MOSFET-C method
elliptic LC low-pass filter, 16-12
implementation, 16-13–16-14
transfer function magnitude, 16-12
V–I relationship, 16-12–16-13
signal-flow graph methods
block diagram, 15-13–15-14
fourth-order all-pole lowpass ladder, 15-17–15-18
impedance- and signal-level scaling, 15-16
inverting lossy Miller integrator, 15-14
ladder structure, 15-13
noninverting lossy phase-lead integrator, 15-14
series and shunt ladder branch, 15-15–15-16
signal-flow graph (SFG) methods
fifth-order elliptic low-pass function, 16–29–16-30
third-order elliptic low-pass filter, 16–30–16-31
Ladder structures
analysis–synthesis ladder structure, 25-17–25-18
biorthogonal property, 25-19–25-20
closed-form Lagrange formula, 25-18
digital fan-shaped filter, 25-17
magnitude response, 25-18–25-19
regular biorthogonal wavelet generation, 25-18
two-channel biorthogonal analysis filter bank, 25-19
wavelet construction, 25-16
Laplace transform
analog filter characteristics, 1-3–1-4
analog IIR filter, 19-11
Laplacian statistics
filtering structure, 26-8–26-9
location estimation, 26-9–26-10
α-stable distributed samples, 26-10–26-11
weighted median (WM) filter, 26-9
Lattice digital filters
injected numerator structure, 19-41–19-42
injector gains, 19-44–19-45

reflection coefficient, 19-42–19-44
tap gains, 19-44
tapped numerator structure, 19-41–19-42
third-order one-multiplier injected numerator filter, 19-42–19-43
LC ladder
zero-producing step, 8-4–8-5
zero-shifting step, 8-2–8-4
LCM one-port network
admittance function, 7-3
driving-point immittance, 7-2
first and second Cauer canonical form, 7-4
first and second Foster canonical form, 7-3
input impedance, 7-1
reactance function, 7-2–7-3, 7-5–7-6
Leapfrog topology, 15-12
Learning vector quantization (LVQ) algorithm, 25-25
Least squares ($\mathcal{L}_2$-norm) criterion
adaptive weighted least square (WLS), 19-37–19-38
eigenfilter approach, 19-39–19-40
quadratic programming, 19-38
transfer function, IIR filter, 19-36
weighted square error, 19-36–19-37
LeGall 3,5-tap filter, 24-14–24-15
Lifting schemes; *see also* Ladder structures
construction
analysis polyphase matrix types, 24-34–24-35
elementary matrix, 24-34
perfect reconstruction (PR) filter bank, 24-35
synthesis polyphase matrix, 24-35–24-36
implementation
division process, 24-37–24-38
dual lifting, 24-38–24-39
Matlab function, 24-37
scaling matrix, 24-39
Linear filtering, 26-7–26-8
Linear programming, 22-10–22-11
Linear shift-invariant system, 22-2
LM-estimation
definition, 26-11
generalized Cauchy density, 26-11–26-14
arg min and log functions, 26-13–26-14
error norm, 26-14–26-15
GCD function, 26-11–26-12
heavy-tailed distribution, 26-12
impulsive sequence, 26-12
minimization problem, 26-13
real-valued weights and optimization, 26-26–26-27
running meridian smoothers, 26-21–26-26
medianity parameter, 26-21–26-22
meridian estimator, 26-22
mode–type estimator, $\delta \to 0$, 26-23–26-24
properties, 26-25
sample median, $\delta \to \infty$, 26-22–26-23
sample meridian, 26-21
weighted meridian, 26-25–26-26
running myriad smoothers, 26-14–26-21
fixed point weighted myriad search, 26-21
linearity parameter, 26-14–26-15
minimization problem, 26-18
mode–type estimator, $\gamma \to 0$, 26-17
normalized linear estimation, $\gamma \to \infty$, 26-19
no undershoot/overshoot, 26-17
objective function, 26-15–26-16
sample mean, $\gamma \to \infty$, 26-16–26-17
sample myriad, 26-14
shift and sign invariance, 26-18
unbiasedness, 26-18
weighted mode-type estimation, $\gamma \to 0$, 26-19–26-20
weighted myriad, 26-18
Low-gain active filters
first-order filter realizations, 11-6–11-7
first-order transfer functions, 11-1–11-2
frequency response, 11-4–11-5, 11-6, 11-7
higher order filters, 11-16–11-17
nonideality influence
gain-bandwidth, 11-23–11-27
noise analysis, 11-27–11-29
sensitivity analysis, 11-22–11-23
slew rate, 11-27
second-order biquadratic filters
biquadratic transfer function, 11-11–11-12
RC network realization, 11-14–11-15
single-amplifier low-gain realization, 11-12–11-13
two-amplifier low-gain biquad configuration, 11-14
second-order positive-gain filter
bandpass circuit, 11-8–11-9
high-pass filter, 11-9–11-10
low-pass filter, 11-6–11-8
notch filter circuits, 11-10–11-11
second-order transfer functions
biquadratic transfer function, 11-3–11-4
high-pass and bandpass transfer functions, 11-3
low-pass filter, 11-2
tuning active filters, 11-6
Low-pass filters (LPF)
analog prototype filters
bilinear transformation, 19-26
impulse invariant method, 19-24–19-25
1-D analog transfer function
2-D zero-phase filter, 23-21
rotation angle, 23-20–23-21
subfilter stability and realization, 23-20
1-D discrete transfer function, 23-21–23-22
digital frequency transformation, 19-33

nonessential singularity elimination, 23-22–23-23
Sallen and Key filter, 11-6–11-8
Low-pass transfer function
bandpass transformation
bandpass network transformation, 3-9–3-10
bandpass transfer function, 3-7
imaginary frequency axis, 3-8
narrow-band approximation, 3-8
passband and stopband specifications, 3-8–3-9
band-reject transformation
band-rejection network transformation, 3-12–3-13
LPP transfer function, 3-12
transformation responses, 3-11–3-12
high-pass transformation
frequency range, 3-5–3-6
HP circuit, 3-6
imaginary axis, 3-5
network transformation, 3-6
LPF, *see* Low-pass filters

M

Magnitude truncation, 3
Maximally flat magnitude (MFM), 2-9–2-10
Maximum likelihood estimation
generalized Cauchy density
arg min and log functions, 26-13–26-14
error norm, 26-14–26-15
GCD function, 26-11–26-12
heavy-tailed distribution, 26-12
impulsive sequence, 26-12
minimization problem, 26-13
generalized Gaussian density
additive noise, 26-5
definition, 26-5–26-6
minimization problem, 26-6–26-7
M-band filter bank
lifting, 24-42–24-43
type 1 polyphase decomposition, 24-41–24-42
type 2 polyphase decomposition, 24-42
types, 24-43
McClellan transformation
Chebyshev polynomial, 22-17
frequency response, 22-17–22-21
nonlinear optimization problem, 22-21
window-based 2-D FIR filter design, 22-7–22-8
zero-phase filter, 22-17
M-estimation
filtering structure, 26-5
Gaussian statistics, linear filtering, 26-7–26-8
generalized Gaussian density, 26-5–26-7
additive noise, 26-5
definition, 26-5–26-6
minimization problem, 26-6–26-7
Laplacian statistics
filtering structure, 26-8–26-9
location estimation, 26-9–26-10
median filtering, 26-8–26-11
α-stable distributed samples, 26-10–26-11
weighted median (WM) filter, 26-9
MFM, *see* Maximally flat magnitude
Mikhael–Bhattacharyya biquad, 13-18
Minimal mean-square-error approximation, 23-38
Minimax algorithm
Charalambous minimax algorithm, 23-32–23-33
objective function, 23-31–23-32
Minimum-phase filters, 1-17–1-19
MOSFET-*C* method
balanced linear MOSFET conductance, 16-3–16-4
balanced symmetrical op amp, 16-4
cascade realization
aspect ratios, 16-11
gain constant, 16-10–16-11
high-order function, 16-9–16-10
*n*th-order transfer function, 16-10
equivalent resistor circuit, 16-4
first- and second-order sections
Tow–Thomas biquad, 16-7–16-8
transfer function, 16-8–16-9
integrators
active *RC* integrator, 16-5
differential MOSFET-*C* integrator, 16-5–16-6
equivalent circuit, 16-5
four-MOSFET configuration, 16-6
lossy integrator, 16-6–16-7
ladder simulations
elliptic *LC* low-pass filter, 16-12
implementation, 16-13–16-14
transfer function magnitude, 16-12
V–*I* relationship, 16-12–16-13
MOS transistor, 16-3
Mother wavelet, 24-8–24-11
Multiband filter, 18-35, 18-37
Multiple-amplifier biquads
all-pass-based biquads, 13-25–13-26
decoupled time constants
dual-amplifier bandpass filter, 13-2
dual op-amp bandpass filter section, 13-4–13-5
dual op-amp high-pass filter section, 13-4
dual op-amp low-pass filter section, 13-3
Sallen and Key bandpass circuit, 13-2
transfer voltage ratio, 13-2, 13-4
dual-amplifier twin-*T* biquads
twin-*T* band-rejection filter, 13-6–13-7
twin-*T* Cauer low-pass and high-pass filter, 13-8–13-9
twin-*T* high-pass filter, 13-7–13-8
twin-*T* low-pass filter, 13-7
twin-*T* resonator, 13-5–13-6
voltage transfer ratios, 13-7
GIC-derived dual-amplifier biquads
all-pass building block, 13-14–13-15

bandpass building block, 13-12–13-13
Cauer high-pass building block, 13-17–13-18
Cauer low-pass building block, 13-15–13-16
high-pass building block, 13-13–13-14
low-pass building block, 13-10
notch filter building block, 13-14–13-15
resonator, 13-9
time constant and conductance ratio, 13-11
transfer function, 13-12
voltage transfer ratio, 13-9–13-10
GIC-derived three-amplifier biquads, 13-18–13-19
single- to multiple-amplifier biquadratic filter, 13-1
state-variable-based biquads
Åkerberg–Mossberg biquad, 13-24–13-25
analog computer structure, 13-20–13-21
Berka–Herpy biquad, 13-24–13-25
low-pass transfer function, 13-21
noninverting integrator, 13-24
op-amp realization, Tow–Thomas filter, 13-22–13-23
output amplifier, 13-21–13-22
Tow–Thomas biquad, 13-23–13-24
Multiple-feedback (MFB) filters
all-pole designs
equal-capacitor design, 12-5
low-pass structure, 12-3–12-4
realization, 12-3–12-5
transfer ratio, 12-4–12-5
design
finite amplifier gain effects, 12-8–12-9
sensitivity, 12-8
tuning function, 12-9–12-10
MMFB structure
bandpass transfer function, 12-10–12-11
biquadratic structure, 12-12–12-13

N

Narrowband filter
amplitude response, 18-51–18-52
Jing–Fam approach, 18-49, 18-55, 18-57–18-58
multiplier, minimization, 18-52–18-53
order of $F(z)$ and $G(z)$, 18-52
passband and stopband ripples, 18-56
single-stage design, 18-57
subfilter design, 18-51
synthesis, 18-50–18-51, 18-57–18-58
transfer function, 18-49, 18-51, 18-55–18-56
NMOS transistor, switch resistance, 17-19–17-20
Nondiagonal nonsingular integer matrix, 22-22
Nonlinear filtering
image denoising
adaptive transform, 27-27
best basis selection, 27-16–27-17
brushlets and wave atoms, 27-20, 27-22
butterfly-based digital implementation, 27-9–27-12
continuous planelet transform, 27-23–27-24
continuous transform, 27-5–27-7
definition, 27-1
denoising algorithm, 27-25
discrete planelet transform (DPT), 27-24
discrete transform, 27-7–27-8
fixed/adaptive transform, 27-4
image data corruption, 27-1–27-2
inverse problem, 27-2
mapping function, 27-3
multiresolution Fourier transform, 27-19–27-20
multiscale polar cosine transform, 27-17–27-19, 27-22
nonlinear approximation, 27-12–27-16
planar feature extraction, 27-25–27-26
polar cosine packets, 27-16
radon-based digital implementation, 27-9
restoration algorithm, 27-27–27-28
ridgelet and curvelet transform, 27-20, 27-22
separable/nonseparable transform, 27-3–27-4
spatiotemporal denoising method, 27-2
standard images, 27-21
standard video sequences, 27-28–27-31
texture representation, 27-5
thresholding, 27-26
transform domain, 27-3
wavelet denoising, 27-26–27-27
statistical signal models
additive noise extraction, 26-27
baseband communication, 26-28–26-29
corrupted image sample, 26-2–26-3
definition, LM-estimation, 26-11
filtering structure, 26-5
Gaussian statistics, linear filtering, 26-7–26-8
generalized Cauchy density, 26-11–26-14
generalized Gaussian density, 26-5–26-7
highpass filtering, 26-29–26-31
Laplacian statistics, median filtering, 26-8–26-11
optimal estimation-based filter development, 26-2
power line communications (PLCs), 26-29–26-30

real-valued weights and optimization, 26-26–26-27
running meridian smoothers, 26-21–26-26
running myriad smoothers, 26-14–26-21
signal processing, 26-1–26-2
symmetric α-stable distributions, 26-28
Nonlinear-phase low-pass filter, 22-4–22-5
Nonminimum-phase transfer functions, 1-20–1-21
Norton equivalent, 9-1–9-2
Nyquist points, 23-25

O

On-chip spectrum/vector analyzer, 17-31
2-Opamp (OA) CGIC biquad
composite GIC biquad, 14-18, 14-19
configuration, 14-2–14-3
design
second-order BP filter, 14-8–14-9
second-order Butterworth LPF, 14-7–14-8
second-order HP filter, 14-9
sixth-order Chebyshev low-pass filter, 14-10–14-11
sixth-order elliptic bandpass filter, 14-11–14-12
and tuning procedure, 14-6–14-7
3*G* and 4*G* ports, 14-2
realization, 14-4–14-5
second-order transfer function, 14-3–14-4
sensitivity analysis, 14-4, 14-6
stability properties, 14-4
universal biquad, 14-11, 14-13–14-14
3-Opamp (OA) CGIC biquad
design and tuning procedure, 14-15–14-16
sixth-order elliptic BP filter design, 14-16–14-17
structure, 14-11, 14-14–14-15
Operational transconductance amplifier (OTA) filters, *see* g_m-*C* filters

P

Padukone–Mulawka–Ghausi biquad, 13-19
Parallel/series ladders
Cauer canonical form, 8-14–8-15
open-circuit voltage ratio, 8-15–8-16
RC α- and β-ladder, 8-12–8-13
realization, 8-13–8-14
scale factors, 8-12
transfer admittance level, 8-13–8-14
two ladder networks, 8-11
Parks–McClellan algorithm, 25-18
Passive cascade synthesis
Darlington type-D Section
cascade connection, 6-12
impedance matrix, 6-13–6-14
positive-real impedance, 6-11–6-12
transmission matrix, 6-12–6-13
index set, 6-4
Richards section, 6-10–6-11
two-port network, 6-1–6-2
type-E section
Brune section, 6-7–6-9
Darlington type-C section, 6-7
impedance and transmission matrix, 6-6–6-7
Passive immittances, one-port network
average electric and magnetic energy, 5-3
driving-point impedance, 5-2, 5-4
Hurwitz polynomial, 5-7–5-8
Kirchhoff current law equation, 5-1
positive-real function, 5-4–5-5
real rational function, 5-5–5-6
resistive, capacitive and inductive branch, 5-2–5-3
RLCM one-port network, 5-1–5-2
Sturm's theorem, 5-8
Passive RLC filters, 1-23
PLCs, *see* Power line communications
Polar cosine transform
best basis selection, 27-16–27-17
brushlets and wave atoms, 27-20, 27-22
butterfly-based digital implementation
Cartesian separable basis function, 27-10–27-11
forward and inverse transforms, 27-9–27-10
Fourier magnitude spectrum, 27-11–27-12
frequency spectrum, 27-10
continuous transform
Fourier slice theorem, 27-6–27-7
polar Fourier transform, 27-6
Radon transform, 27-6
real-to-complex ridge profile, 27-5
ridge function, 27-5–27-6
discrete transform, 27-7–27-8
multiresolution Fourier transform, 27-19–27-20
multiscale polar cosine transform
basis functions and frequency tiling, 27-18–27-19
construction, 27-17–27-18
denoising results, 27-22
nonlinear approximation
ghosting artifact, 27-14, 27-16
image patches, 27-12
PSNR curves, 27-12–27-14
reconstructions, 27-12, 27-14–27-15
polar cosine packets, 27-16
radon-based digital implementation, 27-9
ridgelet and curvelet transforms, 27-20, 27-22
standard images, 27-21
texture representation, 27-5
Polar Fourier transform, 27-6–27-7
Polar trigonometric transforms, 27-7
Polynomial filter, 2-11
Positive-real function, 5-4–5-5

Power line communications (PLCs), 26-29–26-30
Primary resonator block (PRB), 15-10–15-11
Programmable switched-capacitor filter
  block diagram representation, 17-29
  topology versatility, 17-29–17-30
  transfer function, 17-28–17-29
Programmable switched-capacitor oscillator
  BPF-based SC oscillator, 17-32–17-33
  finite-impulse-response (FIR) filter, 17-34
  four-level quasi-sinusoidal wave generator, 17-33–17-34
  high-quality-factor (*Q*-factor), 17-32–17-33
  linearity, 17-33
  measured frequency spectrum, 17-34–17-35
  sinewave oscillator, 17-32

Q

Q factors, *see* Quality factors
Quadrant fan filter, 23-9–23-10
Quadrature mirror filter (QMF), 25-12–25-16
Quality factors, 1-26
Quantization error
  coefficient quantization error
    alternate realization structure, 20-16–20-17
    clustered root, 20-16
    definition, 20-15
    grid, 20-15–20-16
    realizable pole locations, 20-16–20-17
    transfer function, 20-15
  fixed-point quantization error
    sum of products, 20-2–20-3
    types, 20-3–20-4
  floating-point quantization error, 20-4–20-5
Quasi-Newton algorithm, 23-29–23-30
Quincunx sampling matrix
  diamond and fan filter banks, 25-8
  nonsingular real matrix, 25-3–25-5
  quadrature mirror filter (QMF), 25-13–25-14

R

Raised-cosine window, *see* Hanning window
Rational function, 5-6–5-8
RC one-port networks
  continued-fraction expansion, 7-8–7-9
  first and second Cauer canonical form, 7-9
  first and second Foster canonical form, 7-8–7-10
  input impedance, 7-6–7-7
  RC impedance function, 7-7
Reactance function, 7-2–7-3, 7-5–7-6
Rectangular window
  causal and desired impulse response, 18-7
  definition, 18-7
  2-D FIR filter design, 22-9
  log–magnitude response, 18-8–18-9
  spectrum, 18-8
Relative coefficient sensitivity, 4-4
Relative sensitivity, 4-1–4-2
Remez exchange algorithm
  exhaustive step-by-step search, 18-21–18-23
  extremal initialization, 18-20–18-21
  iterative multivariable algorithm, 18-20
  maxima of error function, 18-21
  superfluous potential external rejection, 18-23–18-24
Resistively terminated network design
  Butterworth networks
    element values, 9-6–9-7
    input impedance, 9-5
    input reflection coefficient, 9-4–9-5
    LC ladder, 9-6
    minimum-phase solution, 9-5
    recurrence formula, 9-6
    transducer power gain, 9-4
  Chebyshev networks
    element values, 9-10–9-11
    input impedance, 9-10
    input reflection coefficient, 9-9
    minimum-phase solution, 9-9–9-10
    transducer power-gain, 9-9
  input and output reflection coefficients, 9-1
  maximum average power, 9-3
  Norton equivalent, 9-1–9-2
  sinusoidal steady state, 9-2
  Thévenin equivalent, 9-1–9-2
  transfer voltage ratio and impedance, 9-4
Richards function
  arbitrary positivereal constant, 6-2–6-3
  real nonnegative constant
    impedance matrix, 6-5–6-6
    index set, 6-4
    transmission matrix, 6-5
RLC load formulas
  Butterworth networks
    all-pass function, 10-15–10-16
    Butterworth transducer power-gain, 10-15
    equalizer back-end impedance, 10-16–10-17
    final reactive element, 10-17
    LC ladder network, 10-17–10-18
    load impedance, 10-14
    minimum-phase solution, 10-15
  Chebyshev networks
    all-pass function, 10-23
    equalizer back-end impedance, 10-24
    Laurent series expansion, 10-23
    $R C \omega_c \sinh a < 2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-25
    $R C \omega_c \sinh a < 2 \sin \gamma_1$ and $L_{b1} < L$, 10-25
    $R C \omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1} < L$, 10-24–10-25
    $R C \omega_c \sinh a \geq 2 \sin \gamma_1$ and $L_{b1} \geq L$, 10-24
    transducer power-gain, 10-22
RLCM one-port network, 5-1–5-2
Robust nonlinear estimators/filters
  additive noise extraction, 26-27

baseband communication, 26-28–26-29
highpass filtering, 26-29–26-31
power line communications (PLCs), 26-29–26-30
symmetric α-stable distributions, 26-28
Root sensitivity
pole sensitivity, 4-5–4-7
semirelative root sensitivities, 4-5
Rotation-invariant classification system, 25-25

S

Sallen and Key filter
bandpass circuit, 11-8–11-9, 13-2
high-pass filter, 11-9–11-10
low-pass filter, 11-6–11-8
Second-order filters
biquadratic filters
biquadratic transfer function, 11-11–11-12
RC network realization, 11-14–11-15
single-amplifier low-gain realization, 11-12–11-13
two-amplifier low-gain biquad configuration, 11-14
Butterworth LPF, 14-7–14-8
positive-gain filter
bandpass circuit, 11-8–11-9
high-pass filter, 11-9–11-10
low-pass filter, 11-6–11-8
notch filter circuits, 11-10–11-11
Semirelative coefficient sensitivity, 4-4–4-5
Sensivity and selectivity
active RC filters, 4-19, 4-30
bounds, 4-18–4-19
cascaded realization, active filters
CAD methods, 4-24
operational amplifier, 4-26
polynomial coefficient, 4-23
Q-factor, 4-26–4-28
stages, 4-25–4-26
coefficient sensitivity, 4-4–4-5
comparative errors, 4-30–4-31
computations, adjoint network
branch impedance and branch admittance matrices, 4-21–4-22
port variable separation, 4-21
Tellegen theorem, 4-20
transfer function calculation, 4-22
definition, 4-1–4-2
doubly terminated matched lossless filters
Fettweis-Orchard theorem, 4-29–4-30
high-*Q* transfer function poles, 4-28
steady-state sinusoidal operation, 4-28–4-29
transducer function, 4-29
function sensitivity, 4-3–4-4
invariants
filter function, 4-16
frequency scaling, 4-17–4-18
homogeneous function, 4-15
impedance scaling, 4-17
passive RC circuits, 4-17
scaling operation, 4-15–4-16
limitations, 4-19
multiparameter sensitivities and sensitivity measures
average sensitivity index, 4-9, 4-11
multivariable statistical measure, 4-14
quadratic sensitivity index, 4-10, 4-12–4-13
sensitivity row vector, 4-9–4-10
worst-case sensitivity index, 4-10, 4-12–4-13
reduction methods, 4-22–4-23
root sensitivities
pole sensitivity, 4-5–4-7
semirelative root sensitivities, 4-5
statistical model, 4-7–4-8
Separable product filters
half-plane symmetric filter, 23-14
prescribed specification, 23-15–23-16
quadrantally symmetric filter, 23-12–23-13
transfer function, 23-11–23-12
Sinewave generator, 17-31–17-33
Single-amplifier multiple-feedback filters
all-pole designs
equal-capacitor design, 12-5
low-pass structure, 12-3–12-4
transfer ratio, 12-4–12-5
design
finite amplifier gain effects, 12-8–12-9
sensitivity, 12-8
tuning function, 12-9–12-10
modified multiple-loop feedback (MMFB) structure
bandpass transfer function, 12-10–12-11
biquadratic structure, 12-12–12-13
structures
all-pole realization, 12-3–12-5
double ladder multiple-feedback network, 12-2–12-3
infinite-gain single-amplifier structure, 12-1–12-2
voltage transfer function, 12-2
Singular-value decomposition (SVD)
Antoniou and Lu method
amplitude response matrix, 23-35–23-36
error compensation procedure, 23-39
error correction section, 23-36–23-39
Deng and Kawamata method
design procedure, 23-41–23-43
2-D filter realization, 23-43
matrix decomposition, 23-41
problem formulation, 23-34–23-35
Sinusoidal response
*N*th-order analog filter, 1-11
steady-state sinusoidal response, 1-11–1-12
Sixth-order Chebyshev low-pass filter, 14-10–14-11
Sixth-order elliptic BP filter
2-OA CGIC biquad, 14-11–14-12

3-OA CGIC biquad, 14-16–14-17
Slew rate, 11-27
Spatially adaptive wavelet thresholding (SAWT) algorithm, 25-27
Spectral transformation
complex transformation, 23-7–23-8
Constantinides transformation
application, 23-11
1-D digital filter design, 23-9–23-10
2-D discrete transfer function, 23-10
low-pass transfer function, 23-10–23-11
linear transformation
algebraic structure, 23-5
frequency response, multiplication table, 23-5–23-6
product of functions, 23-6–23-7
zero-phase filter, 23-7
quadrant fan filter, 23-9–23-10
symmetric fan filter, 23-8–23-9
Standard video sequences
computational complexity, 27-30
piecewise planar approximation, 27-31
SNR results, 27-28–27-29
Step response, 1-8
SVD, *see* Singular-value decomposition
Switched-capacitor filters
analog switches, 17-19–17-21
biquadratic filter, 17-23–17-24
channel mobile charge, 17-17
clock feedthrough, 17-15–17-16
continuous-time filter, 17-1
dynamic range, 17-17–17-18
first-order blocks
offset and gain compensated integrator, 17-6
reduced capacitance spread, 17-5–17-6
standard stray-insensitive integrator, 17-4–17-5
transfer function, 17-5
fully differential filter
charge redistribution analysis, 17-23
folded cascode amplifier and CMFB circuit, 17-22–17-23
hold and integrate functions, 17-21
single-ended amplifier, 17-22
low-voltage operational amplifiers, 17-18–17-19
noise
1/*f* noise, 17-12
op-amp noise contribution, 17-13–17-14
switched-capacitor integrator, 17-14
switches, 17-14–17-15
thermal noise, 17-12
op-amp finite parameters, 17-10–17-11
programmable filter
block diagram representation, 17-29
topology versatility, 17-29–17-30
transfer function, 17-28–17-29
programmable oscillator
BPF-based SC oscillator, 17-32–17-33
finite-impulse-response (FIR) filter, 17-34
four-level quasi-sinusoidal wave generator, 17-33–17-34
high-quality-factor (*Q*-factor), 17-32–17-33
linearity, 17-33
measured frequency spectrum, 17-34–17-35
sinewave oscillator, 17-32
properties, 17-1–17-2
reduced area, 17-30–17-31
sixth-order bandpass ladder filter
floating LC tank circuit, 17-26
frequency response, 17-27–17-28
grounded LC tank circuit, 17-25
lossless discrete integrator (LDI) transformation, 17-27
passive LC tank, 17-26
RC active implementation, 17-25–17-26
RLC low-pass prototype, 17-24–17-25
spectrum analyzer, 17-31–17-33
switched-capacitor biquadratic section
capacitance spread, 17-7–17-8
decoupled structure., 17-8–17-9
high-sampling rate and high-*Q* condition, 17-9–17-10
ideal transfer function, 17-9
Q-sensitivity and total capacitance, 17-10
z-domain, 17-6–17-7
voltage gain amplifiers
clock phase, 17-3–17-4
leakage current, 17-2–17-3
open-loop dc gain, 17-4
z-domain transfer function, 17-3

T

Tarmy–Ghausi circuit, 13-25–13-26
Texas Instruments
ADSP-21020 chip architecture, 21-4–21-5
Motorola 56001 architecture, 21-6–21-7
TMS320C30 architecture, 21-3–21-4
TMS320C50 architecture, 21-5
TMS320C80 chip, 21-7
Texture analysis and segmentation
Bamberger pyramid, 25-25–25-26
classification system, 25-25
multichannel method, 25-24–25-25
Thévenin's theorem, 1-24
Third-order Butterworth transfer function, 2-11
Time-shifting theorem, 1-17
Tow–Thomas biquad, 16-7–16-8
Translation invariant (TI) wavelet denoising, 27-27
Transmission coefficient, 9-1–9-4
Two-channel undecimated checkerboard filter bank (UCFB), 25-21–25-22
Two-dimensional FIR filters

arbitrary specification
Chebyshev approximation, 22-6
frequency sampling and linear programming, 22-10–22-11
iterative method, approximate minimax design, 22-13–22-14
L_p norm, 22-11–22-12
window-based filter design, 22-6–22-10
filter banks, 22-25–22-26
frequency transformation method
Chebyshev polynomial, 22-17
frequency response, 22-17–22-21
nonlinear optimization problem, 22-21
zero-phase filter, 12-17
implementation, 22-24
nonrectangular transformations and sampling rate conversions
diamond filter transfer function, 22-22–22-23
half-band filter, 22-23
parallelogram-shaped passband region, 22-21, 22-22
prototype filter impulse response, 22-22
straight-line boundaries, 22-21
upsampling transformation and decimation, 22-22
preliminary design considerations
design techniques, 22-6
filter specification and approximation criteria, 22-3–22-4
linear shift-invariant system, 22-2
zero-phase FIR filter and symmetry consideration, 22-4–22-5
separable filter design, 22-16–22-17
Two-dimensional IIR filters
analog filter transformation
piecewise-constant amplitude response specification, 23-2
rotated filter, 23-3
two-variable reactance function, 23-3–23-5
discrete transfer function, 23-26
HP, BP, and BS filter realization configuration, 23-24–23-25
Nyquist point, 23-25
subfilter contour plot, 23-23–23-24
least *p*th optimization
elemental error function, 23-27
minimax algorithm, 23-31–23-33
problem formulation, 23-28–23-29
quasi-Newton algorithm, 23-29–23-30
linear digital filter
amplitude and phase response, 23-1–23-2
group delay function, 23-2
transfer function, 23-1–23-2
lowpass (LP) filter
1-D analog transfer function, 23-18–23-21
1-D discrete transfer function, 23-21–23-22
nonessential singularity elimination, 23-22–23-23
separable product filter design
half-plane symmetric filter, 23-14
prescribed specification, 23-15–23-16
quadrantally symmetric filter, 23-12–23-13
transfer function, 23-11–23-12
singular-value decomposition (SVD)
Antoniou and Lu method, 23-35–23-39
Deng and Kawamata method, 23-41–23-43
problem formulation, 23-34–23-35
spectral transformation
complex transformation, 23-7–23-8
Constantinides transformation, 23-9–23-11
linear transformation, 23-5–23-7
quadrant fan filter, 23-9–23-10
symmetric fan filter, 23-8–23-9
two-variable network theory design procedure, 23-44–23-45
input admittance, 23-44
$(1+N_1+N_2+N_r)$-port 2-V lossless network, 23-43–23-44
skew-symmetric matrix, 23-44
Two dimensional (2-D) two-channel fan filter bank
BDFB design, ladder structure
analysis–synthesis ladder structure, 25-17–25-18
biorthogonal property, 25-19–25-20
closed-form Lagrange formula, 25-18
digital fan-shaped filter, 25-17
magnitude response, 25-18–25-19
regular biorthogonal wavelet generation, 25-18
two-channel biorthogonal analysis filter bank, 25-19
wavelet construction, 25-16
FFB design, 1-D quadrature mirror filter (QMF)
change-of-variables scheme, 25-12–25-13
checkerboard magnitude response, 25-13–25-14
2-D polyphase filter, 25-14
FIR PR filter, 25-12
four directional filter, 25-15–25-16
image decomposition, 25-15, 25-17
polyphase representation, 25-13
two-channel analysis structure, 25-14
Two-port synthesis
LC ladder

zero-producing step, 8-4–8-5
zero-shifting step, 8-2–8-4
parallel/series ladders
Cauer canonical form, 8-14–8-15
open-circuit voltage ratio, 8-15–8-16
RC α- and β-ladder, 8-12–8-13
realization, 8-13–8-14
scale factors, 8-12
transfer admittance level, 8-13
two ladder networks, 8-11
RC ladder, 8-7–8-9
sigan transmission, 8-1–8-2
transfer current and voltage ratio, 8-1
Type-E section
Brune section
positive-real impedance, 6-9–6-10
two-port network, 6-8
type-A and type-B section, 6-8–6-9
Darlington type-C section, 6-7
impedance and transmission matrix, 6-6
Type-I Chebyshev filters, *see* Chebyshev filters
Type-II Chebyshev filters, *see* Inverse Chebyshev filters

U

Undecimated *á trous* wavelet transform, 27-20
Undecimated checkerboard filter bank (UCFB), 25-21–25-22
Undecimated discrete wavelet transform (UDWT), 25-20, 25-27–25-28
Undecimated fan filter bank (UFFB), 25-21–25-22

V

VCCS, *see* Voltage-controlled current sources
VCVS, *see* Voltage-controlled voltage sources
Video demosaicking filters
bicubic interpolation, 28-2, 28-4
camera identification and forgery detection, 28-19
compression, 28-18
constant-hue-based interpolation, 28-5–28-6
demosaicking definition, 28-1
denoising, 28-18
digital camera, 28-1
edge-directed interpolation
direction selection, 28-7–28-8
edge indicator, 28-7
green channel, 28-6–28-7
hard decision process, 28-6
evaluation, 28-19–28-20
frequency-domain approach
alias canceling interpolation, 28-10–28-11
filtering, 28-11–28-12
high correlation, 28-9–28-10
luminance channel sampling, 28-10
gradients
gain parameter, 28-8–28-9
graphical illustration, 28-9–28-10
green and blue gradients, 28-8
linear FIR filter coefficients, 28-9
linear interpolation, 28-7–28-8
red channel, 28-8
homogeneity-directed interpolation
block diagram, 28-13
definition, 28-13
luminance and chrominance values, 28-12–28-13
neighborhood parameter, 28-13–28-14
human visual system, 28-2–28-3
imaging model
cosine function, 28-4
full-color channel, 28-3
subsampling, 28-2
optimal spectral sensitivity function, 28-19
projections onto convex sets (POCS) technique, 28-14–28-15
single-channel interpolation method, 28-4
spectral response modeling, 28-15–28-16
super-resolution reconstruction
Bayer sampling, 28-16
CFA sampling pattern, 28-17
high-resolution image, 28-17–28-18
missing sample estimation, 28-16
multiframe interpolation, 28-16–82-17
regularization function, 28-18
VLSI implementation, digital filters
architectural approach, 21-1
computational structure, 21-1
design automation, 21-1–21-2
general-purpose processor
ADSP-21020 chip architecture, 21-4–21-5
digital equipment company (DEC) Alpha family, 21-8
fixed- and floating-point processor, 21-2
GNURadio, 21-7–21-8
historical perspective, 21-2
Intel Pentium family, 21-8
Motorola 56001 architecture, 21-6–21-7
TMS320C30 architecture, 21-3–21-4
TMS320C50 architecture, 21-5
TMS320C80 chip, 21-7
wireless personal communications system (PCS), 21-8
programmable logic implementation
application-specific integrated circuit (ASIC), 21-11
field programmable gate array (FPGA), 21-12
FIR filter, 21-12–21-13
IIR filter, 21-13–21-14
special-purpose implementation
bit-serial implementation, 21-9
extensive pipelining, 21-8
FIR filter, 21-9–21-11
IIR filter, 21-11
multiplier architecture, 21-9
parallelism, 21-8

Voltage-controlled current sources (VCCS), 4-18
Voltage-controlled voltage sources (VCVS), 4-18
Voltage doubler, 17-19–17-21
Voltage gain amplifiers
  clock phase, 17-3–17-4
  leakage current, 17-2–17-3
  open-loop dc gain, 17-4
  z-domain transfer function, 17-3
Voltage transfer function, 1-24
2-*V* strictly Hurwitz polynomial, 23-43–23-45

W

Weierstrass theorem, 2-25–2-26
Weighted Chebyshev ($\mathcal{L}\infty$-norm) criterion, 19-40
Weighted median (WM) filter, 26-9
Wideband filter, 18-53–18-54
Window-based filter design
  cutoff boundary, 22-9
  Hamming window, 22-8–22-9
  IIR sequence, 22-6–22-7
  impulse and frequency responses
    circularly symmetric low-pass filter, 22-8–22-9
    2-D filter, 22-9–22-10
    ideal filter, 22-7
  McClellan transformation, 22-7–22-8
  rectangular window, 22-9
  separable window, 22-7, 22-9–22-10
  surface sampling, 22-7

Z

Zero-mean white noise filter, 20-5
Zero of transmission
  binomial expansion formula, 10-4
  classification, 10-3–10-4
  definition, 10-2
  input impedance, 10-3
  Laurent series expansion, 10-4
Zero-phase filter
  causal and noncausal filters, 23-7
  1-D analog transfer function, 23-21
  singular-value decomposition, 23-38–23-39

Electrical Engineering

Upon its initial publication, *The Circuits and Filters Handbook* broke new ground. It quickly became **the** resource for comprehensive coverage of issues and practical information that can be put to immediate use. Not content to rest on his laurels, in addition to updating the second edition, editor Wai-Kai Chen divided it into tightly focused texts that made the information easily accessible and digestible. These texts have been revised, updated, and expanded so that they continue to provide solid coverage of standard practices and enlightened perspectives on new and emerging techniques.

Passive, Active, and Digital Filters provides an introduction to the characteristics of analog filters and a review of the design process and the tasks that need to be undertaken to translate a set of filter specifications into a working prototype. Highlights include discussions of the passive cascade synthesis and the synthesis of LCM and RC one-port networks; a summary of two-port synthesis by ladder development; a comparison of the cascade approach, the multiple-loop feedback topology, and ladder simulations; an examination of four types of finite wordlength effects; and coverage of methods for designing two-dimensional finite-extent impulse response (FIR) discrete-time filters. The book includes coverage of the basic building blocks involved in low- and high-order filters, limitations and practical design considerations, and a brief discussion of low-voltage circuit design.

Revised Chapters:

- Sensitivity and Selectivity
- Switched-Capacitor Filters
- FIR Filters
- IIR Filters
- VLSI Implementation of Digital Filters
- Two-Dimensional FIR Filters

New Chapters:

- 1-D Multirate Filter Banks
- Directional Filter Banks
- Nonlinear Filtering Using Statistical Signal Models
- Nonlinear Filtering for Image Denoising
- Video Demosaicking Filters

This volume will undoubtedly take its place as the engineer's first choice in looking for solutions to problems encountered when designing filters.

58851

ISBN: 978-1-4200-5885-7

90000

9 781420 058857

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

The Circuits and Filters Handbook

Third Edition

Fundamentals of Circuits and Filters

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

Fundamentals of Circuits and Filters

The Circuits and Filters Handbook

Third Edition

Edited by

Wai-Kai Chen

Fundamentals of Circuits and Filters

Feedback, Nonlinear, and Distributed Circuits

Analog and VLSI Circuits

Computer Aided Design and Design Automation

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Fundamentals of Circuits and Filters

Edited by

Wai-Kai Chen

University of Illinois

Chicago, U. S. A.

CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the

Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5887-1 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Fundamentals of circuits and filters / edited by Wai-Kai Chen.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-4200-5887-1
ISBN-10: 1-4200-5887-8
1. Electronic circuits. 2. Electric filters. I. Chen, Wai-Kai, 1936- II. Title.

TK7867.F835 2009
621.3815--dc22 2008048126

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface vii

Editor-in-Chief ix

Contributors xi

SECTION I Mathematics

1 Linear Operators and Matrices **1**-1
Cheryl B. Schrader and Michael K. Sain

2 Bilinear Operators and Matrices **2**-1
Michael K. Sain and Cheryl B. Schrader

3 Laplace Transformation **3**-1
John R. Deller, Jr.

4 Fourier Methods for Signal Analysis and Processing **4**-1
W. Kenneth Jenkins

5 *z*-Transform **5**-1
Jelena Kovačević

6 Wavelet Transforms **6**-1
P. P. Vaidyanathan and Igor Djokovic

7 Graph Theory **7**-1
Krishnaiyan Thulasiraman

8 Signal Flow Graphs **8**-1
Krishnaiyan Thulasiraman

9 Theory of Two-Dimensional Hurwitz Polynomials **9**-1
Hari C. Reddy

10 Application of Symmetry: Two-Dimensional Polynomials, Fourier Transforms, and Filter Design **10**-1
Hari C. Reddy, I-Hung Khoo, and P. K. Rajan

SECTION II Circuit Elements, Devices, and Their Models

11 Passive Circuit Elements **11**-1
Stanisław Nowak, Tomasz W. Postupolski, Gordon E. Carlson, and Bogdan M. Wilamowski

12 RF Passive IC Components **12**-1
Tomas H. Lee, Maria del Mar Hershenson, Sunderarajan S. Mohan, Hirad Samavati, and C. Patrick Yue

13 Circuit Elements, Modeling, and Equation Formulation **13**-1
Josef A. Nossek

14 Controlled Circuit Elements **14**-1
Edwin W. Greeneich and James F. Delansky

15 Bipolar Junction Transistor Amplifiers **15**-1
David J. Comer and Donald T. Comer

16 Operational Amplifiers **16**-1
David G. Nairn and Sergio B. Franco

17 High-Frequency Amplifiers **17**-1
Chris Toumazou and Alison Payne

SECTION III Linear Circuit Analysis

18 Fundamental Circuit Concepts **18**-1
John Choma, Jr.

19 Network Laws and Theorems **19**-1
Ray R. Chen, Artice M. Davis, and Marwan A. Simaan

20 Terminal and Port Representations **20**-1
James A. Svoboda

21 Signal Flow Graphs in Filter Analysis and Synthesis **21**-1
Pen-Min Lin

22 Analysis in the Frequency Domain **22**-1
Jiri Vlach and John Choma, Jr.

23 Tableau and Modified Nodal Formulations **23**-1
Jiri Vlach

24 Frequency-Domain Methods **24**-1
Peter B. Aronhime

25 Symbolic Analysis **25**-1
Benedykt S. Rodanski and Marwan M. Hassoun

26 Analysis in the Time Domain **26**-1
Robert W. Newcomb

27 State-Variable Techniques **27**-1
Kwong S. Chao

Index **IN-1**

Preface

The purpose of this book is to provide in a single volume a comprehensive reference work covering the broad spectrum of mathematics for circuits and filters; circuits elements, devices, and their models; and linear circuit analysis. This book is written and developed for the practicing electrical engineers in industry, government, and academia. The goal is to provide the most up-to-date information in the field.

Over the years, the fundamentals of the field have evolved to include a wide range of topics and a broad range of practice. To encompass such a wide range of knowledge, this book focuses on the key concepts, models, and equations that enable the design engineer to analyze, design, and predict the behavior of large-scale circuits. While design formulas and tables are listed, emphasis is placed on the key concepts and theories underlying the processes.

This book stresses fundamental theories behind professional applications and uses several examples to reinforce this point. Extensive development of theory and details of proofs have been omitted. The reader is assumed to have a certain degree of sophistication and experience. However, brief reviews of theories, principles, and mathematics of some subject areas are given. These reviews have been done concisely with perception.

The compilation of this book would not have been possible without the dedication and efforts of Professors Yih-Fang Huang and John Choma, Jr., and most of all the contributing authors. I wish to thank them all.

Wai-Kai Chen

Editor-in-Chief



Wai-Kai Chen is a professor and head emeritus of the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. He received his BS and MS in electrical engineering at Ohio University, where he was later recognized as a distinguished professor. He earned his PhD in electrical engineering at the University of Illinois at Urbana–Champaign.

Professor Chen has extensive experience in education and industry and is very active professionally in the fields of circuits and systems. He has served as a visiting professor at Purdue University, the University of Hawaii at Manoa, and Chuo University in Tokyo, Japan. He was the editor-in-chief of the *IEEE Transactions on Circuits and Systems, Series I and II*, the president of the IEEE Circuits and Systems Society, and is the founding editor and the editor-in-chief of the *Journal of Circuits, Systems and Computers*. He received the Lester R. Ford Award from the Mathematical Association of America; the Alexander von Humboldt Award from Germany; the JSPS Fellowship Award from the Japan Society for the Promotion of Science; the National Taipei University of Science and Technology Distinguished Alumnus Award; the Ohio University Alumni Medal of Merit for Distinguished Achievement in Engineering Education; the Senior University Scholar Award and the 2000 Faculty Research Award from the University of Illinois at Chicago; and the Distinguished Alumnus Award from the University of Illinois at Urbana–Champaign. He is the recipient of the Golden Jubilee Medal, the Education Award, and the Meritorious Service Award from the IEEE Circuits and Systems Society, and the Third Millennium Medal from the IEEE. He has also received more than a dozen honorary professorship awards from major institutions in Taiwan and China.

A fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the American Association for the Advancement of Science (AAAS), Professor Chen is widely known in the profession for the following works: *Applied Graph Theory* (North-Holland), *Theory and Design of Broadband Matching Networks* (Pergamon Press), *Active Network and Feedback Amplifier Theory* (McGraw-Hill), *Linear Networks and Systems* (Brooks/Cole), *Passive and Active Filters: Theory and Implements* (John Wiley), *Theory of Nets: Flows in Networks* (Wiley-Interscience), *The Electrical Engineering Handbook* (Academic Press), and *The VLSI Handbook* (CRC Press).

Contributors

Peter B. Aronhime
Electrical and Computer Engineering Department
University of Louisville
Louisville, Kentucky

Gordon E. Carlson
Department of Electrical and Computer Engineering
University of Missouri–Rolla
Rolla, Missouri

Kwong S. Chao
Department of Electrical and Computer Engineering
Texas Tech University
Lubbock, Texas

Ray R. Chen
Department of Electrical Engineering
San Jose State University
San Jose, California

Wai-Kai Chen
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, Illinois

John Choma, Jr.
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

David J. Comer
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah

Donald T. Comer
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah

Artice M. Davis
Department of Electrical Engineering
San Jose State University
San Jose, California

James F. Delansky
Department of Electrical Engineering
Pennsylvania State University
University Park, Pennsylvania

John R. Deller, Jr.
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan

Igor Djokovic
PairGain Technologies
Tustin, California

Sergio B. Franco
Division of Engineering
San Francisco State University
San Francisco, California

Edwin W. Greeneich
Department of Electrical Engineering
Arizona State University
Tempe, Arizona

Marwan M. Hassoun
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa

Maria del Mar Hershenson
Center for Integrated Systems
Stanford University
Stanford, California

Yih-Fang Huang
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana

W. Kenneth Jenkins
Department of Electrical Engineering
Pennsylvania State University
University Park, Pennsylvania

I-Hung Khoo
Department of Electrical Engineering
California State University
Long Beach, California

Jelena Kovačević
AT&T Bell Laboratories
Murray Hill, New Jersey

Tomas H. Lee
Center for Integrated Systems
Stanford University
Stanford, California

Pen-Min Lin
School of Electrical Engineering
Purdue University
West Lafayette, Indiana

Sunderarajan S. Mohan
Center for Integrated Systems
Stanford University
Stanford, California

David G. Nairn
Department of Electrical Engineering
Queen's University
Kingston, Canada

Robert W. Newcomb
Electrical Engineering Department
University of Maryland
College Park, Maryland

Josef A. Nossek
Institute for Circuit Theory and Signal Processing
Technical University of Munich
Munich, Germany

Stanisław Nowak
Institute of Electronics
University of Mining and Metallurgy
Krakow, Poland

Alison Payne
Institute of Biomedical Engineering
Imperial College of Science, Technology and Medicine
London, England

Tomasz W. Postupolski
Institute of Electronic Materials Technology
Warsaw, Poland

P. K. Rajan
Department of Electrical and Computer Engineering
Tennessee Tech University
Cookeville, Tennessee

Hari C. Reddy
Department of Electrical Engineering
California State University
Long Beach, California

and

Department of Computer Science/Electrical and Control Engineering
National Chiao-Tung University, Taiwan

Benedykt S. Rodanski
Faculty of Engineering
University of Technology, Sydney
Sydney, New South Wales, Australia

Michael K. Sain
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana

Hirad Samavati
Center for Integrated Systems
Stanford University
Stanford, California

Cheryl B. Schrader
College of Engineering
Boise State University
Boise, Idaho

Marwan A. Simaan
Department of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, Pennsylvania

James A. Svoboda
Department of Electrical Engineering
Clarkson University
Potsdam, New York

Krishnaiyan Thulasiraman
School of Computer Science
University of Oklahoma
Norman, Oklahoma

Chris Toumazou
Institute of Biomedical Engineering
Imperial College of Science, Technology and Medicine
London, England

P. P. Vaidyanathan
Department of Electrical Engineering
California Institute of Technology
Pasadena, California

Jiri Vlach
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada

Bogdan M. Wilamowski
Alabama Nano/Micro Science and Technology Center
Department of Electrical and Computer Engineering
Auburn University
Auburn, Alabama

C. Patrick Yue
Center for Integrated Systems
Stanford University
Stanford, California

I

Mathematics

Yih-Fang Huang
University of Notre Dame

1 Linear Operators and Matrices *Cheryl B. Schrader and Michael K. Sain* **1**-1
Introduction • Vector Spaces over Fields • Linear Operators and Matrix Representations • Matrix Operations • Determinant, Inverse, and Rank • Basis Transformations • Characteristics: Eigenvalues, Eigenvectors, and Singular Values • On Linear Systems • References

2 Bilinear Operators and Matrices *Michael K. Sain and Cheryl B. Schrader* **2**-1
Introduction • Algebras • Bilinear Operators • Tensor Product • Basis Tensors • Multiple Products • Determinants • Skew Symmetric Products • Solving Linear Equations • Symmetric Products • Summary • Reference

3 Laplace Transformation *John R. Deller, Jr.* **3**-1
Introduction • Motivational Example • Formal Developments • Laplace Transform Analysis of Linear Systems • Conclusions and Further Readings • Appendix A: The Dirac Delta (Impulse) Function • Appendix B: Relationships among the Laplace, Fourier, and z-Transforms • References

4 Fourier Methods for Signal Analysis and Processing *W. Kenneth Jenkins* **4**-1
Introduction • Classical Fourier Transform for Continuous-Time Signals • Fourier Series Representation of Continuous-Time Periodic Signals • Discrete-Time Fourier Transform • Discrete Fourier Transform • Family Tree of Fourier Transforms • Selected Applications of Fourier Methods • Summary • References

5 z-Transform *Jelena Kovačević* **5**-1
Introduction • Definition of the z-Transform • Inverse z-Transform • Properties of the z-Transform • Role of the z-Transform in Linear Time-Invariant Systems • Variations on the z-Transform • Concluding Remarks • Acknowledgments • References

6 Wavelet Transforms *P. P. Vaidyanathan and Igor Djokovic* **6**-1
Introduction • Signal Representation Using Basis Functions • Short-Time Fourier Transform • Digital Filter Banks and Subband Coders • Deeper Study of Wavelets, Filter Banks, and Short-Time Fourier Transforms • Space of L^1 and L^2 Signals • Riesz Basis, Biorthogonality, and Other Fine Points • Frames in Hilbert Spaces • Short-Time Fourier Transform: Invertibility, Orthonormality, and Localization • Wavelets and Multiresolution • Orthonormal Wavelet Basis from Paraunitary Filter Banks • Compactly Supported Orthonormal Wavelets • Wavelet Regularity • Concluding Remarks • Acknowledgments • References

7 Graph Theory *Krishnaiyan Thulasiraman* **7**-1
Introduction • Basic Concepts • Cuts, Circuits, and Orthogonality • Incidence, Circuit, and Cut Matrices of a Graph • Orthogonality Relation and Ranks of Circuit and Cut Matrices • Spanning Tree Enumeration • Graphs and Electrical Networks • Tellegen's Theorem and Network Sensitivity Computation • Arc Coloring Theorem and the No-Gain Property • References

8 Signal Flow Graphs *Krishnaiyan Thulasiraman* **8**-1
Introduction • Adjacency Matrix of a Directed Graph • Coates' Gain Formula • Mason's Gain Formula • References

9 Theory of Two-Dimensional Hurwitz Polynomials *Hari C. Reddy* **9**-1
Introduction • Preliminaries and Notations • Value of a Two-Variable Polynomial at Infinity • Various Analog Hurwitz Polynomials • Testsets for Analog Hurwitz Polynomials • Two-Variable Very Strict Hurwitz Polynomials • Application of Two-Dimensional Hurwitz Polynomials for Two-Variable Passive Networks and Stability • Conclusion • Acknowledgments • References

10 Application of Symmetry: Two-Dimensional Polynomials, Fourier Transforms, and Filter Design *Hari C. Reddy, I-Hung Khoo, and P. K. Rajan* **10**-1
Introduction • Basic Symmetry Definitions • Two-Dimensional Fourier Transform Pairs with Symmetry • Symmetry and 2-D Fast Fourier Transform • Symmetry in 2-D Magnitude Response • Polynomial Operations and Definitions • Spectral Forms for Magnitude-Squared Function • Determination of 2-D Polynomials Possessing Various Symmetries • Symmetry and Stability • Filter Design Procedure • Filter Design Examples • Appendix: MATLAB Programs for Filter Design • References

1

Linear Operators and Matrices

1.1 Introduction **1**-1
1.2 Vector Spaces over Fields **1**-2
1.3 Linear Operators and Matrix Representations **1**-4
1.4 Matrix Operations **1**-6
1.5 Determinant, Inverse, and Rank **1**-8
1.6 Basis Transformations **1**-12
1.7 Characteristics: Eigenvalues, Eigenvectors, and Singular Values **1**-15
1.8 On Linear Systems **1**-18
References **1**-20

Cheryl B. Schrader
Boise State University

Michael K. Sain
University of Notre Dame

1.1 Introduction

It is only after the engineer masters' linear concepts—linear models and circuit and filter theory—that the possibility of tackling nonlinear ideas becomes achievable. Students frequently encounter linear methodologies, and bits and pieces of mathematics that aid in problem solution are stored away. Unfortunately, in memorizing the process of finding the inverse of a matrix or of solving a system of equations, the essence of the problem or associated knowledge may be lost. For example, most engineers are fairly comfortable with the concept of a vector space, but have difficulty in generalizing these ideas to the module level. Therefore, the intention of this section is to provide a unified view of key concepts in the theory of linear circuits and filters, to emphasize interrelated concepts, to provide a mathematical reference to the handbook itself, and to illustrate methodologies through the use of many and varied examples.

This chapter begins with a basic examination of vector spaces over fields. In relating vector spaces, the key ideas of linear operators and matrix representations come to the fore. Standard matrix operations are examined as are the pivotal notions of determinant, inverse, and rank. Next, transformations are shown to determine similar representations, and matrix characteristics such as singular values and eigenvalues are defined. Finally, solutions to algebraic equations are presented in the context of matrices and are related to this introductory chapter on mathematics as a whole.

Standard algebraic notation is introduced first. To denote an element s in a set S, use $s \in S$. Consider two sets S and T. The set of all ordered pairs (s, t) where $s \in S$ and $t \in T$ is defined as the Cartesian product set $S \times T$. A function f from S into T, denoted by $f : S \rightarrow T$, is a subset U of ordered pairs $(s, t) \in S \times T$ such that for every $s \in S$, one and only one $t \in T$ exists such that $(s, t) \in U$. The function evaluated at the element s gives t as a solution ($f(s) = t$), and each $s \in S$ as a first element in U appears exactly once.

A binary operation is a function acting on a Cartesian product set $S \times T$. When $T = S$, one speaks of a binary operation on S.

1.2 Vector Spaces over Fields

A field F is a nonempty set F and two binary operations, sum ($+$) and product, such that the following properties are satisfied for all $a, b, c \in F$:

1. Associativity: $(a + b) + c = a + (b + c)$; $(ab)c = a(bc)$
2. Commutativity: $a + b = b + a$; $ab = ba$
3. Distributivity: $a(b + c) = (ab) + (ac)$
4. Identities: (Additive) $0 \in F$ exists such that $a + 0 = a$
   (Multiplicative) $1 \in F$ exists such that $a1 = a$
5. Inverses: (Additive) For every $a \in F$, $b \in F$ exists such that $a + b = 0$
   (Multiplicative) For every nonzero $a \in F$, $b \in F$ exists such that $ab = 1$

Examples

- Field of real numbers R
- Field of complex numbers C
- Field of rational functions with real coefficients $R(s)$
- Field of binary numbers

The set of integers Z with the standard notions of addition and multiplication does not form a field because a multiplicative inverse in Z exists only for ± 1. The integers form a commutative ring. Likewise, polynomials in the indeterminate s with coefficients from F form a commutative ring $F[s]$. If field property 2 also is not available, then one speaks simply of a ring. An additive group is a nonempty set G and one binary operation $+$ satisfying field properties 1, 4, and 5 for addition, that is, associativity and the existence of additive identity and inverse. Moreover, if the binary operation $+$ is commutative (field property 2), then the additive group is said to be abelian. Common notation regarding inverses is that the additive inverse for $a \in F$ is $b = -a \in F$. In the multiplicative case $b = a^{-1} \in F$.

An F-vector space V is a nonempty set V and a field F together with binary operations $+$: $V \times V \to V$ and $\star$: $F \times V \to V$ subject to the following axioms for all elements $v, w \in V$ and $a, b \in F$:

1. V and $+$ form an additive abelian group
2. $a \star (v+w) = (a \star v)+(a \star w)$
3. $(a+b) \star v = (a \star v)+(b \star v)$
4. $(ab) \star v = a \star (b \star v)$
5. $1 \star v = v$

Examples

- Set of all n-tuples $(v_1, v_2, \ldots, v_n)$ for $n > 0$ and $v_i \in F$
- Set of polynomials of degree less than n with real coefficients ($F = R$)

Elements of V are referred to as vectors, whereas elements of F are scalars. Note that the terminology vector space V over the field F is used often. A module differs from a vector space in only one aspect; the underlying field in a vector space is replaced by a ring. Thus, a module is a direct generalization of a vector space.

When considering vector spaces of n-tuples, $+$ is vector addition defined element by element using the scalar addition associated with F. Multiplication ($\star$), which is termed scalar multiplication, also is defined

element by element using multiplication in F. The additive identity in this case is the zero vector (n-tuple of zeros) or null vector, and F^n denotes the set of n-tuples with elements in F, a vector space over F. A nonempty subset $\widetilde{V} \subset V$ is called a subspace of V if for each v, $w \in \widetilde{V}$ and every $a \in F$, $v + w \in \widetilde{V}$, and $a * v \in \widetilde{V}$. When the context makes things clear, it is customary to suppress the $*$, and write av in place of $a * v$.

A set of vectors $\{v_1, v_2, \ldots, v_m\}$ belonging to an F-vector space V is said to span the vector space if any element $v \in V$ can be represented by a linear combination of the vectors v_i. That is, scalars $a_1, a_2, \ldots, a_m \in F$ are such that

$$v = a_1 v_1 + a_2 v_2 + \cdots + a_m v_m \tag{1.1}$$

A set of vectors $\{v_1, v_2, \ldots, v_p\}$ belonging to an F-vector space V is said to be linearly dependent over F if scalars $a_1, a_2, \ldots, a_p \in F$, not all zero, exist such that

$$a_1 v_1 + a_2 v_2 + \cdots + a_p v_p = 0 \tag{1.2}$$

If the only solution for Equation 1.2 is that all $a_i = 0 \in F$, then the set of vectors is said to be linearly independent.

Examples

- (1, 0) and (0, 1) are linearly independent.
- (1, 0, 0), (0, 1, 0), and (1, 1, 0) are linearly dependent over R. To see this, simply choose $a_1 = a_2 = 1$ and $a_3 = -1$.
- $s^2 + 2s$ and $2s + 4$ are linearly independent over R, but are linearly dependent over $R(s)$ by choosing $a_1 = -2$ and $a_2 = s$.

A set of vectors $\{v_1, v_2, \ldots, v_n\}$ belonging to an F-vector space V is said to form a basis for V if it both spans V and is linearly independent over F. The number of vectors in a basis is called the dimension of the vector space, and is denoted as $\dim(V)$. If this number is not finite, then the vector space is said to be infinite dimensional.

Examples

- In an n-dimensional vector space, any n linearly independent vectors form a basis.
- Natural (standard) basis

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \ldots, \quad e_{n-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}, \quad e_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

both spans F^n and is linearly independent over F.

Consider any basis $\{v_1, v_2, \ldots, v_n\}$ in an n-dimensional vector space. Every $v \in V$ can be represented uniquely by scalars $a_1, a_2, \ldots, a_n \in F$ as

$$v = a_1 v_1 + a_2 v_2 + \cdots + a_n v_n \tag{1.3}$$

$$= [v_1 \ v_2 \ \cdots \ v_n] \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{1.4}$$

$$= [v_1 \ v_2 \ \cdots \ v_n] a \tag{1.5}$$

Here, $a \in F^n$ is a coordinate representation of $v \in V$ with respect to the chosen basis. The reader will be able to discern that each choice of basis will result in another representation of the vector under consideration. Of course, in the applications, some representations are more popular and useful than others.

1.3 Linear Operators and Matrix Representations

First, recall the definition of a function f: $S \rightarrow T$. Alternate terminology for a function is mapping, operator, or transformation. The set S is called the domain of f, denoted by $D(f)$. The range of f, $R(f)$, is the set of all $t \in T$ such that $(s, t) \in U$ ($f(s) = t$) for some $s \in D(f)$.

Examples

Use $S = \{1, 2, 3, 4\}$ and $T = \{5, 6, 7, 8\}$.

- $\widetilde{U} = \{(1, 5), (2, 5), (3, 7), (4, 8)\}$ is a function. The domain is {1, 2, 3, 4} and the range is {5, 7, 8}.
- $\hat{U} = \{(1, 5), (1, 6), (2, 5), (3, 7), (4, 8)\}$ is not a function.
- $\overline{U} = \{(1, 5), (2, 6), (3, 7), (4, 8)\}$ is a function. The domain is {1, 2, 3, 4} and the range is {5, 6, 7, 8}.

If $R(f) = T$, then f is said to be surjective (onto). Loosely speaking, all elements in T are used up. If $f: S \rightarrow T$ has the property that $f(s_1) = f(s_2)$ implies $s_1 = s_2$, then f is said to be injective (one-to-one). This means that any element in $R(f)$ comes from a unique element in $D(f)$ under the action of f. If a function is both injective and surjective, then it is said to be bijective (one-to-one and onto).

Examples

- $\widetilde{U}$ is not onto because $6 \in T$ is not in $R(f)$. Also $\widetilde{U}$ is not one-to-one because $f(1) = 5 = f(2)$, but $1 \neq 2$.
- $\overline{U}$ is bijective.

Now consider an operator L: $V \rightarrow W$, where V and W are vector spaces over the same field F. L is said to be a linear operator if the following two properties are satisfied for all v, $w \in V$ and for all $a \in F$:

$$L(av) = aL(v) \tag{1.6}$$

$$L(v + w) = L(v) + L(w) \tag{1.7}$$

Equation 1.6 is the property of homogeneity and Equation 1.7 is the property of additivity. Together they imply the principle of superposition, which may be written as

$$L(a_1 v_1 + a_2 v_2) = a_1 L(v_1) + a_2 L(v_2) \tag{1.8}$$

for all v_1, $v_2 \in V$ and a_1, $a_2 \in F$. If Equation 1.8 is not satisfied, then L is called a nonlinear operator.

Examples

- Consider $V = C$ and $F = C$. Let $L: V \to V$ be the operator that takes the complex conjugate: $L(v) = \bar{v}$ for $v, \bar{v} \in V$. Certainly

$$L(v_1 + v_2) = \overline{v_1 + v_2} = \overline{v_1} + \overline{v_2} = L(v_1) + L(v_2)$$

  However,

$$L(a_1 v_1) = \overline{a_1 v_1} = \overline{a_1} L(v_1) \neq a_1 L(v_1)$$

  Then L is a nonlinear operator because homogeneity fails.

- For F-vector spaces V and W, let V be F^n and W be F^{n-1}. Examine $L: V \to W$, the operator that truncates the last element of the n-tuple in V, that is,

$$L((v_1, v_2, \ldots, v_{n-1}, v_n)) = (v_1, v_2, \ldots, v_{n-1})$$

  Such an operator is linear.

The null space (kernel) of a linear operator $L: V \to W$ is the set

$$\ker L = \{v \in V \text{ such that } L(v) = 0\} \tag{1.9}$$

Equation 1.9 defines a vector space. In fact, ker L is a subspace of V. The mapping L is injective if and only if ker $L = 0$; that is, the only solution in the right member of Equation 1.9 is the trivial solution. In this case, L is also called monic.

The image of a linear operator $L: V \to W$ is the set

$$\text{im } L = \{w \in W \text{ such that } L(v) = w \text{ for some } v \in V\} \tag{1.10}$$

Clearly, im L is a subspace of W, and L is surjective if and only if im L is all of W. In this case, L is also called epic.

A method of relating specific properties of linear mappings is the exact sequence. Consider a sequence of linear mappings

$$\cdots V \xrightarrow{L} W \xrightarrow{\tilde{L}} U \to \cdots \tag{1.11}$$

This sequence is said to be exact at W if im $L = \ker \tilde{L}$. A sequence is called exact if it is exact at each vector space in the sequence. Examine the following special cases:

$$0 \to V \xrightarrow{L} W \tag{1.12}$$

$$W \xrightarrow{\tilde{L}} U \to 0 \tag{1.13}$$

Sequence (Equation 1.12) is exact if and only if L is monic, whereas Equation 1.13 is exact if and only if $\tilde{L}$ is epic.

Further, let $L: V \to W$ be a linear mapping between finite-dimensional vector spaces. The rank of L, $\rho(L)$, is the dimension of the image of L. In such a case

$$\rho(L) + \dim(\ker L) = \dim V \tag{1.14}$$

Linear operators commonly are represented by matrices. It is quite natural to interchange these two ideas, because a matrix with respect to the standard bases is indistinguishable from the linear operator it represents. However, insight may be gained by examining these ideas separately. For V and W, n- and m-dimensional vector spaces over F, respectively, consider a linear operator L: $V \rightarrow W$. Moreover, let $\{v_1, v_2, \ldots, v_n\}$ and $\{w_1, w_2, \ldots, w_m\}$ be respective bases for V and W. Then L: $V \rightarrow W$ can be represented uniquely by the matrix $M \in F^{m \times n}$ where

$$M = \begin{bmatrix} m_{11} & m_{12} & \ldots & m_{1n} \\ m_{21} & m_{22} & \ldots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \ldots & m_{mn} \end{bmatrix} \tag{1.15}$$

The ith column of M is the representation of $L(v_i)$ with respect to $\{w_1, w_2, \ldots, w_m\}$. Element $m_{ij} \in F$ of Equation 1.15 occurs in row i and column j.

Matrices have a number of properties. A matrix is said to be square if $m = n$. The main diagonal of a square matrix consists of the elements m_{ii}. If $m_{ij} = 0$ for all $i > j$ ($i < j$), a square matrix is said to be upper (lower) triangular. A square matrix with $m_{ij} = 0$ for all $i \neq j$ is diagonal. Additionally, if all $m_{ii} = 1$, a diagonal M is an identity matrix. A row vector (column vector) is a special case in which $m = 1$ ($n = 1$). Also, $m = n = 1$ results essentially in a scalar.

Matrices arise naturally as a means to represent sets of simultaneous linear equations. For example, in the case of Kirchhoff equations, Chapter 7 shows how incidence, circuit, and cut matrices arise. Or consider a π network having node voltages v_i, $i = 1, 2$ and current sources i_i, $i = 1, 2$ connected across the resistors R_i, $i = 1, 2$ in the two legs of the π. The bridge resistor is R_3. Thus, the unknown node voltages can be expressed in terms of the known source currents in the manner

$$\frac{(R_1 + R_3)}{R_1 R_3} v_1 - \frac{1}{R_3} v_2 = i_1 \tag{1.16}$$

$$\frac{(R_2 + R_3)}{R_2 R_3} v_2 - \frac{1}{R_3} v_1 = i_2 \tag{1.17}$$

If the voltages, v_i, and the currents, i_i, are placed into a voltage vector $v \in R^2$ and current vector $i \in R^2$, respectively, then Equations 1.16 and 1.17 may be rewritten in matrix form as

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} \frac{(R_1+R_3)}{R_1 R_3} & -\frac{1}{R_3} \\ -\frac{1}{R_3} & \frac{(R_2+R_3)}{R_2 R_3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{1.18}$$

A conductance matrix G may then be defined so that $i = Gv$, a concise representation of the original pair of circuit equations.

1.4 Matrix Operations

Vector addition in F^n was defined previously as an element-wise scalar addition. Similarly, two matrices both M and N in $F^{m \times n}$ can be added (subtracted) to form the resultant matrix $P \in F^{m \times n}$ by

$$m_{ij} \pm n_{ij} = p_{ij}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n \tag{1.19}$$

Matrix addition, thus, is defined using addition in the field over which the matrix lies. Accordingly, the matrix, each of whose entries is $0 \in F$, is an additive identity for the family. One can set up additive inverses along similar lines, which, of course, turn out to be the matrices each of whose elements is the negative of that of the original matrix.

Recall how scalar multiplication was defined in the example of the vector space of n-tuples. Scalar multiplication can also be defined between a field element $a \in F$ and a matrix $M \in F^{m\times n}$ in such a way that the product aM is calculated element-wise:

$$aM = P \Leftrightarrow am_{ij} = p_{ij}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n \tag{1.20}$$

Examples

$$(F = R)\text{: } M = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 2 & -3 \\ 1 & 6 \end{bmatrix}, \quad a = -0.5$$

- $M + N = P = \begin{bmatrix} 4+2 & 3-3 \\ 2+1 & 1+6 \end{bmatrix} = \begin{bmatrix} 6 & 0 \\ 3 & 7 \end{bmatrix}$
- $M - N = \tilde{P} = \begin{bmatrix} 4-2 & 3+3 \\ 2-1 & 1-6 \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ 1 & -5 \end{bmatrix}$
- $aM = \hat{P} = \begin{bmatrix} (-0.5)4 & (-0.5)3 \\ (-0.5)2 & (-0.5)1 \end{bmatrix} = \begin{bmatrix} -2 & -1.5 \\ -1 & -0.5 \end{bmatrix}$

To multiply two matrices M and N to form the product MN requires that the number of columns of M equal the number of rows of N. In this case the matrices are said to be conformable. Although vector multiplication cannot be defined here because of this constraint, Chapter 2 examines this operation in detail using the tensor product. The focus here is on matrix multiplication. The resulting matrix will have its number of rows equal to the number of rows in M and its number of columns equal to the number of columns of N. Thus, for $M \in F^{m\times n}$ and $N \in F^{n\times p}$, $MN = P \in F^{m\times p}$. Elements in the resulting matrix P may be determined by

$$p_{ij} = \sum_{k=l}^{n} m_{ik} n_{kj} \tag{1.21}$$

Matrix multiplication involves one row and one column at a time. To compute the p_{ij} term in P, choose the ith row of M and the jth column of N. Multiply each element in the row vector by the corresponding element in the column vector and sum the result. Notice that in general, matrix multiplication is not commutative, and the matrices in the reverse order may not even be conformable. Matrix multiplication is, however, associative and distributive with respect to matrix addition. Under certain conditions, the field F of scalars, the set of matrices over F, and these three operations combine to form an algebra. Chapter 2 examines algebras in greater detail.

Examples

$$(F = R)\text{: } M = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

- $MN = P = \begin{bmatrix} 10 & 24 & 38 \\ 4 & 10 & 16 \end{bmatrix}$

  To find p_{11}, take the first row of M, [4 3], and the first column of N, $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$, and evaluate Equation 1.21: $4(1) + 3(2) = 10$. Continue for all i and j.

- NM does not exist because that product is not comformable.
- Any matrix $M \in F^{m\times n}$ multiplied by an identity matrix $I \in F^{n\times n}$ such that $MI \in F^{m\times n}$ results in the original matrix M. Similarly, $IM = M$ for I an $m \times m$ identity matrix over F. It is common to interpret I as an identity matrix of appropriate size, without explicitly denoting the number of its rows and columns.

The transpose $M^{\mathrm{T}} \in F^{n\times m}$ of a matrix $M \in F^{m\times n}$ is found by interchanging the rows and columns. The first column of M becomes the first row of M^{T}, the second column of M becomes the second row of M^{T}, and so on. The notations M^{T} and M' are also used. If $M = M^{\mathrm{T}}$ the matrix is called symmetric. Note that two matrices $M, N \in F^{m\times n}$ are equal if and only if all respective elements are equal: $m_{ij} = n_{ij}$ for all i, j. The Hermitian transpose $M_\star \in C^{n\times m}$ of $M \in C^{m\times n}$ is also termed the complex conjugate transpose. To compute $M_\star$, form M^{T} and take the complex conjugate of every element in M^{T}. The following properties also hold for matrix transposition for all $M, N \in F^{m\times n}$, $P \in F^{n\times p}$, and $a \in F$: $(M^{\mathrm{T}})^{\mathrm{T}} = M$, $(M+N)^{\mathrm{T}} = M^{\mathrm{T}} + N^{\mathrm{T}}$, $(aM)^{\mathrm{T}} = aM^{\mathrm{T}}$, and $(MP)^{\mathrm{T}} = P^{\mathrm{T}}\, M^{\mathrm{T}}$.

Examples

$$(F = C):\ M = \begin{bmatrix} j & 1-j \\ 4 & 2+j3 \end{bmatrix}$$

- $M^{\mathrm{T}} = \begin{bmatrix} j & 4 \\ 1-j & 2+j3 \end{bmatrix}$
- $M^{*} = \begin{bmatrix} -j & 4 \\ 1+j & 2-j3 \end{bmatrix}$

1.5 Determinant, Inverse, and Rank

Consider square matrices of the form $[m_{11}] \in F^{1\times 1}$. For these matrices, define the determinant as m_{11} and establish the notation $\det([m_{11}])$ for this construction. This definition can be used to establish the meaning of $\det(M)$, often denoted by $|M|$, for $M \in F^{2\times 2}$. Consider

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{1.22}$$

The minor of m_{ij} is defined to be the determinant of the submatrix which results from the removal of row i and column j. Thus, the minors of m_{11}, m_{12}, m_{21}, and m_{22} are m_{22}, m_{21}, m_{12}, and m_{11}, respectively. To calculate the determinant of this M, (1) choose any row i (or column j), (2) multiply each element m_{ik} (or m_{kj}) in that row (or column) by its minor and by $(-1)^{i+k}$ (or $(-1)^{k+j}$), and (3) add these results. Note that the product of the minor with the sign $(-1)^{i+k}$ (or $(-1)^{k+j}$) is called the cofactor of the element in question. If row 1 is chosen, the determinant of M is found to be $m_{11}(+m_{22}) + m_{12}(-m_{21})$, a well-known result. The determinant of 2×2 matrices is relatively easy to remember: multiply the two elements along the main diagonal and subtract the product of the other two elements. Note that it makes no difference which row or column is chosen in step 1.

A similar procedure is followed for larger matrices. Consider

$$\det(M) = \begin{vmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{vmatrix} \tag{1.23}$$

Expanding about column 1 produces

$$\det(M) = m_{11}\begin{vmatrix} m_{22} & m_{23} \\ m_{32} & m_{33} \end{vmatrix} - m_{21}\begin{vmatrix} m_{12} & m_{13} \\ m_{32} & m_{33} \end{vmatrix} + m_{31}\begin{vmatrix} m_{12} & m_{13} \\ m_{22} & m_{23} \end{vmatrix} \tag{1.24}$$

$$= m_{11}(m_{22}m_{33} - m_{23}m_{32}) - m_{21}(m_{12}m_{33} - m_{13}m_{32}) + m_{31}(m_{12}m_{23} - m_{13}m_{22}) \tag{1.25}$$

$$= m_{11}m_{22}m_{33} + m_{12}m_{23}m_{31} + m_{13}m_{21}m_{32} - m_{13}m_{22}m_{31} - m_{11}m_{23}m_{32} - m_{12}m_{21}m_{33} \tag{1.26}$$

An identical result may be achieved by repeating the first two columns next to the original matrix:

$$\begin{matrix} m_{11} & m_{12} & m_{13} & m_{11} & m_{12} \\ m_{21} & m_{22} & m_{23} & m_{21} & m_{22} \\ m_{31} & m_{32} & m_{33} & m_{31} & m_{32} \end{matrix} \tag{1.27}$$

Then, form the first three products of Equation 1.26 by starting at the upper left corner of Equation 1.27 with m_{11}, forming a diagonal to the right, and then repeating with m_{12} and m_{13}. The last three products are subtracted in Equation 1.26 and are formed by starting in the upper right corner of Equation 1.27 with m_{12} and taking a diagonal to the left, repeating for m_{11} and m_{13}. Note the similarity to the 2×2 case. Unfortunately, such simple schemes fail above the 3×3 case.

Determinants of $n \times n$ matrices for $n > 3$ are computed in a similar vein. As in the earlier cases, the determinant of an $n \times n$ matrix may be expressed in terms of the determinants of $(n-1) \times (n-1)$ submatrices; this is termed as Laplace's expansion. To expand along row i or column j in $M \in F^{n \times n}$, write

$$\det(M) = \sum_{k=l}^{n} m_{ik}\tilde{m}_{ik} = \sum_{k=l}^{n} m_{kj}\tilde{m}_{kj} \tag{1.28}$$

where the m_{ik} (m_{kj}) are elements of M. The $\tilde{m}_{ik}$ ($\tilde{m}_{kj}$) are cofactors formed by deleting the ith (kth) row and the kth (jth) column of M, forming the determinant of the $(n-1) \times (n-1)$ resulting submatrix, and multiplying by $(-1)^{i+k}$ ($(-1)^{k+j}$). Notice that minors and their corresponding cofactors are related by ± 1.

Examples

$$(F = R)\text{: } M = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 2 & 3 & 6 \end{bmatrix}$$

- Expanding about row 1 produces

$$\det(M) = 0\begin{vmatrix} 4 & 5 \\ 3 & 6 \end{vmatrix} - 1\begin{vmatrix} 3 & 5 \\ 2 & 6 \end{vmatrix} + 2\begin{vmatrix} 3 & 4 \\ 2 & 3 \end{vmatrix}$$
$$= -(18 - 10) + 2(9 - 8) = -6$$

- Expanding about column 2 yields

$$\det(M) = -1\begin{vmatrix} 3 & 5 \\ 2 & 6 \end{vmatrix} + 4\begin{vmatrix} 0 & 2 \\ 2 & 6 \end{vmatrix} - 3\begin{vmatrix} 0 & 2 \\ 3 & 5 \end{vmatrix}$$
$$= -(18 - 10) + 4(0 - 4) - 3(0 - 6) = -6$$

- Repeating the first two columns to form Equation 1.27 gives

$$\begin{matrix} 0 & 1 & 2 & 0 & 1 \\ 3 & 4 & 5 & 3 & 4 \\ 2 & 3 & 6 & 2 & 3 \end{matrix}$$

  Taking the appropriate products,

$$0 \cdot 4 \cdot 6 + 1 \cdot 5 \cdot 2 + 2 \cdot 3 \cdot 3 - 1 \cdot 3 \cdot 6 - 0 \cdot 5 \cdot 3 - 2 \cdot 4 \cdot 2$$

  results in -6 as the determinant of M.

- Any square matrix with a zero row and/or zero column will have zero determinant. Likewise, any square matrix with two or more identical rows and/or two or more identical columns will have determinant equal to zero.

Determinants satisfy many interesting relationships. For any $n \times n$ matrix, the determinant may be expressed in terms of determinants of $(n-1) \times (n-1)$ matrices or first-order minors. In turn, determinants of $(n-1) \times (n-1)$ matrices may be expressed in terms of determinants of $(n-2) \times (n-2)$ matrices or second-order minors, etc. Also, the determinant of the product of two square matrices is equal to the product of the determinants:

$$\det(MN) = \det(M)\det(N) \tag{1.29}$$

For any $M \in F^{n \times n}$ such that $|M| \neq 0$, a unique inverse $M^{-1} \in F^{n \times n}$ satisfies

$$MM^{-1} = M^{-1}M = I \tag{1.30}$$

For Equation 1.29 one may observe the special case in which $N = M^{-1}$, then $(\det(M))^{-1} = \det(M^{-1})$. The inverse M^{-1} may be expressed using determinants and cofactors in the following manner. Form the matrix of cofactors

$$\tilde{M} = \begin{bmatrix} \tilde{m}_{11} & \tilde{m}_{12} & \cdots & \tilde{m}_{1n} \\ \tilde{m}_{21} & \tilde{m}_{22} & \cdots & \tilde{m}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{m}_{n1} & \tilde{m}_{n2} & \cdots & \tilde{m}_{nn} \end{bmatrix} \tag{1.31}$$

The transpose of Equation 1.31 is referred to as the adjoint matrix or adj(M). Then,

$$M^{-1} = \frac{\tilde{M}^{\mathrm{T}}}{|M|} = \frac{\mathrm{adj}(M)}{|M|} \tag{1.32}$$

Examples

- Choose M of the previous set of examples. The cofactor matrix is

$$\begin{bmatrix} 9 & -8 & 1 \\ 0 & -4 & 2 \\ -3 & 6 & -3 \end{bmatrix}$$

  Because $|M| = -6$, M^{-1} is

$$\begin{bmatrix} -\frac{3}{2} & 0 & \frac{1}{2} \\ \frac{4}{3} & \frac{2}{3} & -1 \\ -\frac{1}{6} & -\frac{1}{3} & \frac{1}{2} \end{bmatrix}$$

  Note that Equation 1.32 is satisfied.

- $M = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}$, $\text{adj}(M) = \begin{bmatrix} 3 & -1 \\ -4 & 2 \end{bmatrix}$, $M^{-1} = \begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -2 & 1 \end{bmatrix}$

  In the 2×2 case, this method reduces to interchanging the elements on the main diagonal, changing the sign on the remaining elements, and dividing by the determinant.

- Consider the matrix equation in Equation 1.18. Because $\det(G) \neq 0$, whenever the resistances are nonzero, with R_1 and R_2 having the same sign, the node voltages may be determined in terms of the current sources by multiplying on the left of both members of the equation using G^{-1}. Then $G^{-1}i = v$.

The rank of a matrix M, $\rho(M)$, is the number of linearly independent columns of M over F, or using other terminology, the dimension of the image of M. For $M \in F^{m \times n}$ the number of linearly independent rows and columns is the same, and is less than or equal to the minimum of m and n. If $\rho(M) = n$, M is of full-column rank; similarly, if $\rho(M) = m$, M is of full-row rank. A square matrix with all rows (and all columns) linearly independent is said to be nonsingular. In this case, $\det(M) \neq 0$. The rank of M also may be found from the size of the largest square submatrix with a nonzero determinant. A full-rank matrix has a full-size minor with a nonzero determinant.

The null space (kernel) of a matrix $M \in F^{m \times n}$ is the set

$$\ker\ M = \{v \in F^n \text{ such that } Mv = 0\} \tag{1.33}$$

Over F, ker M is a vector space with dimension defined as the nullity of M, $\nu(M)$. The fundamental theorem of linear equations relates the rank and nullity of a matrix $M \in F^{m \times n}$ by

$$\rho(M) + \nu(M) = n \tag{1.34}$$

If $\rho(M) < n$, then M has a nontrivial null space.

Examples

- $M = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

  The rank M is 1 because only one linearly independent column of M is found. To examine the null space of M, solve $Mv = 0$. Any element in ker M is of the form $\begin{bmatrix} f_1 \\ 0 \end{bmatrix}$ for $f_1 \in F$. Therefore, $\nu(M) = 1$.

- $M = \begin{bmatrix} 1 & 4 & 5 & 2 \\ 2 & 5 & 7 & 1 \\ 3 & 6 & 9 & 0 \end{bmatrix}$

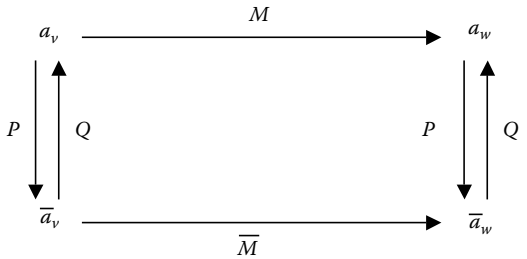
  The rank of M is 2 and the nullity is 2.

1.6 Basis Transformations

This section describes a change of basis as a linear operator. Because the choice of basis affects the matrix of a linear operator, it would be most useful if such a basis change could be understood within the context of matrix operations. Thus, the new matrix could be determined from the old matrix by matrix operations. This is indeed possible. This question is examined in two phases. In the first phase the linear operator maps from a vector space to itself. Then a basis change will be called a similarity transformation. In the second phase, the linear operator maps from one vector space to another, which is not necessarily the same as the first. Then a basis change will be called an equivalence transformation. Of course, the first situation is a special case of the second, but it is customary to make the distinction and to recognize the different terminologies. Philosophically, a fascinating special situation exists in which the second vector space, which receives the result of the operation, is an identical copy of the first vector space, from which the operation proceeds. However, in order to avoid confusion, this section does not delve into such issues.

For the first phase of the discussion, consider a linear operator that maps a vector space into itself, such as L: $V \rightarrow V$, where V is n-dimensional. Once a basis is chosen in V, L will have a unique matrix representation. Choose $\{v_1, v_2, \ldots, v_n\}$ and $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$ as two such bases. A matrix $M \in F^{n\times n}$ may be determined using the first basis, whereas another matrix $\overline{M} \in F^{n\times n}$ will result in the latter choice. According to the discussion following Equation 1.15, the ith column of M is the representation of $L(v_i)$ with respect to $\{v_1, v_2, \ldots, v_n\}$, and the ith column of $\overline{M}$ is the representation of $L(\bar{v}_i)$ with respect to $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$. As in Equation 1.4, any basis element v_i has a unique representation in terms of the basis $\{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$. Define a matrix $P \in F^{n\times n}$ using the ith column as this representation. Likewise, $Q \in F^{n\times n}$ may have as its ith column the unique representation of $\bar{v}_1$ with respect to $\{v_1, v_2, \ldots, v_n\}$. Either represents a basis change which is a linear operator. By construction, both P and Q are nonsingular. Such matrices and linear operators are sometimes called basis transformations. Notice that $P = Q^{-1}$.

If two matrices M and $\overline{M}$ represent the same linear operator L, they must somehow carry essentially the same information. Indeed, a relationship between M and $\overline{M}$ may be established. Consider $a_v, a_w, \bar{a}_v, \bar{a}_w \in F^n$ such that $Ma_v = a_w$ and $\overline{M}\bar{a}_v = \bar{a}_w$. Here, a_v denotes the representation of v with respect to the basis v_i, $\bar{a}_v$ denotes the representation of v with respect to the basis $\bar{v}_i$, and so forth. In order to involve P and Q in these equations, it is possible to make use of a sketch:



In this sketch, a vector at a given corner can be multiplied by a matrix on an arrow leaving the corner and set equal to the vector that appears at the corner at which that arrow arrives. Thus, for example, $a_w = Ma_v$ may be deduced from the top edge of the sketch. It is interesting to perform "chases" around such sketches. By way of illustration, consider the lower right corner. Progress around the sketch

counterclockwise so as to reach the lower left corner and set the result equal to that obtained by progressing clockwise to the lower left corner. In equations this is carried out as follows:

$$\bar{a}_w = Pa_w = PMa_v = PMQ\bar{a}_v = \overline{M}\bar{a}_v \tag{1.35}$$

Inasmuch as $\bar{a}_v \in F^n$ is arbitrary, it follows that

$$\overline{M} = PMP^{-1} \tag{1.36}$$

Sketches that have this type of property, namely the same result when the sketch is traversed from a starting corner to an finishing corner by two paths, are said to be commutative. It is perhaps more traditional to show the vector space F^n instead of the vectors at the corners. Thus, the sketch would be called a commutative diagram of vector spaces and linear operators. M and $\overline{M}$ are said to be similar because a nonsingular matrix $P \in F^{n\times n}$ is such that Equation 1.36 is true. The matrix P is then called a similarity transformation. Note that all matrix representations associated with the same linear operator, from a vector space to itself, are similar. Certain choices of bases lead to special forms for the matrices of the operator, as are apparent in the following examples.

Examples

- Choose L: $R^2 \rightarrow R^2$ as the linear operator that rotates a vector by 90°. Pick $\{v_1, v_2\}$ as the natural basis $\{[1 \quad 0]^T, [0 \quad 1]^T\}$ and $\{\bar{v}_1, \bar{v}_2\}$ as $\{[1 \quad 1]^T, [1 \quad 0]^T\}$. Then,

$$L(v_1) = [v_1 \quad v_2]\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$L(v_2) = [v_1 \quad v_2]\begin{bmatrix} -1 \\ 0 \end{bmatrix}$$
$$L(\bar{v}_1) = [\bar{v}_1 \quad \bar{v}_2]\begin{bmatrix} 1 \\ -2 \end{bmatrix}$$
$$L(\bar{v}_2) = [\bar{v}_1 \quad \bar{v}_2]\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

  so that

$$M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad \overline{M} = \begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}$$

  To find the similarity transformation P, determine the representations of the basis vectors v_i in terms of the basis $\{\bar{v}_1, \bar{v}_2\}$. Then,

$$P = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, \quad \bar{P} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

  so that $PMP^{-1} = \overline{M}$.

- Suppose that $M \in F^{n\times n}$ is a representation of L: $V \rightarrow V$. Assume a $v \in F^n$ exists such that the vectors $\{v, Mv, \ldots, M^{n-1}v\}$ are linearly independent. Thus, these n vectors can be chosen as an alternate basis. (Section 1.7 discusses the characteristic equation of a matrix M.) Using the Cayley–Hamilton

theorem, which states that every matrix satisfies its own characteristic equation, it is always possible to write $M^n v$ as a linear combination of these alternate basis vectors

$$-\alpha_n v - \alpha_{n-1} Mv - \cdots - \alpha_1 M^{n-1} v$$

for $\alpha_i \in F$. The matrix representation of L with respect to the alternate basis given by this set of linearly independent vectors is

$$\overline{M} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\alpha_n \\ 1 & 0 & \cdots & 0 & -\alpha_{n-1} \\ 0 & 1 & \cdots & 0 & -\alpha_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -\alpha_2 \\ 0 & 0 & \cdots & 1 & -\alpha_1 \end{bmatrix}$$

which is the transpose of what is known as the companion form.

For the second phase of the discussion, select a pair of bases, one for the vector space V and one for the vector space W, and construct the resulting matrix representation M of L: $V \rightarrow W$. Another choice of bases exists for V and W, with the property that the resulting matrix $\overline{M}$ representing L is of the form

$$\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \tag{1.37}$$

where I is $\rho(M) \times \rho(M)$. Such a matrix is said to be in normal form. It is possible to transform M into $\overline{M}$ with the assistance of three types of transformations which are called elementary: (1) interchange any two rows or any two columns, (2) scale any row or column by a nonzero element in F, and (3) add any F-multiple of any row (column) to any other row (column). It is apparent that each of the three transformations involving rows may be accomplished by multiplying M on the left by a nonsingular matrix. Column operations require a corresponding multiplication on the right. The secret to understanding elementary transformations is to recognize that each of them will carry one basis into another. Not as easy to see, but equally true, is that any transformation that carries one basis into another basis must be a product of such elementary transformations. The elementary column transformations are interpreted as changing the basis in the vector space from which the operator takes vectors, whereas the elementary row transformations correspond to changing the basis in the vector space into which the operator places its result. It stands to reason that a simultaneous adjustment of both sets of basis vectors could lead to some quite special forms for the matrices of an operator. Of course, a great deal of linear algebra and its applications is concerned with just such constructions and forms. Space does not permit a complete treatment here.

If a matrix M has been transformed into normal form, certain types of key information become available. For example, one knows the rank of M because $\rho(M)$ is the number of rows and columns of the identity in Equation 1.37. Perhaps more importantly, the normal form is easily factored in a fundamental way, and so such a construction is a natural means to construct two factors of minimal rank for a given matrix. The reader is cautioned, however, to be aware that computational linear algebra is quite a different subject than theoretical linear algebra. One common saying is that ''if an algorithm is straightforward, then it is not numerically desirable.'' This may be an exaggeration, but it is well to recognize the implications of finite precision on the computer. Space limitations prevent addressing numerical issues.

Many other thoughts can be expressed in terms of elementary basis transformations. By way of illustration, elementary basis transformations offer an alternative in finding the inverse of a matrix. For a nonsingular matrix $M \in F^{n \times n}$, append to M an $n \times n$ identity I to form the $n \times 2n$ matrix

$$\hat{M} = [M \quad I] \tag{1.38}$$

Perform elementary row transformations on Equation 1.38 to transform M into normal form. Then M^{-1} will appear in the last n columns of the transformed matrix.

Examples

$$(F = R)\text{: } M = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}$$

- Transform M into normal form. The process can be carried out in many ways. For instance, begin by scaling row 1 by $\frac{1}{2}$,

$$L_1 M = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} \\ 4 & 3 \end{bmatrix}$$

  Clear the first element in row 2 by

$$L_2 L_1 M = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$$

  Finally, perform a column operation to produce $\overline{M}$:

$$L_2 L_1 M R_1 = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

  The rank of M is 2.

- Recall M^{-1} from previous examples. Form $\hat{M}$ and transform M into normal form by the following row operations:

$$L_1 \hat{M} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 4 & 3 & 0 & 1 \end{bmatrix}$$

$$L_2 L_1 \hat{M} = \begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 4 & 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}$$

$$L_3 L_2 L_1 \hat{M} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{3}{2} & -\frac{1}{2} \\ 0 & 1 & -2 & 1 \end{bmatrix}$$

1.7 Characteristics: Eigenvalues, Eigenvectors, and Singular Values

A matrix has certain characteristics associated with it. Of these, characteristic values or eigenvalues may be determined through the use of matrix pencils. In general, a matrix pencil may be formed from two matrices M and $N \in F^{m \times n}$ and an indeterminate λ in the manner

$$[\lambda N - M] \in F[\lambda]^{n\times n} \tag{1.39}$$

In determining the eigenvalues of a square matrix $M \in F^{n\times n}$, one assumes the special case in which $N = I \in F^{n\times n}$.

Assume that M is a square matrix over the complex numbers. Then, $\lambda \in C$ is called an eigenvalue of M if some nonzero vector $v \in C^n$ exists such that

$$Mv = \lambda v \tag{1.40}$$

Any such $v \neq 0$ satisfying Equation 1.40 is said to be an eigenvector of M associated with λ. It is easy to see that Equation 1.40 can be rewritten as

$$(\lambda I - M)v = 0 \tag{1.41}$$

Because Equation 1.41 is a set of n linear homogeneous equations, a nontrivial solution ($v \neq 0$) exists if and only if

$$\Delta(\lambda) = \det(\lambda I - M) = 0 \tag{1.42}$$

In other words, $(\lambda I - M)$ is singular. Therefore, λ is an eigenvalue of M if and only if it is a solution of Equation 1.42. The polynomial $\Delta(\lambda)$ is the characteristic polynomial and $\Delta(\lambda) = 0$ is the characteristic equation. Moreover, every $n \times n$ matrix has n eigenvalues that may be real, complex, or both, where complex eigenvalues occur in complex–conjugate pairs. If two or more eigenvalues are equal they are said to be repeated (not distinct). It is interesting to observe that although eigenvalues are unique, eigenvectors are not. Indeed, an eigenvector can be multiplied by any nonzero element of C and still maintain its essential features. Sometimes this lack of uniqueness is resolved by selecting unit length for the eigenvectors with the aid of a suitable norm.

Recall that matrices representing the same operator are similar. One may question if these matrices indeed contain the same characteristic information. To answer this question, examine

$$\det(\lambda I - \overline{M}) = \det(\lambda PP^{-1} - PMP^{-1}) = \det(P(\lambda I - M)P^{-1}) \tag{1.43}$$

$$= \det(P)\det(\lambda I - M)\det(P^{-1}) = \det(\lambda I - M) \tag{1.44}$$

From Equation 1.44 one may deduce that similar matrices have the same eigenvalues because their characteristic polynomials are equal.

For every square matrix M with distinct eigenvalues, a similar matrix $\overline{M}$ is diagonal. In particular, the eigenvalues of M, and hence $\overline{M}$, appear along the main diagonal. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues (all distinct) of M and let $v_1, v_2, \ldots, v_n$ be corresponding eigenvectors. Then, the vectors $\{v_1, v_2, \ldots, v_n\}$ are linearly independent over C. Choose $P^{-1} = Q = [v_1\ v_2\ \cdots\ v_n]$ as the modal matrix. Because $Mv_i = \lambda_i v_i$, $\overline{M} = PMP^{-1}$ as before.

For matrices with repeated eigenvalues, a similar approach may be followed wherein $\overline{M}$ is block diagonal, which means that matrices occur along the diagonal with zeros everywhere else. Each matrix along the diagonal is associated with an eigenvalue and takes a specific form depending upon the characteristics of the matrix itself. The modal matrix consists of generalized eigenvectors, of which the aforementioned eigenvector is a special case; thus the modal matrix is nonsingular. The matrix $\overline{M}$ is the Jordan canonical form. Space limitations preclude a detailed analysis of such topics here; the reader is directed to Chen (1984) for further development.

Examples

$$(F = C):\ M = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

- The characteristic polynomial is $\Delta(\lambda) = (\lambda - 1)(\lambda - 3) - 8 = (\lambda - 5)(\lambda + 1)$. The eigenvalues are $\lambda_1 = 5$, $\lambda_2 = -1$. To find the associated eigenvectors recall that for each λ_i, $(\lambda_i I - M)$ is singular, and write Equation 1.41

$$(\lambda_1 I - M)v_1 = \begin{bmatrix} 4 & -4 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$(\lambda_2 I - M)v_2 = \begin{bmatrix} -2 & -4 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  Then, $v_{11} = v_{12}$ and $v_{21} = -2v_{22}$ so that $v_1 = [1 \quad 1]^T$ and $v_2 = [-2 \quad 1]^T$ are eigenvectors associated with λ_1 and λ_2, respectively.

- Because the eigenvalues of M are distinct, M may be diagonalized. For verification, choose $P^{-1} = [v_1\ v_2]$. Then,

$$\begin{aligned} \overline{M} = PMP^{-1} &= \frac{1}{3}\begin{bmatrix} 1 & 2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}\begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix} \\ &= \frac{1}{3}\begin{bmatrix} 15 & 0 \\ 0 & -3 \end{bmatrix} = \begin{bmatrix} 5 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \end{aligned}$$

In general, a matrix $M \in F^{m \times n}$ of rank r can be written in terms of its singular-value decomposition (SVD),

$$M = U\Sigma V^{\star} \tag{1.45}$$

For any M, unitary matrices U and V of dimension $m \times m$ and $n \times n$, respectively, form the decomposition; that is, $UU_{\star} = U_{\star}U = I$ and $VV_{\star} = V_{\star}V = I$. The matrix $\sum \in F^{m \times n}$ of the form

$$\begin{bmatrix} \sum_r & 0 \\ 0 & 0 \end{bmatrix} \tag{1.46}$$

for $\sum_r \in F^{r \times r}$, a diagonal matrix represented by

$$\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \tag{1.47}$$

The elements σ_i, called singular values, are related by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, and the columns of U (V) are referred to as left (right) singular vectors. Although the unitary matrices U and V are not unique for a given M, the singular values are unique.

SVD is useful in the numerical calculation of rank. After performing an SVD, the size of the matrix $\sum_r$ may be decided. Additionally, the generalized inverse of a matrix M may be found by

$$M^{\dagger} = V \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^{\star} \tag{1.48}$$

It can be verified easily that $MM^{\dagger} M = M$ and $M^{\dagger} MM^{\dagger} = M^{\dagger}$. In the special case in which M is square and nonsingular,

$$M^{\dagger} = M^{-1} = V\Sigma^{-1} U^{\star} \tag{1.49}$$

1.8 On Linear Systems

Consider a set of n simultaneous algebraic equations, in m unknowns, written in the customary matrix form $w = Mv$ where

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{m1} & m_{m2} & \dots & m_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \tag{1.50}$$

In the context of the foregoing discussion, Equation 1.50 represents the action of a linear operator. If the left member is a given vector, in the usual manner, then a first basic issue concerns whether the vector represented by the left member is in the image of the operator or not. If it is in the image, the equation has at least one solution; otherwise the equation has no solution. A second basic issue concerns the kernel of the operator. If the kernel contains only the zero vector, then the equation has at most one solution; otherwise more than one solution can occur, provided that at least one solution exists.

When one thinks of a set of simultaneous equations as a ''system'' of equations, the intuitive transition to the idea of a linear system is quite natural. In this case the vector in the left member becomes the input to the system, and the solution to Equation 1.50, when it exists and is unique, is the output of the system.

Other than being a description in terms of inputs and outputs, as above, linear systems may also be described in terms of sets of other types of equations, such as differential equations or difference equations. When that is the situation, the familiar notion of initial condition becomes an instance of the idea of state, and one must examine the intertwining of states and inputs to give outputs. Then, the idea of Equation 1.50, when each input yields a unique output, is said to define a system function.

If the differential (difference) equations are linear and have constant coefficients, the possibility exists of describing the system in terms of transforms, for example, in the s- or z-domain. This leads to fascinating new interpretations of the ideas of the foregoing sections, this time, for example, over fields of rational functions. Colloquially, such functions are best known as transfer functions.

Associated with systems described in the time-, s-, or z-domain, some characteristics of the system also aid in analysis techniques. Among the most basic of these are the entities termed poles and zeros, which have been linked to the various concepts of system stability. Both poles and zeros may be associated with matrices of transfer functions, and with the original differential or difference equations themselves. A complete and in-depth treatment of the myriad meanings of poles and zeros is a challenging undertaking, particularly in matrix cases. For a survey of the ideas, see Schrader and Sain (1989). However, a great many of the definitions involve such concepts as rank, pencils, eigenvalues, eigenvectors, special matrix forms, vector spaces, and modules—the very ideas sketched out in the sections preceding.

Algebraic methods are of growing importance in system theory. Of particular interest is the historical survey by Schrader and Wyman (2008) which describes the module-theoretic approach to zeros of a linear system and the application of these ideas to inverse systems and system design. The longstanding notion that zeros of a linear system become the poles of its inverse system is resolved using these methods. Additionally, for the first time, by employing module-theoretic approaches, a structural algebraic meaning is given to the principle that the number of zeros of a transfer function matrix equals the number of poles. Central to this result is the Fundamental Pole-Zero Exact Sequence, an exact sequence of finite-dimensional vector spaces over the same field, represented by

$$0 \to \mathcal{Z}(G) \to \frac{\mathcal{X}(G)}{\mathcal{W}(\ker G(z))} \to \mathcal{W}(\text{im } G(z)) \to 0$$

where

$\mathcal{Z}(G)$ is the global space of zeros of a transfer function matrix $G(z)$
$\mathcal{X}(G)$ is the global space of poles of $G(z)$
$\mathcal{W}()$ is the Wedderburn–Forney construction

The sequence is presented here to encourage further investigation into these powerful methods.

One very commonly known idea for representing solutions to Equation 1.50 is Cramer's rule. When $m = n$, and when M has an inverse, the use of Cramer's rule expresses each unknown variable individually by using a ratio of determinants. Choose the ith unknown v_i. Define the determinant M_i as the determinant of a matrix formed by replacing column i in M with w. Then,

$$v_i = \frac{M_i}{\det(M)} \tag{1.51}$$

It turns out that this very interesting idea makes fundamental use of the notion of *multiplying* vectors, which is not part of the axiomatic framework of the vector space. The reader may want to reflect further on this observation, with respect to the foregoing treatment of determinants. When the framework of vector spaces is expanded to include vector multiplication, as in the case of determinants, one gets to the technical subject of algebras. Chapter 2 returns to this concept.

The concepts presented previously allow for more detailed considerations in the solution of circuit and filter problems, using various approaches outlined in the remainder of this book. Chapter 2 discusses the multiplication of vectors by means of the foundational idea of bilinear operators and matrices. Chapters 3 through 5 on transforms—Fourier, z, and Laplace—provide the tools for analysis by allowing a set of differential or difference equations describing a circuit to be written as a system of linear algebraic equations. Moreover, each transform itself can be viewed as a linear operator, and thus becomes a prime example of the ideas of this chapter. The remaining chapters focus on graph–theoretical approaches to the solution of systems of algebraic equations. From this vantage point, one can see the entire Section I in the context of linear operators, their addition, and multiplication.

A brief treatment cannot deal with all the interesting questions and answers associated with linear operators and matrices. For a more detailed treatment of these standard concepts, see any basic algebra text, for example, Greub (1967).

References

C.-T. Chen, *Linear System Theory and Design*, New York: GBS College Publishing, 1984.

W. H. Greub, *Linear Algebra*, New York: Springer-Verlag, 1967.

C. B. Schrader and M. K. Sain, Research on system zeros: A survey, *Int. J. Control*, 50, 4, 1407–1433, Oct. 1989.

C. B. Schrader and B. F. Wyman, Modules of zeros for linear multivariable systems, in *Advances in Statistical Control, Algebraic System Theory, and Dynamic Systems Characteristics*. C.-H Won, C. B. Schrader and A. N. Michel, eds., New York: Birkhauser Boston, Inc., pp. 145–158, 2008.

2

Bilinear Operators and Matrices

2.1 Introduction **2**-1
2.2 Algebras **2**-2
2.3 Bilinear Operators **2**-3
2.4 Tensor Product **2**-4
2.5 Basis Tensors **2**-5
2.6 Multiple Products **2**-8
2.7 Determinants **2**-9
2.8 Skew Symmetric Products **2**-11
2.9 Solving Linear Equations **2**-14
2.10 Symmetric Products **2**-16
2.11 Summary **2**-18
Reference **2**-19

Michael K. Sain
University of Notre Dame

Cheryl B. Schrader
Boise State University

2.1 Introduction

The key player in Chapter 1 was the F-vector space V, together with its associated notion of bases, when they exist, and linear operators taking one vector to another. The idea of basis is, of course, quite central to the applications because it permits a vector v in V to be represented by a list of scalars from F. Such lists of scalars are the quantities with which one computes. No doubt the idea of an F-vector space V is the most common and widely encountered notion in applied linear algebra. It is typically visualized, on the one hand, by long lists of axioms, most of which seem quite reasonable, but none of which is particularly exciting, and on the other hand by images of classical addition of force vectors, velocity vectors, and so forth. The notion seems to do no harm, and helps one to keep his or her assumptions straight. As such it is accepted by most engineers as a plausible background for their work, even if the ideas of matrix algebra are more immediately useful. Perhaps some of the least appreciated but most crucial of the vector space axioms are the four governing the scalar multiplication of vectors. These link the abelian group of vectors to the field of scalars. Along with the familiar distributive covenants, these four agreements intertwine the vectors with the scalars in much the same way that the marriage vows bring about the union of man and woman. This section brings forth a new addition to the marriage.

As useful as it is, the notion of an F-vector space V fails to provide for one of the most important ideas in the applications—the concept of multiplication of vectors. In a vector space one can add vectors and multiply vectors by scalars, but one cannot multiply vectors by vectors. Yet, there are numerous situations in which one faces exactly these operations. Consider, for instance, the cross and dot products from field theory. Even in the case of matrices, the ubiquitous and crucial matrix multiplication is available, when it is defined. The key to the missing element in the discussion lies in the terminology for

matrix operations, which will be familiar to the reader as the matrix algebra. What must occur in order for vector-to-vector multiplication to be available is for the vector space to be extended into an algebra.

Unfortunately, the word "algebra" carries a rather imprecise meaning from the most elementary and early exposures from which it came to signify the collection of operations done in arithmetic, at the time when the operations are generalized to include symbols or literals such as a, b, and c or x, y, and z. Such a notion generally corresponds closely with the idea of a field, F, as defined in Chapter 1, and is not much off the target for an environment of scalars. It may, however, come as a bit of a surprise to the reader that algebra is a technical term, in the same spirit as fields, vector spaces, rings, etc. Therefore, if one is to have available a notion of multiplication of vectors, then it is appropriate to introduce the precise notion of an algebra, which captures the desired idea in an axiomatic sense.

2.2 Algebras

Chapter 1 mentioned that the integers I and the polynomials $F[s]$ in s with coefficients in a field F were instances of a ring. In this section, it is necessary to carry the concept of ring beyond the example stage. Of course, a long list of axioms could be provided, but it may be more direct just to cite the changes necessary to the field axioms already provided in Section 1.2. To be precise, the axioms of a commutative ring differ from the axioms of a field only by the removal of the multiplicative inverse. Intuitively, this means that one cannot always divide, even if the element in question is nonzero. Many important commutative rings are found in the applications; however, this chapter is centered on rings, wherein one more axiom is removed—the commutativity of multiplication. The ring of $n \times n$ matrices with elements from a field is a classic and familiar example of such a definition. It may be remarked that in some references a distinction is made between rings and rings with identity, the latter having a multiplicative identity and the former not being so equipped. This treatment has no need for such a distinction, and hereafter the term "ring" is understood to mean ring with identity, or, as described above, a field with the specified two axioms removed.

It is probably true that the field is the most comfortable of axiomatic systems for most persons because it corresponds to the earliest and most persistent of calculation notions. However, it is also true that the ring has an intuitive and immediate understanding as well, which can be expressed in terms of the well-known phrase "playing with one arm behind one's back." Indeed, each time an axiom is removed, it is similar to removing one of the options in a game. This adds to the challenge of a game, and leads to all sorts of new strategies. Such is the case for algebras, as is clear from the next definition. What follows is not the most general of possible definitions, but probably that which is most common.

An algebra A is an F-vector space A, which is equipped with a multiplication a_1a_2 of vectors a_1 and a_2 in such a manner that it is also a ring. First, addition in the ring is simply addition of vectors in the vector space. Second, a special relationship exists between multiplication of vectors and scalar multiplication in the vector space. If a_1 and a_2 are vectors in A, and if f is a scalar in F, then the following identity holds:

$$f(a_1a_2) = (fa_1)a_2 = a_1(fa_2) \tag{2.1}$$

Note that the order of a_1 and a_2 does not change in the above equalities. This must be true because no axiom of commutativity exists for multiplication. The urge to define a symbol for vector multiplication is resisted here so as to keep things as simple as possible. In the same way the notation for scalar multiplication, as introduced in Chapter 1, is suppressed here in the interest of simplicity. Thus, the scalar multiplication can be associated either with the vector product, which lies in A, or with one or other of the vector factors. This is exactly the familiar situation with the matrix algebra.

Hidden in the definition of the algebra A above is the precise detail arising from the statement that A is a ring. Associated with that detail is the nature of the vector multiplication represented above with the juxtaposition a_1a_2. Because all readers are familiar with several notions of vector multiplication, the question arises as to just what constitutes such a multiplication. It turns out that a precise notion for

multiplication can be found in the idea of a bilinear operator. Thus, an alternative description of Section 2.3 is that of vector spaces equipped with vector multiplication. Moreover, one is tempted to inquire whether a vector multiplication exists that is so general in nature that all the other vector multiplications can be derived from it. In fact, this is the case, and the following section sets the stage for introducing such a multiplication.

2.3 Bilinear Operators

Suppose that there are three F-vector spaces: U, V, and W. Recall that $U \times V$ is the Cartesian product of U with V, and denotes the set of all ordered pairs, the first from U and the second from V. Now, consider a mapping b from $U \times V$ into W. For brevity of notation, this can be written b: $U \times V \rightarrow W$. The mapping b is a bilinear operator if it satisfies the following pair of conditions

$$b(f_1 u_1 + f_2 u_2, v) = f_1 b(u_1, v) + f_2 b(u_2, v) \tag{2.2}$$

$$b(u, f_1 v_1 + f_2 v_2) = f_1 b(u, v_1) + f_2 b(u, v_2) \tag{2.3}$$

for all f_1 and f_2 in F, for all u, u_1, and u_2 in U, and for all v, v_1, and v_2 in V. The basic idea of the bilinear operator is apparent from this definition. It is an operator with two arguments, having the property that if either of the two arguments is fixed, the operator becomes linear in the remaining argument. A moment's reflection will show that the intuitive operation of multiplication is of this type.

One of the important features of a bilinear operator is that its image need not be a subspace of W. This is in marked contrast with the image of a linear operator, whose image is always a subspace. This property leads to great interest in the manipulations associated with vector products. At the same time, it brings about a great deal of nontriviality. The best way to illustrate the point is with an example.

Example

Suppose that U, V, and W have bases $\{u_1, u_2\}$, $\{v_1, v_2\}$, and $\{w_1, w_2, w_3, w_4\}$, respectively. Then, vectors u in U and v in V can be represented in the following manner

$$u = f_1 u_1 + f_2 u_2 \tag{2.4}$$

$$v = g_1 v_1 + g_2 v_2 \tag{2.5}$$

where f_i and g_i are elements of F for $i = 1, 2$. Define a bilinear map by the action

$$b(u, v) = 2 f_1 g_1 w_1 + 3 f_1 g_2 w_2 + 3 f_2 g_1 w_3 + 2 f_2 g_2 w_4 \tag{2.6}$$

It is clear that every vector

$$h_1 w_1 + h_2 w_2 + h_3 w_3 + h_4 w_4 \tag{2.7}$$

in the image of b has the property that $9 h_1 h_4 = 4 h_2 h_3$. If the $\{h_i, i = 1, 2, 3, 4\}$ are given so as to satisfy the latter condition, consider the task of showing that this vector in W is a vector in the image of b. Suppose that $h_1 = 0$. Then either h_2 or h_3, is zero, or both are zero. If h_2 is zero, one may choose $f_1 = 0$, $f_2 = 1$, $g_1 = h_3/3$, and $g_2 = h_4/2$. If $h_3 = 0$, one may choose $g_1 = 0$, $g_2 = 1$, $f_1 = h_2/3$, and $f_2 = h_4/2$. An analogous set of constructions is available when $h_4 = 0$. For the remainder of the argument, it is assumed that neither h_1 nor h_2 is zero. Accordingly, none of the coordinates $\{h_i, i = 1, 2, 3, 4\}$ is zero. Without loss, assume that $f_1 = 1$. Then, g_1 is given by $h_1/2$, g_2 is found from $h_2/3$, and f_2 is constructed from $h_3/3g_1$, which is then $2h_3/3h_1$. It is easy to check that these choices produce the correct first three coordinates;

the last coordinate is $4h_3h_2/9h_1$, which by virtue of the property $9h_1h_4 = 4h_2h_3$ is equal to h_4 as desired. Thus, a vector in W is in the image of b if and only if the relation $9h_1h_4 = 4h_2h_3$ is satisfied. Next, it is shown that the vectors in this class are not closed under addition. For this purpose, simply select a pair of vectors represented by (1, 1, 9, 4) and (4, 9, 1, 1). The sum, (5, 10, 10, 5), does not satisfy the condition.

It is perhaps not so surprising that the image of b in this example is not a subspace of W. After all, the operator b is nonlinear, when both of its arguments are considered. What may be surprising is that a natural and classical way can be used to circumvent this difficulty, at least to a remarkable degree. The mechanism that is introduced in order to address such a question is the tensor. The reader should bear in mind that many technical personnel have prior notions and insights on this subject emanating from areas such as the theory of mechanics and related bodies of knowledge. For these persons, the authors wish to emphasize that the following treatment is algebraic in character and may exhibit, at least initially, a flavor different from that to which they may be accustomed. This difference is quite typical of the distinctive points of view that often can be found between the mathematical areas of algebra and analysis. Such differences are fortunate insofar as they promote progress in understanding.

2.4 Tensor Product

The notions of tensors and tensor product, as presented in this treatment, have the intuitive meaning of a very general sort of bilinear operator, in fact, the *most* general such operator. Once again, F-vector spaces U, V, and W are assumed. Suppose that b: $U \times V \rightarrow W$ is a bilinear operator. Then the pair (b, W) is said to be a tensor product of U and V if two conditions are met. The first condition is that W is the smallest F-vector space that contains the image of b. Using alternative terminology this could be expressed as W being the vector space generated by the image of b. The term "generated" in this expression refers to the formation of all possible linear combinations of elements in the image of b. The second condition relates b to an arbitrary bilinear operator $\breve{b}$: $U \times V \rightarrow X$ in which X is another F-vector space. To be precise, the second condition states that for every such $\breve{b}$, a linear operator $\breve{B}$: $W \rightarrow X$ exists with the property that

$$\breve{b}(u, v) = \breve{B}(b(u, v)) \tag{2.8}$$

for all pairs (u, v) in $U \times V$. Intuitively, this means that the arbitrary bilinear operator $\breve{b}$ can be factored in terms of the given bilinear operator b, which does not depend upon $\breve{b}$, and a linear operator $\breve{B}$ which does depend upon $\breve{b}$.

The idea of the tensor product is truly remarkable. Moreover, for any bilinear operator $\breve{b}$, the induced linear operator $\breve{B}$ is unique. The latter result is easy to see. Suppose that there are two such induced linear operators, e.g., $\breve{B}_1$ and $\breve{B}_2$. It follows immediately that

$$(\breve{B}_1 - \breve{B}_2)(b(u, v)) = 0 \tag{2.9}$$

for all pairs (u, v). However, the first condition of the tensor product assures that the image of b contains a set of generators for W, and thus that $(\breve{B}_1 - \breve{B}_2)$ must in fact be the zero operator. Therefore, once the tensor product of U and V is put into place, bilinear operations are in a one-to-one correspondence with linear operations. This is the essence of the tensor idea, and a very significant way to parameterize product operations in terms of matrices. In a certain sense, then, the idea of this chapter is to relate the fundamentally nonlinear product operation to the linear ideas of Chapter 1. That this is possible is, of course, classical; nonetheless, it remains a relatively novel idea for numerous workers in the applications. Intuitively, what happens here is that the idea of *product* is abstracted in the bilinear operator b, with all the remaining details placed in the realm of the induced linear operator $\breve{B}$.

When a pair (b, W) satisfies the two conditions above, and is therefore a tensor product for U and V, it is customary to replace the symbol b with the more traditional symbol $\otimes$. However, in keeping the notion

that $\otimes$ represents a product and not just a general mapping it is common to write $u \otimes v$ in place of the more correct, but also more cumbersome, $\otimes(u, v)$. Along the same lines, the space W is generally denoted $U \otimes V$. Thus, a tensor product is a pair $(U \otimes V, \otimes)$. The former is called the tensor product of U with V, and $\otimes$ is loosely termed the tensor product. Clearly, $\otimes$ is the most general sort of product possible in the present situation because all other products can be expressed in terms of it by means of linear operators $\breve{B}$. Once again, the colloquial use of the word "product" is to be identified with the more precise algebraic notion of bilinear operation. In this way the tensor product becomes a sort of "grandfather" for all vector products.

Tensor products can be constructed for arbitrary vector spaces. They are not, however, unique. For instance, if $U \otimes V$ has finite dimension, then W obviously can be replaced by any other F-vector space of the same dimension, and $\otimes$ can be adjusted by a vector space isomorphism. Here, the term "isomorphism" denotes an invertible linear operator between the two spaces in question. It can also be said that the two tensor product spaces $U \otimes V$ and W are isomorphic to each other. Whatever the terminology chosen, the basic idea is that the two spaces are essentially the same within the axiomatic framework in use.

2.5 Basis Tensors

Attention is now focused on the case in which U and V are finite-dimensional vector spaces over the field F. Suppose that $\{u_1, u_2, \ldots, u_m\}$ is a basis for U and $\{v_1, v_2, \ldots, v_n\}$ is a basis for V. Consider the following vectors

$$u_1 \otimes v_1 \quad u_1 \otimes v_2 \quad \cdots \quad u_1 \otimes v_n \quad u_2 \otimes v_1 \quad \cdots \quad u_m \otimes v_n \tag{2.10}$$

which can be represented in the manner $\{u_i \otimes v_j,\ i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n\}$. These vectors form a basis for the vector space $U \otimes V$. To understand the motivation for this, note that vectors in U and V, respectively, can be written uniquely in the following forms

$$u = \sum_{i=1}^{m} f_i u_i \tag{2.11}$$

$$v = \sum_{j=1}^{n} g_j v_j \tag{2.12}$$

Recall that $\otimes$, which is an alternate notation for b, is a bilinear operator. It follows then that

$$u \otimes v = \sum_{i=1}^{m} \sum_{j=1}^{n} f_i g_j u_i \otimes v_j \tag{2.13}$$

which establishes that the proposed basis vectors certainly span the image of $\otimes$, and thus that they span the tensor product space $U \otimes V$. It also can be shown that the proposed set of basis vectors is linearly independent. However, in the interest of brevity for this summary exposition, the details are omitted.

From this point onward, inasmuch as the symbol $\otimes$ has replaced b, it will be convenient to use b in place of $\breve{b}$ and B in place of $\breve{B}$. It is hoped that this leads to negligible confusion. Thus, in the sequel b refers simply to a bilinear operator and B to its induced linear counterpart.

Example

Consider the bilinear form b: $R^2 \times R^3 \rightarrow R$ with action defined by

$$b(f_1, f_2, g_1, g_2, g_3) = 2 f_2 g_3 \tag{2.14}$$

Observe that this can be put into the more transparent form

$$[f_1 \quad f_2]\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 2\end{bmatrix}\begin{bmatrix}g_1\\ g_2\\ g_3\end{bmatrix} \tag{2.15}$$

which, in turn, can be written in the compact notation $u^{\mathrm{T}}Mv$. Clearly, U has dimension two, and V has dimension three. Thus, $U \otimes V$ has a basis with six elements. The operator b maps into R, which has a basis with one element. All bases are chosen to be standard. Thus, an ith basis vector contains the multiplicative field element 1 in its ith row, and the additive field element 0 in its other rows. Therefore, the matrix of B has one row and six columns. To compute the entries, it is necessary to agree upon the order of the basis elements in $R^2 \otimes R^3$. It is customary to choose the natural ordering as introduced previously

$$u_1 \otimes v_1 \quad u_1 \otimes v_2 \quad u_1 \otimes v_3 \quad u_2 \otimes v_1 \quad u_2 \otimes v_2 \quad u_2 \otimes v_3 \tag{2.16}$$

The coordinate h_1 associated with the basis vector [1] in R, considered to be a vector space, is given by

$$h_1 = b(1, 0, 1, 0, 0) = 0 \tag{2.17}$$

when u and v are given by the respective first basis vectors in R^2 and R^3, respectively,

$$u = \begin{bmatrix}1\\ 0\end{bmatrix} \tag{2.18}$$

$$v = \begin{bmatrix}1\\ 0\\ 0\end{bmatrix} \tag{2.19}$$

Similarly, for the other five pairings in order, one obtains

$$h_1 = b(1, 0, 0, 1, 0) = 0 \tag{2.20}$$

$$h_1 = b(1, 0, 0, 0, 1) = 0 \tag{2.21}$$

$$h_1 = b(0, 1, 1, 0, 0) = 0 \tag{2.22}$$

$$h_1 = b(0, 1, 0, 1, 0) = 0 \tag{2.23}$$

$$h_1 = b(0, 1, 0, 0, 1) = 2 \tag{2.24}$$

in order. In view of these calculations, together with the definitions of matrices in Chapter 1, it follows that the matrix description of $B : R^2 \otimes R^3 \rightarrow R$ is given by

$$[B] = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2] \tag{2.25}$$

Observe that all the numerical information concerning B has been arrayed in $[B]$. It becomes increasingly clear then that such numerical entries define all possible bilinear forms of this type.

Example

In order to generalize the preceding example, one has only to be more general in describing the matrix of M. Suppose that

$$[M] = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \tag{2.26}$$

so that the bilinear operator b has action

$$b(f_1, f_2, g_1, g_2, g_3) = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{2.27}$$

Thus, it is easy to determine that

$$[B] = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{21} & m_{22} & m_{23} \end{bmatrix} \tag{2.28}$$

The two examples preceding help in visualizing the linear operator B by means of its matrix. They do not, however, contribute to the understanding of the nature of the tensor product of two vectors. For that purpose, it is appropriate to carry the examples a bit further.

Example

The foregoing example presents the following representations

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{2.29}$$

for u and

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{2.30}$$

for v. From the development of the ideas of the tensor product, it was established that $b(u, v) = B(u \otimes v)$. The construction of $u \otimes v$ proceeds according to definition in the manner

$$u \otimes v = \left(\sum_{i=1}^{2} f_i u_i \right) \otimes \left(\sum_{j=1}^{3} g_j v_j \right) \tag{2.31}$$

$$= \sum_{i=1}^{2} \sum_{j=1}^{3} f_i g_j u_i v_j \tag{2.32}$$

From this and the basis ordering chosen above, it is clear that the representation of $u \otimes v$ is given by

$$[u \otimes v] = [f_1 g_1 \quad f_1 g_2 \quad f_1 g_3 \quad f_2 g_1 \quad f_2 g_2 \quad f_2 g_3]^{\mathrm{T}} \tag{2.33}$$

The total picture for the tensor representation of $b(u, v)$, then, is

$$b(f_1,f_2,g_1,g_2,g_3)=\begin{bmatrix}f_1\\f_2\end{bmatrix}^{\top}\begin{bmatrix}m_{11}&m_{12}&m_{13}\\m_{21}&m_{22}&m_{23}\end{bmatrix}\begin{bmatrix}g_1\\g_2\\g_3\end{bmatrix} \tag{2.34}$$

$$=\begin{bmatrix}m_{11}&m_{12}&m_{13}&m_{21}&m_{22}&m_{23}\end{bmatrix}\left(\begin{bmatrix}f_1\\f_2\end{bmatrix}\otimes\begin{bmatrix}g_1\\g_2\\g_3\end{bmatrix}\right) \tag{2.35}$$

$$=\begin{bmatrix}m_{11}&m_{12}&m_{13}&m_{21}&m_{22}&m_{23}\end{bmatrix}\begin{bmatrix}f_1g_1\\f_1g_2\\f_1g_3\\f_2g_1\\f_2g_2\\f_2g_3\end{bmatrix} \tag{2.36}$$

The reader should have no difficulty in extending the notions of these examples to cases in which the dimensions of U and V differ from those used here. The extension to an X with dimension larger than 1 is similar in nature, and can be carried out row by row.

Example

Another sort of example, which is likely to be familiar to most readers, is the formation of the ordinary matrix product $m(P, Q) = PQ$ for compatible matrices P and Q over the field F. Clearly, the matrix product m is a bilinear operator. Thus, a linear operator M exists that has the property

$$m(P,Q)=M(P\otimes Q) \tag{2.37}$$

The matrix $P \otimes Q$ is known in the applications as the Kronecker product. If the basis vectors are chosen in the usual way, then its computation has the classical form. Thus, the Kronecker product of two matrices is seen to be the most general of all such products. Indeed, any other product, including the usual matrix product, can be found from the Kronecker product by multiplication with a matrix.

2.6 Multiple Products

It may happen that more than two vectors are multiplied together. Thus, certain famous and well-known field formulas include both crosses and dots. While the notion of multiple products is part and parcel of the concept of ring, so that no further adjustments need to be made there, one must undertake the question of how these multiple products are reflected back into the tensor concept. The purpose of this section, therefore, is to sketch the major ideas concerning such questions. A basic and natural step is the introduction of a generalization of bilinear operators.

For obvious reasons, not the least of which is the finite number of characters in the alphabet, it is now necessary to modify notation so as to avoid the proliferation of symbols. With regard to the foregoing discussion, the modification, which is straightforward, is to regard a bilinear operator in the manner b: $U_1 \times U_2 \to V$ in place of the previous $U \times V \to W$. Generalizing, consider p F-vector spaces U_i, $i = 1, 2, \ldots, p$. Let m: $U_1 \times U_2 \times \cdots \times U_p \to V$ be an operator that satisfies the following condition

$$\begin{aligned}m(u_1,u_2,\ldots,u_{i-1},f_iu_i+\breve{f}_i\breve{u}_i,u_{i+1},\ldots,u_p)&=f_im(u_1,u_2,\ldots,u_{i-1},u_i,u_{i+1},\ldots u_p)\\&\quad+\breve{f}_im(u_1,u_2,\ldots,u_{i-1},\breve{u}_i,u_{i+1},\ldots u_p)\end{aligned} \tag{2.38}$$

for $i = 1, 2, \ldots, p$, for all f_i and $\breve{f}_i$ in F, and for all u_i and $\breve{u}_i$ in U_i. Thus, m is said to be a p-linear operator. Observe in this definition that when $p - 1$ of the arguments of m are fixed, m becomes a linear operator in the remaining argument. Clearly, the bilinear operator is a special case of this definition, when $p = 2$. Moreover, the definition captures the intuitive concept of multiplication in a precise algebraic sense.

Next, the notion of tensor product is extended in a corresponding way. To do this, suppose that m and V satisfy two conditions. The first condition is that V is the smallest F-vector space that contains the image of m. Equivalently, V is the F-vector space generated by the image of m. Recall that the image of m is not equal to V, even in the bilinear case $p = 2$. The second condition is that

$$\breve{m}(u_1, u_2, \ldots, u_p) = \breve{M}(m(u_1, u_2, \ldots, u_p)) \tag{2.39}$$

where

$\breve{M}$: $V \rightarrow W$ is a linear operator
W is an F-vector space
$\breve{m}$: $U_1 \times U_2 \times \cdots \times U_p \rightarrow W$ is an arbitrary p-linear operator

If m satisfies these two conditions, the action of m is more traditionally written in the manner

$$m(u_1, u_2, \ldots, u_p) = u_1 \otimes u_2 \otimes \cdots \otimes u_p \tag{2.40}$$

and the space V is given by the notation

$$V = U_1 \otimes U_2 \otimes \cdots \otimes U_p \tag{2.41}$$

Once again, existence of the tensor product pair (m, V) is not a problem, and the same sort of uniqueness holds, that is, up to isomorphism.

It is now possible to give a major example of the multiple product idea. The general import of this example far exceeds the interest attached to more elementary illustrations. Therefore, it is accorded its own section. The reason for this will shortly become obvious.

2.7 Determinants

The body of knowledge associated with the theory of determinants tends to occupy a separate and special part of the memory which one reserves for mathematical knowledge. This theory is encountered somewhat indirectly during matrix inversion, and thus is felt to be related to the matrix algebra. However, this association can be somewhat misleading. Multiplication in the matrix algebra is really a multiplication of linear operators, but determinants are more naturally seen in terms of multiplication of vectors. The purpose of this section is to make this idea apparent, and to suggest that a natural way to correlate this body of knowledge is with the concept of an algebra constructed upon a given F-vector space. As such, it becomes a special case of the ideas previously introduced. Fitting determinants into the larger picture is then much less of a challenge than is usually the case, which can save precious human memory.

Consider at the outset a square array of field elements from F, denoted customarily by

$$D = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1p} \\ d_{21} & d_{22} & \cdots & d_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p1} & d_{p2} & \cdots & d_{pp} \end{pmatrix} \tag{2.42}$$

The determinant of D will be denoted by $\det(D)$. It is assumed that all readers are comfortable with at least one of the algorithms for computing $\det(D)$. The key idea about $\det(D)$ is that it is a p-linear

operator on its columns or upon its rows. In fact, two of the three classical properties of determinants are tantamount precisely to this statement. The third property, which concerns interchanging columns or rows, is also of great interest here (see below).

Without loss of generality, suppose that det(D) is regarded as p-linear function of its columns. If the columns, in order, are denoted by $d_1, d_2, \ldots, d_p$, then it is possible to set up a p-linear operator

$$m(d_1, d_2, \ldots, d_p) = \det(D) \tag{2.43}$$

Accordingly, tensor theory indicates that this operator can be expressed in the manner

$$m(d_1, d_2, \ldots, d_p) = M(d_1 \otimes d_2 \otimes \cdots \otimes d_p) \tag{2.44}$$

It is interesting to inquire about the nature of the matrix [M].

In order to calculate [M], it is necessary to select bases U_i, $i = 1, 2, \ldots, p$. In this case it is possible to identify U_i for each i with a fixed space U of dimension p. Let $\{u_1, u_2, \ldots, u_p\}$ be a basis for U and represent this basis by the standard basis vectors $\{e_1, e_2, \ldots, e_p\}$ in F^p. Moreover, select a basis for F and represent it by the multiplicative unit 1 in F. Then the elements of [M] are found by calculating

$$\det(e_{i_1} \quad e_{i_2} \quad \cdots \quad e_{i_p}) \tag{2.45}$$

for all sequences $i_1 i_2 \ldots i_p$ in the increasing numerical order introduced earlier. Thus if $p = 3$, this set of sequences is 111, 112, 113, 121, 122, 123, 131, 132, 133, 211, 212, 213, 221, 222, 223, 231, 232, 233, 311, 312, 313, 321, 322, 323, 331, 332, 333.

Example

For the case $p = 3$ described previously, it is desired to calculate [M]. The first few calculations are given by

$$\det(e_1 \quad e_1 \quad e_1) = 0 \tag{2.46}$$

$$\det(e_1 \quad e_1 \quad e_2) = 0 \tag{2.47}$$

$$\det(e_1 \quad e_1 \quad e_3) = 0 \tag{2.48}$$

$$\det(e_1 \quad e_2 \quad e_1) = 0 \tag{2.49}$$

$$\det(e_1 \quad e_2 \quad e_2) = 0 \tag{2.50}$$

$$\det(e_1 \quad e_2 \quad e_3) = +1 \tag{2.51}$$

$$\det(e_1 \quad e_3 \quad e_1) = 0 \tag{2.52}$$

$$\det(e_1 \quad e_3 \quad e_2) = -1 \tag{2.53}$$

$$\det(e_1 \quad e_3 \quad e_3) = 0 \tag{2.54}$$

$$\det(e_2 \quad e_1 \quad e_1) = 0 \tag{2.55}$$

$$\det(e_2 \quad e_1 \quad e_2) = 0 \tag{2.56}$$

$$\det(e_2 \quad e_1 \quad e_3) = -1 \tag{2.57}$$

$$\vdots$$

Rather than providing the entire list in this form, it is easier to give the elements in the right members of the equations. Employing determinant theory, it follows that those sequences with repeated subscripts correspond to 0. Moreover, interchanging two columns changes the sign of the determinant, the third property mentioned previously. Thus, the desired results are

$$0,0,0,0,0,+1,0,-1,0,0,0,-1,0,0,0,+1,0,0,0,+1,0,-1,0,0,0,0,0 \tag{2.58}$$

Then, [*M*] is a row matrix having these numerical entries. It is 1×27.

Example

The preceding example indicates that the formation of the determinant in tensor notation results in the appearance of numerous multiplications by zero. This is inefficient. Moreover, if all the zero entries in [*M*] are dropped, the result is a product of the form

$$[+1 \quad -1 \quad -1 \quad +1 \quad +1 \quad -1] \begin{bmatrix} d_{11}d_{22}d_{33} \\ d_{11}d_{32}d_{23} \\ d_{21}d_{12}d_{33} \\ d_{21}d_{32}d_{13} \\ d_{31}d_{12}d_{23} \\ d_{31}d_{22}d_{13} \end{bmatrix} \tag{2.59}$$

easily seen to be the standard formula for classical calculation of the determinant. In view of this result, one immediately wonders what to do about all the dropped zeros. The following section shows how to do away with all the zeros. In the process, however, more things happen than might have been anticipated; as a result, an entirely new concept appears.

2.8 Skew Symmetric Products

The determinant is an instance of skew symmetry in products. Consider a *p*-linear operator m: $U_1 \times U_2 \times \cdots \times U_p \rightarrow V$ with the property that each interchange of two arguments changes the sign of the result produced by m. Thus, for example,

$$m(u_1,\ldots,u_{i-1},u_i,\ldots,u_{j-1},u_j,\ldots,u_p) = -m(u_1,\ldots,u_{i-1},u_j,\ldots,u_{j-1},u_i,\ldots,u_p) \tag{2.60}$$

If a list of k interchanges is performed, the sign is changed k times. Such an operator is described as skew symmetric.

Provided that only skew symmetric multiplications are of interest, the tensor construction can be streamlined. Let (m_{skewsym}, V) be a pair consisting of a skew symmetric *p*-linear operator and an *F*-vector space V. This pair is said to constitute a skew symmetric tensor product for the *F*-vector spaces U_1, $U_2,\ldots,U_p$, if two conditions hold. The reader can probably guess what these two conditions are. Condition 1 is that V is the *F*-vector space generated by the image of m_{skewsym}. Condition 2 is the property that for every skew symmetric *p*-linear operator $\breve{m}_{\text{skewsym}}$: $U_1 \times U_2 \times \cdots \times U_p \rightarrow W$, a linear operator $\breve{M}_{\text{skewsym}}$: $V \rightarrow W$ exists having the feature

$$\breve{m}_{\text{skewsym}}(u_1,u_2,\ldots,u_p) = \breve{M}_{\text{skewsym}}(m_{\text{skewsym}}(u_1,u_2,\ldots,u_p)) \tag{2.61}$$

If these two conditions hold for the pair (m_{skewsym}, V), then the custom is to write

$$m_{\text{skewsym}}(u_1,u_2,\ldots,u_p) = u_1 \wedge u_2 \wedge \cdots \wedge u_p \tag{2.62}$$

for the action of m_{skewsym} and

$$V = U_1 \wedge U_2 \wedge \cdots \wedge U_p \tag{2.63}$$

for the product of the vector spaces involved. Once again, this skew symmetric tensor product exists, and is unique in the usual way.

Now suppose that $U_i = U$, $i = 1, 2, \ldots, p$ and that $\{u_1, u_2, \ldots, u_p\}$ is a basis for U. It is straightforward to show that $u_{i_1} \wedge u_{i_2} \wedge \cdots \wedge u_{i_p}$ vanishes whenever two of its arguments are equal. Without loss, assume that $u_{i_j} = u_{i_k} = u$. If u_{i_j} and u_{i_k} are switched, the sign of the product must change. However, after the switch, the argument list is identical to the previous list. Because the only number whose value is unchanged after negation is zero, the conclusion follows. Accordingly, the basis $U_1 \wedge U_2 \wedge \cdots \wedge U_p$ is the family

$$\{u_{i_1} \wedge u_{i_2} \wedge \cdots \wedge u_{i_p}\} \tag{2.64}$$

where each $i_1\, i_2 \cdots i_p$ consists of p distinct nonzero natural numbers, and where the ordinary convention is to arrange the $\{i_k\}$ so as to increase from left to right. A moment's reflection shows that only one such basis element can exist, which is $u_1 \wedge u_2 \wedge \cdots \wedge u_p$. Thus, if $p = 4$, the basis element in question is $u_1 \wedge u_2 \wedge u_3 \wedge u_4$. If we return to the example of the determinant, and regard it as a skew symmetric p-linear operator, then the following representation

$$m_{\text{skewsym}}(d_1, d_2, \ldots, d_p) = M_{\text{skewsym}}(d_1 \wedge d_2 \wedge \cdots \wedge d_p) \tag{2.65}$$

is obtained. Next observe that each of the p columns of the array D can be written as a unique linear combination of the basis vectors $\{u_1, u_2, \ldots, u_p\}$ in the manner

$$d_j = \sum_{i=1}^{p} d_{ij} u_i \tag{2.66}$$

for $j = 1, 2, \ldots, p$. Then, it follows that $d_1 \wedge d_2 \wedge \cdots \wedge d_p$ is given by

$$\sum_{i_1=1}^{p} \sum_{i_2=1}^{p} \cdots \sum_{i_p=1}^{p} d_{i_1 1} d_{i_2 2} \cdots d_{i_p p} u_{i_1} \wedge u_{i_2} \wedge \cdots \wedge u_{i_p} \tag{2.67}$$

which is a consequence of the fact that $\wedge$ is a p-linear operator. The only nonzero terms in this p-fold summation are those for which the indices $\{i_k,\ k = 1, 2, \ldots, p\}$ are distinct. The reader will correctly surmise that these terms are the building blocks of $\det(D)$. Indeed,

$$d_1 \wedge d_2 \wedge \cdots \wedge d_p = \det(D) u_1 \wedge u_2 \wedge \cdots \wedge u_p \tag{2.68}$$

and, if U is R^p with the $\{u_i\}$ chosen as standard basis elements, then

$$d_1 \wedge d_2 \wedge \cdots \wedge d_p = \det(D) \tag{2.69}$$

because $u_1 \wedge u_2 \wedge \cdots \wedge u_p$ becomes 1 in F. Moreover, it is seen from this example how the usual formula for $\det(D)$ is altered if the columns of D are representations with respect to a basis other than the standard basis. In turn, this shows how the determinant changes when the basis of a space is changed. The main idea is that it changes by a constant that is constructed from the determinant whose columns are the corresponding vectors of the alternate basis. Finally, because this new basis is given by an

invertible linear transformation from the old basis, it follows that the determinant of the transformation is the relating factor.

It can now be observed that the change from a tensor product based upon $\otimes$ to a tensor product based upon $\wedge$ has indeed eliminated the zero multiplications associated with skew symmetry. However, and this could possibly be a surprise, it has reduced everything to one term, which is the coordinate relative to the singleton basis element in a tensor product space of dimension one. This may be considered almost a tautology, except for the fact that it produces a natural generalization of the determinant to arrays in which the number of rows is not equal to the number of columns. Without loss, assume that the number of columns is less than the number of rows.

Example

Consider, then, an array of field elements from F, with fewer columns than rows, denoted by

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1p} \\ d_{21} & d_{22} & \cdots & d_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{q1} & d_{q2} & \cdots & d_{qp} \end{bmatrix} \tag{2.70}$$

where $p < q$. The apparatus introduced in this section still permits the formation of a skew symmetric p-linear operation in the manner $d_1 \wedge d_2 \wedge \cdots \wedge d_p$. This is a natural generalization in the sense that the ordinary determinant is recovered when $p = q$. Moreover, the procedure of calculation is along the same lines as before, with the following representations

$$d_j = \sum_{i=1}^{q} d_{ij} u_i \tag{2.71}$$

for $j = 1, 2, \ldots, p$. Note that the upper limit on the summation has changed from p to q. The reader will observe, then, that $d_1 \wedge d_2 \wedge \cdots \wedge d_p$ can be found once again by the familiar step

$$\sum_{i_1=1}^{q} \sum_{i_2=1}^{q} \cdots \sum_{i_p=1}^{q} d_{i_1 1} d_{i_2 2} \cdots d_{i_p p} u_{i_1} \wedge u_{i_2} \wedge \cdots \wedge u_{i_p} \tag{2.72}$$

which is a consequence once again of the fact that $\wedge$ is a p-linear operator. In this case, however, there is more than one way to form nonzero products in the family

$$\{u_{i1} \wedge u_{i2} \wedge \cdots \wedge u_{ip}\} \tag{2.73}$$

in which $i_1\, i_2 \cdots i_p$ contains p distinct numbers, and where the traditional convention is to arrange the $\{i_k\}$ so that the numbers in each list are increasing, while the numbers which these sequences represent are also increasing. This verbiage is best illustrated quickly.

Example

Suppose that $p = 3$ and $q = 4$. Thus, the sequences $i_1 i_2 i_3$ of interest are 123, 124, 134, 234. It can be observed that each of these sequences describes a 3×3 subarray of D in which the indices are associated with rows. In a sense these subarrays lead to all the interesting 3×3 minors of D, inasmuch as all the others are either zero or negatives of these four. Some of these designated four minors could also be zero, but that would be an accident of the particular problem instead of a general feature.

Example

This example investigates in greater detail the idea of $q > p$. Suppose that $p = 2$ and $q = 3$. Further, let the given array be

$$D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \end{pmatrix} \tag{2.74}$$

Choose the standard basis $\{e_1, e_2, e_3\}$ for $U = F^3$. Then,

$$d_1 = d_{11}e_1 + d_{21}e_2 + d_{31}e_3 \tag{2.75}$$

$$d_2 = d_{12}e_1 + d_{22}e_2 + d_{32}e_3 \tag{2.76}$$

from which one computes that

$$\begin{aligned} d_1 \wedge d_2 &= (d_{11}e_1 + d_{21}e_2 + d_{31}e_3) \wedge (d_{12}e_1 + d_{22}e_2 + d_{32}e_3) \\ &= (d_{11}d_{22} - d_{21}d_{12})e_1 \wedge e_2 \qquad (2.77) \\ &\quad + (d_{11}d_{32} - d_{31}d_{12})e_1 \wedge e_3 + (d_{21}d_{32} - d_{31}d_{22})e_2 \wedge e_3 \qquad (2.78) \end{aligned}$$

that evidently can be rewritten in the form

$$\begin{aligned} d_1 \wedge d_2 = &\det\begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} e_1 \wedge e_2 \\ &+ \det\begin{pmatrix} d_{11} & d_{12} \\ d_{31} & d_{32} \end{pmatrix} e_1 \wedge e_3 \\ &+ \det\begin{pmatrix} d_{21} & d_{22} \\ d_{31} & d_{32} \end{pmatrix} e_2 \wedge e_3 \end{aligned} \tag{2.79}$$

making clear the idea that the 2×2 minors of D become the coordinates of the expansion in terms of the basis $\{e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3\}$ for $R^3 \wedge R^3$.

2.9 Solving Linear Equations

An important application of Section 2.8 is to relate the skew symmetric tensor algebra to one's intuitive idea of matrix inversion. Consider the following linear equation

$$\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1p} \\ d_{21} & d_{22} & \cdots & d_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p1} & d_{p2} & \cdots & d_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{bmatrix} \tag{2.80}$$

With the aid of the usual notation for columns of D, rewrite this equation in the manner

$$\sum_{i=1}^{p} x_i d_i = c \tag{2.81}$$

where c is a vector whose ith element is c_i. To solve for x_k, multiply both members of this equation by the quantity

$$d_1 \wedge d_2 \wedge \cdots \wedge d_{k-1} \wedge d_{k+1} \wedge \cdots \wedge d_p \tag{2.82}$$

which will be denoted by d_{k-}. This multiplication can be done either on the left or the right, and the vector product which is used is $\wedge$. Multiplying on the left provides the result

$$x_k d_{k-} \wedge d_k = d_{k-} \wedge c \tag{2.83}$$

Now if $\det(D)$ is not zero, then this equation solves for

$$x_k = (d_{k-} \wedge c)/(d_{k-} \wedge d_k) \tag{2.84}$$

and this is essentially Cramer's rule. Using this rule conventionally one performs enough interchanges so as to move c to the kth column of the array. If these interchanges are performed in an analogous way with regard to d_k in the denominator, the traditional form of the rule results. This approach to the result shows that the solution proceeds by selecting multiplication by a factor which annihilates all but one of the terms in the equation, where each term is concerned with one column of D.

This is simply a new way of viewing an already known result. However, the treatment of Section 2.8 suggests the possibility of extending the construction to the case in which are found more unknowns than equations. The latter procedure follows via a minor adjustment of the foregoing discussion, and thus it seems instructive to illustrate the steps by means of an example.

Example

Consider the problem corresponding to $q = 3$ and $p = 2$ and given by three equations in two unknowns as follows:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \tag{2.85}$$

Begin by rewriting the equation in the form

$$x_1 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \tag{2.86}$$

To solve for x_2, multiply from the left with $[1\ 3\ 5]^{\mathrm{T}}$. This gives

$$x_2 \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \wedge \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \wedge \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \tag{2.87}$$

which then implies

$$x_2 \begin{bmatrix} -2 \\ -4 \\ -6 \end{bmatrix} = \begin{bmatrix} -13 \\ -26 \\ -13 \end{bmatrix} \tag{2.88}$$

which implies that $x_2 = 13/2$. Next, consider a left multiplication by $[2 \quad 4 \quad 6]^T$. Therefore,

$$x_1 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \wedge \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \wedge \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \tag{2.89}$$

which then implies

$$x_1 \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} -12 \\ -24 \\ -12 \end{bmatrix} \tag{2.90}$$

which implies that $x_1 = -6$. It is easy to check that these values of x_1 and x_2 are the unique solution of the equation under study.

The reader is cautioned that the construction of the preceding example produces necessary conditions on the solution. If any of these conditions cannot be satisfied, no solution can be found. On the other hand, if solutions to the necessary conditions are found, we must check that these solutions satisfy the original equation. Space limitations prevent any further discussion of this quite fascinating point.

2.10 Symmetric Products

The treatment presented in Section 2.8 has a corresponding version for this section. Consider a p-linear operator m: $U_1 \times U_2 \times \cdots \times U_p \to V$ with the property that each interchange of two arguments leaves the result produced by m unchanged. Symbolically, this is expressed by

$$m(u_1, \ldots, u_{i-1}, u_i, \ldots, u_{j-1}, u_j, \ldots, u_p) = m(u_1, \ldots, u_{i-1}, u_j, \ldots, u_{j-1}, u_i, \ldots, u_p) \tag{2.91}$$

Such an operator is said to be symmetric.

If only symmetric multiplications are of interest, the tensor construction can once again be trimmed to fit. Let (m_{sym}, V) be a pair consisting of a symmetric p-linear operator and an F-vector space V. This pair is said to constitute a symmetric tensor product for the F-vector spaces $U_1, U_2, \ldots, U_p$ if two conditions hold. First, V is the F-vector space generated by the image of m_{sym}; and, second, for every symmetric p-linear operator $\breve{m}_{\text{sym}}$: $U_1 \times U_2 \times \cdots \times U_p \to W$, linear operator $\breve{M}_{\text{sym}}$: $V \to W$ exists such that

$$\breve{m}_{\text{sym}}(u_1, u_2, \ldots, u_p) = \breve{M}_{\text{sym}}(m_{\text{sym}}(u_1, u_2, \ldots, u_p)) \tag{2.92}$$

In such a case, one writes

$$m_{\text{sym}}(u_1, u_2, \ldots, u_p) = u_1 \vee u_2 \vee \cdots \vee u_p \tag{2.93}$$

to describe the action of m_{sym} and

$$V = U_1 \vee U_2 \vee \cdots \vee U_p \tag{2.94}$$

for the symmetric tensor product of the vector spaces involved. As before, this symmetric tensor product exists and is essentially unique.

Next, let $U_i = U$, $i = 1, 2, \ldots, p$, and $\{u_1, u_2, \ldots, u_p\}$ be a basis for U. Because the interchange of two arguments does not change the symmetric p-linear operator, the basis elements are characterized by the family

$$\{u_{i_1} \vee u_{i_2} \vee \cdots \vee u_{i_p}\} \tag{2.95}$$

where each $i_1 i_2 \ldots i_p$ consists of all combinations of p nonzero natural numbers, written in increasing order, and where the ordinary convention is to arrange the basis vectors so that the numbers $i_1 i_2 \ldots i_p$ increase. Unlike the skew symmetric situation, quite a few such basis vectors can, in general, exist. For instance, the first basis element is $u_1 \vee u_1 \vee \cdots \vee u_1$ with p factors and the last one is $u_p \vee u_p \vee \cdots \vee u_p$, again with p factors.

Example

Suppose that $p = 3$ and that the dimension of U is 4. The sequences $i_1 i_2 i_3$ of interest in the representation of symmetric p-linear products are 111, 112, 113, 114, 122, 123, 124, 133, 134, 144, 222, 223, 224, 233, 234, 244, 333, 334, 344, 444. Section 2.7 showed a related example which produced 27 basis elements for the tensor product space built upon $\otimes$. In this case, it would be 64. The same situation in Section 2.8 produced four basis elements for a tensor product space constructed with $\wedge$. Twenty basis elements are found for the tensor product space produced with $\vee$. Notice that $20 + 4 \neq 64$. This means that the most general space based on $\otimes$ is not just a direct sum of those based on $\wedge$ and $\vee$.

Example

For an illustration of the symmetric product idea, choose $U = R^2$ and form a symmetric bilinear form in the arrangement of a quadratic form $u^{\mathrm{T}} M u$:

$$m_{\mathrm{sym}}(f_1, f_2) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{2.96}$$

A word about the matrix M is in order. Because of the relationship

$$M = \frac{1}{2}(M + M^{\mathrm{T}}) + \frac{1}{2}(M - M^{\mathrm{T}}) \tag{2.97}$$

it is easy to see that M may be assumed to be symmetric without loss of generality, as the remaining term in this representation of m_{sym} (f_1, f_2) leads to a zero contribution in the result. Thus, one is concerned with a natural candidate for the symmetric tensor mechanism. The tensor construction begins by considering

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \vee \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{2.98}$$

Choose a standard basis $\{e_1, e_2\}$ for R^2. Then the expression introduced previously becomes

$$\sum_{i=1}^{2} \sum_{j=1}^{2} f_i f_j e_i \vee e_j \tag{2.99}$$

which becomes

$$f_1^2 e_1 \vee e_1 + 2 f_1 f_2 e_1 \vee e_2 + f_2^2 e_2 \vee e_2 \tag{2.100}$$

The result may be represented with the matrix

$$\begin{bmatrix} f_1^2 & 2 f_1 f_2 & f_2^2 \end{bmatrix}^{\mathrm{T}} \tag{2.101}$$

Because $p = 2$, the basis vectors of interest are seen to be $\{e_1 \vee e_1, e_1 \vee e_2, e_2 \vee e_2\}$. Inserted into the expression for m_{sym}, these produce the results m_{11}, $m_{12} = m_{21}$, m_{22}, respectively. Thus,

$$[M_{sym}] = [m_{11} \quad m_{12} \quad m_{22}] \tag{2.102}$$

Finally, the symmetric tensor expression for m_{sym} is

$$m_{sym}(f_1, f_2) = [m_{11} \quad m_{12} \quad m_{22}] \begin{bmatrix} f_1^2 \\ 2f_1 f_2 \\ f_2^2 \end{bmatrix} \tag{2.103}$$

Example

If M, as in the previous example, is real and symmetric, then it is known to satisfy the equation

$$ME = E\Lambda \tag{2.104}$$

where E is a matrix of eigenvectors of M, satisfying $E^T E = I$, and Λ is a diagonal matrix of eigenvalues $\{\lambda_i\}$, which are real. Then,

$$M = E\Lambda E^T \tag{2.105}$$

and the quadratic form $u^T M u$ becomes

$$u^T M u = [E^T u]^T \Lambda [E^T u] \tag{2.106}$$

$$= \sum_{i=1}^{p} \lambda_i [E^T u]_i \tag{2.107}$$

When one considers u to be an arbitrary vector in R^p, this quadratic form is nonnegative for all u if and only if the $\{\lambda_i\}$ are nonnegative and is positive for all nonzero u if and only if the $\{\lambda_i\}$ are positive. M is then nonnegative definite and positive definite, respectively.

Example

With reference to the preceding example, it is sometimes of interest to choose $U = C^p$, where the reader is reminded that C denotes the complex numbers. In this case a similar discussion can be carried out, with M as a complex matrix. The quadratic form must be set up a bit differently, with the structure $u^* M u$, in which superscript * denotes a combined transposition and conjugation. Also, without loss one can assume that $M^* = M$. A common instance of this sort of situation occurs when M is a function $M(s)$ of the Laplace variable s, which is under consideration on the axis $s = j\omega$.

2.11 Summary

The basic idea of this chapter is to examine the axiomatic framework for equipping an F-vector space V with a vector multiplication, and thus developing it into an algebra. As vector multiplication is manifestly a nonlinear operation, a very useful matrix theory can be developed for such multiplications. The treatment is based upon notions of algebraic tensor products from which all other multiplications can

be derived. The authors demonstrate that the determinant is nothing but a product of vectors, with a special skew symmetric character attached to it. Specializations of the tensor product to such cases, and to the analogous case of symmetric products, were discussed.

As a final remark, it should be mentioned that tensor products develop into a complete algebra of their own. Although space does not permit treatment here, note that the key idea is the definition

$$(v_1 \otimes \cdots \otimes v_p) \otimes (v_{p+1} \otimes \cdots \otimes v_q) = v_1 \otimes \cdots \otimes v_q \tag{2.108}$$

Reference

1. W. H. Greub, *Multilinear Algebra*, New York: Springer-Verlag, 1967.

3

Laplace Transformation

3.1 Introduction **3**-1

3.2 Motivational Example **3**-2
Series RLC Circuit • Homogeneous Solution and the Natural Response • Nonhomogeneous Solution and the Forced Response • Total Solution • Scrutinizing the Solution • Generalizing the Phasor Concept: Onward to the Laplace Transform

3.3 Formal Developments **3**-8
Definitions of the Unilateral and Bilateral Laplace Transforms • Existence of the Laplace Transform • Example of Laplace Transform Computations and Table of Unilateral Laplace Transforms • Poles and Zeros—Part I • Properties of the Laplace Transform • Inverse Laplace Transform

3.4 Laplace Transform Analysis of Linear Systems **3**-32
Solution of the System Differential Equation • System Function • Poles and Zeros—Part II: Stability Analysis of Systems • Laplace-Domain Phasor Circuits

3.5 Conclusions and Further Readings **3**-42

3.6 Appendix A: The Dirac Delta (Impulse) Function **3**-42

3.7 Appendix B: Relationships among the Laplace, Fourier, and *z*-Transforms **3**-44

References **3**-45

John R. Deller, Jr.
Michigan State University

3.1 Introduction

The Laplace transform (LT) is the cornerstone of classical circuits, systems, and control theory. Developed as a means of rendering cumbersome differential equation solution simple algebraic problems, the engineer has transcended this original motivation and has developed an extensive toolbag of analysis and design methods based on the "*s*-plane." After a motivating differential equation (circuit) problem is presented, this chapter introduces the formal principles of the LT, including its properties and methods for forward and inverse computation. The transform is then applied to the analysis of circuits and systems, exploring such topics as the system function and stability analysis. Two appendices conclude the chapter, one of which relates the LT to other signal transforms to be covered in this book.

3.2 Motivational Example

3.2.1 Series RLC Circuit

Let us motivate our study of the LT with a simple circuit example. Consider the series RLC circuit shown in Figure 3.1, in which we leave the component values unspecified for the moment. With the input and output to the circuit taken to be the voltages x and y, respectively, the input–output dynamics of this circuit are found to be governed by a linear, constant-coefficient differential equation

$$x(t) = LC\frac{d^2y}{dt^2} + RC\frac{dy}{dt} + y(t) \tag{3.1}$$

This equation arises from a circuit example, but is typical of many second-order systems arising in mechanical, fluid, acoustic, biomedical, chemical, and other engineering models. We can, therefore, view the circuit as a "system," and the theory explored here as having broad applicability in modeling and analysis of systems.

Suppose we are asked to find the complete solution for the output voltage y, given input

$$x(t) = M_x e^{\sigma_x t}\cos(\omega_x t + \theta_x)u(t) \tag{3.2}$$

in which u denotes the unit step function,

$$u(t) \stackrel{\text{def}}{=} \{0,\ t < 0 \quad 1/2,\ t = 0 \quad 1,\ t > 0\} \tag{3.3}$$

For convenience and without loss of generality we assume that $M_x > 0$. The initial conditions (at time $t(0^-)$) on the circuit are given to be* $i(0^-) = i_0$ and $y(0^-) = y_0$ where y_0 and i_0 are known quantities.

3.2.2 Homogeneous Solution and the Natural Response

The homogeneous or complementary solution of the differential equation, say $\tilde{y}$, represents the natural, or unforced, response of the system. The natural response occurs because of the inequity between the initial conditions and the conditions imposed upon the system by the input at the instant it is applied. The system must adjust to these new conditions and will do so in accordance with its own physical properties (e.g., circuit values). For stable systems (see Section 3.4.3), the natural response will consist of transients, signals which decay exponentially with time. The unforced response will always be present unless the system is stable, and either the input was applied at time $t = -\infty$ (transient will have diminished in the infinite time interval prior to time zero) or the initial conditions on the system exactly nullify the "shock" of the input so that a transient adjustment is not necessary. The form of the natural response (homogeneous solution) is not dependent on the input (except for its use in determining changes around time $t = 0$), but rather on the inherent properties of the system.

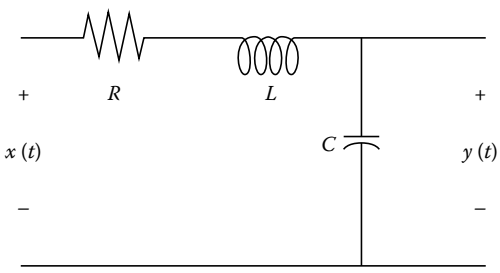


FIGURE 3.1 Series RLC circuit example.

* The notations 0^+ and 0^- indicate limits at $t = 0$ from the right and left, respectively. That $y(0^+) = y(0^-)$, for example, indicates continuity of $y(t)$ at $t = 0$.

The homogeneous solution is found by initially seeking the input-free response which may be written as

$$0 = (LC\mathcal{D}^2 + RC\mathcal{D} + 1)y(t) \quad \text{with } \mathcal{D}^i \stackrel{\text{def}}{=} \frac{\mathrm{d}^i}{\mathrm{d}t^i} \tag{3.4}$$

The characteristic equation is therefore

$$0 = (LCp^2 + RCp + 1) \tag{3.5}$$

Solving for the roots, we find

$$p_1, p_2 = \frac{-RC \pm \sqrt{R^2C^2 - 4LC}}{2LC} = -\frac{R}{2L} \pm \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}} \tag{3.6}$$

In general, p_1 and p_2 can be real and unequal (overdamped case), equal and real (critically damped case), or complex conjugates (underdamped case). Except in the critically damped case (in which $R^2C^2 = 4LC$ so that two identical real roots are found), the homogeneous solution takes the form.

$$\tilde{y}(t) = Ae^{p_1 t} + Be^{p_2 t} \tag{3.7}$$

with A and B to be specified at the end of the solution.

For the sake of discussion, assume the underdamped case in which the natural response will be oscillatory, corresponding to a complex-conjugate pair of roots of Equation 3.5. In this case we have $R^2C^2 < 4LC$ and $p_2 = p_1^*$. Let us define $p_h \stackrel{\text{def}}{=} p_1$, so the two roots are p_h and $= p_h^*$. The meaning of the subscript "h" will become clear later. Some manipulation of Equation 3.7 will simplify our future work. We have

$$\tilde{y}(t) = Ae^{p_h t} + Be^{p_h^* t} \tag{3.8}$$

We observe that A and B must be complex conjugates if $\tilde{y}$ is to be a real signal. Thus, we write

$$\tilde{y}(t) = Ae^{p_h t} + A^* e^{p_h t} \tag{3.9}$$

After some manipulation using Euler's relation, $e^{j\alpha} = \cos(\alpha) + j\,\sin(\alpha)$, we have

$$\begin{aligned} \tilde{y}(t) &= 2A_{\text{re}} e^{\sigma_h t} \cos(\omega_h t) - 2A_{\text{im}} e^{\sigma_h t} \sin(\omega_h t) \\ &= 2|A| e^{\sigma_h t} \cos(\omega_h t + \theta_A) \end{aligned} \tag{3.10}$$

where $A = A_{\text{re}} + jA_{\text{im}} = |A|\, e^{j\theta_A}$ and $p_h = \sigma_h + j\omega_h$. The numbers A_{re} and A_{im}, or, equivalently, $|A|$ and θ_A, are to be determined later.* Note that the number p_h is often called the complex frequency associated with the damped† sinusoid. The complex frequency is simply a convenient mathematical way to hold the damping and frequency information in one quantity. Later in the chapter we see that p_h is also a "pole" of the system being analyzed.

---

* In fact, because A_{re} and A_{im} are unknowns, we could replace $2A_{\text{re}}$, $2A_{\text{im}}$, and $2|A|$ with some simpler notations if desired.

† We use the term "damping" to refer to the real part of any complex frequency, "σ," with the understanding that two other cases are actually possible: If $\sigma = 0$ the signal is undamped, and if $\sigma > 0$, the signal is exponentially increasing.

3.2.3 Nonhomogeneous Solution and the Forced Response

The nonhomogeneous, or particular solution, say $\tilde{\tilde{y}}$, represents the system's forced, or driven, response. For certain prevalent types of inputs, the forced response represents an attempt to "track" the forcing function in some sense. If the natural response in a particular problem is transient and the forced response is not, then a "long time" (theoretically, $t \to \infty$) after the input is applied, only this "tracking" response will remain. In the present circuit example the forced response is uniquely present as $t \to \infty$ if $\sigma_h < 0$ (natural response exponentially decays) and $\sigma_x \geq 0$ (forcing function persists in driving the circuit for all time). Further, in the special case in which the forcing function is a periodic or constant signal (an undamped sinusoid, undamped complex exponential, or a constant (in each case $\sigma_x = 0$)), this "tracking" response, in the absence of any transients, is called the steady-state response of the system. Note that the forced response will be present for all $t > 0$, but may become uniquely evident only after the transient natural response dies out. Also, note that the forced response might never become evident if it is itself a transient ($\sigma_x < 0$), even though in this case the forced response will still represent an attempt to track the input.

For mathematical convenience in finding $\tilde{\tilde{y}}$, we let x be replaced by

$$\begin{aligned} x(t) &= M_x e^{\sigma_x(j\omega_x t + 0_x)} u(t) \\ &= [M_x e^{\sigma_x t}\cos(\omega_x t + \theta_x)] + [jM_x e^{\sigma_x t}\sin(\omega_x t + \theta_x)]u(t) \end{aligned} \tag{3.11}$$

Because of the linearity of the system, the real and imaginary parts of the solution $\tilde{\tilde{y}}$ will correspond to the real and imaginary parts of the complex x. Because we want the response to the real part (the cosine), we simply take the real part of the solution at the end.* It is extremely useful to rewrite Equation 3.11 as

$$x(t) = M_x e^{j\theta_x} e^{(\sigma_x + j\omega_x)t} u(t) = M_x e^{j\theta_x} e^{p_x t} u(t) \tag{3.12}$$

We shall call the complex number $\overline{X} = M_x e^{j\theta_x}$ a generalized phasor for the sinusoid, noting that the quantity is a conventional phasor (see, e.g., [5]) when $\sigma_x = 0$ (i.e., when x represents an undamped sinusoid). The complex frequency associated with the signal x is $p_x \stackrel{\text{def}}{=} \sigma_x + j\omega_x$.

Any signal that can be written as a sum of exponentials will be an eigensignal of a linear, time-variant (LTI) system such as the present circuit. The forced response of a system to an eigensignal is a scaled, time-shifted version of the eigensignal. This means that an eigensignal generally has its amplitude and phase altered by the system, but never its frequency! Many signals used in engineering analysis are eigensignals. This is the case with the present input x.

Because x is an eigensignal, the nonhomogeneous solution will be of the form

$$\tilde{\tilde{y}}(t) = |H(p_x)| M_x e^{\sigma_x t} e^{j(\omega_x t + \theta_x + \arg\{H(p_x)\})} = M_y e^{j\theta_y} e^{P_x t} \tag{3.13}$$

where $|H(p_x)|$ represents the amplitude scaling imposed upon a signal of complex frequency p_x, and arg $\{H(p_x)\}$ is the phase shift. For the moment, do not be concerned about the seemingly excessive notation $|H(p_x)|$ and $\arg\{H(p_x)\}$. The number $H(p_x) = |H(p_x)| e^{j\arg\{H(p_x)\}}$, called the eigenvalue of the system for complex frequency p_x, is a package containing the scaling and phase change induced upon a sinusoid or exponential input of complex frequency p_x. In Equation 3.13 we have implicitly defined $M_y \stackrel{\text{def}}{=} |H(p_x)| M_x$ and $\theta_y = \theta_x + \arg\{\mathrm{H(p_x)}\}$, noting that

$$\overline{Y} = M_y e^{j\theta_y} = H(p_x) M_x e^{j\theta_x} = H(p_x)\overline{X} \tag{3.14}$$

is the generalized phasor for the forced response $\tilde{\tilde{y}}$.

* Alternatively, we could also find the response $x^*(t)$, and average the two responses at each t.

Let us now put the expressions (Equations 3.12 and 3.13) into the differential equation (Equation 3.1) (ignoring the $u(t)$ because we are seeking a solution for $t > 0$)

$$M_x e^{j\theta_x} e^{p_x t} = p_x^2 LCM_y e^{j\theta_y} e^{p_x t} + p_x RCM_y e^{j\theta_y} e^{p_x t} + M_y e^{j\theta_y} e^{p_x t} \tag{3.15}$$

Dividing through by e^{pxt} (note this critical step),

$$M_x e^{j\theta_x} = p_x^2 LCM_y e^{j\theta_y} + p_x RCM_y e^{j\theta_y} + M_y e^{j\theta_y} \tag{3.16}$$

Isolating $M_y e^{j\theta_y}$ on the left side of Equation 3.16, we have

$$M_y e^{j\theta_y} = \frac{M_x e^{j\theta_x}}{LCp_x^2 + RCp_x + 1} \tag{3.17}$$

Because all quantities on the right are known, we can now solve for M_y and θ_y. For example (so that we can compare this result with future work), suppose we have system parameters

$$L = 2 \text{ H}, \quad C = 1 \text{ F}, \quad R = 1\ \Omega, \quad y_0 = \frac{1}{2} \text{ V}, \quad i_0 = 0 \text{ A} \tag{3.18}$$

and signal parameters

$$M_x = 3V, \quad \sigma_x = -0.1/\text{s}, \quad \omega_x = 1\,\text{rad/s} \quad \theta_x = \pi/4\,\text{rad} \tag{3.19}$$

Then we find $M_y e^{j\theta_y} = 2.076e^{-j(0.519\pi)}$.

Whatever the specific numbers, let us now assume that M_y and θ_y are known. We have

$$\tilde{\tilde{y}}(t) = M_y e^{j\theta_y} e^{(\sigma_x + j\omega_x)t} \tag{3.20}$$

Taking the real part,

$$\tilde{\tilde{y}}(t) = M_y e^{\sigma_x t} \cos(\omega_x t + \theta_y) \tag{3.21}$$

This nonhomogeneous solution is valid for $t > 0$

3.2.4 Total Solution

Combining the previous results, we obtain the complete solution for $t > 0$.

$$y(t) = \tilde{y}(t) + \tilde{\tilde{y}}(t) = 2|A|e^{\sigma_h t} \cos(\omega_h t + \theta_A) + M_y e^{\sigma_x t} \cos(\omega_x t + \theta_y) \tag{3.22}$$

We must apply the initial conditions to find the unknown numbers $|A|$ and θ_A. By a physical considerations we know $y(0^+) = y(0^-) = y_0$ and $i(0^+) = i(0^-) = i_0$, so

$$y(0) = y_0 = 2|A| \cos(\theta_A) + M_y \cos(\theta_y) \tag{3.23}$$

and

$$i(0) = i_0 = C\frac{dy}{dt}\bigg|_{t=0} = 2|A|C[\sigma_h \cos(\theta_A) - \omega_h \sin(\theta_A)] + M_y C[\sigma_h \cos(\theta_A) - \omega_h \sin(\theta_A)] \tag{3.24}$$

These two equations can be solved for $|A|$ and θ_A. For example, for the numbers given in Equations 3.18 and 3.19, using Equation 3.6 we find that $p_h = -1/4 + j\sqrt{7}/4$ and $A = 0.416e^{j(0.230\pi)}$. Whatever the specific numbers, let us assume that $|A|$ and θ_A are known. Then, putting all the known numbers back into Equation 3.23 gives a complete solution for $t > 0$.

3.2.5 Scrutinizing the Solution

The first term in the final solution, Equation 3.22, comprises the unforced response and corresponds to the homogeneous solution of the differential equation in conjunction with the information provided by the initial conditions. Notice that this response involves only parameters dependent upon the circuit components, e.g., σ_h and ω_h, and information provided by the initial conditions, $A = |A|e^{j\theta A}$. The latter term in Equation 3.22 is the forced response. We reemphasize that this part of the response, which corresponds to the nonhomogeneous solution to the differential equation, is of the same form as the forcing function x and that the system has only scaled and time-shifted (as reflected in the phase angle) the input.

It is important to understand that the natural or unforced response is not altogether unrelated to the forcing function. The adjustment that the circuit must make (using its own natural modes) depends on the discrepancy at time zero between the actual initial conditions on the circuit components and the conditions the input would "like" to impose on the components as the forcing begins. Accordingly, we can identify two parts of the natural solution, one due to the initial energy storage, the other to the "shock" of the input at time zero. We can see this in the example above by reconsidering Equations 3.23 and 3.24 and rewriting them as

$$y_0 = 2A_{\text{re}} + M_y \cos(\theta_y) \tag{3.25}$$

and

$$\frac{i_0}{C} = 2[A_{\text{re}}\sigma_h - A_{\text{im}}\omega_h] + M_y\left[\sigma_x \cos(\theta_y) - \omega_x \sin(\theta_y)\right] \tag{3.26}$$

Solving

$$A_{\text{re}} = \frac{y_0}{2} - \frac{M_y \cos(\theta_y)}{2} \tag{3.27}$$

$$A_{\text{im}} = \frac{\sigma_h y_0 - (i_0/C)}{2\omega_h} + \frac{M_y \cos(\theta_y)(\sigma_x - \sigma_h) - M_y \sin(\theta_y)\omega_x}{2\omega_h} \tag{3.28}$$

We see that each of the real and imaginary parts of the complex number A can be decomposed into a part depending on initial circuit conditions, y_0 and i_0, and a part depending on the system's interaction with the input at the initial instant. Accordingly, we may write

$$A = A_{\text{ic}} + A_{\text{input}} \tag{3.29}$$

where $A_{\text{ic}} = A_{\text{re,ic}} + jA_{\text{im,ic}}$ and $A_{\text{input}} = A_{\text{re,input}} + jA_{\text{im,input}}$. In polar form $A_{\text{ic}} = |A_{\text{ic}}|e^{j\theta A_{\text{ic}}}$ and $A_{\text{input}} = |A_{\text{input}}|e^{j\theta A_{\text{input}}}$. Therefore, the homogeneous solution can be decomposed into parts

$$\tilde{y}(t) = \tilde{y}_{\text{ic}}(t) + \tilde{y}_{\text{input}}(t) = 2|A_{ic}|e^{\sigma_h t}\cos(\omega_h t + \theta_{A_{\text{ic}}}) + 2\left|A_{\text{input}}\right|e^{\sigma_h t}\cos(\omega_h t + \theta_{A_{\text{input}}}) \tag{3.30}$$

Hence, we observe that a natural response may occur even if the initial conditions on the circuit are zero. The combined natural and forced response in this case, $\tilde{\tilde{y}}(t) + \tilde{y}_{\text{input}}(t)$, is called the zero-state response to indicate the state of zero energy storage in the system at time $t = 0$. On the other hand, the response to initial energy *only*, $\tilde{y}_{\text{ic}}$, is called the zero-input response for the obvious reason.

3.2.6 Generalizing the Phasor Concept: Onward to the Laplace Transform

To begin to understand the meaning of the LT, we reflect on the process of solving the circuit problem above. Although we could examine this solution deeply to understand the LT connections to both the natural and forced responses, it is sufficient for current purposes to examine only the forced solution.

Because the input to the system is an eigensignal previously, we could assume that the form of $\tilde{\tilde{y}}$ would be identical to that of x, with modifications only to the magnitude and phase. In noting that both x and $\tilde{\tilde{y}}$ would be of the form $Me^{j\theta}$, it seems reasonable that the somewhat tedious nonhomogeneous solution would eventually reduce to an algebraic solution to find M_y and θ_y from M_x and θ_x. All information needed and sought is found in the phasor quantities $\overline{X}$ and $\overline{Y}$ in conjunction with the system information. The critical step which converted the differential equation solution to an algebraic one comes in Equation 3.16 in which the superfluous term $e^{p_x t}$ is divided out of the equation.

Also observe that the ratio $H(p_x) = M_y e^{j\theta_y}/M_x e^{j\theta_x}$ depends only on system parameters and the complex frequency of the input, $p_x = \sigma_x + j\omega_x$. In fact, this ratio, when considered a function, e.g., H, of a general complex frequency, say, $s = \sigma + j\omega$, is called the system function for the circuit. In the present example, we see that

$$H(s) = \frac{1}{LCs^2 + RCs + 1} \tag{3.31}$$

The complex number $H(s)$, $s = \sigma + j\omega$, contains the scaling and delay (phase) information induced by the system on a signal with damping σ and frequency ω.

Let us now consider a slightly more general class of driving signals. Suppose we had begun the analysis above with a more complicated input of the form*

$$x(t) = \sum_{i=1}^{N} M_i e^{\sigma_i t} \cos(\omega_i t + \theta_i) \tag{3.32}$$

which, for convenience, would have been replaced by

$$x(t) = \sum_{i=1}^{N} M_i e^{j\theta_i} e^{(\sigma_i + j\omega_i)t} = \sum_{i=1}^{N} M_i e^{j\theta_i} e^{p_i t} \tag{3.33}$$

in the nonhomogeneous solution. It follows immediately from linearity that the solution could be obtained by entering each of the components in the input individually, and then combining the N solutions at the output. In each case we would clearly need to rid the analysis of the superfluous term of the form $e^{p_i t}$ by division. This information is equivalent to the form $e^{\sigma_i t}\cos(\omega_i t)$ which is known to automatically carry through to the output.

Now, recalling that $M_i e^{j\theta i}$ is the generalized phasor for the ith component in Equation 3.33, let us rewrite this expression as

* We omit the unit step u which appears in the input above because we are concerned only with the forced response for $t > 0$.

$$x(t) = \sum_{i=1}^{N} \overline{X}(p_i) e^{p_i t} \tag{3.34}$$

where $\overline{X}(p_i) \stackrel{\text{def}}{=} M_i e^{j\theta_i}$. Expression (Equation 3.34) is similar to a Fourier series (FS) (see Chapter 4), except that here (unless all $\sigma_i = 0$) the signal is only "pseudoperiodic" in that all of its sinusoidal components may be decaying or expanding in amplitude. The generalized phasors $\overline{X}(p_i)$ are similar to FS coefficients and contain all the information (amplitude and phase) necessary to obtain steady-state solutions. These phasors comprise frequency domain information as they contain packets of amplitude and phase information for particular complex frequencies.

With the concepts gleaned from this example, we are now prepared to introduce LT in earnest.

3.3 Formal Developments

3.3.1 Definitions of the Unilateral and Bilateral Laplace Transforms

Most signals used in engineering analysis and design of circuits and filters can be modeled as a sort of limiting case of Equation 3.34. Such a representation includes not just several complex frequency exponentials as in Equation 3.34, but an uncountably infinite number of such exponentials, one for every possible value of frequency ω. Each of these exponentials is weighted by a "generalized phasor" of infinitesimal magnitude. The exponential at complex frequency $s = \sigma + j\omega$, for example, is weighted by phasor $\overline{X}(\sigma + j\omega)\mathrm{d}\omega/2\pi$, where the differential dω assures the infinitesimal magnitude and the scale factor of 2π is included by convention. The uncountably infinite number of terms is "summed" by integration as follows:

$$x(t) = \int_{-\infty}^{\infty} \overline{X}(\sigma + j\omega) \frac{\mathrm{d}\omega}{2\pi} e^{(\sigma + j\omega)t} \tag{3.35}$$

The number σ in this representation is arbitrary as long as the integral exists. In fact, if the integral converges for any σ, then the integral exists for an uncountable infinite number of σ.

The complex function $\overline{X}(\sigma + j\omega)$ in Equation 3.35 is the LT for the signal $x(t)$. Based on the foregoing discussion, we can interpret the LT as a complex-frequency-dependent, uncountable infinite set of "phasor densities" containing all the magnitude and phase information necessary to find forced solutions for LTI systems. We use the word "density" here to indicate that the LT at complex frequency $\sigma + j\omega$ must be multiplied by the differential $\mathrm{d}\omega/2\pi$ to become properly analogous to a phasor. The LT, therefore has, for example, units volts per hertz. However, we find that the LT is much more than just a phasor-like representation, providing a rich set of analysis tools with which to design and analyze systems, including unforced responses, transients, and stability.

As in the preceding simpler examples, the solution of differential equations will be made easier by ridding the signals of superfluous complex exponentials of form $e^{(\sigma + j\omega)}$, that is, by working directly with LTs. Before doing so we change variables, to put Equation 3.35 into a more conventional form. Let s denote the general complex frequency $s \stackrel{\text{def}}{=} \sigma + j\omega$. Then

$$x(t) = \frac{1}{j2\pi} \int_{\sigma - j\infty}^{\sigma + j\infty} X(s) e^{st} \mathrm{d}s \tag{3.36}$$

where we have dropped the bar over the LT, X. This integral, which we have interpreted as an "expansion" of the signal x in terms of an uncountably infinite set of infinitesimal generalized phasors

and complex exponentials, offers a means for obtaining the signal x from the LT, X. Accordingly, Equation 3.36 is known as the inverse Laplace transform (inverse LT). The inverse LT operation is often denoted

$$x(t) = \mathcal{L}^{-1}\{X(s)\} \tag{3.37}$$

How one would evaluate such an integral and for what values of s it would exist are issues we shall address later.

In order to rid the signal x of the superfluous factors e^{st}, we can simply compute the LT. Without any rigorous attempt to derive the transform from Equation 3.36, it is believed that

$$X(s) = \int_{-\infty}^{\infty} \frac{x(t)}{e^{st}}\,\mathrm{d}t = \int_{-\infty}^{\infty} x(t)e^{-st}\,\mathrm{d}t \tag{3.38}$$

This is the bilateral, or two-sided, Laplace transform (BLT). The descriptor ''bilateral'' or ''two-sided'' is a reference to the fact that the signal may be nonzero in both positive and negative time. In contrast, the unilateral, or one-sided, Laplace transform (ULT) is defined as

$$X(s) = \int_{0^-}^{\infty} \frac{x(t)}{e^{st}}\,\mathrm{d}t = \int_{0^-}^{\infty} x(t)e^{-st}\,\mathrm{d}t \tag{3.39}$$

When a signal is zero all for $t < 0$, the ULT and BLT are identical. The same inverse LT, Equation 3.36, is applied in either case, with the understanding that the resulting time signal is zero by assumption in the ULT case. While the BLT can be used to treat a more general class of signals, we find that ULT has the advantage of allowing us to find the component of the natural response due to nonzero initial conditions. In other words, ULT is used to analyze signals that ''start'' somewhere, a time we conventionally call* $t = 0$.

These transformations are reminiscent of the process of dividing through by the complex exponential which was first encountered in the forced solution in the motivating circuit problem (see Equation 3.16).

3.3.2 Existence of the Laplace Transform

The variable $s = \sigma + j\omega$ is a complex variable over which the LT is calculated. The complex plane with σ along the abscissa and $j\omega$ on the ordinate, is called the s-plane. We find some powerful tools centered on the s-plane below. Note that the s-plane is not the LT, nor can the LT be ''plotted'' in the s-plane. The LT is a complex function of the complex variable s, and a plot of the LT would require another two dimensions ''over'' the s-plane. For this reason, we need to place some constraints on either s or $X(s)$ or both to create a plot. For example, we could use the LT to plot $|X(j\omega)|$ as a function of ω, by evaluating the magnitude of $X(s)$ along the $j\omega$-axis in the s-plane.† An illustration of these points is found in Figure 3.2.

We now address the question: For what values of s (i.e., ''where'' in the s-*p*lane) does the LT exist? Consider first a two-sided (in time) signal, *x*. x is assumed piecewise continuous in every finite interval of the real line. We assert that the BLT X ordinarily exists in the s-plane in a strip of the form

$$\sigma_+ < \mathrm{Re}\{s\} = \sigma < \sigma_- \tag{3.40}$$

as illustrated in Figure 3.3. In special cases the BLT may converge in the half-plane $\sigma_+ < \sigma$, or the half-plane $\sigma < \sigma_-$, or even in the entire s-plane.

---

* Note that if we apply the ULT to a signal $x(t)$ that ''starts'' in negative time, the result will be identical to that for the signal $x(t)\,u(t)$.

† This particular plot is equivalent to the magnitude spectrum of the signal that could be obtained using Fourier techniques discussed in Chapter 4.

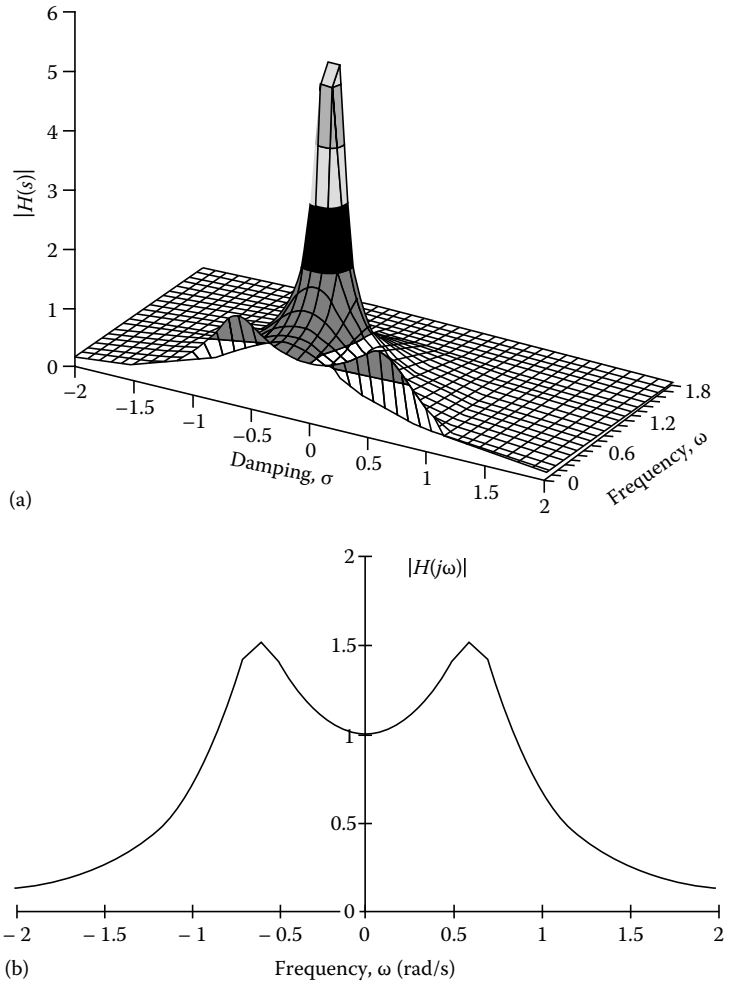


FIGURE 3.2 LT is called "H" instead of "X" in this figure for a reason to be discovered later. (a) A plot of $|H(s)|$ vs. s for the LT $H(s) = (0.5)/(s^2 + 0.5s + 0.5)$. Only the upper-half s-plane is shown ($\omega \geq 0$). Note that the peak occurs near the value $p_h = -1/4 + j\sqrt{7}/4$, as a root of the denominator of $H(s)$, which we shall later call a pole of the LT. The LT is theoretically infinite at $s = p_h$. (b) Evaluation of $|H(s)|$ along the $j\omega$-axis (corresponding to $\sigma = 0$) with the magnitude plotted as function of ω.

The boundary values σ_+ and σ_- are associated with the positive-time and negative-time portions of the signal, respectively. The minimum possible value of σ_+ and maximum possible value of σ_- are called the abscissas of absolute convergence. We henceforth use the notations σ_+ and σ_- to explicitly mean these extreme values. The vertical strip between but exclusive of, these abscissas is called the region of convergence (ROC) for the LT. In the special cases the ROC extends indefinitely from a single abscissa to the right ($\sigma_+ < \sigma$) or left ($\sigma < \sigma_-$) or covers the entire s-plane.

To verify that Equation 3.40 is a correct description of the ROC consider the positive-time part of x first. We maintain that $X(s)$ will exist on any s such that $\mathrm{Re}\{s\} = \sigma > \sigma_+$, if and only if (iff) a positive M exists such x is bounded by $Me^{\sigma_+ t}$ on $t \geq 0$,

$$|x(t)| < Me^{\sigma_+ t} \quad t \geq 0 \tag{3.41}$$

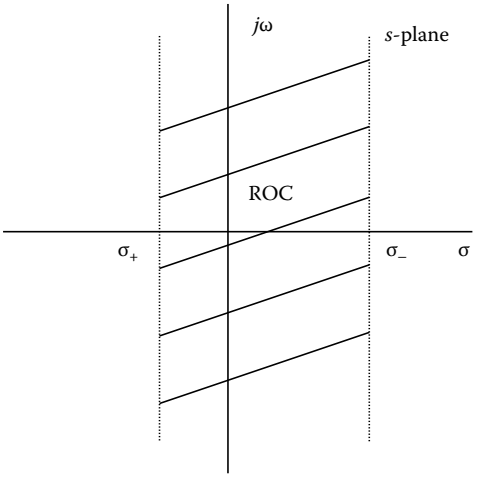


FIGURE 3.3 Except in special cases, the region of convergence for the BLT will be a vertical strip in the *s*-plane. This strip need not contain the *j*ω-axis as is the case in this illustration.

Under this condition (letting X_+ denote the LT of nonnegative-time part of x),

$$\begin{aligned}|X_+(s)| &= \left|\int_0^\infty x(t)e^{-st}\mathrm{d}t\right| \\ &\le \int_0^\infty |x(t)e^{-st}|\mathrm{d}t \\ &\le \int_0^\infty |x(t)|e^{-\sigma t}\mathrm{d}t < \int_0^\infty Me^{(\sigma_+-\sigma)t}\mathrm{d}t \\ &= \begin{cases} \dfrac{M}{\sigma-\sigma_+}, & \sigma > \sigma_+ \\ \infty, & \text{otherwise} \end{cases}\end{aligned} \tag{3.42}$$

If there is no finite, positive M such that Equation 3.30 holds, then the signal grows faster than $e^{-\sigma+t}$ and the LT integral (area under $x(t)e^{-\sigma+t}e^{j\omega t}$) will not converge for at least some value of s in the neighborhood of the vertical line $s=\sigma_+$. In this case, σ_+ is not a proper abscissa.

By similar means, we can argue that the negative-time part of x must be bounded as

$$|x(t)| < Me^{\sigma_- t}, \quad t < 0 \tag{3.43}$$

for some positive M, if the corresponding part of the LT, say X_-, is to converge at s such that $\mathrm{Re}\{s\}=\sigma<\sigma_-$. Similarly to Equation 3.42 it is shown that

$$\begin{aligned}|X_-(s)| &= \left|\int_{-\infty}^0 x(t)e^{-st}\mathrm{d}t\right| < \int_{-\infty}^0 Me^{(\sigma_--\sigma)t}\mathrm{d}t \\ &= \begin{cases} \dfrac{M}{\sigma_--\sigma}, & \sigma > \sigma_- \\ \infty, & \text{otherwise} \end{cases}\end{aligned} \tag{3.44}$$

and X_- will not converge in some left neighborhood of $\sigma=\sigma_-$ if the condition (Equation 3.43) is not met.

It should be clear from the discussion of the BLT that a ULT will ordinarily converge in the half-plane $\mathrm{Re}\{s\}=\sigma>\sigma_+$ iff a positive M exists such that Equation 3.41 is met. This follows immediately from the fact that the ULT of $x(t)$ is equivalent to the BLT of $x(t)u(t)$. The ''negative-time'' part of $x(t)$ yields $X_-(s)=0$ which converges everywhere in the s-plane. The ULT may also converge everywhere in the s-plane in special cases.

3.3.3 Example of Laplace Transform Computations and Table of Unilateral Laplace Transforms

A listing of commonly used ULTs with ROCs is given in Table 3.1. Each entry in the table can be verified by direct computation of the appropriate LT integral, or in many cases, properties of the LT can be exploited to make the computation easier. These properties are developed in Section 3.3.5.

TABLE 3.1 Table of ULT Pairs

Signal, $x(t)$	ULT, $X(s)$	ROC
$d(t)$	1	Entire s-plane
$\frac{d^k\delta}{dt^k}$	s^k	Entire s-plane
$u(t)$	$\frac{1}{s}$	$\{s : \sigma > 0\}$
$t^{n-1}u(t)$	$\frac{(n-1)!}{s^n}$	$\{s : \sigma > 0\}$
$e^{p_x t}u(t)$	$\frac{1}{(s-p_x)}$	$\{s : \sigma > \sigma_x\}$
$t^n e^{p_x t}\ u(t)$	$\frac{n!}{(s-p_x)^{n+1}}$	$\{s : \sigma > \sigma_x\}$
$\cos(\omega_x t + \theta_x)u(t)$	$\frac{s\cos(\theta_x) - \omega_x \sin(\theta_x)}{s^2 + \omega_x^2}$	$\{s : \sigma > 0\}$
$e^{\sigma_x t}\cos(\omega_x t + \theta_x)\ u(t)$	$\frac{(s-\sigma_x)\cos(\theta_x) - \omega_x \sin(\theta_x)}{(s-\sigma_x)^2 + \omega_x^2}$	$\{s : \sigma > \sigma_x\}$
$M_x t^{n-1} e^{\sigma_x t}\cos(\omega_x t + \theta_x)u(t)$	$\frac{E}{(s-p_x)^n} + \frac{E*}{(s-p_x*)^n}$; M_x real and positive, $E = \frac{M_x}{2}(n-1)!e^{j\theta_x}$	$\{s : \sigma > \sigma_x\}$

Note: In each entry, s is the general complex number $\sigma + j\omega$, and, where relevant, p_x is the specific complex number $\sigma_x + j\omega_x$.

It is rare to find a table of BLTs in engineering material because most of our work is done with the ULT. However, BLTs can often be found by summing the results of two ULTs in the following way. Suppose x is written as

$$x(t) = x_+(t) + x_-(t) \tag{3.45}$$

where x_+ and x_- are the causal and anticasual parts of the signal, respectively. To obtain X_+, we can use a ULT table. To obtain X_-, note the following easily demonstrated property of the BLT: If $\mathcal{L}\{y(t)\} = Y(s)$ with ROC $\{s$: Re$\{s\} > a\}$, then $\mathcal{L}\{y(-t)\} = Y(-s)$ with ROC $\{s$: Re$\{s\} > -a\}$. Therefore, we can find $\mathcal{L}\{x_-(-t)\}$ in a ULT table, and replace the argument s by $-s$ to obtain X_(s). The $X(s) = X_+(s) + X_-(s)$. This strategy is illustrated in Example 3.1. The ROC of the sum will ordinarily be the intersection of the individual LTs X_+ and X_-, but the total ROC may be larger than this if a pole–zero cancellation occurs (see Section 3.3.4).

Let us consider some examples which illustrate the direct forward computation of the LT and the process discussed in the preceding paragraph.

Example 3.1

Find the BLT and ULT for the signal

$$x(t) = Ae^{at}u(-t) + Be^{bt}u(t) \quad |A|,|B| < \infty \tag{3.46}$$

Note that A, B, a, and b may be complex.

Solution

In the bilateral case, we have

$$\begin{aligned} X(s) &= \int_{-\infty}^{0_-} Ae^{at}e^{-st} + \int_{-\infty}^{0_-} Be^{bt}e^{-st}\mathrm{d}t \\ &= \frac{Ae^{(a-s)t}}{a-s}\Big|_{-\infty}^{0-} + \frac{Be^{(b-s)t}}{b-s}\Big|_{0-}^{\infty} \\ &= \begin{bmatrix} -\dfrac{A}{s-a}, & \mathrm{Re\{s\}} < \mathrm{Re\{a\}} \\ \infty, & \text{otherwise} \end{bmatrix} \\ &\quad + \begin{bmatrix} \dfrac{B}{s-b}, & \mathrm{Re\{s\}} > \mathrm{Re\{b\}} \\ \infty, & \text{otherwise} \end{bmatrix} \end{aligned} \tag{3.47}$$

The ROC for this LT is $\{s\colon \mathrm{Re}\{b\} < \mathrm{Re}\{s\} < \mathrm{Re}\{a\}\}$. The LT does not exist for any s for which $\mathrm{Re}\{s\} \geq \mathrm{Re}\{a\}$ or $\mathrm{Re}\{s\} \leq \mathrm{Re}\{b\}$. Note also that when $\mathrm{Re}\{b\} \geq \mathrm{Re}\{a\}$, then no ROC can be found, meaning that the BLT does not exist anywhere for the signal.

The ULT follows immediately from the previous work. We have

$$X(s) = \int_{0_-}^{\infty} Be^{bt}e^{-st}\mathrm{d}t = \begin{cases} \dfrac{B}{s-b}, & \mathrm{Re\{s\}} > \mathrm{Re\{b\}} \\ \infty, & \text{otherwise} \end{cases} \tag{3.48}$$

The ROC in this case is $\{s\colon \mathrm{Re}\{s\} > \mathrm{Re}\{b\}\}$. We need not be concerned about the negative-time part of the signal (and the associated ROC) because the LT effectively zeros the signal on $t < 0$.

Note. The result of this example is worth committing to memory because it will reappear frequently.

Note that if we had found the ULT, Equation 3.48, for the casual part of x in a table (call it $X_+(s)$), then we could employ the trick suggested above to find the LT for the anticausal part. Let x_- denote the negative-time part: $x_-(t) = Ae^{at}u(-t)$. We know that the LT of $x_-(-t) = Ae^{-at}u(t)$ (a casual signal) is

$$X_-(-s) = \frac{A}{s+a}, \quad \text{with ROC } \{s\colon \mathrm{Re\{s\}} > -\mathrm{a}\} \tag{3.49}$$

Therefore,

$$X_-(s) = \frac{-A}{s-a}, \quad \text{with ROC } \{s\colon \mathrm{Re\{s\}} < \mathrm{a}\} \tag{3.50}$$

The overall BLT result is then $X(s) = X_+(s) + X_-(s)$ with ROC equal to the intersection of the individual results. This is consistent with the BLT found by direct integration.

The preceding simple example suggests that the BLT can treat a broader class of signals at the expense of greater required care in locating its ROC. A further and related complication of the BLT is the nonuniqueness of the transform with respect to the time signals. Consider Example 3.2.

Example 3.2

Find the BLT for the following signals:

$$x_1(t) = e^{bt}u(t) \quad \text{and} \quad x_2(t) = -e^{bt}u(-t) \tag{3.51}$$

Solution

From our work in Example 3.1, we find immediately that

$$X_1(s) = \frac{1}{s-b}, \quad \text{Re\{s\}} > \text{Re\{b\}} \quad \text{and}$$
$$X_2(s) = \frac{1}{s-b}, \quad \text{Re\{s\}} < \text{Re\{b\}} \tag{3.52}$$

Neither X_1 nor X_2 can be unambiguously associated with a time signal without knowledge of its ROC.

Another drawback of the BLT is its inability to handle initial conditions in problems like the one that motivated our discussion. For this, reason, and also because signals tend to be casual (occurring only in positive time) in engineering problems, the ULT is more widely used and we shall focus on it exclusively after treating one more important topic in Section 3.3.4. Before moving to Section 3.3.4, let us tackle a few more example computations.

Example 3.3

Find the ULT of the impulse function, $\delta(t)$ (see Section 3.6).

Solution

$$\Delta(s) = \int_{0^-}^{\infty} \delta(t) e^{-st} dt = 1 \quad \text{for all } s \tag{3.53}$$

The LT converges everywhere in the s-plane. We note that the lower limit 0^- is important here to yield the answer 1 (which will provide consistency of the theory) instead of 1/2.

Example 3.4

Find the ULT of the unit step function, $u(t)$ (see Equation 3.3).

Solution

$$U(s) = \int_{0^-}^{\infty} 1 e^{-st} dt = \frac{-e^{-st}}{s}\bigg|_{0^-}^{\infty} = \frac{1}{s} \quad \text{for Re}\{s\} > 0 \tag{3.54}$$

The ROC for this transform consists of the entire right-half s-plane exclusive of the $j\omega$-axis.

Example 3.5

Find the ULT of the damped ($\sigma_x < 0$), undamped ($\sigma_x = 0$), or expanding ($\sigma_x > 0$) sinusoid, $x(t) = M_x e^{\sigma_x t} \cos(\omega_x t \mid \theta_x)\, u(t)$.

Solution

Using Euler's relation, write x as

$$x(t) = \frac{M_x}{2} e^{\sigma_x t}\left[e^{j(\omega_x t + \theta_x)} + e^{-j(\omega_x t + \theta_x)}\right] = \frac{M_x}{2}\left[e^{j\theta_x} e^{p_x t} + e^{-j\theta_x} e^{p_x^* t}\right] \tag{3.55}$$

with $p_x \stackrel{\text{def}}{=} \sigma_x + j\omega_x$. Taking the LT,

$$\begin{aligned} X(s) &= \frac{M_x}{2}\int_{0-}^{\infty}\left[e^{j\theta_x}e^{p_x t} + e^{-j\theta_x}e^{p_x^* t}\right]e^{-st}\,\mathrm{d}t \\ &= \int_{0-}^{\infty}\frac{M_x}{2}e^{j\theta_x}e^{p_x t}e^{-st}\,\mathrm{d}t + \int_{0-}^{\infty}\frac{M_x}{2}e^{-j\theta_x}e^{p_x^* t}e^{-st}\,\mathrm{d}t \end{aligned} \tag{3.56}$$

Now using Equation 3.48 on each of the integrals

$$X(s) = \frac{(M_x/2)e^{j\theta_x}}{(s-p_x)} + \frac{(M_x/2)e^{-j\theta_x}}{(s-p_x^*)} \tag{3.57}$$

with the ROC associated with each of the terms being $\mathrm{Re}\{s\} > \mathrm{Re}\{p_x\} = \sigma_x$. Putting the fractions over a common denominator yields

$$\begin{aligned} X(s) &= \frac{M_x}{2}\left[\frac{(s-p_x^*)e^{j\theta_x} + (s-p_x)e^{-j\theta_x}}{(s-p_x)(s-p_x^*)}\right] \\ &= \frac{M_x}{2}\left[\frac{\left(se^{j\theta_x} + se^{-j\theta_x} - p_x^*e^{j\theta_x} - p_x e^{-j\theta_x}\right)}{s^2 - 2\mathrm{Re}\{p_x\}s + |p_x|^2}\right] \\ &= M_x\left[\frac{(s-\sigma_x)\cos(\theta_x) - \omega_x\sin(\theta_x)}{(s-\sigma_x)^2 + \omega_x^2}\right] \end{aligned} \tag{3.58}$$

The ROC of X is $\{s\colon \mathrm{Re}\{s\} > \sigma_x\}$.

Note. The chain of denominators in Equation 3.58 is worth noting because these relations occur frequently in LT work.

$$(s-p_x)(s-p_x^*) = s^2 - 2\mathrm{Re}\{p_x\}s + |p_x|^2 = s^2 - 2\sigma_x s + |p_x|^2 = (s-\sigma_x)^2 + \omega_x^2 \tag{3.59}$$

3.3.4 Poles and Zeros—Part I

"Pole–zero" analysis is among the most important uses of the LT in circuit and system design and analysis. We need to take a brief look at some elementary theory of functions of complex variables in order to carefully describe the meaning of a pole or zero. When we study methods of inverting the LT in a future section, this side trip will prove to be especially useful.

Let us begin with a general function, F, of a complex variable s. We stress that $F(s)$ may or may not be an LT. F is said to be analytic at $s = a$ if it is differentiable at a and in a neighborhood of a. For example, $F(s) = s - 1$ is analytic everywhere (or entire), but $G(s) = |s|$ is nowhere analytic because its derivative exists only at $s = 0$. On the other hand, a point p is an isolated singular point of F if the derivative F does not exist at p, but F is analytic in a neighborhood of p. The function $F(s) = e^{-s}/(s-1)$ has a singular point at $s = 1$. There is a circular analytic domain around any singular point, p, of F, say $\{s\colon |s-p| < p\}$, in which the function F can be represented by a Laurent series [3],

$$F(s) = \sum_{i=0}^{\infty} q_{i,p}(s-p)^i + \sum_{i=1}^{\infty}\frac{r_{i,p}}{(s-p)^i} \tag{3.60}$$

The second sum in Equation 3.60 is called the principle part of the function F at p. When the principle part of F at p contains terms up to order n, the isolated singular point p is called an nth-order pole of F.

Evidently from Equation 3.60, F tends to infinity at a pole and the order of infinity is n. For future reference, we note that the complex number $r_p \stackrel{\text{def}}{=} r_{1,p}$ is called the residue of F at $s = p$.

A zero of F is more simply defined as a value of s, say z, at which F is analytic and for which $F(z) = 0$. If all the derivatives up to the $(m-1)$st are also zero at z, but mth is nonzero, then z is called an mth-order zero of F. It can be shown that the zeroes of an analytic function F are isolated, except in trivial case $F(s) = 0$ for all s [3].

Most LTs encountered in signal and system problems are quotients of polynomials in s, say

$$X(s) = \frac{N(s)}{D(s)} \tag{3.61}$$

because of the signals employed in engineering work, and because (as we shall see later) rational LTs are naturally associated with LTI systems. N and D connote "numerator" and "denominator." In this case both N and D are analytic everywhere in the s-plane, and the poles and zeroes of X are easily found by factoring the polynomials N and D, to express X in the form

$$X(s) = C\left(\prod_{i=1}^{n_N}(s - z_i) \Big/ \prod_{i=1}^{n_D}(s - p_i)\right) \tag{3.62}$$

where

n_N is the number of simple factors in $N(s)$ (order of N in s)
n_D is the number of simple factors in $D(s)$ (order of D in s)
C is a constant
X is called a proper rational LT if $n_D > n_N$

After canceling all factors common to the numerator and denominator, if m terms $(s - z)$ are left in the numerator, then X has an mth order zero at $s = z$. Similarly, if n terms $(s - p)$ are left in the denominator, then X has an nth order pole at $s = p$.

Although the LT does not exist outside the ROC, all of the poles will occur at values of s outside the ROC. None, some, or all of the zeros may also occur outside the ROC. This does not mean that the LT is valid outside the ROC, but that its poles and zeroes may occur there. A pole is ordinarily indicated in the s-plane by the symbol $\times$; whereas a zero is marked with a small circle $\circ$.

Example 3.6

Find the poles and zeros of the LT

$$X(s) = \frac{3s^2 + 9s + 9}{(s+2)(s^2 + 2s + 2)} \tag{3.63}$$

Solution

Factoring the top and bottom polynomials to put X in form Equation 3.62, we have

$$X(s) = 3\frac{\left(s + \left(3 - j\sqrt{3}\right)/2\right)\left(s + \left(3 + j\sqrt{3}\right)/2\right)}{(s+2)(s+1+j)(s+1-j)} \tag{3.64}$$

There are first-order zeros at $s = (-3 + j\sqrt{3})/2$ and $s = (-3 - j\sqrt{3})/2$, and first order-poles at $s = -2$, $s = -1 + j$, and $s = -1 - j$. The pole–zero diagram appears in Figure 3.4.

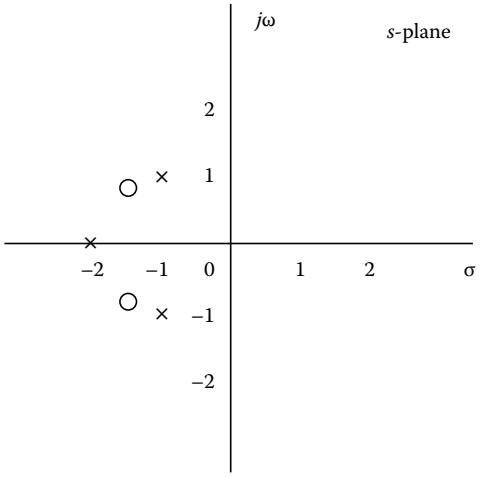


FIGURE 3.4 Pole–zero diagram for Example 3.6.

Two points are worth noting. First, complex poles and zeros will always occur in conjugate pairs, as they have here, if the LT corresponds to a real signal. Second, the denominator of Equation 3.64 also can be expressed as $(s+2)[(s+1)^2+1]$ (recall Equation 3.59). Comparing this form with the LT obtained in Example 3.5 suggests that the latter form might prove useful.

The purpose of introducing poles and zeros at this point in our discussion is to note the relationship of these singularities to the ROC. The preceding examples illustrate the following facts:

1. For a "right-sided" (nonnegative-time only) signal, x, the ROC of LT X (either ULT or BLT) is $\{s\colon \mathrm{Re}\{s\} > \mathrm{Re}\{p_+\} = \sigma_+\}$, where p_+ is the pole of X with maximum real part, namely, σ_+. If X has no poles, then the ROC is the entire s-plane.
2. For a "left-sided" (negative-time only) signal, x, the ROC of BLT X is $\{s\colon \mathrm{Re}\{s\} < \mathrm{Re}\{p_-\} = \sigma_-\}$, where p_- is the pole of X with minimum real part, namely, σ_-. If X has no poles, then the ROC is the entire s-plane.
3. For a "two-sided" signal x, the ROC of the BLT X is $\{s\colon \mathrm{Re}\{p_+\} = \sigma_+ < \mathrm{Re}\{s\} < \mathrm{Re}\{p_-\} = \sigma_-\}$ where p_+ is the pole of maximum real part associated with the right-sided part of x, and p_- is the pole of minimum real part associated with the left-sided part of x. If the right-sided signal has no pole, then the ROC extends indefinitely to the right in the s-plane. If the left-sided signal has no pole, then the ROC extends indefinitely to the left in the s-plane. Therefore, if neither part of the signal has a pole, then the ROC is the entire s-plane.

Let us revisit three of the previous examples to verify these claims. In Example 3.1, we found the ROC for the BLT to be $\{s\colon \mathrm{Re}\{b\} < \mathrm{Re}\{s\} < \mathrm{Re}\{a\}\}$. The only pole associated with the left-sided sequence is at $s = a$. The only pole associated with the right-sided signal occurs at $s = b$. Following rule 3 in the list above yields exactly the ROC determined by analytical means. The poles of X as well as the ROC are shown in Figure 3.5a.

In Example 3.4, we found the ROC to be the entire right-half s-plane, exclusive of the $j\omega$-axis. The single pole of $U(s) = 1/s$ occurs at $s = 0$. Figure 3.5b is consistent with rule 1.

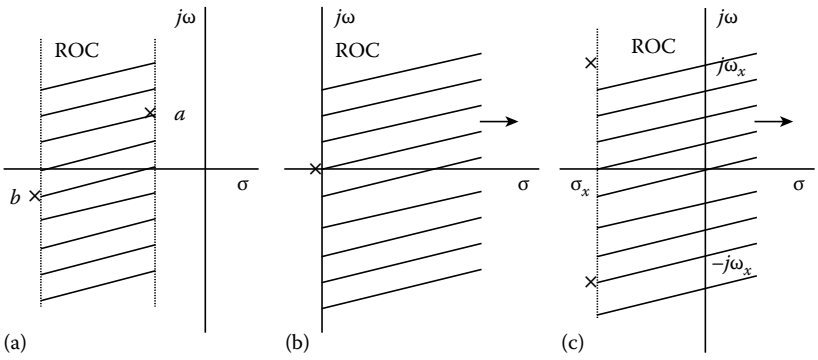


FIGURE 3.5 Pole–zero plots ROCs for the LTs of (a) Examples 3.1, (b) 3.4, and (c) 3.5.

The ULT of Example 3.5 has poles at $s = \sigma_x \pm j\omega_x$ and a zero at $s = \sigma_x + \omega_x \tan(\theta_x)$. Rule 1 therefore specifies that the ROC should be $\{s\colon \mathrm{Re}\{s\} > \mathrm{Re}\{\sigma_x \pm j\omega_x\} = \sigma_x\}$, which is consistent with the solution to Example 3.5. The pole–zero plot and ROC are illustrated in Figure 3.5c.

3.3.5 Properties of the Laplace Transform*

This section considers some properties of the LT which are useful in computing forward and inverse LTs, and in other manipulations occurring in signal and system design and analysis. A list of properties appears in Table 3.2.

In most cases, the verification of these properties follows in a straightforward manner from the definition of the transform. Consider the following examples. For convenience, we define the notation

$$x(t) \leftrightarrow X(s) \tag{3.65}$$

to mean that x and X are an LT pair, $X(s) = \mathcal{L}\{x(t)\}$.

TABLE 3.2 Operational Properties of the ULT

Description of Operation	Formal Operation	Corresponding LT
Linearity	$\alpha x(t) + \beta y(t)$	$\alpha X(s) + \beta Y(s)$
Time delay ($t_0 > 0$)	$x(t - t_0)u(t - t_0)$	$e^{-st_0} X(s)$
Exponential modulation in time (or complex frequency ("s") shift)	$e^{s_0 t} x(t)$	$X(s - s_0)$
Multiplication by t^k, $k = 1, 2, \ldots$	$t^k x(t)$	$(-1)^k \dfrac{\mathrm{d}^k X}{\mathrm{d}s^k}$
Time differentiation	$\dfrac{\mathrm{d}^k x}{\mathrm{d}t^k}$	$s^k X(s) - \displaystyle\sum_{i=0}^{k-1} s^i x^{(k-1-i)}(0^-)$
		$x^{(i)}(0^-) \stackrel{\text{def}}{=} \left.\dfrac{\mathrm{d}^i x}{\mathrm{d}t^i}\right\|_{t=0^-}$
Time integration	$\displaystyle\int_{-\infty}^{t} x(\lambda)\mathrm{d}(\lambda)$	$\dfrac{X(s)}{s} + \dfrac{x^{(-1)}(0^-)}{s}$
		$x^{(-1)}(0^-) \stackrel{\text{def}}{=} \left.\displaystyle\int_{-\infty}^{t} x(\lambda)\mathrm{d}(\lambda)\right\|_{t=0^-}$
Convolution	$\displaystyle\int_0^{\infty} x(\lambda) y(t - \lambda)\mathrm{d}\lambda$	$\boldsymbol{X(s)Y(s)}$
Correlation	$\displaystyle\int_0^{\infty} x(t) y(t + \tau)\mathrm{d}t$	$\boldsymbol{X(s)Y(-s)}$
Product (s-domain convolution)	$x(t)y(t)$	$\dfrac{1}{j2\pi}\displaystyle\int_{\sigma - j\infty}^{\sigma + j\infty} X(\lambda)Y(s - \lambda)\mathrm{d}\lambda$
Initial signal value (if time limit exists)	$\lim_{t \to 0^+} x(t)$	$\lim_{s \to \infty} sX(s)$
Final signal value (if time limit exists)	$\lim_{t \to \infty} x(t)$	$\lim_{s \to 0} sX(s)$
Time scaling	$x(\alpha t)$, $\alpha > 0$	$\dfrac{1}{\alpha} X\left(\dfrac{s}{\alpha}\right)$
Periodicity (period T)	$\displaystyle\sum_{i=0}^{\infty} x(t - iT)$	$\dfrac{X(s)}{(1 - e^{-sT})}$
	$x(t) = 0,\ t \notin [0, T]$	

Note: Throughout, X and Y are LTs of signals x and y, respectively; x and y are casual signals.

* Henceforth, this study restricts attention to the ULT and uses the acronym "LT" only.

Example 3.7

Verify the modulation property of the LT which states that if $x(t) \leftrightarrow X(s)$, then $e^{s0t}\, x(t) \leftrightarrow X(s - s_0)$.

Solution

By definition,

$$\begin{aligned}\mathcal{L}\{e^{s_0 t}x(t)\} &= \int_{0^-}^{\infty} e^{s_0 t}x(t)e^{-st}\,\mathrm{d}t \\ &= \int_{0^-}^{\infty} x(t)e^{-(s-s_0)}\,\mathrm{d}t = X(s - s_0)\end{aligned} \tag{3.66}$$

Example 3.8

Verify the periodicity property and find the LT for a square wave of period $T = 2$ and duty cycle 1/2.

Solution

Using the linearity and time-delay properties of the LT

$$\mathcal{L}\left\{\sum_{i=0}^{\infty} x(t - iT)\right\} = \sum_{i=0}^{\infty} X(s)e^{-siT} = X(s)\sum_{i=0}^{\infty} e^{-siT} = \frac{X(s)}{(1 - e^{-sT})} \tag{3.67}$$

Let us call the square wave $z(t)$ and its LT $Z(s)$. Now one period of z can be written as $x(t) = u(t) - u(t-1)$, $0 \le t < 2$ (see Figure 3.6). Using the delay property, therefore, $X(s) = (1/s) - (e^{-s}/s)$. Using Equation 3.67 with $T = 2$, we have

$$Z(s) = \frac{(1/s) - (e^{-s}/s)}{(1 - e^{-2s})} = \frac{(1 - e^{-s})}{s(1 - e^{-2s})} \tag{3.68}$$

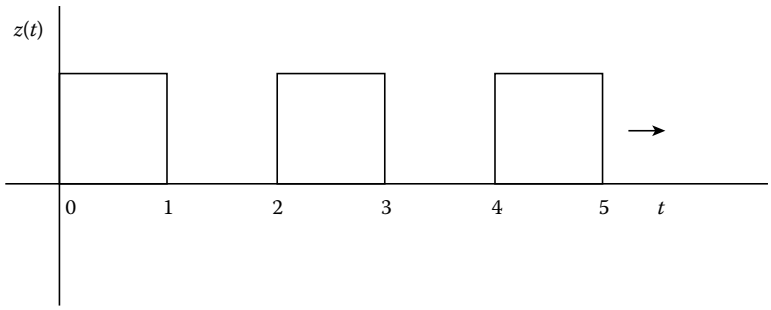


FIGURE 3.6 Square wave of Example 3.8.

Example 3.9

Verify the time-differentiation property of the LT which states that if $x(t) \leftrightarrow X(s)$, then $dx/dt \leftrightarrow sX(s) - x(0^-)$.

Solution

By definition, $\mathcal{L}\{dx/dt\} = \int_{0^-}^{\infty} (dx/dt)e^{-st}dt$. Integrating by parts yields

$$\mathcal{L}\left\{\frac{dx}{dt}\right\} = x(t)e^{-st}\Big|_{0^-}^{\infty} + s\int_{0^-}^{\infty} x(t)e^{-st}dt = sX(s) - x(0^-) \tag{3.69}$$

Example 3.10

Verify the initial value theorem of the LT which states that if $x(0^+) = \lim_{t\downarrow 0} x(t) < \infty$, then $\lim_{s\to\infty} sX(s) = x(0^+)$.

Solution

In case a discontinuity exists in x at $t = 0$, define the signal

$$y(t) = x(t) - c\delta(t) \tag{3.70}$$

where c is the amplitude shift at the discontinuity, $c = x(0^+) - x(0^-)$. Then y will be continuous at $t = 0$ (see Figure 3.7). Further,

$$\frac{dx}{dt} = \frac{dy}{dt} + c\delta(t) \tag{3.71}$$

so that using the time-differentiation property and the fact that $\mathcal{L}\{c\delta(t)\} = c$, we have

$$sX(s) - x(0^-) = \int_{0^-}^{\infty} \frac{dy}{dt}e^{-st}dt + c \tag{3.72}$$

Because $c = x(0^+) - x(0^-)$,

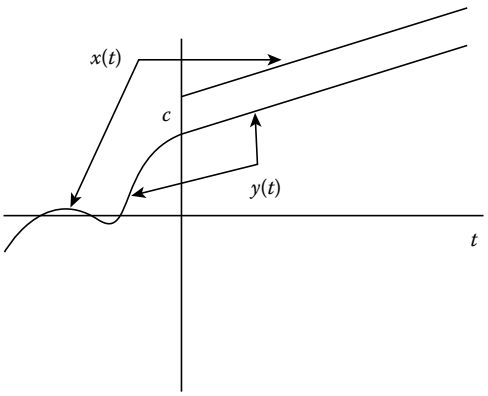


FIGURE 3.7 Signals x and y used in Example 3.10.

$$sX(s) = \int_{0^-}^{\infty} \frac{dy}{dt}e^{-st}dt + x(0^+) \tag{3.73}$$

Assuming that the LT of the signal y has a ROC (y is of exponential order), the integral in Equation 3.73 vanishes as $s \to \infty$. Finally, therefore, we obtain that $\lim_{s\to\infty} sX(s) = x(0^+)$.

Example 3.11

Verify the convolution property of the LT which states that if x and h are casual signals with $x(t) \leftrightarrow X(s)$, and $h(t) \leftrightarrow H(s)$, then

$$x(t) * h(t) = \int_{-\infty}^{\infty} x(\xi)h(t-\xi)\mathrm{d}\xi \leftrightarrow X(s)H(s) \tag{3.74}$$

Solution

Because $x(t) = 0$ for $t < 0$, we can write

$$\int_{-\infty}^{\infty} x(\xi)h(t-\xi)\mathrm{d}\xi = \int_{0^-}^{\infty} x(\xi)h(t-\xi)\mathrm{d}\xi \tag{3.75}$$

Now

$$\begin{aligned}\mathcal{L}\left\{\int_{-\infty}^{\infty} x(\xi)h(t-\xi)\mathrm{d}\xi\right\} &= \int_{0^-}^{\infty}\int_{0^-}^{\infty} x(\xi)h(t-\xi)e^{-st}\,\mathrm{d}\xi\mathrm{d}t \\ &= \int_{0^-}^{\infty}\int_{0^-}^{\infty} x(\xi)h(\beta)e^{-s\beta}e^{-s\xi}\,\mathrm{d}\beta\mathrm{d}\xi \\ &= \int_{0^-}^{\infty} x(\xi)e^{-s\xi}\mathrm{d}\xi \int_{0^-}^{\infty} x(\beta)e^{-s\beta}\,\mathrm{d}\beta \\ &= X(s)H(s)\end{aligned} \tag{3.76}$$

The causality of h is used in line (Equation 3.76) in setting the lower limit of integration over β to 0^-.

The operational properties are used to simplify forward and inverse transform computations and other manipulations involving transforms. To briefly illustrate, three examples follow.

Example 3.12

Using operational properties, rederive the LT for $x(t) = M_x \cos(\omega_x t + \theta_x)u(t)$, which was first considered in Example 3.5.

Solution

Write x as

$$x(t) = \frac{M_x}{2}e^{j\theta_x}e^{(\sigma_x+j\omega_x)t}u(t) + \frac{M_x}{2}e^{-j\theta_x}e^{(\sigma_x-j\omega_x)t}u(t) \tag{3.77}$$

The linearity property allows us to ignore the factors $M_x e^{\pm j\theta x}$ in the process of taking the transform, and returning them afterward. Recalling the previous result, Equation 3.48, we can write immediately

$$\mathcal{L}\left\{e^{(\sigma_x+j\omega_x)t}u(t)\right\} = \frac{1}{s-(\sigma_x+j\omega_x)}, \quad \mathrm{Re}\{s\} > \sigma_x \tag{3.78}$$

$$\mathcal{L}\left\{e^{(\sigma_x-j\omega_x)t}u(t)\right\} = \frac{1}{s-(\sigma_x-j\omega_x)}, \quad \mathrm{Re}\{s\} > \sigma_x \tag{3.79}$$

so

$$X(s) = \frac{M_x}{2} e^{j\theta_x} \frac{1}{s - (\sigma_x + j\omega_x)} + \frac{M_x}{2} e^{-j\theta_x} \frac{1}{s - (\sigma_x - j\omega_x)}, \quad \text{Re}\{s\} > \sigma_x \tag{3.80}$$

Placing the fractions over a common denominator yields the same result as that found using direct integration in Example 3.5.

Example 3.13

Find the time signals corresponding to the following LTs:

$$\begin{aligned} X(s) &= e^{-\pi s} \quad \text{for all } s \\ Y(s) &= \log(7) \frac{e^{-32s}}{s}, \quad \text{Re}\{s\} > 0 \\ Z(s) &= \frac{e^{\sqrt{2}s}}{s+5} + \frac{\sqrt{3}}{s-5}, \quad \text{Re}\{s\} > 5 \end{aligned} \tag{3.81}$$

Solution

Recognize that $X(s) = e^{-\pi s}\Delta(s)$, where $\Delta(s) = 1$ is the LT for the impulse function $\delta(t)$. Using the time-shift property, therefore, we have $x(t) = \delta(t - \pi)$.

Recognize that $Y(s) = \log(7)e^{-32s}\ U(s)$ where $U(s) = 1/s$ is the LT for the step function $u(t)$. Using linearity and the time-shift properties, therefore, $y(t) = \log(7)\ u(t - 32)$.

In finding z, linearity allows us to treat the two terms separately. Further, from Equation 3.48, we know that $\mathcal{L}^{-1}\{1/(s-5)\} = e^{5t}u(t)$ and $\mathcal{L}^{-1}\{1/(s-5)\} = e^{5t}u(t)$. Therefore, $x(t) = e^{-5(t+\sqrt{2})}u(t + \sqrt{2}) + \sqrt{3}e^{5t}u(t)$. Note that the first term has a ROC $\{s\text{: Re}\{s\} > -5\}$, while the second has ROC $\{s\text{: Re}\{s\} < 5\}$. The overall ROC is therefore consistent with these two components.

3.3.6 Inverse Laplace Transform

In principle, finding a time function corresponding to a given LT requires that we compute the integral in Equation 3.36:

$$x(t) = \frac{1}{j2\pi} \int_{\sigma - j\infty}^{\sigma + j\infty} X(s)e^{st}\,ds \tag{3.82}$$

Recall that σ is constant and taken to be in the ROC of X. Direct computation of this line integral requires a knowledge of the theory of complex variables. However, several convenient computational procedures are available that circumvent the need for a detailed understanding of the complex calculus. These measures are the focus of this section. The reader interested in more detailed information on complex variable theory is referred to [3]. ''Engineering'' treatments of this subject are also found in Ref. [10].

We first study the most challenging of the inversion methods, and the one that most directly solves the inversion integral above. The reader interested in quick working knowledge of LT inversion might wish to proceed immediately to Section 3.3.6.2.

3.3.6.1 Residue Theory

It is important to be able to compute residues of a function of a complex variable. Recall the Laurent series expansion of a complex function, say, F, which was introduced in Section 3.3.4, Equation 3.60. Also, recall that the coefficient $r_{1,p}$ is called the residue of F at p, and that we defined the simplified notation $r_p \stackrel{\text{def}}{=} r_{1,p}$ to indicate the residue because the subscript "1" is not useful outside the Laurent series. In the analytic neighborhood of singular point $s = p$ (an nth-order pole) we define the function

$$\varphi_p(s) = (s-p)^n F(s) = r_{1,p}(s-p)^{n-1} + r_{2,p}(s-p)^{n-2} + \cdots + r_{n,p} + \sum_{i=0}^{\infty} q_{i,p}(s-p)^{n+i} \tag{3.83}$$

in which it is important to note that $r_{n,p} \neq 0$. Because F is not analytic at $s = p$, φ_p is not defined, and is therefore not analytic, at $s = p$. We can, however, make φ_p analytic at p by simply defining $\varphi_p(p) \stackrel{\text{def}}{=} r_{n,p}$. In this φ_p is said to have a removable singular point (at p). Note that Equation 3.83 can be interpreted as the Taylor series expansion of φ_p about the point $s = p$. Therefore, the residue is apparently given by

$$r_p \stackrel{\text{def}}{=} r_{1,p} = \frac{\varphi_p^{(n-1)}(p)}{(n-1)!} \tag{3.84}$$

where $\varphi_p^{(i)}$ indicates the ith derivative of φ_p. When $n = 1$ (first-order pole), which is frequently the case in practice, this expression reduces to

$$r_p = \varphi_p(p) = \lim_{s \to p} (s-p)F(s) \tag{3.85}$$

The significance of the residues appears in the following key result (e.g., see Ref. [3]):

THEOREM 3.1: (Residue theorem)

Let C be a simple closed contour within and on which a function F is analytic except for a finite number of singularity points, $p_1, p_2, \ldots, p_k$ interior to C. If the respective residues at the singularities are $r_{p1}, r_{p2}, \ldots, r_{pk}$, then

$$\oint_C F(s)\mathrm{d}s = j2\pi(r_{p_1} + r_{p_2} + \cdots + r_{p_k}) \tag{3.86}$$

where the contour C is traversed in the counterclockwise direction.

The relevance of this theorem in our work is as follows: In principle, according to Equation 3.82, we want to integrate the complex function $F(s) = X(s)e^{st}$ on some vertical line in the ROC, for example, $c - j\infty$ to $c + j\infty$, where $c > \sigma_+$. Instead, suppose we integrate over the contour shown in Figure 3.8. By the residue theorem, we have

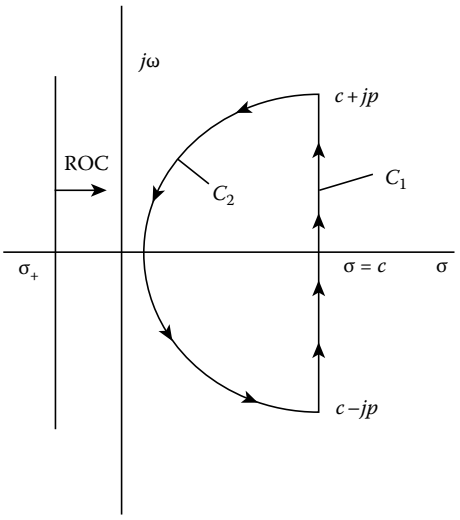


FIGURE 3.8 Contour in the s-plane for evaluating the inverse LT.

$$\oint_C X(e)e^{st}\,ds = j2\pi(r_{p_1} + r_{p_2} + \cdots + r_{p_k}) \tag{3.87}$$

where $r_{p_1}, r_{p_2}, \ldots, r_{p_k}$ are the k residues of the function $X(s)e^{st}$. The integral can be decomposed as

$$\oint_C X(s)e^{st}\,ds = \oint_{C_1} X(s)e^{st}\,ds + \oint_{C_2} X(s)e^{st}\,ds \tag{3.88}$$

where, as $\rho \to \infty$, C_1 approaches the line over which we wish to integrate according to Equation 3.82. It can be shown* that the integral over C_2 contributes nothing to the answer for $t > 0$, provided that X approaches zero uniformly† on C_2. Therefore,

$$\begin{aligned}\lim_{\rho\to\infty} \oint_C X(s)e^{st}\,ds &= \lim_{\rho\to\infty} \oint_{C_1} X(s)e^{st}\,ds \\ &= \lim_{\omega\to\infty} \frac{1}{j2\pi} \int_{c-j\omega}^{c+j\omega} X(s)e^{st}\,ds\end{aligned} \tag{3.89}$$

From Equation 3.87, we have

$$\frac{1}{j2\pi} \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}\,ds = r_{p_1} + r_{p_2} + \cdots + r_{p_k}, \quad t > 0 \tag{3.90}$$

Thus, recalling that the left side is the original inversion integral, we have

$$x(t) = \sum_{i=1}^{k} r_{p_i}, \quad t > 0 \tag{3.91}$$

where the r_{p_i} are the residues of $X(s)e^{st}$ at its k singular points.

Note that the residue method returns a time function only over the positive time range. We might expect a result beginning at $t = 0$ or $t = 0^-$ because we have defined the forward LT as an integral beginning at $t = 0^-$. The reason for this lower limit is so that an impulse function at the time origin will be transformed "properly." Another important place at which the initial condition "$x(0^-)$" appears is in the LT of a differentiated time function (see Table 3.2 and Example 3.9). Again, the condition is included to properly handle the fact that if x has a discontinuity at $t = 0$, its derivative should include an impulse and the LT of the derivative should include a corresponding constant. The residue cannot properly invert LTs of impulse functions (constants over the s-plane) because such LTs do not converge uniformly to zero over the semicircular part of the contour C_2. Such constants in the LT must be inverted in a more ad hoc way. If no apparent impulses occur and the residue method has provided x for $t > 0$, it is, in fact, possible, to assign an arbitrary finite value to $x(0)$. This is because one point in the time signal will not affect the LT, so the proper correspondence between x and X remains.‡ If it is necessary to assign a value to $x(0)$, the most natural procedure is to let $x(0) \stackrel{\text{def}}{=} x(0^+)$. Then when we

* A rigorous mathematical discussion appears in Ref. [3], while a particularly clear "engineering" discussion appears in Appendix B of Ref. [6].

† At the same rate regardless of the angle considered along C_2.

‡ In fact, any two signals that differ only on a set of measure zero (e.g., see Ref. [7]) will have the same LT.

write the final answer as $x(t) = [\text{time signal determined by residues}]\ u(t)$, the signal takes an implied value $x(0^+)/2$ at $t = 0$.

The preceding discussion emphasizes that, in general, successful application of the residue method depends on the uniform convergence of the LT to zero on the contour segment C_2. In principle each LT to be inverted must be checked against this criterion. However, this seemingly foreboding process is usually not necessary in linear signal and system analysis. The reason is that a practical LT ordinarily will take the form of a ratio of polynomials in s like (Equation 3.61). The check for proper convergence is a simple matter of assuring that the order of the denominator in s exceeds that of the numerator. When this not the case, a simply remedy exists which we illustrate in Example 3.15. Another frequent problem is the occurrence of an LT of form

$$X(s) = \frac{N(s)e^{st_0}}{D(s)}, \quad t_0 < 0 \tag{3.92}$$

where N and D are polynomials. The trick here is to recognize the factor e^{st_0} as corresponding to a time shift which can be taken care of at the end of the problem, once the rational part of the transform is inverted.

Finally, we remark that similar results apply to the bilateral LT. When the signal is two-sided in time, the casual part is obtained using the procedure above, but including only residues of poles known to be associated with the nonnegative-time part of the signal. The noncausal part of the signal is found by summing residues belonging to ''noncausal'' poles. Note that the association of poles with the casual and noncausal parts of the signal follows from a specification of the ROC. Poles belonging to the causal signal are to the left of the ROC, while noncausal poles are to the right.

Let us now illustrate the procedure with two examples.

Example 3.14

In Example 3.5, we showed that

$$\mathcal{L}\{M_x e^{\sigma_x t}\cos(\omega_x t + \theta_x)u(t)\} = M_x\left[\frac{(s-\sigma_x)\cos(\theta_x) - \omega_x \sin(\theta_x)}{(s-\sigma_x)^2 + \omega_x^2}\right] \tag{3.93}$$

with ROC $\{s\colon \mathrm{Re}\{s\} > \sigma_x\}$. Verify that this is correct by finding the inverse LT using residues. Call the signal x and LT X.

Solution

We can ignore the scalar M_x until the end due to linearity. Two poles are in the transform: $p_x = \sigma_x + j\omega_x$ and $p_x^* = \sigma_x + j\omega_x$ ($k = 2$ in the discussion above). These can be obtained by expanding the denominator and using the quadratic equation, but it is useful to remember the relationship between a quadratic polynomial written in this form and the conjugate roots (recall Equation 3.59). The residue for the pole at p_x is given by

$$\begin{aligned}\varphi_{px}(p_x) = X(s)e^{st}(s-p_x)\Big|_{s=p_x} &= \frac{[(s-\sigma_x)\cos(\theta_x) - \omega_x\sin(\theta_x)]e^{st}}{(s-p_x^*)}\Big|_{s=p_x} \\ &= \frac{[(p_x-\sigma_x)\cos(\theta_x) - \omega_x\sin(\theta_x)]e^{p_x t}}{(p_x - p_x^*)} \\ &= \frac{[(j\omega_x)\cos(\theta_x) - \omega_x\sin(\theta_x)]e^{(\sigma_x + j\omega_x)t}}{(2j\omega_x)}\end{aligned} \tag{3.94}$$

Similarly, we have for the pole at p_x^*, $\varphi_{p_x^*}(p_x^*) = [(-j\omega_x)\cos(\theta_x) - \omega_x \sin(\theta_x)]e^{(\sigma_x - j\omega_x)t}/(-2j\omega_x)$. For $t > 0$, therefore,

$$\frac{x(t)}{M_x} = \varphi_{p_x}(p_x) + \varphi_{p_x^*}(p_x^*)$$

$$= \frac{e^{\sigma_x t}[j\omega_x \cos(\theta_x)(e^{j\omega_x t} + e^{-j\omega_x t}) - \omega_x t \sin(\theta_x)(e^{j\omega_x t} - e^{-j\omega_x t})]}{2j\omega_x} \tag{3.95}$$

$$= e^{\sigma_x t}[\cos(\theta_x)\cos(\omega_x t) - \sin(\theta_x)\sin(\omega_x t)] = e^{\sigma_x t}\cos(\omega_x t + \theta_x) \tag{3.96}$$

and the transform is verified.

Example 3.15 illustrates the technique for handling polynomial quotients in which the order of the denominator does not exceed the numerator (X is not a proper rational LT).

Example 3.15

Find the causal time signal corresponding to LT

$$X(s) = \frac{N(s)}{D(s)} = \frac{Gs^2}{(s-p)^2}, \quad G, p \text{ are real} \tag{3.97}$$

Solution

To use the residue method, we must reduce the order of the numerator to at most unity. By dividing polynomial D and N using long division, we can express X as

$$X(s) = G + \frac{2Gps - Gp^2}{(s-p)^2} \stackrel{\text{def}}{=} X_1(s) + X_2(s) \tag{3.98}$$

First note that we can use linearity and invert the two terms separately. The first is simple because (see Table 3.1) $\mathcal{L}^{-1}\{X_1(s)\} = \mathcal{L}^{-1}\{G\} = G\delta(t)$. In the second term we find a pole of order $n = 2$ at $s = p$. To use Equation 3.84 to compute the residue, we require $\varphi_p(s) = X_2(s)e^{st}(s-p)^2 = [2Gps - Gp^2]e^{st}$. Then, for $t > 0$, the residue is

$$r_p = \frac{1}{1!}\frac{d\varphi_p}{ds}\bigg|_{s=p} = G[2pe^{pt} + 2p^2te^{pt} - p^2te^{pt}] = Gpe^{pt}[2 + pt] \tag{3.99}$$

Because r_p is the only residue, we have $x(t) = G\delta(t) + Gpe^{pt}[2 + pt]u(t)$.

Remark. If the order of the numerator *exceeds* the denominator by $k > 0$, the long division will result in a polynomial of the form $A_k s^k + A_{k-1}s^{k-1} + \cdots + A_0$. Consequently, the time domain signal will contain derivatives of the impulse function. In particular, if $k = 1$, and a "doublet" will be present in the time signal. (See Table 3.1 and Appendix A for more information.)

We can usually find several ways to solve inverse LT problems, residues often being among the most challenging. In this example, for instance, we could use Equation 3.98 to write

$$X(s) = G + \frac{2Gps}{(s-p)^2} - \frac{Gp^2}{(s-p)^2} = G + X_3(s) - X_4(s) \tag{3.100}$$

then use Table 3.1 to find $\mathcal{L}^{-1}\{G\} = G\delta(t)$ and $\mathcal{L}^{-1}\{X_4\} = x_4(t)$ [or use residues to find $x_4(t)$]. Noting that $X_3(s) = 2\ psX_4(s)$, we could then use the s-differentiation property (Table 3.2) to find x_3 from x_4. This alternative solution illustrates a general method that ordinarily is used regardless of the fundamental inversion technique. Linearity allows us to divide the problem into a sum of smaller, easier problems to invert, and then combine solutions at the end. The method presented next is probably the most popular, and clearly follows this paradigm.

3.3.6.2 Partial Fraction Expansion

The partial fraction expansion (PFE) method can be used to invert only rational LTs, with the exception that factors of the form e^{st_0} can be handled as discussed near Equation 3.92. As noted previously, this is not practically restricting for most engineering analyses. Partial fraction expansion is closely related to the residue method, a relationship evident via our examples.

As in the case of residues, a rational LT to be inverted must be proper, having a numerator polynomial whose order is strictly less than that of the denominator. If this is not the case, long division should be employed in the same manner as in the residue method (see Example 3.15).

Consider the LT $X(s) = N(s)/D(s)$. Suppose X has k poles, $p_1, p_2, \ldots, p_k$ and D is factored as

$$D(s) = (s - p_1)^{n_1}(s - p_2)^{n_2} \cdots (s - p_k)^{n_k} \tag{3.101}$$

where n_i is the order of the ith pole. (Note that $\Sigma_i^k = {}_1 n_i = n_D$.) Now if x is a real signal, the complex poles will appear in the conjugate pairs. It will sometimes be convenient to combine the corresponding factors into a quadratic following Equation 3.59,

$$(s - p)(s - p^*) = s^2 + \beta s + \gamma = s^2 - \mathrm{Re}\{p\}s + \left|p\right|^2 = s^2 - 2\sigma_p s + \left|p\right|^2 \tag{3.102}$$

for each conjugate pair. Assuming that the poles are ordered in Equation 3.101 so that the first k' are real and the last $2k''$ are complex ($k = k' + 2k''$), D can be written

$$\begin{aligned} D(s) = {} & (s - p_1)^{n_{re,1}}(s - p_2)^{n_{re,2}} \cdots (s - p_{k'})^{n_{re,k'}}(s^2 + \beta_1 s + \gamma_1)^{n_{c,1}} \\ & \times (s^2 + \beta_2 s + \gamma_2)^{n_{c,2}} \cdots (s^2 + \beta_{k''} s + \gamma_{k''})^{n_{c,k''}} \end{aligned} \tag{3.103}$$

Terms of the form $(s - p)$ are called simple linear factors, while those of form $(s^2 + \beta s + \gamma)$ are simple quadratic factors. When a factor is raised to a power greater than one, we say that there are repeated linear (or quadratic) factors. We emphasize that linear factors need not always be combined into quadratics when they represent complex poles. In other words, in the term $(s - p_i)^{ni}$, the pole p_i may be complex. Whether quadratics are used depends on the approach taken to solution.

The idea behind PFE is to decompose the larger problem into the sum of smaller ones. We expand the LT as

$$X(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{\mathrm{factor}_1 \mathrm{factor}_2 \cdots \mathrm{factor}_{k'} \mathrm{factor}_{k'+1} \cdots \mathrm{factor}_{k'+k''}}$$

$$= \frac{N_1(s)}{\mathrm{factor}_1} + \frac{N_2(s)}{\mathrm{factor}_2} + \cdots + \frac{N_{k'}(s)}{\mathrm{factor}_{k'}} + \frac{N_{k'+1}(s)}{\mathrm{factor}_{k'+1}} \tag{3.104}$$

$$+ \frac{N_{k'+k''}(s)}{\mathrm{factor}_{k'+k''}} \tag{3.105}$$

Now, because of linearity, we can invert each of the partial functions in the sum individually, then add the results. Each of these ''small'' problems is easy and ordinarily can be looked up in a table or solved by memory.

We now consider a series of cases and examples.

Case 1: Simple linear factors. Let $X(s) = N(s)/D(s)$. Assume that the order of N is strictly less than the order of D. Without loss of generality (for the case under consideration), assume that the first factor in D is a simpler linear factor so that D can be written $D(s) = (s - p_1)D_{\text{other}}(s)$ where D_{other} is the product of all remaining factors. Then the PFE will take the form

$$\begin{aligned} X(s) &= \frac{N(s)}{D(s)} = \frac{N(s)}{(s-p_1)D_{\text{other}}(s)} \\ &= \frac{N_1(s)}{(s-p_1)} + [\text{other PFs corresponding to factors in } D_{\text{other}}(s)] \end{aligned} \tag{3.106}$$

Now note that

$$\begin{aligned} (s-p_1)X(s) &= \frac{N(s)}{D_{\text{other}}(s)} \\ &= N_1(s) + (s-p_1)[\text{other PFs corresponding to factors in } D_{\text{other}}(s)] \end{aligned} \tag{3.107}$$

Letting $s = p_1$ reveals that

$$N_1(p_1) = A_1 = \frac{N(p_1)}{D_{\text{other}}(p_1)} = [(s-p_1)X(s)]|_{s=p_1} \tag{3.108}$$

Note that the number A_1 is the residue of the pole at $s = p_1$. In terms of our residue notation

$$\varphi_{p_1}(s) = (s-p_1)X(s) = \frac{N(s)}{D_{\text{other}}(s)} \quad \text{and} \quad r_{p_1} = A_1 = \varphi_{p_1}(p_1) \tag{3.109}$$

Carefully note that we are computing residues of poles of $X(s)$, not $X(s)e^{st}$, in this case.

Example 3.16

Given LT

$$X(s) = \frac{s^2 + 3s}{(s+1)(s+2)(s+4)} \tag{3.110}$$

with ROC $\{s: \text{Re}\{s\} > -1\}$, find the corresponding time signal x.

Solution

Check that the order of the denominator exceeds that of the numerator. Because it does, we can proceed by writing

$$X(s) = \frac{A_1}{(s+1)} + \frac{A_2}{(s+2)} + \frac{A_3}{(s+4)} \tag{3.111}$$

Using the previous method, we find that

$$A_1 = \left.\frac{s^2 + 3s}{(s+2)(s+4)}\right|_{s=-1} = -\frac{2}{3} \tag{3.112}$$

In a similar manner, we find that $A_2 = 1$ and $A_3 = 2/3$. Therefore,

$$X(s) = -\frac{2/3}{(s+1)} + \frac{1}{(s+2)} + \frac{2/3}{(s+4)} \tag{3.113}$$

Now, using linearity and Table 3.1 (or recalling Equation 3.48), we can immediately write

$$x(t) = \left[-\frac{2}{3}e^{-t} + e^{-2t} + \frac{2}{3}e^{-4t}\right]u(t) \tag{3.114}$$

Case 2: Simple quadratic factors. When $D(s)$ contains a simple quadratic factor, the LT can be expanded as

$$\begin{aligned} X(s) &= \frac{N(s)}{D(s)} = \frac{N(s)}{(s^2 - \beta s + \gamma)D_{\text{other}}(s)} \\ &= \frac{Bs + C}{(s^2 - \beta s + \gamma)} + [\text{other PFs corresponding to factors in } D_{\text{other}}(s)] \end{aligned} \tag{3.115}$$

The usefulness of this form is illustrated next.

Example 3.17

Find the time signal x corresponding to LT

$$X(s) = \frac{(s+4)}{(s+2)(s^2+6s+34)}, \quad ROC\{s: \text{Re}\{s\} > -3\} \tag{3.116}$$

Solution

The order of D exceeds the order of N, so we may proceed. The roots of the quadratic term are $p, p* = (-6 \pm \sqrt{36 - 136}/2 = -3 \pm j5)$, so we leave it as a quadratic. (If the roots were real, we would use the simple linear factor approach.) Expand the LT into PFs

$$X(s) = \frac{A}{(s+2)} + \frac{Bs + C}{(s^2 + 6s + 34)} \tag{3.117}$$

Using the method for simple linear factors, we find that $A = 1/13$. Now multiply both sides of Equation 3.117 by $D(s) = (s+2)\ (s^2 + 6s + 34)$ to obtain

$$(s+4) = \frac{1}{13}(s^2 + 6s + 34) + (Bs + C)(s+2) \tag{3.118}$$

Equating like powers of s on the two sides of the equation yields $B = -1/13$ and $C = -1/13$, so

$$X(s) = \frac{1/13}{(s+2)} + \frac{[(-1/13)s + (9/13)]}{(s^2 + 6s + 34)} \tag{3.119}$$

The first fraction has become familiar by now and corresponds to time function $(1/13)e^{-2t}\ u(t)$. Let us focus on the second fraction. Note that (recall Equation 3.59)

$$s^2 + 6s + 34 = (s - p)(s - p^*) = (s + 3 - j5)(s + 3 + j5) = (s+3)^2 + 5^2 \tag{3.120}$$

The second fraction, therefore, can be written as

$$-\frac{1}{13}\left[\frac{(s+3)}{(s+3)^2+5^2}-\frac{12}{(s+3)^2+5^2}\right]$$
$$=-\frac{1}{13}\left[\frac{(s+3)}{(s+3)^2+5^2}-\frac{(12/5)5}{(s+3)^2+5^2}\right] \tag{3.121}$$

The terms in brackets correspond to a cosine and sine, respectively, according to Table 3.1. Therefore,

$$X(t)=\left[\frac{1}{13}e^{-2t}-\frac{1}{13}e^{-3t}\cos(5t)+\frac{12}{5(13)}e^{-3t}\sin(5t)\right]u(t) \tag{3.122}$$

Simple quadratic factors need not be handled by the above procedure, but can be treated as simple linear factors, as illustrated by Example 3.18.

Example 3.18

Repeat the previous problem using simple linear factors.

Solution

Expand X into simple linear PFs

$$X(s)=\frac{A}{(s+2)}+\frac{E}{(s-p)}+\frac{E^*}{(s-p^*)},\quad p=-3+j5 \tag{3.123}$$

Note that the PFE coefficients corresponding to complex-conjugate pole pairs will themselves be complex conjugates. In order to derive a useful general relationship, we ignore the specific numbers for the moment. Using the familiar inverse transform, we can write

$$x(t)=[Ae^{-2t}+Ee^{pt}+E^*e^{p^*t}]u(t) \tag{3.124}$$

Letting $E=|E|e^{j\theta E}$ and $p=\sigma_p+j\omega_p$, we have

$$Ee^{pt}+E^*e^{p^*t}=|E|e^{j\theta_E}e^{(\sigma_p+j\omega_p)t}+|E|e^{-j\theta_E}e^{(\sigma_p+j\omega_p)t}=2|E|e^{\sigma_p t}\cos(\omega_p t+\theta_E) \tag{3.125}$$

This form should be noted for use with complex-conjugate pairs.

Using the method for finding simple linear factor coefficients, we find that $A=1/13$. Also

$$E=X(s)\ (s-p)|_{s=p}=\left.\frac{(s+4)}{(s+2)(s-p^*)}\right|_{s=p}=\frac{(p+4)}{(p+2)(j2\mathrm{Im}\{p\})}$$
$$=\frac{(1+j5)}{(-1+j5)(j10)}=0.1e^{-j0.626\pi} \tag{3.126}$$

Therefore, the answer can be written as

$$x(t)=\left[\frac{1}{13}e^{-2t}+0.2e^{-3t}\cos(5t-0.626\pi)\right]u(t) \tag{3.127}$$

This solution is shown to be consistent with that of the previous example using the trigonometric identity $\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$.

Case 3: Repeated linear factors. When $D(s)$ contains a repeated linear factor, for example, $(s-p)^n$ the LT must be expanded as

$$X(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s-p)^n D_{\text{other}}(s)} = \frac{A_n}{(s-p)^n} + \frac{A_{n-1}}{(s-p)^{n-1}} + \cdots + \frac{A_1}{(s-p)} + [\text{other PFs corresponding to factors in } D_{\text{other}}(s)] \tag{3.128}$$

The PFE coefficients of the n fractions are found as follows. Define

$$\varphi_p(s) = (s-p)^n X(s) \tag{3.129}$$

Then

$$A_{n-i} = \frac{1}{i!}\frac{d^i}{ds^i}\varphi_p\bigg|_{s=p} \tag{3.130}$$

The similarity of these computations to residues is apparent, but only A_1 can be interpreted as the residue of pole p. We illustrate this method by example.

Example 3.19

Find the time signal x corresponding to LT $X(s) = (s+4)/(s+1)^3$.

Solution

The order of D exceeds the order of N, so we may proceed. X has a third-order pole at $s=-1$, so we expand X as

$$X(s) = \frac{(s+4)}{(s+1)^3} = \frac{A_3}{(s+1)^3} + \frac{A_2}{(s+1)^2} + \frac{A_1}{(s+1)} \tag{3.131}$$

Let $\varphi_{-1}(s) = (s+1)^3\, X(s) = (s+4)$. Then

$$A_3 = \varphi_{-1}(-1) = 3, \quad A_2 = \frac{d\varphi_{-1}}{ds}\bigg|_{s=-1} = 1, \quad \text{and} \quad A_1 = \frac{1}{2}\frac{d^2\varphi_{-1}}{ds^2}\bigg|_{s=-1} = 0 \tag{3.132}$$

So

$$X(s) = \frac{3}{(s+1)^3} + \frac{1}{(s+1)^2} \tag{3.133}$$

Using Table 3.1, we have

$$x(t) = \left[\frac{3}{2}t^2 e^{-t} + te^{-t}\right]u(t) \tag{3.134}$$

Case 4: Repeated quadratic factors. When $D(s)$ contains a repeated quadratic factor, e.g., $(s^2+\beta s+\gamma)^n$, the LT may be either inverted by separating the quadratic into repeated linear factors (one nth-order factor for each of the complex roots; see Equation 3.155), then treated using the method of Case 3, or it can be expanded as

$$X(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{(s^2 + \beta s + \gamma)^n D_{\text{other}}(s)}$$
$$= \sum_{i=0}^{n-1} \frac{(B_{n-i}s + C_{n-i})}{(s^2 + \beta s + \gamma)^n} + [\text{other PFs corresponding to factors in } D_{\text{other}}(s)] \quad (3.135)$$

The PFE coefficients of the n factions are found algebraically as we illustrate by example.

Example 3.20

Find the time signal x corresponding to $X(s) = 4s^2/(s^2 + 1)^2\,(s + 1)$ [10].

Solution

Recognize the factor $(s^2 + 1)^2$ as an $n = 2$ order quadratic factor with $\beta = 0$ and $\gamma = 1$. We know from our previous work that $\beta = -2\,\text{Re}\{p\}$, where p and p^* are the poles associated with the quadratic factor, in this case $\pm j$. In turn, the real part of p provides the damping term in front of the sinusoid represented by the quadratic factor. In this case, therefore, we should expect one of the terms in the time signal to be a pure sinusoid.

Write X as

$$X(s) = \frac{(B_2 s + C_2)}{(s^2 + 1)^2} + \frac{(B_1 s + C_1)}{(s^2 + 1)} + \frac{A}{(s + 1)} \quad (3.136)$$

Using the familiar technique for simple linear factors, we find that $A = 1$. Now multiplying both sides of Equation 3.136 by $(s^2 + 1)^2\,(s + 1)$, we obtain $4s^2 = (B_2 s + C_2)(s + 1) + (B_1 s + C_1)(s^2 + 1)(s + 1) + (s^2 + 1)^2$. Equating like powers of s, we obtain $B_2 = -1$, $C_2 = 1$, $B_1 = 2$, and $C_1 = -2$. Hence,

$$X(s) = \frac{1}{(s + 1)} + \frac{(2s - 2)}{(s^2 + 1)^2} + \frac{(s - 1)}{(s^2 + 1)} \quad (3.137)$$

We can now use Tables 3.1 and 3.2 to invert the three fractions. Note that the third term will yield an undamped sinusoid as predicted. Also note that the middle term is related to the third by differentiation. This fact can be used in obtaining the inverse.

3.4 Laplace Transform Analysis of Linear Systems

Let us return to the example circuit problem, which originally motivated our discussion, and discover ways in which the LT can be used in system analysis. Three fundamental means are available for using the LT in such problems. The most basic is the use of LT theory to solve the differential equation governing the circuit dynamics. The differential equation solution methods are quite general and apply to linear, constant coefficient differential equations arising in any context. The second method involves the use of the "system function," an LT-based representation of an LTI system, which embodies all the relevant information about the system dynamics. Finally, we preview "LT equivalent circuits," an LT method that is primarily used for circuit analysis, and is treated more completely in Chapter 20.

3.4.1 Solution of the System Differential Equation

Consider a system is governed by a linear, constant coefficient differential equation of the form

$$\sum_{\ell=0}^{n_D} a_\ell \frac{d^\ell}{dt^\ell} y(t) = \sum_{\ell=0}^{n_N} b_\ell \frac{d^\ell}{dt^\ell} x(t) \quad (3.138)$$

with appropriate initial conditions given. (The numbers n_N and n_D should be considered fixed integers for now, but are later seen to be consistent with similar notation used in the discussion of rational LTs.) Using the linearity and time-differentiation properties of the LT, we can transform both sides of the equation to obtain

$$\sum_{\ell=0}^{n_D} a_\ell \left[s^\ell Y(s) - \sum_{i=0}^{\ell-1} y^{(i)}(0^-) s^{\ell-i-1} \right] = \sum_{\ell=0}^{n_N} b_\ell \left[s^\ell X(s) - \sum_{i=0}^{\ell-1} x^{(i)}(0^-) s^{\ell-i-1} \right] \tag{3.139}$$

where $y^{(i)}$ and $x^{(i)}$ are the ith derivatives of y and x. Rearranging, we have

$$Y(s) = \frac{X(s) \sum_{\ell=0}^{n_N} b_\ell s^\ell - \sum_{\ell=0}^{n_N} b_\ell \sum_{i=0}^{\ell-1} x^{(i)}(0^-) s^{\ell-i-1} + \sum_{\ell=0}^{n_D} a_\ell \sum_{i=0}^{\ell-1} y^{(i)}(0^-) s^{\ell-i-1}}{\sum_{\ell=0}^{n_D} a_\ell s^\ell} \tag{3.140}$$

Given the input signal x and all necessary initial conditions on x and y, all quantities on the right side of Equation 3.140 are known and can be combined to yield Y. Our knowledge of LT inversion will then, in principle, allow us to deduce y. This process often turns an unwieldy differential equation solution into simpler algebraic operations. The price paid for this simplification, however, is that the process of inverting Y to obtain y is sometimes challenging.

Recall that in the motivating example, a similar conversion of the (nonhomogeneous) differential equation solution to algebraic operations occurred (recall Equation 3.16 and surrounding discussion), as the superfluous term e^{pxt} was divided out of the equation. Except for the lack of attention paid to initial conditions, what remains in Equation 3.16 is tantamount to an LT equation of form Equation 3.139, as shown below. The homogeneous solution and related initial conditions were not included in the earlier discussion to avoid obfuscating the main issue. The reader was encouraged to think of the LT as a process of "dividing" the "e^{st}" term out of the signals before starting the solution. We can now clearly see this fundamental connection between the differential equation and LT solutions.

Example 3.21

Return to the motivating example in Section 3.2 and solve the problem using LT analysis.

Solution

Recall the differential equation governing the circuit, Equation 3.1, $x(t) = LC(d^2 y/dt^2) + RC(dy/dt) + y(t)$. The initial conditions are $y(0^-) = y_0$ and $i(0^-) = i_0$. Recall also that for convenience we seek the solution for $x(t) = M_x e^{j\theta_x} e^{(\sigma + j\omega_x)t} u(t) = M_x e^{j\theta_x} e^{p_x t} u(t)$ recognizing that the "correct" solution will be the real part of that obtained.

Taking the LT of each side of the differential equation, we have

$$X(s) = LC[s^2 Y(s) - sy(0^-) - y^{(1)}(0^-)] + RC[sY(s) - y(0^-)] + Y(s) \tag{3.141}$$

or

$$Y(s) = \frac{X(s)}{(LCs^2 + RCs + 1)} + \frac{(sLC - RC)y(0^-) + LCy^{(1)}(0^-)}{(LCs^2 + RCs + 1)} \tag{3.142}$$

Dividing both numerator and denominator of each fraction by LC, and inserting the LT $X(s) = M_x e^{j\theta x}/(s - p_x)$ and the initial conditions [recall that $Cy^{(1)} = i(t) \Rightarrow Cy^{(1)}(0^-) = i_0$], we have

$$Y(s) = \frac{M_x e^{j\theta_x}/LC}{(s - p_x)[s^2 + (R/L)s + 1]} + \frac{[s - (R/L)]y_0 + i_0/C}{[s^2 + (R/L)s + 1]} \tag{3.143}$$

Using PFE, this can be written

$$Y(s) = \frac{\dfrac{M_x e^{j\theta_x}/LC}{\left[p_x^2 + (R/L)p_x + 1\right]}}{(s - p_x)} + \frac{\dfrac{M_x e^{j\theta_x}/LC}{(p_h - p_x)}}{(s - p_h)} + \frac{\left(\dfrac{M_x e^{j\theta_x}/LC}{(p_h - p_x)}\right)^*}{(s - p_h^*)} + \frac{[s - (R/L)]y_0 + i_0/C}{(s - p_h)(s - p_h^*)} \tag{3.144}$$

where p_h and $p_h{}^\star$ are the system poles. We have expanded the first fraction in Equation 3.143 using simpler linear factors involving the poles p_x, p_h, and $p_h{}^\star$.The latter two poles correspond to the system (roots of $(s^2 + (r/L)s + 1)$), and the resulting terms in Equation 3.144 are part of the natural response. The first pole, p_x, is attributable to the forcing function, and the resulting term in Equation 3.144 will yield the forced response in the time domain. Finally, the last term in Equation 3.144, which arises from the last term in Equation 3.143, is also part of the natural response. The separation of the natural response into these three LT terms harkens back to the time-domain discussion about the distinct contributions of the input and initial conditions to the natural response. The last term is clearly related to initial conditions and the circuit's natural means of dissipating that energy. The former, terms, which will also yield a damped sinusoid of identical complex frequency to that of the third term, are clearly "caused" by the input.

Remark. The reader may wonder why the first fractions in Equations 3.17 and 3.144, both of which represent the forced response, are not identical. After all, we have been encouraged to view the LT as a kind of generalized phasor representation. Recall, however, that the LT at a particular s must be thought of as a "phasor density." If an eigensignal of the system represents one complex frequency, e.g., $s = p_x$, the LT is infinitely dense at that point in the s-plane, corresponding to the existence of a pole there. If the signal does have a conventional phasor representation such as $\overline{Y} = M_y e^{j\theta_y}$, then this phasor will be related to the LT as

$$\bar{Y} = Y(s)\frac{\mathrm{d}s}{2\pi}\bigg|_{s=p_x} = \lim_{s \to p_x} Y(s)(s - p_x) \tag{3.145}$$

which we recognize as the residue of the LT Y at the pole p_x. The reader can easily verify this assertion using Equation 3.143. This discussion is closely related to the interconnection between the FS coefficients (which are similar to conventional sinusoidal phasors) and the Fourier transform (FT). These signal representations are discussed in Chapter 4.

3.4.2 System Function

Definition 3.1: In our motivating example an input sinusoid with generalized phasor $M_x e^{j\theta x}$ produced an output sinusoid with generalized phasor $M_y e^{j\theta y}$. We discovered that the ratio of phasors was dependent only upon system parameters and the complex frequency, p_x. We noted that when considered as a function of the general complex frequency, s, this ratio is called the system function. That is, the system function is a complex function, H, of complex frequency, s, such that if a damped sinusoid

of complex frequency $s = p_x$ and with generalized phasor $\overline{X} = M_x e^{j\theta_x}$ is used as input to the system, the forced response will be a sinusoid of complex frequency p_x and with generalized phasor

$$M_y e^{j\theta_y} = H(p_x) M_x e^{j\theta_x} \tag{3.146}$$

3.4.2.1 Preview of Magnitude and Phase Responses

It is often important to know how a system will respond to a pure sinusoid of radian frequency, e.g., ω_x. In particular, we would like to know the amplitude and phase changes imposed on the sinusoid by the system. In terms of the definition just given, we see that this information is contained in the system function evaluated at frequency $p_x = j\omega_x$. In particular, $^*H(j\omega_x)^*$ represents the magnitude factor and arg $\{H(j\omega_x)\}$ the phase change at this frequency. When plotted as functions of general frequency ω, the real functions $^*H(j\omega)^*$ and arg $\{H(j\omega_x)\}$ are called the magnitude (or sometimes frequency) response, and phase response of the system, respectively. The complex function $H(j\omega)$ will be seen to be the FT of the impulse response (see Appendix B and Chapter 4) and is sometimes called the transfer function for the system. Returning to Figure 3.2b, the reader will discover that we have plotted the magnitude response for the series RLC circuit of Section 3.2 with numerical values given in Equation 3.18.

The magnitude and phase responses of the system can be obtained graphically from the pole–zero diagram. Writing $H(s)$ similarly to Equation 3.62,
$H(s) = \mathcal{C} \prod_{i=1}^{n_N} (s - z_i) / \prod_{i=1}^{n_D} (s - p_i)$. Therefore,

$$\begin{aligned} |H(j\omega)| &= \mathcal{C} \frac{\prod_{i=1}^{n_N} |j\omega - z_i|}{\prod_{i=1}^{n_D} |j\omega - p_i|} \quad \text{and} \\ \arg\{H(j\omega)\} &= \sum_{i=1}^{n_N} \arg\{(j\omega - z_i)\} - \sum_{i=1}^{n_D} \arg\{(j\omega - p_i)\} \end{aligned} \tag{3.147}$$

where we have assumed $\mathcal{C} > 0$ (if not, add π radians to arg$\{H(j\omega_x)\}$). By varying ω the desired plots are obtained as illustrated in Figure 3.9.

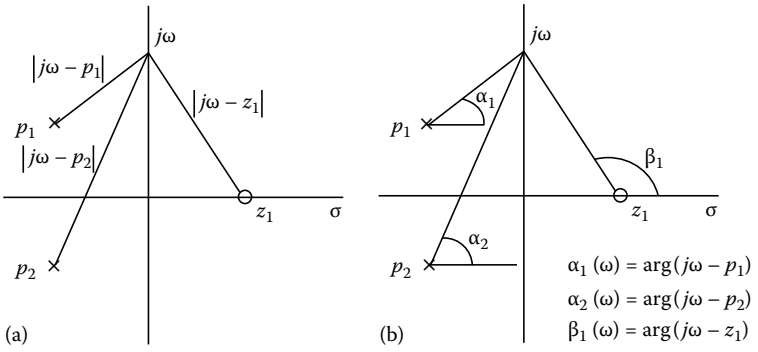


FIGURE 3.9 Pole–zero plot for an example system function, $H(s) = C(s - z_1)/(s - p_1)(s - p_2)$. (a) To obtain the magnitude response $|H(s\omega)|$ at frequency ω, the product of lengths from all zeroes to $s = j\omega$ is divided by the product of lengths from all poles to $s = j\omega$. The result must be multiplies by the gain term $|C|$. (b) To obtain the phase response $\{H(j\omega_x)\}$ at frequency ω, the sum of angles from all poles to $s = j\omega$ is subtracted from the sum of angles from all zeroes to $s = j\omega$. An additional π radians is added if $C < 0$.

Definition 3.2: More generally, the system function, $H(s)$, for an LTI system can be defined as the ratio of the LT, for example, $Y(s)$, of the output, resulting from any input with LT, for example, $X(s)$, when the system is initially at rest (zero initial conditions). In other words, H is the ratio of the LT of the zero-state response to the LT of the input,

$$H(s) \stackrel{\text{def}}{=} \frac{Y(s)}{X(s)} \text{when all initial conditions are zero} \tag{3.148}$$

While the latter definition is more general, the two definitions are consistent when the input is an eigensignal, as we show by example.

Example 3.22

Show that the two definitions of the system function described previously are consistent for the RLC circuit of Section 3.2.

Solution

Replacing the specific frequency p_x by a general complex frequency s in the initial example using phasor analysis, we found that (recall Equation 3.31) $H(s) = 1/(LCs^2 + RCs + 1)$. On the other hand, using more formal LT analysis on the same problem, we derived Equation 3.142. Forming the ratio $Y(s)/X(s)$ where both initial conditions are set to zero, yields an identical result.

Finally, another very useful definition of the system is as follows.

Definition 3.3: The output of LTI system to an impulse excitation, $x(t) = \delta(t)$ (see Section 3.6, Appendix A), when all initial conditions are zero, is called the impulse response of the system. The impulse response is usually denoted $h(t)$. The system function can be defined as the LT of the impulse response $H(s) = \mathcal{L}\{h(t)\}$.

The consistency of this definition with the first two is easy to demonstrate. Let H denote the system function for the system, regardless of how it might be related to $h(t)$. From Definition 3.2, we have $H(s) = Y(s)/X(s)$ for any valid LTs X and Y. Let $x(t) = \delta(t)$, in which $X(s) = 1$. By definition, $y(t) = h(t)$, so $\mathcal{L}\{h(t)\} = \mathcal{L}\{y(t)\} = H(s)X(s) = H(s)$.

This interpretation of H enables us to find the impulse response of the system, a task which is not always easy in the time domain because of the pitfalls of working with impulse functions.

Example 3.23

Find the impulse response, h, for the circuit of Section 3.2.

Solution

Using various means we showed that the system function is $H(s) = 1/(LCs^2 + RCs + 1)$. Let us find h by computing $h(t) = \mathcal{L}^{-1}\{H(s)\}$. Using the quadratic equation, we find the roots of the denominator (poles of the system) to be (cf., Equation 3.6)

$$p_1, p_2 = \frac{-RC \pm \sqrt{R^2C^2 - 4LC}}{2LC} = -\frac{R}{2L} \pm \sqrt{\frac{R^2}{4L^2} - \frac{1}{LC}} \tag{3.149}$$

Assume that these poles are complex conjugates and call them p_h, p_h^*, where

$$p_h = -\frac{R}{2L} + j\sqrt{\frac{1}{LC} - \frac{R^2}{4L^2}} \stackrel{\text{def}}{=} \sigma_h + j\omega_h \tag{3.150}$$

Comparing to our initial work on finding the homogeneous solution of the differential equation, we see that these system poles are the roots of the characteristic equation. The reason for using subscript "h" in our early work should now be clear. In terms of Equation 3.150 we can rewrite H as

$$H(s) = \frac{1/LC}{(s-\sigma_h)^2 + \omega_h^2} \tag{3.151}$$

(recall Equation 3.59). Now using Table 3.1: $\mathcal{L}\{e^{\sigma_b t}\cos(\omega_b t + \theta_b)\} = [(s-\sigma_b)\cos(\theta_b) - \omega_b \sin(\theta_b)]/[(s-\sigma_h)^2 + \omega_h^2]$. Letting $\theta_h = \pi/2$ and using linearity, we have

$$h(t) = -\frac{1}{LC\omega_h} e^{\sigma_h t}\cos\left(\omega_h t + \frac{\pi}{2}\right)u(t) = \frac{1}{LC\omega_h} e^{\sigma_h t}\sin(\omega_h t)u(t) \tag{3.152}$$

We could also note that $|p_h| = \sqrt{1/LC}$ and write the initial scale factor as $|p_h|^2/\omega_h$.

We see clearly that the impulse response is closely related to the natural responses of a system, which in turn are tied to the homogeneous solution of the differential equation. These transient responses depend only on properties of the system, and not on properties of the input signal (beyond the initial instant of excitation). The form of the homogeneous differential equation solution is specified by the number and values of the roots of the characteristic equation. These roots are, thus, the poles of the system function H. The system function offers an extremely valuable tool for the design and analysis of systems. We now turn to this important topic.

3.4.3 Poles and Zeros—Part II: Stability Analysis of Systems

We return to the issue of poles and zeroes, this time with attention restricted to rational LTs. In particular, we focus on the poles and zeroes of a system function and their effects on the performance of the system.

3.4.3.1 Natural Modes

The individual time responses corresponding to the poles of H are often called natural modes of the system. These modes are indicators of the physical properties of the system (e.g., circuit values), which in turn determine the natural way in which the system will dissipate, store, amplify, or respond to energy of various frequencies. Consider two general cases. Suppose H has a real pole of order n at $s = p$, so that it can be written

$$H(s) = \sum_{i=1}^{n} \frac{A_i}{(s-p)^i} + [\text{other terms}] \tag{3.153}$$

We know from previous work, therefore, that (see Table 3.1) h has corresponding modal components,

$$h(t) = \sum_{i=1}^{n} \frac{A_i}{(i-1)!} t^{i-1} e^{pt} u(t) + [\text{other terms}] \tag{3.154}$$

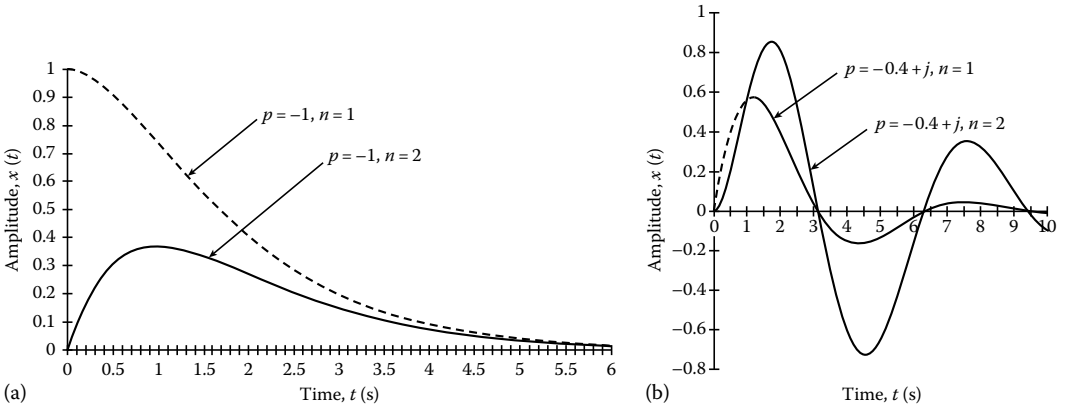


FIGURE 3.10 Modal components in the impulse response corresponding to (a) a real pole of orders $n = 1$ and $n = 2$ at $s = p$, and (b) a complex pole pair of orders $n = 1$ and $n = 2$ and $s = p, p^{\star}$.

When $|p| < 0$, the modal components due to pole p will decay exponentially with time (modulated by the terms t^{i-1}). When $|p| > 0$, these modal components will increase exponentially with time (modulated by the terms t^{i-1}). When $p = 0$, the term will either remain bounded if $n = 1$, or terms will increase with power n as t increases. These cases are illustrated in Figure 3.10a.

Next, let H have a complex pole pair of order n at $s = p, p^{\star}$, so that it can be written

$$H(s) = \sum_{i=1}^{n} \left[\frac{E_i}{(s-p)^i} + \frac{E_i^{\star}}{(s-p^{\star})^i} \right] + [\text{other terms}] \tag{3.155}$$

Using Table 3.1, we see that h has a corresponding modal component

$$h(t) = \sum_{i=1}^{n} \frac{2|E_i|}{(i-1)!} t^{i-1} e^{\sigma_p t} \cos(\omega_p t + \theta_{E_i}) u(t) + [\text{other terms}] \tag{3.156}$$

where $p = \sigma_p + j\omega_p$, and $\theta_{E_i} = \arg\{E_i\}$. In this case, when $|p| < 0$, the sinusoidal components will decay exponentially with time (modulated by the terms t^{i-1}); when $|p| > 0$, these terms will increases exponentially with time (modulated by the terms t^{i-1}); and when $p = 0$, the terms due to p will either represent a constant or increasing sinusoid, depending on the value of n. These cases are illustrated in Figure 3.10b.

3.4.3.2 BIBO Stability

We digress momentarily to discuss the concept of stability. To avoid some unnecessary complications, we restrict the discussion to causal systems, and, as usual, to the unilateral LT. There are various ways to define stability, but the most frequent and useful definition for an LTI system is that of BIBO stability. A system is said to be bounded-input–bounded-output (BIBO) stable iff every bounded input produces a bounded output. Formally, any bounded input, for example, x such that $|x(t)| \leq B_x < \infty$ for all t, must result in an output, y, for which $B_y < \infty$ exists so that $|y(t)| \leq B_y$ for all t.

A necessary and sufficient condition for BIBO stability of an LTI system is that its impulse response be absolutely integrable

$$\int_0^\infty |h(t)|\mathrm{d}t < \infty \tag{3.157}$$

This is easy to show. First, assume $|x(t)| < B_x$ for all t. Then, form the convolution integral

$$|y(t)| = \left| \int_0^\infty x(t-\lambda)h(\lambda)\mathrm{d}\lambda \right| \tag{3.158}$$

Now, using the Schwarz inequality (see Ref. [7]), we have

$$|y(t)| \leq \int_0^\infty \left\| x(t-\lambda) \right\| h(\lambda)|\mathrm{d}\lambda \leq B_x \int_0^\infty |h(\lambda)|\mathrm{d}\lambda \tag{3.159}$$

Consequently, it is sufficient that Equation 3.157 be true for $B_y < \infty$ to exist. On the other hand, suppose that condition (Equation 3.157) were not true, but the system were BIBO stable. For a fixed t, consider the input

$$x(t-\lambda) = \begin{cases} 1, & h(\lambda) > 0 \\ -1, & h(\lambda) < 0 \\ 0, & h(\lambda) = 0 \end{cases} \tag{3.160}$$

For this input, the output at time t is

$$y(t) = \int_0^\infty \left| h(\lambda) \right| \mathrm{d}\lambda \tag{3.161}$$

which is not bounded according to assumption. Therefore, we encounter a contradiction showing that the condition (Equation 3.157) is also necessary for BIBO stability.

3.4.3.3 Stability and Natural Modes

Now we tie the stability analysis to the discussion of modal components above. We assert that a ''causal LTI system with proper rational system function H will be BIBO stable iff all of its poles are in the left-half s-plane.'' That this is true is easily seen. If any pole is in the right-half s-plane, we know that h will contain at least one mode that will increase without bound. Therefore, Equation 3.157 cannot hold and the system is not BIBO stable. Conversely, if all poles are in the left-half s-plane, Equation 3.157 will hold. The case in which one or more simple poles fall exactly on the $s = j\omega$-axis (and none in the right-half s-plane) is called marginal BIBO stability. In this case Equation 3.157 does not hold, so the system is not strictly BIBO stable. However, the system does theoretically produce bounded outputs for some inputs. Finally, note that we show by example below Example 3.24(a) that an improper rational system function cannot represent a BIBO stable system.

Based on our earlier discussion of LT ROCs, an equivalent way to state the BIBO stability condition is as follows: ''A causal, LTI system is BIBO stable iff the ROC of its system function H includes the $s = j\omega$-axis in the s-plane.'' We recall that the failure to include the $j\omega$- axis would imply at least one pole of H in the right-half s-plane.

Let us conclude this discussion by considering some examples.

Example 3.24

Comment on the BIBO stability of the following systems:

$$\text{(a)}\quad H(s)=\frac{N(s)}{D(s)}=\frac{(1/2)s^3}{s^2+2s+1}=\frac{(1/2)s^3}{(s+1)^2}$$

$$\text{(b)}\quad H(s)=\frac{(s-3)}{s^2+5s+4}$$

$$\text{(c)}\quad H(s)=\frac{(s-3)}{(s^2+5s+4)(s^2-2s-3)}$$

$$\text{(d)}\quad H(s)=\frac{6}{\left[(s+3)^2+25\right](s+8)}$$

$$\text{(e)}\quad H(s)=\frac{s}{s^2+4}$$

Solution

(a) H is not a proper rational fraction because the order of $N(s)>$ order of $D(s)$. We illustrate that such a system is not BIBO stable. Dividing D into N, we can write

$$H(s)=\frac{1}{2}s-1+\frac{3/2}{s(s^2+2s+1)} \tag{3.162}$$

Suppose we enter $x(t)=u(t)$ as a (bounded) input. Then

$$Y(s)=H(s)X(s)=\frac{H(s)}{s}=\frac{1}{2}-\frac{1}{s}+\frac{3/2}{s(s^2+2s+1)} \tag{3.163}$$

Therefore,

$$y(t)=\frac{1}{2}\delta(t)-u(t)+\left[\text{terms resulting from}\frac{3/2}{s(s^2+2s+1)}\right] \tag{3.164}$$

The output is unbounded in response to a bounded input and the system is therefore not BIBO stable. "A similar result will occur whenever H is not proper."

(b) H has poles at $s=-1,-4$, and zero at $s=3$. The system is BIBO stable. Note that the right-half plane zero has no adverse effect on stability.

(c) $H(s)=N(s)/D(s)$ has a second-order pole at $s=-1$, and a simple pole at $s=-4$. Both N and D have roots $s=3$, thus neither a pole nor zero is found there (they cancel). The system is therefore BIBO stable.

Remark. The response to nonzero initial conditions of a system which has one or more "cancelled poles" in the right-half s-plane will increase without bound, and therefore could be considered "unstable" in some sense, even though it is BIBO stable. The reader is invited to show that the present system will respond in this undesirable manner to nonzero initial conditions. A system is said to be asymptotically stable if it is BIBO stable and its response to initial conditions approaches zero as $t\to\infty$. We see that asymptotic stability implies BIBO stability but the converse is not true.

(d) Recalling Equation 3.59, we find H has poles $s=-3\pm j5$ and $s=-8$. No finite zeros are included. The system is BIBO stable and will have both an oscillatory mode and a damped exponential mode.

(e) H has poles at $s=\pm j2$ which are on the $j\omega$-axis. The system is marginally BIBO stable.

Example 3.25

Return to the series RLC circuit of Section 3.2 and discuss the system's stability as R varies for a fixed L and C. Assume that $R < 2\sqrt{L/C}$.

Solution

In light of Equation 3.149, the upper bound on R means that the poles of the system will always be complex conjugates, e.g., p_h, p_h^* (h is oscillatory). From Equation 3.150 we see that the poles of the system can be written in polar form as

$$p_h, p_h^* = \sqrt{\sigma_h^2 + \omega_h^2}\, e^{\pm j \tan^{-1}(\omega_h/\sigma_h)} = \frac{1}{\sqrt{LC}} e^{\pm j \tan^{-1}(\omega_h/\sigma_h)} \tag{3.165}$$

For fixed L and C, the poles remain on a circle of radius $1/\sqrt{LC}$ in the s-plane. Having established this fact, recall the Cartesian form of p_h given in Equation 3.149 and note that as $R \to 0, p_h \to j\sqrt{1/LC}$. On other hand, as $R \to 2\sqrt{LC}, p_h \to -\sqrt{1/LC}$ (see Figure 3.11). Therefore, the poles remain in the left-half s-plane over the specified range for R, except when $R = 0$, in which case the poles are exactly on the $j\omega$-axis at $s = \pm j(1/\sqrt{LC})$. Therefore, the system is BIBO stable, except when $R = 0$, in which case it is marginally stable. Only if R were to take negative values would the circuit go unstable. This is not realistic unless active circuit elements are present which effectively present negative resistance. The reader is encouraged to explore what happens as R continues to increase beyond the given upper bound.

3.4.4 Laplace-Domain Phasor Circuits

After completing a sufficient number of circuit problems involving differential equations like Example 3.21, certain patterns would become apparent in a Laplace-transformed differential equation like Equation 3.141. These patterns occur precisely because of the invariant relationships between currents and voltages across lumped parameter components. For example, because the current and voltage

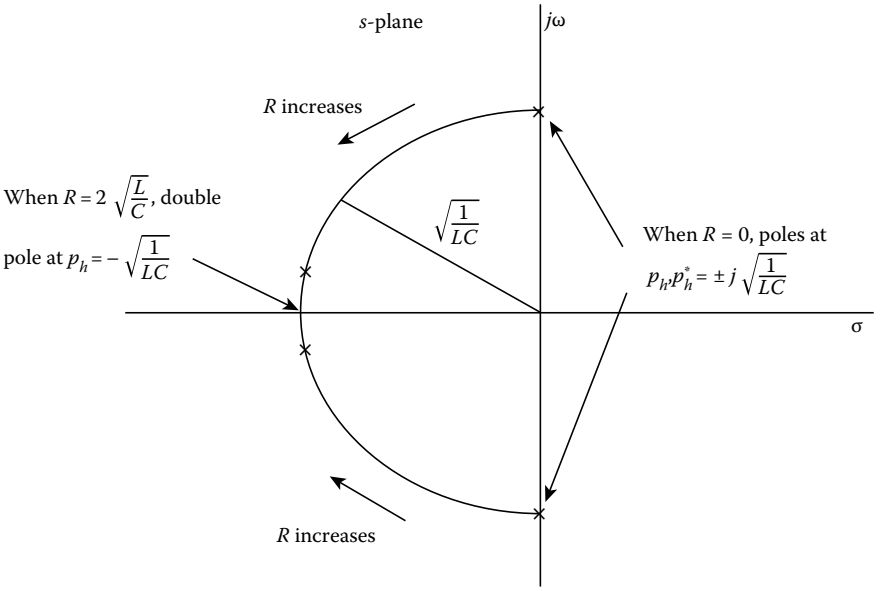


FIGURE 3.11 Locus of the poles as R varies in Example 3.25.

through an inductor are related as $v_L(t) = L(\mathrm{d}i_L/\mathrm{d}t)$, whenever this relationship is encountered and transformed in a circuit problem, it becomes $V_L(s) = sLI_L(s) - Li_L(0^-)$. This relationship may be written without recourse to the time domain by treating $V_L(s)$ and $I_L(s)$ as a ''DC voltage'' and a ''DC current,'' by replacing the inductor with a ''resistor'' with value sL, by adding a voltage source in series with value $Li_L(0^-)$ having proper polarity, a small DC circuit problem.

In general, we can make ''Laplace'' substitutions for each component in a circuit, then use ''DC'' analysis to write the Laplace-transformed differential equation directly. The appropriate substitutions and variations for both mesh and nodal analysis are discussed in detail in Chapter 12 of *Feedback, Nonlinear, and Distributed Circuits*, along with more advanced frequency-domain uses of such LT replacements. After studying that material, the reader may wish to provide closure to the present discussion by solving the following problem.

Example 3.26

Derive the Laplace-transformed differential equation (Equation 3.141) using Laplace circuit equivalents.

3.5 Conclusions and Further Readings

The LT is a powerful tool in the analysis and design of linear, time-invariant circuits and systems. In this exposition we have developed the LT by appealing to the manner in which it turns differential equation solutions into simple algebraic problems. We have focused not only on the technique for exploiting this advantage, but also the reasons that this happens. Along the way we have discovered some useful properties of the LT, as well as the meaning of poles and zeros. We also discussed numerous techniques for recovering the time signal from the LT, noting that this sometime difficult task is the price paid for alleviating the difficulties of time-domain solution. In the latter part of the chapter we turned our attention to the analysis of systems, in particular the meaning and derivation of the system function. This study led to the understanding of the relationship between the system function and impulse response, and between the pole locations of the system function and stability of the system.

Most of our work has focused on the ULT, although the BLT was carefully discussed at the outset. The advantage of the ULT is that it provides a way to incorporate initial condition information, a very important property in many design and analysis problems, particularly because the initial conditions play an important role in the transient response of the system. On the other hand, the BLT can handle a much broader class of signals, a feature that is often advantageous in theoretical and formal developments. The ROC of the LT in the s-plane becomes much more important for the BLT as the LT itself is not unique without it. Accordingly, we spent a significant amount of effort studying the ROC.

The LT is closely related to the Fourier and z-transforms (ZTs), which are the subjects of subsequent chapters. Section 3.7, Appendix B previews these ideas.

Finally, we note that several LT topics have not been treated here. These topics, such as the analysis of feedback systems and state-space models, deal principally with control theory and applications and are therefore outside of the scope of this book. Many fine books on signals and systems are available to which the reader can turn to explore these subjects. Some of these texts are referenced below. Many excellent books on circuit analysis and filter design, too numerous to cite here, are also available.

3.6 Appendix A: The Dirac Delta (Impulse) Function

The Dirac delta, or impulse, function, $\delta(t)$, is defined as the function with the following properties: If signal x is continuous at t_0 (''where the impulse is located in time''), then

$$\int_a^b x(t)\delta(t - t_0)\mathrm{d}t = \begin{cases} 0, & t_0 \notin [a,b] \\ x(t_0), & t_0 \in (a,b) \end{cases} \tag{3.166}$$

Two special cases are noted:

1. Note that Equation 3.166 does not cover the case in which the impulse is located exactly at one of the limits of integration. In such cases whether x is continuous at t_0.
   The integral takes the value $\frac{1}{2}x(t_0^+)$ if t_0 is the lower limit, $t_0 = a$.
   The integral takes the value $\frac{1}{2}x(t_0^-)$ if t_0 is the upper limit, $t_0 = b$.
2. The only case not explicitly covered is one in which x is discontinuous at t_0 and t_0 is not a limit of integration. In this case the integral takes the value $\frac{1}{2}[x(t_0^-) + x(t_0^+)]$ if $t_0 \in (a, b)$, and 0 otherwise. Note that this answer is also valid if x is continuous at t_0, but it is unnecessarily complicated.

Note what happens in the special case in which $x(t) = 1$, $a = -\infty$, $b = t$, and $t_0 = 0$. From the definition, we can write

$$\int_{-\infty}^{t} \delta(\lambda)\mathrm{d}\lambda = \begin{cases} 0, & t < 0 \\ 1, & 5t > 0 \\ \dfrac{1}{2}, & t = 0 \end{cases} \tag{3.167}$$

We see that

$$\int_{-\infty}^{t} \delta(\lambda)\mathrm{d}\lambda = u(t) \tag{3.168}$$

Therefore, except at $t = 0$,

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \delta(t) \tag{3.169}$$

What emerges here is a very strange function. We see from Equation 3.168 that δ must be zero everywhere except at $t = 0$ because, apparently we accumulate area at only that point. The area under that one point is unity because the integral takes a jump from 0 to 1 as t crosses zero. Because δ has zero width and unity total area, it must have infinite amplitude (at that one point!). To indicate the delta function, therefore, we draw ''arrows'' as shown in Figure 3.12. It is sometimes mathematically useful to indicate a delta function with area other than unity. In this case we simply label the arrow with a number called the ''weight'' of the impulse. Note that it makes no sense to draw taller and shorter arrows for

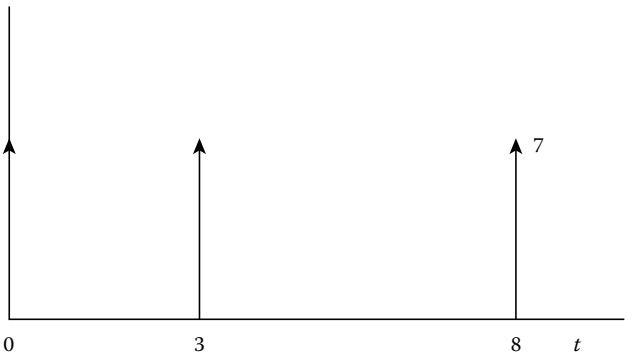


FIGURE 3.12 Impulse functions $\delta(t)$, $\delta(t-3)$, and $7\delta(t-8)$.

different weights, although this is sometimes done in textbooks, because the weight does not indicate the "height" of the function (∞!), but instead, its area!

Finally, note that computing integrals with delta functions in them is very easy because one need only follow the rules of the definition. For example,

$$\int_{-18}^{\infty} e^{\pi(\lambda-4)}\delta(\lambda-6)\mathrm{d}\lambda = e^{2\pi} \tag{3.170}$$

and

$$\int_{0}^{2} 3u(t)\delta(t)\mathrm{d}t = 3\frac{u(0^+)}{2} = \frac{3}{2} \tag{3.171}$$

Remark. The kth derivative of the impulse (usually $k \leq 2$) occasionally appears in LT work. The notation

$$\delta^{(k)}(t) \stackrel{\text{def}}{=} \frac{\mathrm{d}^k\delta}{\mathrm{d}t^k} \tag{3.172}$$

is often used to denote this signal. The signal $\delta^{(1)}(t)$ is called a doublet and is plotted as shown in Figure 3.13.

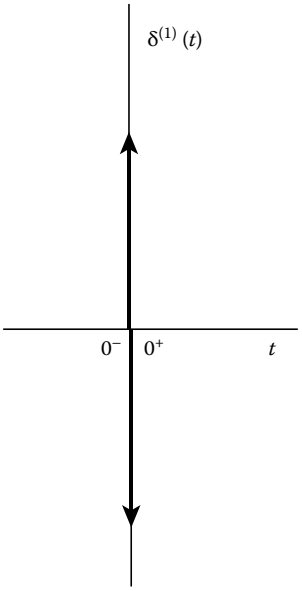


FIGURE 3.13 Doublet.

3.7 Appendix B: Relationships among the Laplace, Fourier, and z-Transforms

The transforms previewed here are most frequently defined and discussed for two-sided signals. Therefore, it is most natural to base this discussion on the BLT, as defined in Equation 3.38.

The FT, X_F, of a signal x is defined by the integral

$$X_F(\omega) = \int_{-\infty}^{\infty} x(t)e^{j\omega t}\mathrm{d}t \tag{3.173}$$

Upon comparison with Equation 3.38, it is apparent that the FT evaluated at radian frequency ω is equivalent to the BLT of x evaluated at $s = j\omega$ in the s-plane. The FT can, therefore, be obtained over all ω by evaluating the BLT, e.g., X_L along the $j\omega$-axis:

$$X_F(\omega) = X_L(s)\big|_{s=j\omega} \tag{3.174}$$

Evidently, the FT will only exist for a signal x if its BLT has an ROC that includes the $j\omega$-axis.

One very important class of signals whose BLT ROCs include the entire left-half s-plane, but not the $j\omega$-axis, is the periodic signals. For this purpose, the FS can be used to expand the signal on a set of discrete, harmonically related, basis functions (either complex exponentials or sinusoids). The complex version is

$$x(t) = \sum_{\ell=-\infty}^{\infty} c_\ell e^{j\ell\omega_0 t} \tag{3.175}$$

where $\omega_0 = 2\pi/T_0$ is the fundamental radian frequency, with T_0 the period of the waveform. The complex numbers c_ℓ, $\ell = \ldots, -1, 0, 1, 2, \ldots$ are the FS coefficients computed as

$$c_\ell = \frac{1}{T_0} \int_{T_0} x(t) e^{-j\ell\omega_0 t} \mathrm{d}t \tag{3.176}$$

where the integral is taken over any period of the waveform.

Comparing Equations 3.176 and 3.173, we see that the FS coefficients are equivalent to (scaled) samples of the FT of one period of x, where the samples are taken at frequencies $\ell\omega_0$. If we have a periodic waveform, therefore, we can always represent it by samples of the FT of one period. Similarly, if we have "short" signal and want to represent it using only frequency samples, we can let it be periodic and represent it using the FS coefficients. In this case, we simply need to recall that the signal is not truly periodic and work with only one period.

Conversely, the FT may be represented using only samples of the time waveform by artificially letting the FT become periodic, then letting the time samples play the role of the FS coefficients. In this case, we in effect, let the BLT become periodic along the $j\omega$-axis. This "backward FS" is what is known as the discrete-time Fourier transform (DTFT). The DTFT is discussed in Chapter 4 along with the discrete Fourier transform, a Fourier-type transform which is discrete and periodic in both time and frequency. For the latter, the connections to the BLT are too obtuse to describe in brief terms here.

Finally, if we let the BLT become periodic in ω with some fixed period along each σ line, the BLT can also be represented by discrete-time samples. This is similar to writing a "FS" which changes for each σ. This discrete-time Laplace transform (DTLT) could, in principle, be used in the design and analysis of discrete-time systems in much the same way the BLT is used incontinuous-time work. For historical reasons and for mathematical convenience, however, the ZT is almost universally used. The ZT is obtained from the DTLT using the mapping $e^{sT} \to z$, where T is the sample period on the time signal. As a consequence of this mapping, "strips" in the s-plane map into annuli in the z-plane. Therefore, the ROC of a ZT takes the form of an annulus and the unit circle in the z-plane plays the role of the $j\omega$-axis in the s-plane.

These ideas will become clearer and more precise through the study of successive Chapters 4 and 5. The reader is also encouraged to see Ref. [2] for an elementary approach to discrete FTs.

References

1. G. E. Carlson, *Signal and Linear System Analysis*, Boston: Houghton-Mifflin, 1992.
2. J. R. Deller, Tom, Dick and Mary discover the DFT, *IEEE Signal Process. Mag.*, 11, 36–50, Apr. 1994.
3. R. V. Churchill, J. W. Brown, and R. F. Verhey, *Complex Variables and Applications,* 3rd edn., New York: McGraw-Hill, 1976.
4. G. Doetsch, *Guide to the Applications of Laplace Transforms*, New York: Van Nostrand, 1961.
5. W. H. Hayt and J. E. Kemmerly, *Engineering Circuit Analysis*, New York: McGraw-Hill, 1971.
6. R. C. Houts, *Signal Analysis in Linear Systems*, Philadelphia: Saunders, 1989.
7. A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*, New York: Dover, 1975. Translated and edited by R. A. Silverman.
8. P. Kraniauskas, *Transforms in Signals and Systems*, Reading, MA: Addision-Wesley, 1992.
9. W. LePage, *Complex Variables and the Laplace Transform for Engineers*, New York: McGraw-Hill, 1961.
10. C. D. MacGillem and G. R. Copper, *Continuous and Discrete Signal and System Analysis*, 3rd edn., Philadelphia: Saunders, 1991.

4

Fourier Methods for Signal Analysis and Processing

4.1 Introduction **4**-1
4.2 Classical Fourier Transform for Continuous-Time Signals **4**-2
Properties of the Continuous-Time Fourier Transform • Sampling Models for Continuous- and Discrete-Time Signals • Fourier Spectrum of a Continuous-Time Sampled Signal • Generalized Complex Fourier Transform
4.3 Fourier Series Representation of Continuous-Time Periodic Signals **4**-7
Exponential Fourier Series • Trigonometric Fourier Series • Convergence of the Fourier Series • Fourier Transform of Periodic Continuous-Time Signals
4.4 Discrete-Time Fourier Transform **4**-12
Properties of the Discrete-Time Fourier Transform • Relationship between the CT and DT Spectra
4.5 Discrete Fourier Transform **4**-15
Properties of the DFT • Fast Fourier Transform Algorithms
4.6 Family Tree of Fourier Transforms **4**-19
Walsh–Hadamard Transform
4.7 Selected Applications of Fourier Methods **4**-21
DFT (FFT) Spectral Analysis • FIR Digital Filter Design • Fourier Block Processing in Real-Time Filtering Applications • Fourier Domain Adaptive Filtering • Adaptive Fault Tolerance via Fourier Domain Adaptive Filtering
4.8 Summary **4**-28
References **4**-28

W. Kenneth Jenkins
Pennsylvania State University

4.1 Introduction

The Fourier transform is a mathematical tool that is used to expand signals into a spectrum of sinusoidal components to facilitate the signal representation and the analysis of system performance. In certain applications, the Fourier transform is used for spectral analysis, and while in others it is used for spectrum shaping that adjusts the relative contributions of different frequency components in the filtered result. In certain applications, the Fourier transform is used for its ability to decompose the input signal into uncorrelated components, so that signal processing can be more effectively implemented on the

individual spectral components. Different forms of the Fourier transform, such as the continuous-time (CT) Fourier series, the CT Fourier transform, the discrete-time (DT) Fourier transform (DTFT), the discrete Fourier transform (DFT), and the fast Fourier transform (FFT), are applicable in different circumstances. The goal of this section is to clearly define the various Fourier transforms, to discuss their properties, and to illustrate how each form is related to the others in the context of a family tree of Fourier signal processing methods.

Classical Fourier methods such as the Fourier series and the Fourier integral are used for CT signals and systems, i.e., systems in which the signals are defined at all values of t on the continuum $-\infty < t < \infty$. A more recently developed set of discrete Fourier methods, including the DTFT and the DFT, are extensions of basic Fourier concepts for DT signals and systems. A DT signal is defined only for integer values of n in the range $-\infty < n < \infty$. The class of DT Fourier methods is particularly useful as a basis for digital signal processing (DSP) because it extends the theory of classical Fourier analysis to DT signals and leads to many effective algorithms that can be directly implemented on general computers or special-purpose DSP devices.

4.2 Classical Fourier Transform for Continuous-Time Signals

A CT signal $s(t)$ and its Fourier transform $S(j\omega)$ form a transform pair that are related by the following equations for any $s(t)$ for which the integral (Equation 4.1a) converges:

$$S(j\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t}dt \tag{4.1a}$$

$$s(t) = \frac{1}{2\Pi}\int_{-\infty}^{\infty} S(j\omega)e^{j\omega t}d\omega \tag{4.1b}$$

In most literatures, Equation 4.1a is simply called the *Fourier transform*, whereas Equation 4.1b is called the *Fourier integral*. The relationship $S(j\omega) = F\{s(t)\}$ denotes the Fourier transformation of $s(t)$, where $F\{\cdot\}$ is a symbolic notation for the integral operator, and where ω is the continuous frequency variable expressed in rad/s. A transform pair $s(t) \leftrightarrow S(j\omega)$ represents a one-to-one invertible mapping as long as $s(t)$ satisfies conditions which guarantee that the Fourier integral converges.

In the following discussion, the symbol $\delta(t)$ is used to denote a *CT impulse function* that is defined to be zero for all $t \neq 0$, undefined for $t = 0$, and has unit area when integrated over the range $-\infty < t < \infty$. From Equation 4.1a, it is found that $F\{\delta(t - t_0)\} = e^{-j\omega t_0}$ due to the well-known shifting property of $\delta(t)$. Similarly, from Equation 4.1b we find that $F^{-1}\{2\pi\delta(\omega - \omega_0)\} = e^{j\omega_0 t}$, so that $\delta(t - t_0) \leftrightarrow e^{-j\omega t_0}$ and $e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega - \omega_0)$ are Fourier transform pairs. Using these relationships, it is easy to establish the Fourier transforms of $\cos(\omega_0 t)$ and $\sin(\omega_0 t)$, as well as many other useful waveforms, many of which are listed in Table 4.1.

The CT Fourier transform is useful in the analysis and design of CT systems, i.e., systems that process CT signals. Fourier analysis is particularly applicable to the design of CT filters, which are characterized by Fourier magnitude and phase spectra, i.e., by $|H(j\omega)|$ and $\arg H(j\omega)$, where $H(j\omega)$ is commonly called the frequency response of the filter.

4.2.1 Properties of the Continuous-Time Fourier Transform

The CT Fourier transform has many properties that make it useful for the analysis and design of linear CT systems. Some of the more useful properties are summarized in this section, while a more complete

TABLE 4.1 Continous Time Fourier Transform Pairs

Signal	Fourier Transform	Fourier Series Coefficients (if Periodic)
$\sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t}$	$2\pi \sum_{k=-\infty}^{+\infty} a_k \delta(\omega - k\omega_0)$	a_k
$e^{j\omega_0 t}$	$2\pi\delta(\omega - \omega_0)$	$a_1 = 1$ $a_k = 0$, otherwise
$\cos \omega_0 t$	$\pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)]$	$a_1 = a_{-1} = \frac{1}{2}$ $a_k = 0$, otherwise
$\sin \omega_0 t$	$\frac{\pi}{j}[\delta(\omega - \omega_0) - \delta(\omega + \omega_0)]$	$a_1 = -a_{-1} = \frac{1}{2j}$ $a_k = 0$, otherwise
$x(t) = 1$	$2\pi\delta(\omega)$	$a_0 = 1, a_k = 0, k \neq 0$ (has this Fourier series representation for any choice of $T_0 > 0$)
Periodic square wave $x(t) = \begin{cases} 1, & \lvert t\rvert < T_1 \\ 0, & T_1 < \lvert t\rvert \leq \frac{T_0}{2} \end{cases}$ and $x(t + T_0) = x(t)$	$\sum_{k=-\infty}^{+\infty} \frac{2 \sin k\omega_0 T_1}{k} \delta(\omega - k\omega_0)$	$\frac{\omega_0 T_1}{\pi} \operatorname{sinc}\left(\frac{k\omega_0 T_1}{\pi}\right) = \frac{\sin k\omega_0 T_1}{k\pi}$
$\sum_{n=-\infty}^{+\infty} \delta(t - nT)$	$\frac{2\pi}{T} \sum_{k=-\infty}^{+\infty} \delta\left(\omega - \frac{2\pi k}{T}\right)$	$a_k = \frac{1}{T}$ for all k
$x(t) = \begin{cases} 1, & \lvert t\rvert < T_1 \\ 0, & \lvert t\rvert > T_1 \end{cases}$	$2T_1 \operatorname{sinc}\left(\frac{\omega T_1}{\pi}\right) = \frac{2 \sin \omega T_1}{\omega}$	—
$\frac{W}{\pi} \operatorname{sinc}\left(\frac{Wt}{\pi}\right) = \frac{\sin Wt}{\pi t}$	$X(\omega) = \begin{cases} 1, & \lvert\omega\rvert < W \\ 0, & \lvert\omega\rvert > W \end{cases}$	—
$\delta(t)$	1	—
$u(t)$	$\frac{1}{j\omega} + \pi\delta(\omega)$	—
$\delta(t - t_0)$	$e^{-j\omega t_0}$	—
$e^{-at}u(t)$, $\mathcal{R}e\{a\} > 0$	$\frac{1}{a + j\omega}$	—
$te^{-at}u(t)$, $\mathcal{R}e\{a\} > 0$	$\frac{1}{(a + j\omega)^2}$	—
$\frac{t^{n-1}}{(n-1)!} e^{-at}u(t)$, $\mathcal{R}e\{a\} > 0$	$\frac{1}{(a + j\omega)^n}$	—

Source: Oppenheim, A.V. et al., *Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ, 1983. With permission.

list of the CT Fourier transform properties is given in Table 4.2. Proofs of these properties are found in Oppenheim et al. (1983) and Bracewell (1986). Note that $F\{\cdot\}$ denotes the Fourier transform operation, $F^{-1}\{\cdot\}$ denotes the inverse Fourier transform operation, and "$\star$" denotes the linear convolution operation defined as

$$f_1(t) \star f_2(t) = \int_{-\infty}^{\infty} f_1(t) f_2(t - \tau) d\tau$$

TABLE 4.2 Properties of the CT Fourier Transform

Name	If $\mathcal{F} f(t) = F(j\omega)$, then:
Definition	$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$
	$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(j\omega)e^{j\omega t}d\omega$
Superposition	$\mathcal{F}[af_1(t) + bf_2(t)] = aF_1(j\omega) + bF_2(j\omega)$
Simplification if:	
(*a*) $f(t)$ is even	$F(j\omega) = 2\int_0^{\infty} f(t)\cos\omega t\, dt$
(*b*) $f(t)$ is odd	$F(j\omega) = 2j\int_0^{\infty} f(t)\sin\omega t\, dt$
Negative t	$\mathcal{F} f(-t) = F^*(j\omega)$
Scaling:	
(*a*) Time	$Ff(at) = \frac{1}{\lvert a\rvert}F\left(\frac{j\omega}{a}\right)$
(*b*) Magnitude	$\mathcal{F}af(t) = aF(j\omega)$
Differentiation	$\mathcal{F}\left[\frac{d^n}{dt^n}f(t)\right] = (j\omega)^n F(j\omega)$
Integration	$\mathcal{F}\left[\int_{-\infty}^{t} f(x)dx\right] = \frac{1}{j\omega}F(j\omega) + \pi F(0)\delta(\omega)$
Time shifting	$\mathcal{F} f(t-a) = F(j\omega)e^{-j\omega a}$
Modulation	$\mathcal{F}f(t)e^{j\omega_0 t} = F[j(\omega - \omega_0)]$
	$\mathcal{F}f(t)\cos\omega_0 t = \frac{1}{2}\{F[j(\omega-\omega_0)] + F[j(\omega+\omega_0)]\}$
	$\mathcal{F}f(t)\sin\omega_0 t = \frac{1}{2}j\{F[j(\omega-\omega_0)] + F[j(\omega+\omega_0)]\}$
Time convolution	$\mathcal{F}^{-1}[F_1(j\omega)F_2(j\omega)] = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau$
Frequency convolution	$\mathcal{F}[f_1(t)f_2(t)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(j\lambda)F_2[j(\omega-\lambda)]d\lambda$

Source: Van Valkinburg, M.E., *Network Analysis*, 3rd edn., Prentice Hall, Englewood Cliffs, NJ, 1974. With permission.

1. Linearity (a and b are complex constants)	$F\{af_1(t) + bf_2(t)\} = aF\{f_1(t)\} + bF\{f_2(t)\}$
2. Time-shifting	$F\{f(t-t_0)\} = e^{-j\omega t_0}F\{f(t)\}$
3. Frequency-shifting	$e^{j\omega_0 t}F^{-1}\{F\{j(\omega-\omega_0)\}$
4. Time-domain convolution	$F\{f_1(t)^* f_2(t)\} = F\{f_1(t)\}\cdot F\{f_2(t)\}$
5. Frequency-domain convolution	$F\{f_1(t)\cdot f_2(t)\} = \frac{1}{2\Pi}F\{f_1(t)\}^* F\{f_2(t)\}$
6. Time-differentiation	$-j\omega F(j\omega) = F\{d(f(t))/dt\}$
7. Time-integration	$F\left\{\int_{-\infty}^{t} f(\tau)d\tau\right\} = \frac{1}{j\omega}F(j\omega) + \pi F(0)\delta(\omega)$

The above properties are particularly useful in CT system analysis and design, especially when the system characteristics are easily specified in the frequency domain, as in linear filtering. Note that properties (1), (6), and (7) are useful for solving differential or integral equations. Property (4) (time-domain

convolution) provides the basis for many signal processing algorithms, since many systems can be specified directly by their impulse or frequency response. Property (3) (frequency-shifting) is useful for analyzing the performance of communication systems where different modulation formats are commonly used to shift spectral energy among different frequency bands.

4.2.2 Sampling Models for Continuous- and Discrete-Time Signals

The relationship between the CT and the DT domains is characterized by the operations of sampling and reconstruction. If $s_a(t)$ denotes a signal $s(t)$ that has been uniformly sampled every T seconds, then the mathematical representation of $s_a(t)$ is given by

$$S_a(t) = \sum_{n=-\infty}^{n=\infty} s(t)\delta(t - nT) \tag{4.2a}$$

where $\delta(t)$ is the CT impulse function defined previously. Since the only places where the product $s(t)\delta(t - nT)$ is not identically equal to zero are at the sampling instances, $s(t)$ in Equation 4.2a can be replaced with $s(nT)$ without changing the overall meaning of the expression. Hence, an alternate expression for $s_a(t)$ that is often useful in Fourier analysis is

$$S_a(t) = \sum_{n=-\infty}^{n=\infty} s(nT)\delta(t - nT) \tag{4.2b}$$

The CT sampling model $s_a(t)$ consists of a sequence of CT impulse functions uniformly spaced at intervals of T seconds and weighted by the values of the signal $s(t)$ at the sampling instants, as depicted in Figure 4.1. Note that $s_a(t)$ is not defined at the sampling instants because the CT impulse function itself is not defined at $t = 0$. However, the values of $s(t)$ at the sampling instants are imbedded as "area under the curve" of $s_a(t)$, and as such represent a useful mathematical model of the sampling process. In the DT domain, the sampling model is simply the sequence defined by taking the values of $s(t)$ at the sampling instants, i.e.,

$$s[n] = s(t)|_{t=nT} \tag{4.3}$$

In contrast to $s_a(t)$, which is not defined at the sampling instants, $s[n]$ is well defined at the sampling instants, as illustrated in Figure 4.2. From this discussion, it is now clear that $s_a(t)$ and $s[n]$ are different but equivalent models of the sampling process in the CT and DT domains, respectively. They are both useful for signal analysis in their corresponding domains. It will be shown later that their equivalence is established by the fact that they have equal spectra in the Fourier domain, and that the underlying CT

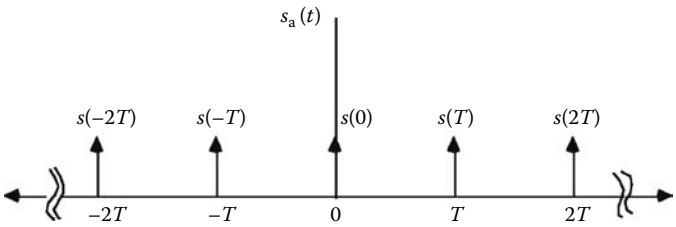


FIGURE 4.1 CT model of a sampled CT signal.

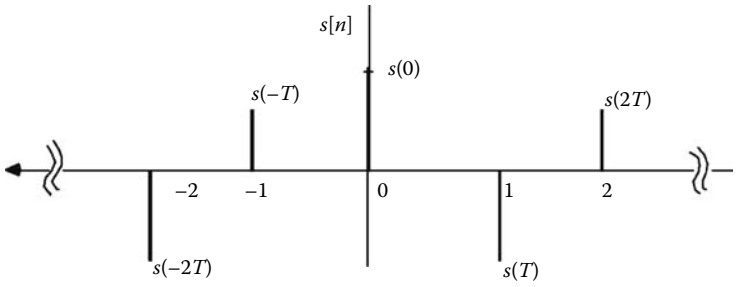


FIGURE 4.2 DT model of a sampled CT signal.

signal from which $s_a(t)$ and $s[n]$ are derived can be recovered from either sampling representation provided that a sufficiently high sampling rate is used in the sampling operation.

4.2.3 Fourier Spectrum of a Continuous-Time Sampled Signal

The operation of uniformly sampling a CT signal $s(t)$ at every T seconds is characterized by Equation 4.2a and b, where $\delta(t)$ is the CT time impulse function defined earlier:

$$s_a(t) = \sum_{n=-\infty}^{\infty} s_a(t)\delta(t - nT) = \sum_{n=-\infty}^{\infty} s_a(nT)\delta(t - nT)$$

Since $s_a(t)$ is a CT signal, it is appropriate to apply the CT Fourier transform to obtain an expression for the spectrum of the sampled signal:

$$F\{s_a(t)\} = F\left\{\sum_{n=-\infty}^{\infty} s_a(nT)\delta(t - nT)\right\} = \sum_{n=-\infty}^{\infty} s_a(nT)[e^{j\omega T}]^{-n} \tag{4.4}$$

Since the expression on the right-hand side of Equation 4.4 is a function of $e^{j\omega T}$, it is customary to express the transform as $F(e^{j\omega T}) = F\{s_a(t)\}$. If ω is replaced with a normalized frequency $\omega' = \omega/T$, so that $-\pi < \omega' < \pi$, then the right side of Equation 4.4 becomes identical to the DTFT that is defined directly for the sequence $s[n] = s_a(nT)$.

4.2.4 Generalized Complex Fourier Transform

The CT Fourier transform characterized by Equation 4.1 can be generalized by considering the variable $j\omega$ to be the special case of $u = \sigma + j\omega$ with $\sigma = 0$, writing Equation 4.1 in terms of u, and interpreting u as a complex frequency variable. The resulting complex Fourier transform pair is given by Equation 4.5a and b (Bracewell 1986):

$$s(t) = \frac{1}{2\Pi j}\int_{\sigma - j\infty}^{\sigma + j\infty} S(u)e^{jut}du \tag{4.5a}$$

$$S(u) = \int_{-\infty}^{\infty} s(t)e^{-jut}dt \tag{4.5b}$$

The set of all values of u for which the integral of Equation 4.5b converges is called the region of convergence (ROC). Since the transform $S(u)$ is defined only for values of u within the ROC, the path of integration in Equation 4.5a must be defined so the entire path lies within the ROC. In some literature this transform pair is called the bilateral Laplace transform because it is the same result obtained by including both the negative and positive portions of the time axis in the classical Laplace transform integral. The complex Fourier transform (bilateral Laplace transform) is not often used in solving practical problems, but its significance lies in the fact that it is the most general form that represents the place where Fourier and Laplace transform concepts merge together. Identifying this connection reinforces the observation that Fourier and Laplace transform concepts share common properties because they result from placing different constraints on the same parent form.

4.3 Fourier Series Representation of Continuous-Time Periodic Signals

The classical Fourier series representation of a periodic time-domain signal $s(t)$ involves an expansion of $s(t)$ into an infinite series of terms that consist of sinusoidal basis functions, each weighted by a complex constant (Fourier coefficient) that provides the proper contribution of that frequency component to the complete waveform. The conditions under which a periodic signal $s(t)$ can be expanded in a Fourier series are known as the *Dirichlet conditions*. They require that in each period $s(t)$ has a finite number of discontinuities, a finite number of maxima and minima, and satisfies the absolute convergence criterion of Equation 4.6 (VanValkenburg 1974):

$$\int_{-T/2}^{T/2} |s(t)|\mathrm{d}t < \infty \tag{4.6}$$

It is assumed throughout the following discussion that the Dirichlet conditions are satisfied by all functions that will be represented by a Fourier series.

4.3.1 Exponential Fourier Series

If $s(t)$ is a CT periodic signal with period T the exponential Fourier series expansion of $s(t)$ is given by

$$s(t) = \sum_{n=-\infty}^{\infty} a_n \mathrm{e}^{jn\omega_0 t} \tag{4.7a}$$

where $\omega_0 = 2\pi/T$. The a_ns are the complex Fourier coefficients given by

$$a_n = \frac{1}{T} \int_{-T/2}^{T/2} s(t)\mathrm{e}^{-jn\omega_0 t}\mathrm{d}t, \quad -\infty < n < \infty \tag{4.7b}$$

For every value of t where $s(t)$ is continuous the right side of Equation 4.7a converges to $s(t)$. At values of t where $s(t)$ has a finite jump discontinuity, the right side of Equation 4.7a converges to the average of $s(t^-)$ and $s(t^+)$, where

$$s(t^-) = \lim_{\varepsilon \to 0} (t - \varepsilon) \quad \text{and} \quad s(t^+) = \lim_{\varepsilon \to 0} (t + \varepsilon).$$

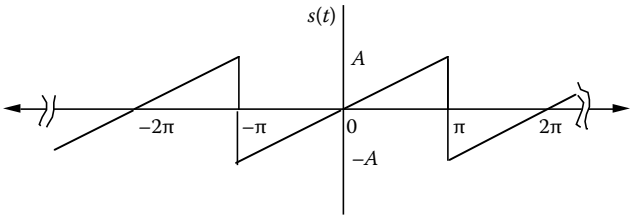


FIGURE 4.3 Periodic CT signal used in Fourier series Example 1.

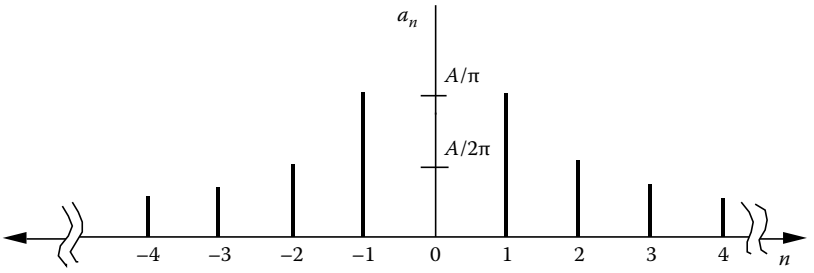


FIGURE 4.4 Magnitude of the Fourier coefficients for Example 1.

For example, the Fourier series expansion of the sawtooth waveform (Example 1) illustrated in Figure 4.3 is characterized by $T = 2\pi$, $\omega_0 = 1$, $a_0 = 0$, and $a_n = a_{-n} = A\ \cos(n\pi)/(jn\pi)$ for $n = 1, 2, \ldots$. The coefficients of the exponential Fourier series given by Equation 4.7b can be interpreted as a spectral representation of $s(t)$, since the a_nth coefficient represents the contribution of the $(n\omega_0)$th frequency component to the complete waveform. Since the a_ns are complex valued, the Fourier domain (spectral) representation has both magnitude and phase spectra. For example, the magnitudes of the a_ns are plotted in Figure 4.4 for the sawtooth waveform of Figure 4.3. The fact that the a_ns constitute a discrete set is consistent with the fact that a periodic signal has a spectrum that contains only integer multiples of the fundamental frequency ω_0. The equation pair given by Equation 4.7a and b can be interpreted as a transform pair that is similar to the CT Fourier transform for periodic signals. This leads to the observation that the classical Fourier series can be interpreted as a special transform that provides a one-to-one invertible mapping between the discrete-spectral domain and the CT domain.

4.3.2 Trigonometric Fourier Series

Although the complex form of the Fourier series expansion is useful for complex periodic signals, the Fourier series can be more easily expressed in terms of real-valued sine and cosine functions for real-valued periodic signals. In the following discussion it is assumed that the signal $s(t)$ is real-valued. When $s(t)$ is periodic and real-valued, it is convenient to replace the complex exponential Fourier series with a *trigonometric expansion* that contains $\sin(\omega_0 t)$ and $\cos(\omega_0 t)$ terms with corresponding real-valued coefficients (VanValkenburg 1974). The trigonometric form of the Fourier series for a real-valued signal $s(t)$ is given by

$$s(t) = \sum_{n=0}^{\infty} b_n \cos(n\omega_0) + \sum_{n=1}^{\infty} c_n \sin(n\omega_0) \tag{4.8a}$$

where $\omega_0 = 2\pi/T$. In Equation 4.8a the b_ns and c_ns are real-valued Fourier coefficients determined by

$$b_0 = \frac{1}{T}\int_{-T/2}^{T/2} s(t)\mathrm{d}t$$
$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} s(t)\cos(n\omega_0 t)\mathrm{d}t, \quad n = 1, 2, \ldots \tag{4.8b}$$

and

$$c_n = \frac{2}{T}\int_{-T/2}^{T/2} s(t)\sin(n\omega_0 t)\mathrm{d}t, \quad n = 1, 2, \ldots$$

An arbitrary real-valued signal $s(t)$ can be expressed as a sum of even and odd components, $s(t) = s_{\text{even}}(t) + s_{\text{odd}}(t)$, where $s_{\text{even}}(t) = s_{\text{even}}(-t)$ and $s_{\text{odd}}(t) = -s_{\text{odd}}(-t)$, and where $s_{\text{even}}(t) = [s(t) + s(-t)]/2$ and $s_{\text{odd}}(t) = [s(t) - s(-t)]/2$. For the trigonometric Fourier series, it can be shown that $s_{\text{even}}(t)$ is represented by the (even) cosine terms in the infinite series, $s_{\text{odd}}(t)$ is represented by the (odd) sine terms, and b_0 is the DC level of the signal. Therefore, if it can be determined by inspection that a signal has a DC level, or if it is even or odd, then the correct form of the trigonometric series can be chosen to simplify the analysis. For example, it is easily seen that the signal shown in Figure 4.5 (Example 2) is an even signal with a zero DC level, and therefore, can be accurately represented by the cosine series with $b_n = 2A\ \sin(\pi n/2)/(\pi n/2)$, $n = 1, 2, \ldots$ as shown in Figure 4.6. In contrast note that the sawtooth waveform used in the previous example is an odd signal with zero DC level, so that it can be completely

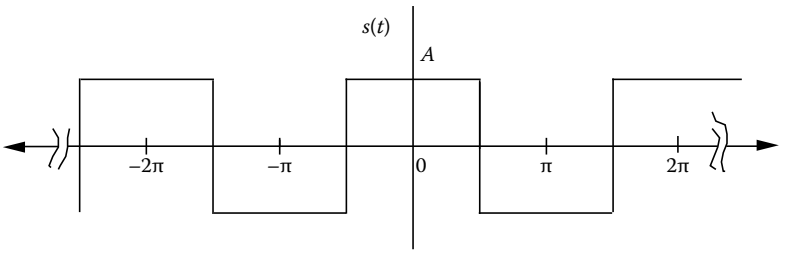


FIGURE 4.5 Periodic CT signal used in Fourier series Example 2.

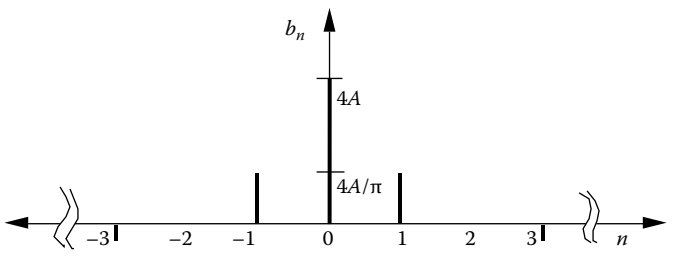


FIGURE 4.6 Fourier coefficients for example of Figure 4.5.

specified by the sine terms of the trigonometric series. This result can be demonstrated by pairing each positive frequency component from the exponential series with its conjugate partner, i.e., $c_n = \sin(n\omega_0 t) = a_n e^{jn\omega_0 t} + a_{-n} e^{-jn\omega_0 t}$, whereby it is found that $c_n = 2A\cos(n\pi)/(n\pi)$ for this example. In general, it is found that $a_n = (b_n - jc_n)/2$ for $n = 1, 2, \ldots$, $a_0 = b_0$, and $a_{-n} = a_n^*$. The trigonometric Fourier series is common in the signal processing literature because it replaces complex coefficients with real ones and often results in a simpler and more intuitive interpretation of the results.

4.3.3 Convergence of the Fourier Series

The Fourier series representation of a periodic signal is an approximation that exhibits mean squared convergence to the true signal. If $s(t)$ is a periodic signal of period T, and $s'(t)$ denotes the Fourier series approximation of $s(t)$, then $s(t)$ and $s'(t)$ are equal in the mean square sense if

$$\text{mse} = \int_{-T/2}^{T/2} |s(t) - s'(t)|^2 \, dt = 0 \tag{4.9}$$

Even if Equation 4.9 is satisfied, *mean square error* (MSE) convergence does not guarantee that $s(t) = s'(t)$ at every value of t. In particular, it is known that at values of t where $s(t)$ is discontinuous, the Fourier series converges to the average of the limiting values to the left and right of the discontinuity. For example, if t_0 is a point of discontinuity, then $s'(t_0) = [s(t_0^-) + s(t_0^+)]/2$, where $s(t_0^-)$ and $s(t_0^+)$ were defined previously (note that at points of continuity, this condition is also satisfied by the very definition of continuity). Since the Dirichlet conditions require that $s(t)$ have at most a finite number of points of discontinuity in one period, the set S_t such that $s(t) \neq s'(t)$ within one period contains a finite number of points, and S_t is a set of measure zero in the formal mathematical sense. Therefore $s(t)$ and its Fourier series expansion $s'(t)$ are *equal almost everywhere*, and $s(t)$ can be considered identical to $s'(t)$ for analysis in most practical engineering problems.

The condition of convergence almost everywhere is satisfied only in the limit as an infinite number of terms are included in the Fourier series expansion. If the infinite series expansion of the Fourier series is truncated to a finite number of terms, as it must always be in practical applications, then the approximation will exhibit an oscillatory behavior around the discontinuity, known as the *Gibbs phenomenon* (VanValkenburg 1974). Let $s'_N(t)$ denote a truncated Fourier series approximation of $s(t)$, where only the terms in Equation 4.7a from $n = -N$ to $n = N$ are included if the complex Fourier series representation is used, or where only the terms in Equation 4.8a from $n = 0$ to $n = N$ are included if the trigonometric form of the Fourier series is used. It is well known that in the vicinity of a discontinuity at t_0 the Gibbs phenomenon causes $s'_N(t)$ to be a poor approximation to $s(t)$. The peak magnitude of the Gibbs oscillation is 13% of the size of the jump discontinuity $s(t_0^-) - s(t_0^+)$ regardless of the number of terms used in the approximation. As N increases, the region that contains the oscillation becomes more concentrated in the neighborhood of the discontinuity, until, in the limit as N approaches infinity, the Gibbs oscillation is squeezed into a single point of mismatch at t_0. The Gibbs phenomenon is illustrated in Figure 4.7 where an ideal low-pass frequency response is approximated by impulse response function that has been limited to having only N nonzero coefficients, and hence the Fourier series expansion contains only a finite number of terms.

An important property of the Fourier series is that the exponential basis functions $e^{jn\omega_0 t}$ (or $\sin(n\omega_0 t)$ and $\cos(n\omega_0 t)$ for the trigonometric form) for $n = 0, \pm 1, \pm 2, \ldots$ (or $n = 0, 1, 2, \ldots$ for the trigonometric form) constitute an *orthonormal set*, i.e., $t_{nk} = 1$ for $n = k$, and $t_{nk} = 0$ for $n \neq k$, where

$$t_{nk} = \frac{1}{T} \int_{-T/2}^{T/2} (e^{-jn\omega_0 t})(e^{jk\omega_0 t}) \, dt$$

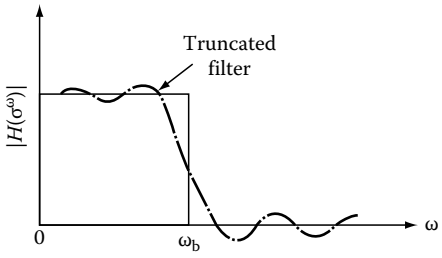


FIGURE 4.7 Gibbs phenomenon in a low-pass digital filter caused by truncating the impulse response to N terms.

As terms are added to the Fourier series expansion, the orthogonality of the basis functions guarantees that the approximation error decreases in the mean square sense, i.e., that MSE_N decreases monotonically as N is increased, where

$$\text{MSE}_N = \int_{-T/2}^{T/2} \left| s(t) - s'_N(t) \right|^2 \mathrm{d}t$$

Therefore, when applying Fourier series analysis including more terms always improves the accuracy of the signal representation.

4.3.4 Fourier Transform of Periodic Continuous-Time Signals

For a periodic signal $s(t)$, the CT Fourier transform can then be applied to the Fourier series expansion of $s(t)$ to produce a mathematical expression for the "line spectrum" that is characteristic of periodic signals:

$$F\{s(t)\} = F\left\{ \sum_{n=-\infty}^{\infty} a_n \mathrm{e}^{jn\omega_0 t} \right\} = 2\pi \sum_{n=-\infty}^{\infty} a_n \delta(\omega - \omega_0) \tag{4.10}$$

The spectrum is shown in Figure 4.8. Note the similarity between the spectral representation of Figure 4.8 and the plots of the Fourier coefficients in Figures 4.4 and 4.6, which were heuristically interpreted as a line spectrum. Figures 4.4 and 4.6 are different from Figure 4.8 but they are equivalent representations of the Fourier line spectrum that is characteristic of periodic signals.

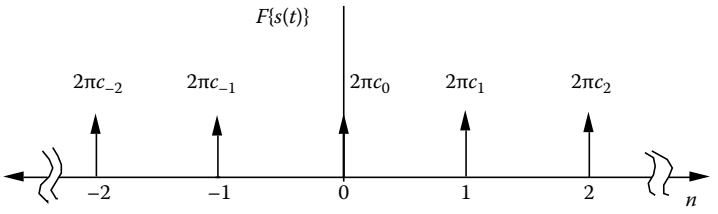


FIGURE 4.8 Spectrum of the Fourier representation of a periodic signal.

4.4 Discrete-Time Fourier Transform

The DTFT is obtained directly in terms of the sequence samples $s[n]$ by taking the relationship obtained in Equation 4.4 to be the definition of the DTFT. Letting $T = 1$ so that the sampling period is removed from the equations and the frequency variable is replaced with a normalized frequency $\omega' = \omega T$, the DTFT pair is defined by Equation 4.11. In order to simplify notation, it is not customary to distinguish between ω and ω', but rather to rely on the context of the discussion to determine whether ω refers to the normalized ($T = 1$) or the unnormalized ($T \neq 1$) frequency variable.

$$S(e^{j\omega'}) = \sum_{n=-\infty}^{\infty} s[n]e^{-j\omega' n} \tag{4.11a}$$

$$s[n] = \frac{1}{2\Pi} \int_{-\Pi}^{\Pi} S(e^{j\omega'})e^{jn\omega'} d\omega' \tag{4.11b}$$

The spectrum $S(e^{j\omega'})$ is periodic in ω' with period 2π. The fundamental period in the range $-\pi < \omega' \leq \pi$, referred to as the baseband, is the useful frequency range of the DT system because frequency components in this range can be represented unambiguously in sampled form (without aliasing error). In much of the signal processing literature the explicit primed notation is omitted from the frequency variable. However, the explicit primed notation will be used throughout this section because there is a potential for confusion when so many related Fourier concepts are discussed within the same framework.

By comparing Equations 4.4 and 4.11a, and noting that $\omega' = \omega T$, it is seen that

$$F\{s_a(t)\} = \text{DTFT}\{s[n]\}$$

where $s[n] = s_a(t)|_{t=nT}$. This demonstrates that the spectrum of $s_a(t)$ as calculated by the CT Fourier transform is identical to the spectrum of $s[n]$ as calculated by the DTFT. Therefore although $s_a(t)$ and $s[n]$ are quite different sampling models, they are equivalent in the sense that they have the same Fourier domain representation. A list of common DTFT pairs is presented in Table 4.3. Just as the CT Fourier transform is useful in CT signal system analysis and design, the DTFT is equally useful for DT system analysis and design.

In the same way that the CT Fourier transform was found to be a special case of the complex Fourier transform (or bilateral Laplace transform), the DTFT is a special case of the *bilateral z-transform* with $z = e^{j\omega' t}$. The more general bilateral z-transform is given by

$$S(z) = \sum_{n=-\infty}^{\infty} s[n]z^{-n} \tag{4.12a}$$

$$s[n] = \frac{1}{2\pi j} \oint_c S(z)z^{n-1} dz \tag{4.12b}$$

where C is a counter clockwise contour of integration which is a closed path completely contained within the ROC of $S(z)$. Recall that the DTFT was obtained by taking the CT Fourier transform of the CT sampling model $s_a(t)$. Similarly, the bilateral z-transform results by taking the bilateral Laplace transform of $s_a(t)$. If the lower limit on the summation of Equation 4.12a is taken to be $n = 0$, then Equation 4.12a and b becomes the one-sided z-transform, which is the DT equivalent of the one-sided Laplace transform for CT signals.

TABLE 4.3 Some Basic Discrete Time Fourier Transform Pairs

Sequence	Fourier Transform
1. $\delta[n]$	1
2. $\delta[n - n_0]$	$e^{-j\omega n_0}$
3. $1\ (-\infty < n < \infty)$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega + 2\pi k)$
4. $a^n u[n]\ (\lvert a\rvert < 1)$	$\dfrac{1}{1 - ae^{-j\omega}}$
5. $u[n]$	$\dfrac{1}{1 - e^{-j\omega}} + \sum_{k=-\infty}^{\infty} \pi\delta(\omega + 2\pi k)$
6. $(n + 1)a^n u[n]\ (\lvert a\rvert < 1)$	$\dfrac{1}{(1 - ae^{-j\omega})^2}$
7. $\dfrac{r^n \sin\omega_p(n + 1)}{\sin\omega_p} u[n]\ (\lvert r\rvert < 1)$	$\dfrac{1}{1 - 2r\cos\omega_p e^{-j\omega} + r^2 e^{-j2\omega}}$
8. $\dfrac{\sin\omega_c n}{\pi n}$	$X(e^{j\omega}) = \begin{cases} 1, & \lvert\omega\rvert < \omega_c, \\ 0, & \omega_c < \lvert\omega\rvert \le \pi \end{cases}$
9. $x[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$	$\dfrac{\sin[\omega(M + 1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}$
10. $e^{j\omega_0 n}$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega - \omega_0 + 2\pi k)$
11. $\cos(\omega_0 n + \phi)$	$\pi \sum_{k=-\infty}^{\infty} [e^{j\phi}\delta(\omega - \omega_0 + 2\pi k) + e^{-j\phi}\delta(\omega + \omega_0 + 2\pi k)]$

Source: Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989. With permission.

4.4.1 Properties of the Discrete-Time Fourier Transform

Since the DTFT is a close relative of the classical CT Fourier transform, it should come as no surprise that many properties of the DTFT are similar to those of the CT Fourier transform. In fact, for many of the properties presented earlier, there is an analogous property for the DTFT. The following list parallels the list that was presented earlier for the CT Fourier transform, to the extent that the same properties exist (a more complete list of DTFT properties is given in Table 4.4). Note that $F\{\cdot\}$ denotes the DT Fourier transform operation, $F^{-1}\{\cdot\}$ denotes the inverse DT Fourier transform operation, and "$\star$" denotes the DT convolution operation defined as

$$f_1[n]\star f_2[n] = \sum_{k=-\infty}^{+\infty} f_1[n] f_2[n-k]$$

1. Linearity (a and b are complex constants)	$\text{DTFT}\{af_1[n] + bf_2[n]\} = a \cdot \text{DTFT}\{f_1[n]\} + b \cdot \text{DTFT}\{f_2[n]\}$
2. Index-shifting	$\text{DTFT}\{f[n - n_0]\} = e^{-j\omega n_0}\ \text{DTFT}\{f[n]\}$
3. Frequency-shifting	$e^{j\omega_0 n} f[n] = \text{DTFT}^{-1}\{F(j\omega - \omega_0)\}$
4. Time-domain convolution	$\text{DTFT}\{f_1[n]\star f_2[n]\} = F\{f_1[n]\} \cdot F\{f_2[n]\}$
5. Frequency-domain convolution	$\text{DTFT}\{f_1[n] \cdot f_2[n]\} = \frac{1}{2\Pi}\text{DTFT}\{f_1[n]\} * \text{DTFT}\{f_2[n]\}$
6. Frequency-differentiation	$nf[n] = \text{DTFT}^{-1}\{dF(j\omega)/d\omega\}$

Note that the time-differentiation and time-integration properties of the CT Fourier transform do not have analogous counterparts in the DTFT because time-domain differentiation and integration are not

TABLE 4.4 Properties of the Discrete Time Fourier Transform

Sequence $x[n]$ $y[n]$	Fourier Transform $X(e^{j\omega})$ $Y(e^{j\omega})$
1. $ax[n] + by[n]$	$aX(e^{j\omega}) + bY(e^{j\omega})$
2. $x[n - n_d]$ (n_d an integer)	$e^{-j\omega n_d} X(e^{j\omega})$
3. $e^{j\omega_0 n}x[n]$	$X(e^{j(\omega-\omega_0)})$
4. $x[-n]$	$X(e^{-j\omega})$ if $x[n]$ real $X^*(e^{j\omega})$
5. $nx[n]$	$j\dfrac{dX(e^{j\omega})}{d\omega}$
6. $x[n] * y[n]$	$X(e^{j\omega})Y(e^{j\omega})$
7. $x[n]y[n]$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})d\theta$
Parseval's theorem	
8. $\sum_{n=-\infty}^{\infty} \lvert x[n]\rvert^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \lvert X(e^{j\omega})\rvert^2 d\omega$	
9. $\sum_{n=-\infty}^{\infty} x[n]y^*[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})Y^*(e^{j\omega})d\omega$	

Source: Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989. With permission.

defined for DT signals. When working with DT systems practitioners must often manipulate difference equations in the frequency domain. For this purpose the properties of linearity and index-shifting are very important. As with the CT Fourier transform, time-domain convolution is also important for DT systems because it allows engineers to work with the frequency response of the system in order to achieve proper shaping of the input spectrum, or to achieve frequency selective filtering for noise reduction or signal detection.

4.4.2 Relationship between the CT and DT Spectra

Since DT signals often originate by sampling a CT signal, it is important to develop the relationship between the original spectrum of the CT signal and the spectrum of the DT signal that results. First, the CT Fourier transform is applied to the CT sampling model, and the properties are used to produce the following result:

$$F\{s_a(t)\} = F\left\{s_a(t)\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\} = \frac{1}{2\pi}S_a(j\omega)F\left\{\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\} \tag{4.13}$$

Since the sampling function (summation of shifted impulses) on the right-hand side of Equation 4.13 is periodic with period T it can be replaced with a CT Fourier series expansion and the frequency-domain convolution property of the CT Fourier transform can be applied to yield two equivalent expressions for the DT spectrum:

$$S(e^{j\omega T}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} S_a(j[\omega - n\omega_s]) \quad \text{or} \quad S(e^{j\omega'}) = \frac{1}{T}\sum_{n=-\infty}^{\infty} S_a(j[\omega' - n2\pi/T]) \tag{4.14}$$

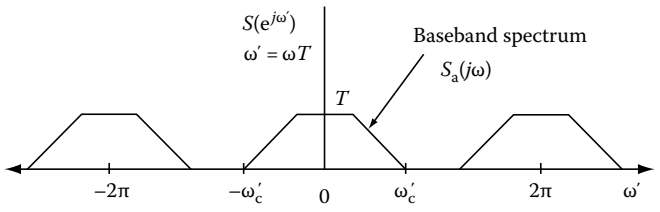


FIGURE 4.9 Relationship between the CT and DT spectra.

In Equation 4.14 $\omega_s = (2\pi/T)$ is the sampling frequency and $\omega' = \omega T$ is the normalized DT frequency axis expressed in radians. Note that $S(e^{j\omega T}) = S(e^{j\omega'})$ consists of an infinite number of replicas of the CT spectrum $S(j\omega)$, positioned at intervals of $(2\pi/T)$ on the ω axis (or at intervals of 2π on the ω' axis), as illustrated in Figure 4.9. Note that if $S(j\omega)$ is band-limited with a bandwidth ω_c, and if T is chosen sufficiently small so that $\omega_s > 2\omega_c$, then the DT spectrum is a copy of $S(j\omega)$ (scaled by $1/T$) in the baseband. The limiting case of $\omega_s = 2\omega_c$ is called the *Nyquist sampling frequency*. Whenever a CT signal is sampled at or above the Nyquist rate, no aliasing distortion occurs (i.e., the baseband spectrum does not overlap with the higher order replicas) and the CT signal can be exactly recovered from its samples by extracting the baseband spectrum of $S(e^{j\omega'})$ with an ideal low-pass filter that recovers the original CT spectrum by removing all spectral replicas outside the baseband and scaling the baseband by a factor of T.

4.5 Discrete Fourier Transform

To obtain the DFT the continuous-frequency domain of the DTFT is sampled at N points uniformly spaced around the unit circle in the z-plane, i.e., at the points $\omega_k = (2\pi k/N)$, $k = 0,\ 1, \ldots, N-1$. The result is the DFT transform pair defined by Equation 4.15a and b:

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1, \ldots, N-1 \tag{4.15a}$$

$$s[k] = \frac{1}{N} \sum_{k=0}^{N-1} S[k] e^{j\frac{2\pi kn}{N}}, \quad n = 0, 1, \ldots, N-1 \tag{4.15b}$$

The signal $s[n]$ is either a finite length sequence of length N, or it is a periodic sequence with period N. Regardless of whether $s[n]$ is a finite length or periodic sequence, the DFT treats the N samples of $s[n]$ as though they are one period of a periodic sequence. This is a peculiar feature of the DFT, and one that must be handled properly in signal processing to prevent the introduction of artifacts.

4.5.1 Properties of the DFT

Important properties of the DFT are summarized in Table 4.5. The notation $([k])_N$ denotes k modulo N, and $R_N[n]$ is a rectangular window such that $R_N[n] = 1$ for $n = 0, \ldots, N-1$, and $R_N[n] = 0$ for $n < 0$ and $n \geq N$. The transform relationship given by Equation 4.15a and b is also valid when $s[n]$ and $S[k]$ are periodic sequences, each of period N. In this case n and k are permitted to range over the complete set of real integers, and $S[k]$ is referred to as the discrete Fourier series (DFS). In some cases, the DFS is developed as a distinct transform pair in its own right (Jenkins 1986). Whether or not the DFT and the DFS are considered identical or distinct is not important in this discussion. The important point to be

TABLE 4.5 Properties of the Discrete Fourier Transform (DFT)

Finite-Length Sequence (Length N)	N-Point DFT (Length N)
1. $x[n]$	$X[k]$
2. $x_1[n]$, $x_2[n]$	$X_1[k]$, $X_2[k]$
3. $ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
4. $X[n]$	$Nx[((-k))_N]$
5. $x[(\,(n-m))_N]$	$W_N^{km}X[k]$
6. $W_N^{-\ell n}x[n]$	$X[((k-\ell))_N]$
7. $\sum_{m=0}^{N-1} x_1(m)x_2[((n-m))_N]$	$X_1[k]X_2[k]$
8. $x_1[n]x_2[n]$	$\frac{1}{N}\sum_{\ell=0}^{N-1} X_1(\ell)X_2[((k-\ell))_N]$
9. $x^*[n]$	$X^*[(\,(-k))_N]$
10. $x^*[((-n))_N]$	$X^*[k]$
11. $\mathcal{R}e\{x[n]\}$	$X_{ep}[k] = \frac{1}{2}\{X[((k))_N] + X^*[((-k))_N]\}$
12. $j\mathcal{I}\mathrm{m}\{x[n]\}$	$X_{op}[k] = \frac{1}{2}\{X[((k))_N] - X^*[((-k))_N]\}$
13. $x_{ep}[n] = \frac{1}{2}\{x[n] + x^*[((-n))_N]\}$	$\mathcal{R}\mathrm{e}\ \{X[k]\}$
14. $x_{op}[n] = \frac{1}{2}\{x[n] - x^*[((-n))_N]\}$	$j\mathcal{I}\mathrm{m}\ \{X[k]\}$
Properties 15–17 apply only when $x[n]$ is real.	
15. Symmetry properties	$\begin{cases} X[k] & = X^*[((-k))_N] \\ Re\{X[k]\} & = Re\{X[((-k))_N]\} \\ \mathcal{I}\mathrm{m}\{X[k]\} & = -\mathcal{I}\mathrm{m}\{X[((-k))_N]\} \\ \lvert X[k]\rvert & = \lvert X[((-k))_N]\rvert \\ \measuredangle\{X[k]\} & = -\measuredangle\{X[((-k))_N]\} \end{cases}$
16. $x_{ep}[n] = \frac{1}{2}\{x[n] + x[((-n))_N]\}$	$\mathcal{R}\mathrm{e}\{X[k]\}$
17. $x_{op}[n] = \frac{1}{2}\{x[n] - x[((-n))_N]\}$	$j\mathcal{I}\mathrm{m}\{X[k]\}$

Source: Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989. With permission.

emphasized here is that the DFT treats $s[n]$ as though it were a single period of a periodic sequence, and all signal processing done with the DFT will inherit the consequences of this assumed periodicity.

Most of the properties listed in Table 4.5 for the DFT are similar to those of the Z-transform and the DTFT, although there are important differences. For example, Property (5) (time-shifting property), holds for circular shifts of the finite length sequence $s[n]$, which is consistent with the notion that the DFT treats $s[n]$ as one period of a periodic sequence. Also, the multiplication of two DFTs results in the *circular convolution* of the corresponding DT sequences, as specified by Property (7). This latter property is quite different from *the linear convolution* property of the DTFT. Circular convolution is simply a linear convolution of the periodic extensions of the finite sequences being convolved, where each of the finite sequences of length N defines the structure of one period of the periodic extensions.

For example, suppose it is desired to implement a digital filter with finite impulse response (FIR) $h[n]$. The output in response to $s[n]$ is

$$y[n] = \sum_{k=0}^{N-1} h[k]s[n-k] \tag{4.16}$$

which is obtained by transforming $h[n]$ and $s[n]$ into $H[k]$ and $S[k]$ using the DFT, multiplying the transforms point-wise to obtain $Y[k] = H[k]S[k]$, and then using the inverse DFT to obtain $y[n] = \mathrm{DFT}^{-1}\{Y[k]\}$. If $s[n]$ is a finite sequence of length M, then the results of the circular convolution

implemented by the DFT will correspond to the desired linear convolution if and only if the block length of the DFT, N_{DFT}, is chosen sufficiently large so that $N_{\mathrm{DFT}} > N + M - 1$ and both $h[n]$ and $s[n]$ are padded with zeros to form blocks of length N_{DFT}.

4.5.2 Fast Fourier Transform Algorithms

The DFT is typically implemented in practice with one of the common forms of the FFT algorithm. The FFT is not a Fourier transform in its own right, but rather it is simply a computationally efficient algorithm that reduces the complexity of the computing DFT from Order $\{N^2\}$ to Order $\{N \log_2 N\}$. When N is large, the computational savings provided by the FFT algorithm is so great that the FFT makes real-time DFT analysis practical in many situations which would be entirely impractical without it. There are numerous FFT algorithms, including decimation-in-time (D-I-T) algorithms, decimation-in-frequency (D-I-F) algorithms, bit-reversed algorithms, normally ordered algorithms, mixed-radix algorithms (for block lengths that are not powers-of-2 [PO2]), prime factor algorithms, and Winograd algorithms (Blahut 1985). The D-I-T and the D-I-F radix-2 FFT algorithms are the most widely used in practice. Detailed discussions of various FFT algorithms can be found in Brigham (1974) and Oppenheim and Schafer (1975).

The FFT is easily understood by examining the simple example of $N = 8$. There are numerous ways to develop the FFT algorithm, all of which deal with a nested decomposition of the summation operator of Equation 4.20a. The development presented here is called an *algebraic* development of the FFT because it follows straightforward algebraic manipulation. First, each of the summation indices (k, n) in Equation 4.15a is expressed as explicit binary integers, $k = k_2 4 + k_1 2 + k_0$ and $n = n_2 4 + n_1 2 + n_0$, where k_i and n_i are bits that take on the values of either 0 or 1. If these expressions are substituted into Equation 4.20a, all terms in the exponent that contain the factor $N = 8$ can be deleted because $e^{-j2\pi l} = 1$ for any integer l. Upon deleting such terms and regrouping the remaining terms, the product nk can be expressed in either of two ways:

$$nk = (4k_0)n_2 + (4k_1 + 2k_0)n_1 + (4k_2 + 2k_1 + k_0)n_0 \tag{4.17a}$$

$$nk = (4n_0)k_2 + (4n_1 + 2n_0)k_1 + (4n_2 + 2n_1 + n_0)k_0 \tag{4.17b}$$

Substituting Equation 4.17a into Equation 4.15a leads to the D-I-T FFT, whereas substituting Equation 4.25b leads to the D-I-F FFT. Only the D-I-T FFT is discussed further here. The D-I-F and various related forms are treated in detail in Oppenheim and Schafer (1975).

The D-I-T FFT decomposes into $\log_2 N$ stages of computation, plus a stage of bit reversal,

$$x_1[k_0, n_1, n_0] = \sum_{n_2=0}^{n_2=1} s[n_2, n_1, n_0] W_8^{4k_0 n_2} \text{ (stage 1)} \tag{4.18a}$$

$$x_2[k_0, k_1, n_0] = \sum_{n_1=0}^{n_1=1} x_1[k_0, n_1, n_0] W_8^{4k_1 + 2k_0)n_1} \text{ (stage 2)} \tag{4.18b}$$

$$x_3[k_0, k_1, k_2] = \sum_{n_0=0}^{n_0=1} x_2[k_0, k_1, n_0] W_8^{4k_2 + 2k_1 + k_0)n_0} \text{ (stage 3)} \tag{4.18c}$$

$$s(k_2, k_1, k_0) = x_3(k_2, k_1, k_2) \text{ (bit reversal)} \tag{4.18d}$$

In each summation above, one of the n_is is summed out of the expression, while at the same time a new k_i is introduced.

The notation is chosen to reflect this. For example, in stage 3, n_0 is summed out, k_2 is introduced as a new variable, and n_0 is replaced by k_2 in the result. The last operation, called bit reversal, is necessary to

correctly locate the frequency samples $X[k]$ in the memory. It is easy to show that if the samples are paired correctly, an in-place computation can be done by a sequence of butterfly operations. The term in-place means that each time a butterfly is to be computed, a pair of data samples is read from memory, and the new data pair produced by the butterfly calculation is written back into the memory locations where the original pair was stored, thereby overwriting the original data. An in-place algorithm is designed so that each data pair is needed for only one butterfly, and so the new results can be immediately stored on top of the old in order to minimize memory requirements.

For example, in stage 3 the $k=6$ and $k=7$ samples should be paired, yielding a "butterfly" computation that requires one complex multiply, one complex add, and one subtract.

$$x_3(1,1,0) = x_2(1,1,0) + W_8^3 x_2(1,1,1) \tag{4.19a}$$

$$x_3(1,1,1) = x_2(1,1,0) - W_8^3 x_2(1,1,1) \tag{4.19b}$$

Samples $x_2(6)$ and $x_2(7)$ are read from the memory, the butterfly is executed on the pair, and $x_3(6)$ and $x_3(7)$ are written back to the memory, overwriting the original values of $x_2(6)$ and $x_2(7)$. In general, there are $N/2$ butterflies per stage and $\log_2 N$ stages, so the total number of butterflies is $[N/2]\log_2 N$. Since there is at most one complex multiplication per butterfly, the total number of multiplications is bounded by $(N/2)\log_2 N$ (some of the multiplies involve factors of unity and should not be counted).

Figure 4.10 shows the signal flow graph of the D-I-T FFT algorithm for $N=8$. This algorithm is referred to as an in-place FFT with normally ordered input and bit-reversed output samples. Minor variations that include bit-reversed inputs and normally ordered outputs, and non-in-place algorithms with normally ordered inputs and outputs are also possible. Also, when N is not a "power-of-2" (PO2), a mixed-radix algorithm can be used to reduce the computation. The mixed-radix FFT is most efficient when N is highly composite, i.e., when N can be expressed as $N = p_1^{r_1} p_2^{r_2} \cdots p_L^{r_L}$, where the p_is are small

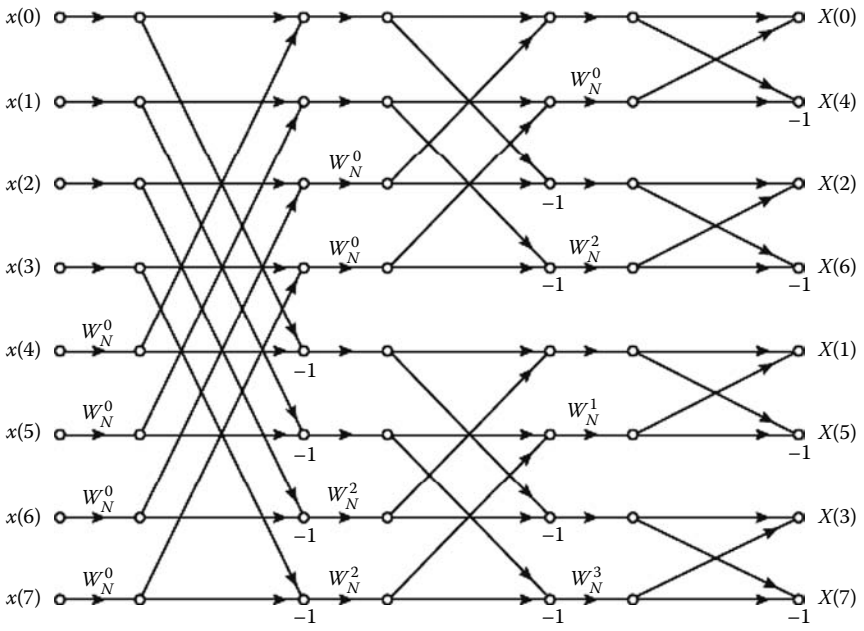


FIGURE 4.10 D-I-T FFT algorithm with normally ordered inputs and bit-reversed outputs.

prime numbers and the r^is are positive integers. It can be shown that the order of complexity of the mixed-radix FFT is Order $\{N(r_1(p_1 - 1) + r_2(p_2 - 1) + \cdots + r_L(p^L - 1)\}$. Because of the lack of uniformity of structure among stages, this algorithm has not received much attention for hardware implementation. However, the mixed-radix FFT is often used in software applications, especially for processing data recorded in laboratory experiments where it is not convenient to restrict the block lengths to be PO2. Many advanced FFT algorithms, such as higher-radix forms, the mixed-radix form, the prime-factor algorithm, and the Winograd algorithm, are described in Blahut (1985). Algorithms specialized for real-valued data reduce the computational cost by a factor-of-2.

4.6 Family Tree of Fourier Transforms

Figure 4.11 illustrates the functional relationships among the various forms of CT and DT Fourier transforms that have been discussed in the previous sections. The family of CT Fourier transforms is shown on the left side of Figure 4.11, whereas the right side of the figure shows the hierarchy of DT Fourier transforms. Note that the most general, and consequently the most powerful, Fourier transform is the classical complex Fourier transform (or equivalently, the bilateral Laplace transform). Note also that the complex Fourier transform is identical to the bilateral Laplace transform, and it is at this level that the classical Laplace transform techniques and Fourier transform techniques become identical. Each special member of the CT Fourier family is obtained by impressing certain constraints on the general

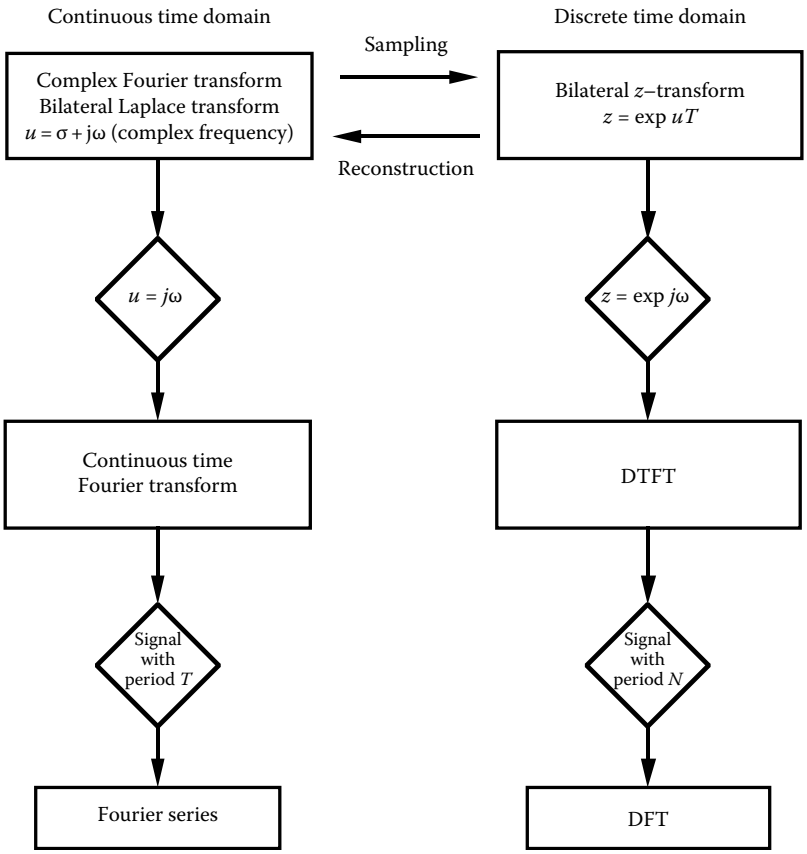


FIGURE 4.11 Functional relationships among various forms of the Fourier transform.

form, thereby producing special transforms that are simpler and more useful in practical problems where the constraints are met.

In Figure 4.11 it is seen that the bilateral z-transform is analogous to the complex Fourier transform, the unilateral z-transform is analogous to the classical (one-sided) Laplace transform, the DTFT is analogous to the classical Fourier (CT) transform, and the DFT is analogous to the classical (CT) Fourier series.

4.6.1 Walsh–Hadamard Transform

The Walsh–Hadamard transform (WHT) is a computationally attractive orthogonal transform that is structurally related to the DFT, and which can be implemented in practical applications without multiplication, and with a computational complexity for addition that is on the same order of complexity as that of an FFT. The t_{mk}th element of the WHT matrix $\mathbf{T}_{\text{WHT}}$ is given by

$$t_{mk} = \frac{1}{\sqrt{N}} \prod_{\ell=0}^{p-1} (-1)^{b_1(m) b_{p-1-\ell(k)}}, \quad m \text{ and } k = 0, \ldots, N-1$$

where $b_\ell(m)$ is the ℓth order bit in the binary representation of m, and $N = 2^p$. The WHT is defined only when N is a PO2. Note that the columns of $\mathbf{T}_{\text{WHT}}$ form a set of orthogonal basis vectors whose elements are all 1s or -1s, so that the calculation of the matrix–vector product $\mathbf{T}_{\text{WHT}}\, X$ can be accomplished with only additions and subtractions. It is well known that $\mathbf{T}_{\text{WHT}}$ of dimension $(N \times N)$, for N a PO2, can be computed recursively according to

$$\mathbf{T}_k = \begin{bmatrix} \mathbf{T}_{k/2} & \mathbf{T}_{k/2} \\ \mathbf{T}_{k/2} & -\mathbf{T}_{k/2} \end{bmatrix} \quad \text{for} \quad K = 4, \ldots, N \text{ (even) and } \mathbf{T}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The above relationship provides a convenient way of quickly constructing the Walsh–Hadamard matrix for any arbitrary (even) size N.

Due to structural similarities between the DFT and the WHT matrices, the WHT transform can be implemented using a modified FFT algorithm. The core of any FFT program is a butterfly calculation that is characterized by a pair of coupled equations that have the following form:

$$X_{i+1}(\ell, m) = X_i(\ell, m) + e^{j\theta(\ell,m,k,s)} X_i(k, s)$$
$$X_{i+1}(\ell, m) = X_i(\ell, m) - e^{j\theta(\ell,m,k,s)} X_i(k, s)$$

If the exponential factor in the butterfly calculation is replaced by a "1," so the "modified butterfly" calculation becomes

$$X_{i+1}(\ell, m) = X_i(\ell, m) + X_i(k, s)$$
$$X_{i+1}(\ell, m) = X_i(\ell, m) - X_i(k, s)$$

the modified FFT program will in fact perform a WHT on the input vector. This property not only provides a quick and convenient way to implement the WHT, but it also establishes clearly that in addition to the WHT requiring no multiplication, the number of additions required has order of complexity of $(N/2)\log_2 N$, i.e., the same as the that of the FFT.

The WHT is used in many applications that require signals to be decomposed in real time into a set of orthogonal components. A typical application in which the WHT has been used in this manner is in code division multiple access (CDMA) wireless communication systems. A CDMA system requires spreading

of each user's signal spectrum using a PN sequence. In addition to the PN spreading codes, a set of length-64 mutually orthogonal codes, called the Walsh codes, are used to ensure orthogonality among the signals for users received from the same base station. The length $N = 64$ Walsh codes can be thought of as the orthogonal column vectors from a (64×64) Walsh–Hadamard matrix, and the process of demodulation in the receiver can be interpreted as performing a WHT on the complex input signal containing all the modulated user's signals so they can be separated for accurate detection.

4.7 Selected Applications of Fourier Methods

4.7.1 DFT (FFT) Spectral Analysis

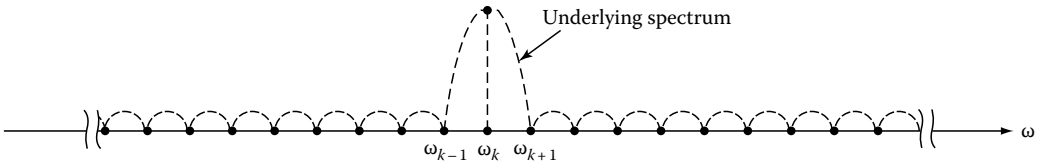
An FFT program is often used to perform spectral analysis on signals that are sampled and recorded as part of laboratory experiments, or in certain types of data acquisition systems. There are several issues to be addressed when spectral analysis is performed on (sampled) analog waveforms that are observed over a finite interval of time.

Windowing: The FFT treats the block of data as though it were one period of a periodic sequence. If the underlying waveform is not periodic, then harmonic distortion may occur because the periodic waveform created by the FFT may have sharp discontinuities at the boundaries of the blocks. This effect is minimized by removing the mean of the data (it can always be reinserted) and by windowing the data so the ends of the block are smoothly tapered to zero. A good rule of thumb is to taper 10% of the data on each end of the block using either a cosine taper or one of the other common windows (e.g., Hamming, Von Hann, Kaiser windows, etc.). An alternate interpretation of this phenomenon is that the finite length observation has already windowed the true waveform with a rectangular window that has large spectral sidelobes. Hence, applying an additional window results in a more desirable window that minimizes frequency-domain distortion.

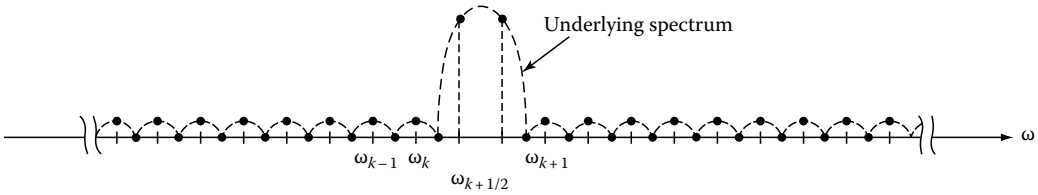
Zero-padding: An improved spectral analysis is achieved if the block length of the FFT is increased. This can be done by (1) taking more samples within the observation interval, (2) increasing the length of the observation interval, or (3) augmenting the original data set with zeros. First, it must be understood that the finite observation interval results in a fundamental limit on the spectral resolution, even before the signals are sampled. The CT rectangular window has a $(\sin x)/x$ spectrum, which is convolved with the true spectrum of the analog signal. Therefore, the frequency resolution is limited by the width of the mainlobe in the $(\sin x)/x$ spectrum, which is inversely proportional to the length of the observation interval. Sampling causes a certain degree of aliasing, although this effect can be minimized by using a sufficiently high sampling rate. Therefore, lengthening the observation interval increases the fundamental resolution limit, while taking more samples within the observation interval minimizes aliasing distortion and provides a better definition (more sample points) on the underlying spectrum.

Padding the data with zeros and computing a longer FFT does give more frequency-domain points (improved spectral resolution), but it does not improve the fundamental limit, nor does it alter the effects of aliasing error. The resolution limits are established by the observation interval and the sampling rate. No amount of zero padding can improve these basic limits. However, zero padding is a useful tool for providing more spectral definition, i.e., it enables one to get a better look at the (distorted) spectrum that results once the observation and sampling effects have occurred.

Leakage and the picket-fence effect: An FFT with block length N can accurately resolve only frequencies $w_k = (2\pi/N)k$, $k = 0, \ldots, N-1$ that are integer multiples of the fundamental $w_1 = (2\pi/N)$. An analog waveform that is sampled and subjected to spectral analysis may have frequency components between the harmonics. For example, a component at frequency $w_{k+1/2} = (2\pi/N)(k + 1/2)$ will appear scattered throughout the spectrum. The effect is illustrated in Figure 4.12 for a sinusoid that is observed through a rectangular window and then sampled a N points. The "picket-fence effect" means that not all



(a) FFT of a windowed sinusoid with frequency $\omega_k = 2\pi k/N$



(b) Leakage for a nonharmonic sinusoidal component

FIGURE 4.12 Illustration of leakage and the picket-fence effects.

frequencies can be seen by the FFT. Harmonic components are seen accurately, but other components "slip through the picket fence" while their energy is "leaked" into the harmonics. These effects produce artifacts in the spectral domain that must be carefully monitored to assure that an accurate spectrum is obtained from FFT processing.

4.7.2 FIR Digital Filter Design

A common method for designing FIR digital filters is by the use of windowing and FFT analysis. In general, window designs can be carried out with the aid of a hand calculator and a table of well-known window functions. Let $h[n]$ be the impulse response that corresponds to some desired frequency response, $H(e^{j\omega})$. If $H(e^{j\omega})$ has sharp discontinuities then $h[n]$ will represent an infinite impulse response (IIR) function. The objective is to time-limit $h[n]$ in such a way as to not distort $H(e^{j\omega})$ any more than necessary. If $h[n]$ is simply truncated, a ripple (Gibbs phenomenon) occurs around the discontinuities in the spectrum, resulting in a distorted filter, as was earlier illustrated in Figure 4.7.

Suppose that $w[n]$ is a window function that time-limits $h[n]$ to create an FIR approximation, $h'[n]$; i.e., $h'[n] = w[n]h[n]$. Then if $W(e^{j\omega})$ is the DTFT of $w[n]$, $h'[n]$ will have a Fourier transform given by $H'(e^{j\omega}) = W(e^{j\omega}) \star H(e^{j\omega})$, Where $\star$. denotes convolution. From this it can be seen that the ripples in $H(e^{j\omega})$ result from the sidelobes of $W(e^{j\omega})$. Ideally, $W(e^{j\omega})$ should be similar to an impulse so that $H'(e^{j\omega})$ is approximately equal to $H(e^{j\omega})$.

Special case: Let $h[n] = \cos n\omega_0$, for all n. Then $h[n] = w[n]\cos n\omega_0$, and

$$H'(e^{j\omega}) = (1/2)W(e^{j(\omega+\tilde{\omega})}) + (1/2)W(e^{j(\omega+\tilde{\omega})}) \tag{4.20}$$

as illustrated in Figure 4.13. For this simple class, the center frequency of the passband is controlled by ω_0, and both the shape of the passband and the sidelobe structure are strictly determined by the choice of the window. While this simple class of FIRs does not allow for very flexible designs, it is a simple technique for determining quite useful low-pass, bandpass, and high-pass FIR filters.

General case: Specify an ideal frequency response, $H(e^{j\omega})$, and choose samples at selected values of w. Use a long inverse FFT of length N' to find $h'[n]$, an approximation to $h[n]$, where if N is the desired length of the final filter, then $N' \gg N$. Then use a carefully selected window to truncate $h'[n]$ to obtain $h[n]$ by

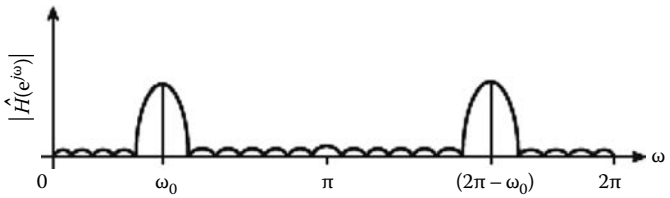


FIGURE 4.13 Design of a simple bandpass FIR filter by windowing.

letting $h[n] = w[n]h'[n]$. Finally, use an FFT of length N' to find $H'(e^{j\omega})$. If $H'(e^{j\omega})$ is a satisfactory approximation to $H(e^{j\omega})$, the design is finished. If not, choose a new $H(e^{j\omega})$, or a new $w[n]$ and repeat. Throughout the design procedure, it is important to choose $N' = kN$, with k an integer that is typically in the range $[4, \ldots, 10]$. Since this design technique is a trial-and-error procedure, the quality of the result depends to some degree on the skill and experience of the designer.

4.7.3 Fourier Block Processing in Real-Time Filtering Applications

In some practical applications, either the value of M is too large for the memory available, or $s[n]$ may not actually be finite in length, but rather a continual stream of data samples that must be processed by a filter at real time rates. Two well-known algorithms are available that partition $s[n]$ into smaller blocks and process the individual blocks with a smaller-length DFT (1) overlap-save partitioning and (2) overlap-add partitioning. Each of these algorithms is summarized below (Burrus 1985; Jenkins 2002).

1. *Overlap-save processing*: In this algorithm, N_{DFT} is chosen to be some convenient value with $N_{\text{DFT}} > N$. The signal $s[n]$, is partitioned into blocks which are of length N_{DFT} and which overlap by $N-1$ data points. Hence, the kth block is $s_k[n] = s[n + k(N_{\text{DFT}} - N + 1)]$, $n = 0, \ldots, N_{\text{DFT}} - 1$. The filter impulse response $h[n]$ is augmented with $N_{\text{DFT}} - N$ zeros to produce

$$h_{\text{pad}}[n] = \begin{cases} h[n], & n = 0, \ldots, N-1 \\ 0, & n = N, \ldots, N_{\text{DFT}} - 1 \end{cases} \tag{4.21}$$

   The DFT is then used to obtain $Y_{\text{pad}}[n] = \text{DFT}\{h_{\text{pad}}[n]\} \cdot \text{DFT}\{s_k[n]\}$, and $y_{\text{pad}}[n] = \text{IDFT}\{Y_{\text{pad}}[n]\}$. From the $y_{\text{pad}}[n]$ array the values that correctly correspond to the linear convolution are saved; values that are erroneous due to wrap-around error caused by the circular convolution of the DFT are discarded. The kth block of the filtered output is obtained by

$$y_k[n] = \begin{cases} y_{\text{pad}}[n], & n = 0, \ldots, N-1 \\ 0, & n = N, \ldots, N_{\text{DFT}} - 1 \end{cases} \tag{4.22}$$

   For the overlap-save algorithm, each time a block is processed there are $N_{\text{DFT}} - N + 1$ points saved and $N-1$ points discarded. Each block moves forward by $N_{\text{DFT}} - N + 1$ data points and overlaps the previous block by $N-1$ points.
2. *Overlap-add processing*: This algorithm is similar to the previous one except that the kth input block is defined to be

$$s_k[n] = \begin{cases} s[n], & n = 0, \ldots, L-1 \\ 0, & n = L, \ldots, N_{\text{DFT}} - 1 \end{cases} \tag{4.23}$$

   where $L = N_{\text{DFT}} - N + 1$. The filter function $h_{\text{pad}}[n]$ is augmented with zeros, as before, to create $h_{\text{pad}}[n]$, and the DFT processing is executed as before. In each block $y_{\text{pad}}[n]$ that is obtained at the

output, the first $N-1$ points are erroneous, the last $N-1$ points are erroneous, and the middle $N_{\mathrm{DFT}}-2(N-1)$ points correctly correspond to the linear convolution. However, if the last $N-1$ points from block k are overlapped with the first $N-1$ points from block $k+1$ and added pairwise, correct results corresponding to linear convolution are obtained from these positions, too. Hence, after this addition, the number of correct points produced per block is $N_{\mathrm{DFT}}-N+1$, which is the same as that for the overlap-save algorithm. The overlap-add algorithm requires approximately the same amount of computation as the overlap-save algorithm, although the addition of the overlapping portions of blocks is extra. This feature, together with the extra delay of waiting for the next block to be finished before the previous one is complete, has resulted in more popularity for the overlap-save algorithm in practical applications.

Block filtering algorithms make it possible to efficiently filter continual data streams in real time because the FFT algorithm can be used to implement the DFT, thereby minimizing the total computation time and permits reasonably high overall data rates. However, block filtering generates data in bursts, i.e., there is a delay during which no filtered data appears, and then suddenly an entire block is generated. In real-time systems, buffering must be used. The block algorithms are particularly effective for filtering very long sequences of data that are prerecorded on magnetic tape or disk.

4.7.4 Fourier Domain Adaptive Filtering

A transform domain adaptive filter (TDAF) is a generalization of the well-known least mean squares (LMS) adaptive filter in which the input signal is passed through a linear transformation in order to decompose it into a set of orthogonal components and to optimize the adaptive step size for each component and thereby maximize the learning rate of the adaptive filter (Jenkins et al. 1996). The LMS algorithm is an approximation to the steepest descent optimization strategy. For a length N FIR filter with the input expressed as a column vector $x(n)=[x(n), x(n-1),\ldots, x(n-N+1)]^{\mathrm{T}}$, the filter output $y(n)$ is expressed as

$$y(n)=\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n)$$

where $\mathbf{w}(n)=[w_0(n), w_1(n),\ldots, w_{N-1}(n)]^{\mathrm{T}}$ is the time varying vector of filter coefficients (tap weights), and the superscript "T" denotes vector transpose. The output error is formed as the difference between the filter output and a training signal $d(n)$, i.e., $e(n)=d(n)-y(n)$. Strategies for obtaining an appropriate $d(n)$ vary from one application to another. In many cases the availability of a suitable training signal determines whether an adaptive filtering solution will be successful in a particular application. The ideal cost function is defined by the MSE criterion, $E\{|e(n)|^2\}$. The LMS algorithm is derived by approximating the ideal cost function by the instantaneous squared error, resulting in $J_{\mathrm{LMS}}(n)=|e(n)|^2$. While the LMS seems to make a rather crude approximation at the very beginning, the approximation results in an unbiased estimator. In many applications the LMS algorithm is quite robust and is able to converge rapidly to a small neighborhood of the Wiener solution.

When a steepest descent optimization strategy is combined with a gradient approximation formed using the LMS cost function $J_{\mathrm{LMS}}(n)=|e(n)|^2$, the conventional LMS adaptive algorithm results:

$$\begin{aligned}\mathbf{w}(n+1)&=\mathbf{w}(n)+\mu e(n)\mathbf{x}(n)\\ e(n)&=d(n)-y(n)\\ y(n)&=\mathbf{x}(n)^{\mathrm{T}}\mathbf{w}(n)\end{aligned}\tag{4.24}$$

The convergence behavior of the LMS algorithm, as applied to a direct form FIR filter structure, is controlled by the autocorrelation matrix $\mathbf{R}_{\mathrm{x}}$ of the input process, where

$$\mathbf{R}_{\mathrm{x}}\equiv E[\mathbf{x}^{*}(n)\mathbf{x}^{\mathrm{T}}(n)]\tag{4.25}$$

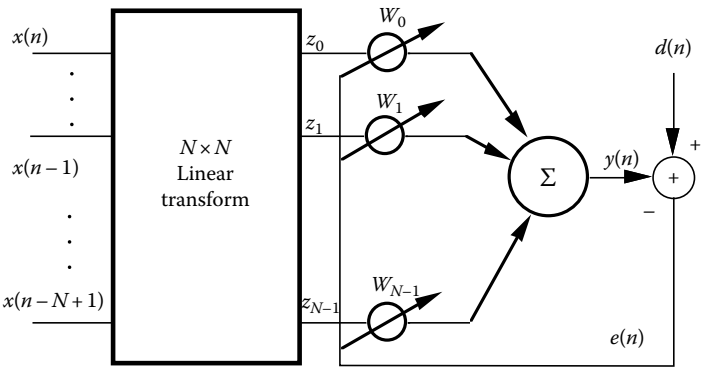


FIGURE 4.14 TDAF structure.

The autocorrelation matrix $\mathbf{R}_x$ is usually positive definite, which is one of the conditions necessary to guarantee convergence to the Wiener solution. Another necessary condition for convergence is $0 < m < 1/l_{max}$, where l_{max} is the largest eigenvalue of $\mathbf{R}_x$. It is well established that the convergence of this algorithm is directly related to the eigenvalue spread of $\mathbf{R}_x$. The eigenvalue spread is measured by the condition number of $\mathbf{R}_x$, defined as $k = l_{max}/l_{min}$, where l_{min} is the minimum eigenvalue of $\mathbf{R}_x$. Ideal conditioning occurs when $k = 1$ (white noise); as this ratio increases, slower convergence results. The eigenvalue spread (condition number) depends on the spectral distribution of the input signal, and is related to the maximum and minimum values of the input power spectrum. From this line of reasoning it becomes clear that white noise is the ideal input signal for rapidly training an LMS adaptive filter. The adaptive process is slower and requires more computation for input signals that are colored.

The TDAF structure is shown in Figure 4.14. The input $x(n)$ and the desired signal $d(n)$ are assumed to be zero mean and jointly stationary. The input to the filter is a vector of N current and past input samples, defined in the previous section and denoted as $\mathbf{x}(n)$. This vector is processed by a unitary transform, such as the DFT. Once the filter Order N is fixed, the transform is simply an $N \times N$ matrix T, which is in general complex, with orthonormal rows. The transformed outputs form a vector $\mathbf{v}(n)$ which is given by

$$\mathbf{v}(n) = [v_0(n), v_1(n), \ldots, v_{N-1}(n)]^T = \mathbf{T}\mathbf{x}(n)$$

With an adaptive tap vector defined as $w(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T$, the filter output is given by

$$y(n) = \mathbf{w}^T(n)\mathbf{v}(n) = \mathbf{W}^T(n)\mathbf{T}\mathbf{x}(n) \tag{4.26}$$

The instantaneous output error is then formed and used to update the adaptive filter taps using a modified form of the LMS algorithm (Jenkins 1996):

$$\begin{aligned} &\mathbf{W}(n+1) = \mathbf{W}(n) + \mu e(n)\boldsymbol{\Lambda}^{-2}\mathbf{v}^*(n) \\ &\boldsymbol{\Lambda}^2 \equiv \text{diag}\left[\sigma_1^2, \sigma_1^2, \ldots, \sigma_N^2\right] \end{aligned} \tag{4.27}$$

where

$$\sigma_i^2 = E\left[|v_i(n)|^2\right]$$

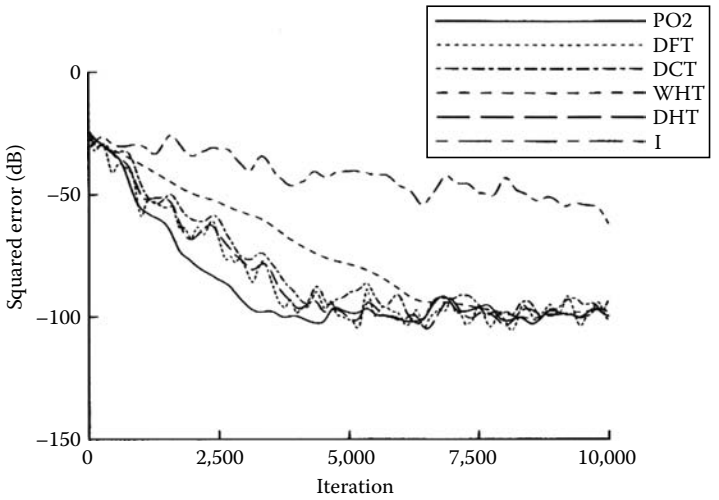
The power estimates σ_i^2 can be developed online by computing an exponentially weighted average of past samples according to

$$\sigma_i^2(n) = \alpha\sigma_i^2(n-1) + |v_i(n)|^2, \quad 0 < a < 1 \tag{4.28}$$

If σ_i^2 becomes too small due to an insufficient amount of energy in the *i*th channel, the update mechanism becomes ill-conditioned due to a very large effective step size. In some cases, the process will become unstable and register overflow will cause the adaptation to catastrophically fail. So the algorithm given by Equation 4.27 should have the update mechanism disabled for the *i*th orthogonal channel if σ_i^2 falls below a critical threshold.

The motivation for using the TDAF adaptive system instead of a simpler LMS-based system is to achieve rapid convergence of the filters coefficients when the input signal is not white, while maintaining a reasonably low computational complexity requirement. The optimal decorrelating transform is composed of the orthonormal eigenvectors of the input autocorrelation matrix, and is known as the Karhunen–Loe'ye transform (KLT). The KLT is signal dependent and usually cannot be easily computed in real time. Throughout the literature the DFT, the discrete cosine transform (DCT), and the WHT have received considerable attention as possible candidates for use in the TDAF.

Figure 4.15 shows learning characteristics for computer generated TDAF examples using six different orthogonal transforms to decorrelate the input signal. The examples presented are for system



Transform	Effective Input Correlation Matrix Eigenvalue Ratio
Identity (I)	681
DFT	210
DCT	200
WHT	216
DHT	218
PO2	128

FIGURE 4.15 Comparison of (smoothed) learning curves for five different transforms operating on a colored noise input signal with condition number 681 fault in any of the coefficients. When *R* redundant coefficients are added as many as *R* coefficients can fail to adjust without any adverse effect on the filter's ability to achieve the minimum MSE condition.

identification experiments, where the desired signal was derived by passing the input through an 8-tap FIR filter that is the "unknown system" to be identified. The filter input was generated by filtering white pseudonoise with a 32-tap linear phase FIR coloring filter to produce an input autocorrelation matrix with a condition number (eigenvalue ratio) of 681. Examples were then produced using the DFT, the DCT, the WHT, discrete Hartley transform (DHT), and a specially designed computationally efficient (PO2) transform. The condition numbers that result from transform processing with each of these transforms are also shown in Figure 4.15. Note that all of the transforms used in this example are able to reduce the input condition number and greatly improve convergence rates, although some transforms are seen to be more effective than others for the coloring chosen for these examples.

4.7.5 Adaptive Fault Tolerance via Fourier Domain Adaptive Filtering

Adaptive systems adjust their parameters to minimize a specified error criterion under normal operating conditions. Fixed errors or hardware faults would prevent the system to minimize the error criterion, but at the same time the system will adapt the parameters such that the best possible solution is reached. In adaptive fault tolerance (AFT), the inherent learning ability of the adaptive system is used to compensate for failure of the adaptive coefficients. This mechanism can be used with specially designed structures whose redundant coefficients have the ability to compensate for the adjustment failures of other coefficients (Jenkins et al. 1996).

The FFT-based transform domain fault tolerant adaptive filter (FTAF) is described by the following equations:

$$\begin{aligned} \mathbf{x}[n] &= [\mathrm{x}_{\mathrm{in}}[n],00\cdots 0] \\ \mathbf{x}_{\mathrm{T}}[n] &= \mathbf{T}\mathbf{x}[n] \\ y[n] &= \mathbf{w}_{\mathrm{T}}^{t}[n]\mathbf{x}_{\mathrm{T}}[n] \end{aligned} \tag{4.29}$$

where $\mathbf{x}_{\mathrm{in}}[n] = [x[n], x[n-1], \ldots, x[n-N+1]]$ is the vector of the current input and $N-1$ past inputs samples, $x[n]$ is $x_{\mathrm{in}}[n]$ zero-padded with R zeros, T is the $M \times M$ DFT matrix where $M = N+R$, $\mathbf{w}_T[n]$ is the vector of M adaptive coefficients in the transform domain, and $d[n]$ is the desired response. The FFT-based transform domain FTAF is similar to a standard TDAF except that the input data vector is zero-padded with R zeros before it is multiplied by the transform matrix. Since the input data vector is zero-padded, the transform domain FTAF maintains a length N impulse response and has R redundant coefficients in the transform domain. When used with the zero padding strategy described above, this structure possesses a property called *full fault tolerance*, where each redundant coefficient is sufficient to compensate for a single "stuck at" fault condition in any of the coefficients. When R redundant coefficients are added as many as R coefficients can fail without any adverse effect on the filter's ability to achieve the minimum MSE condition.

An example of a transform domain FTAF with one redundant filter tap ($R = 1$) is demonstrated below for the identification of a 64-tap FIR low-pass "unknown" system. The training signal is Gaussian white noise with a unit variance and a noise floor of -60 dB. A fixed fault is introduced at iteration 3000 by setting an arbitrary filter coefficient to a random fixed value. Simulated learning curves are shown in Figure 4.16 in which both demonstrated that the redundant tap allows the filter to reconverge after the occurrence of the fault, although the postfault convergence rate slowed somewhat due to an increased condition number of the postfault autocorrelation matrix (Jenkins 1996).

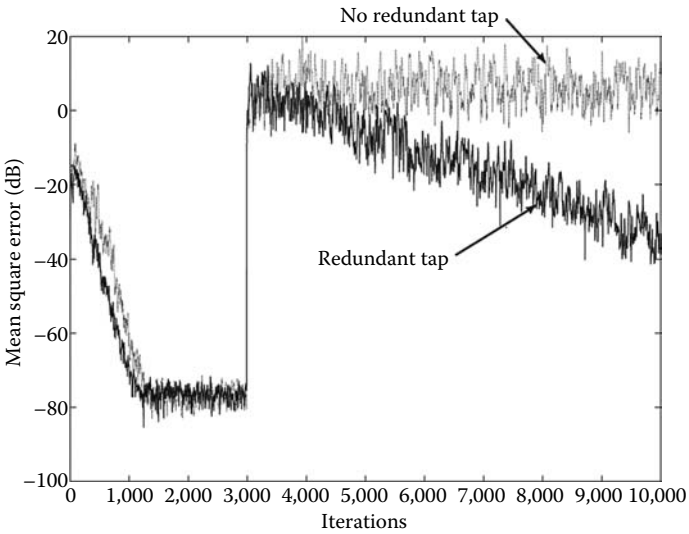


FIGURE 4.16 Learning curve demonstrating postfault behavior both with and without a redundant tap.

4.8 Summary

Numerous Fourier transform concepts have been presented for both CT and DT signals and systems. Emphasis was placed on illustrating how various forms of the Fourier transform relate to one another, and how they are all derived from more general complex transforms, the complex Fourier (or bilateral Laplace) transform for CT, and the bilateral z-transform for DT. It was shown that many of these transforms have similar properties that are inherited from their parent forms, and that there is a parallel hierarchy among Fourier transform concepts in the CT and DT domains. Both CT and DT sampling models were introduced as a means of representing sampled signals in these two different domains and it was shown that the models are equivalent by virtue of having the same Fourier spectra when transformed into the Fourier domain with the appropriate Fourier transform. It was shown how Fourier analysis properly characterizes the relationship between the spectra of a CT signal and its DT counterpart obtained by sampling, and the classical reconstruction formula was obtained as a result of this analysis. Finally, the DFT, the backbone for much of modern DSP, was obtained from more classical forms of the Fourier transform by simultaneously discretizing the time and frequency domains. The DFT, together with the remarkable computational efficiency provided by the FFT algorithm, has contributed to the resounding success that engineers and scientists have had in applying DSP to many practical scientific problems.

References

Blahut, R. E., *Fast Algorithms for Digital Signal Processing*, Reading, MA: Addison-Wesley Publishing Co., 1985.

Bracewell, R. N., *The Fourier Transform*, 2nd edn, New York, NY: McGraw-Hill, 1986.

Brigham, E. O., *The Fast Fourier Transform*, Englewood Cliffs, NJ: Prentice-Hall, 1974.

Burrus, C. S. and T. W. Parks, *DFT/FFT and Convolution Algorithms*, New York: John Wiley and Sons, 1985.

Jenkins, W. K., Discrete-time signal processing, *Reference Data for Engineers: Radio, Electronics, Computers, and Communications*, 9th edn (Chapter 28), Wendy M. Middleton, Editor-in-Chief, Newnes, Carmel, MA: Butterworth-Heinemann, 2002.

Jenkins, W. K. and M. D. Desai, The discrete-frequency Fourier transform, *IEEE Transactions on Circuits and Systems*, CAS-33, 7, 732–734, July 1986.

Jenkins, W. K. et al., *Advanced Concepts in Adaptive Signal Processing*, Boston MA: Kluwer Academic Publishers, 1996.

Oppenheim, A. V. and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.

Oppenheim, A. V. and R. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

Oppenheim, A. V., A. S. Willsky, and I. T. Young, *Signals and Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

VanValkenburg, M. E., *Network Analysis*, 3rd edn, Englewood Cliffs, NJ: Prentice-Hall, 1974.

5

z-Transform

5.1 Introduction **5**-1
5.2 Definition of the z-Transform **5**-2
5.3 Inverse z-Transform **5**-5
Contour Integration • Partial Fraction Expansion • Other Methods for Obtaining the Inverse z-Transform
5.4 Properties of the z-Transform **5**-8
Region of Convergence • Properties of the Transform • Convolution in z-Domain
5.5 Role of the z-Transform in Linear Time-Invariant Systems **5**-12
5.6 Variations on the z-Transform **5**-15
Multidimensional z-Transform • Modified z-Transform • Chirp z-Transform Algorithm
5.7 Concluding Remarks **5**-17
Acknowledgments **5**-17
References **5**-17

Jelena Kovačević
AT&T Bell Laboratories

5.1 Introduction

When analyzing linear systems, one of the problems we often encounter is that of solving linear, constant-coefficient differential equations. A tool used for solving such equations is the Laplace transform. At the same time, to aid the analysis of linear systems, we extensively use Fourier-domain methods. With the advent of digital computers, it has become increasingly necessary to deal with discrete-time signals, or, sequences. These signals can be either obtained by sampling a continuous-time signal, or they could be inherently discrete. To analyze linear discrete-time systems, one needs a discrete-time counterpart of the Laplace transform (LT). Such a counterpart is found in the z-transform, which similarly to the LT, can be used to solve linear constant-coefficient difference equations. In other words, instead of solving these equations directly, we transform them into a set of algebraic equations first, and then solve in this transformed domain. On the other hand, the z-transform can be seen as a generalization of the discrete-time Fourier transform (FT)

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} \tag{5.1}$$

This expression does not always converge, and thus, it is useful to have a representation which will exist for these nonconvergent instances. Furthermore, the use of the z-transform offers considerable notational

simplifications. It also allows us to use the extensive body of work on complex variables to aid in analyzing discrete-time systems.

The z-transform, as pointed out by Jury in his classical text [3], is not new. It can be traced back to the early eighteenth century and the times of DeMoivre, who introduced the notion of the generating function, extensively used in probability theory

$$\Gamma(z) = \sum_{n=-\infty}^{+\infty} p[n]z^n \tag{5.2}$$

where $p[n]$ is the probability that the discrete random variable **n** will take a value n [8]. By comparing Equations 5.2 and 5.3, we can see that the generating function $\Gamma(1/z)$ is the z-transform of the sequence $p[n] = p\{\mathbf{n} = n\}$. After these initial efforts, and due to the fast development of digital computers, a renewed interest in the z-transform occurred in the early 1950s, and z-transform has been used for analyzing discrete-time systems ever since.

This section is intended as a brief introduction to the theory and application of the z-transform. For a rigorous mathematical treatment of the transform itself, the reader is referred to the book by one of the pioneers in the development of analysis of sampled data systems, Jury [3], and the references therein. For a more succinct account of the z-transform, its properties and use in discrete-time systems, consult, for example, Ref. [7]. A few other texts which contain parts on the z-transform include Refs. [1,2,5,6,10].

5.2 Definition of the *z*-Transform

Suppose we are given a discrete-time sequence $x[n]$, either inherently discrete time, or obtained by sampling a continuous-time signal $x_c(t)$, so that $x[n] = x_c(nT)$, $n \in \mathbb{Z}$, where T is the sampling period. Then the two-sided z-transform of $x[n]$ is defined as

$$X(z) = \sum_{n=-\infty}^{+\infty} x[n]z^{-n} \tag{5.3}$$

Here, z is a complex variable, and depending on its magnitude and the sequence $x[n]$, the above sum may or may not converge. The region in the z-plane where the sum does converge is called the region of convergence (ROC), and is discussed in more detail later.

Observe that in Equation 5.3, n ranges from $-\infty$ to $+\infty$. That is why the z-transform defined in this way is called two-sided. One could define a one-sided z-transform, where n would range from 0 to $+\infty$. Obviously, the two definitions are equivalent only if the signal itself is one-sided, that is, if $x[n] = 0$, for $n < 0$. The advantage of using the one-sided z-transform is that it is useful in solving linear constant-coefficient difference equations with nonzero initial conditions and in the study of sampled-data feedback systems, discussed later. However, from now on, we deal mostly with the two-sided z-transform (see also Chapter 3 where the one-sided LT is used).

A power series given in Equation 5.3 is a Laurent series, and thus for it to converge uniformly, it has to be absolutely summable, that is, the following must hold:

$$\sum_{n=-\infty}^{n=+\infty} |x[n]|z|^{-n} < \infty \tag{5.4}$$

where $|z|$ is the magnitude of the complex variable z, i.e., $z = |z|\ e^{j\arg z}$. We can now see that if the z-transform converges for $z = z_1$, it will converge for all z such that $|z| = |z_1|$; that is, for all z on the circle $|z| = |z_1|$. In general, therefore, the ROC will be an annular region in the z-plane, and will be of the form

$$0 \le R_- < |z| < R_+ \le \infty \tag{5.5}$$

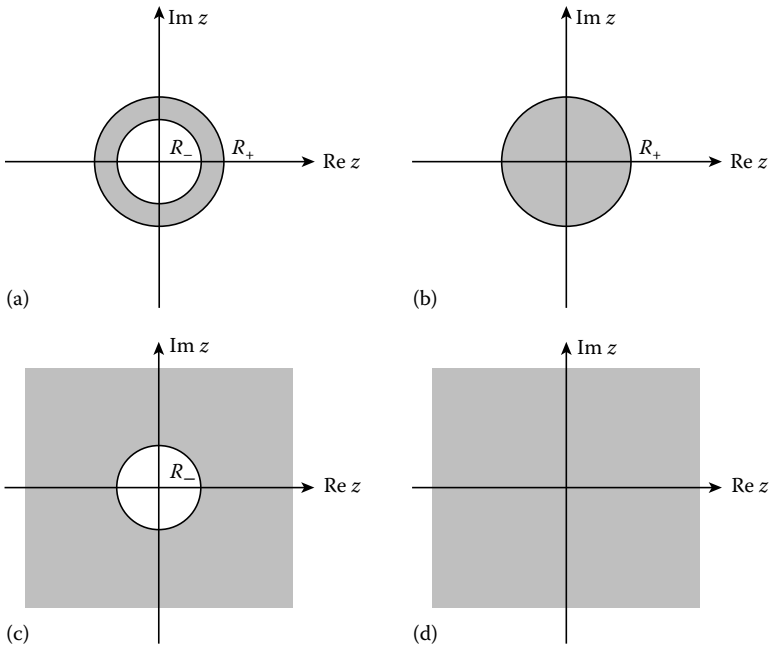


FIGURE 5.1 ROC of the z-transform. (a) General case: $0 \leq R_- < |z| < R_+ \leq \infty$. (b) ROC is the inside of the circle $|z| < R_+$ ($z = 0$ might be excluded). (c) ROC is the outside of the circle $R_- < |z|$ ($z = \infty$ might be excluded). (d) ROC is the whole z-plane (except possibly $z = 0$ or $z = \infty$).

Here, R_- could be zero, and R_+ could be infinity, so that all of the cases given in Figure 5.1 are possible. Note that the points $z = 0$, or $z = \infty$ could be included in the region of convergence.

A very important notion when talking about the ROC of the z-transform is that of the unit circle, because it makes the connection to the discrete-time FT. The unit circle is defined as the set of points in the z-plane for which $|z| = 1$. Let us evaluate the z-transform of a sequence on the unit circle. Because $|z| = 1$, it can be expressed as $z = e^{j\omega}$, where $\omega = \arg z$. Thus,

$$X(z)|_{|z|=1} = X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n} \tag{5.6}$$

Note that here we use $X(e^{j\omega})$ instead of $X(\omega)$ to make it explicit that this is a function of $e^{j\omega}$. As can be seen from the equation, the z-transform evaluated on the unit circle is equivalent to the discrete-time FT given in Equation 5.1. This immediately justifies the statement from the beginning of the section that the *z-transform* can be seen as a generalization of the FT. In other words, instances occur when the FT does not converge, while the z-transform does. One such instance is the unit-step function $u[n]$:

$$u[n] = \begin{cases} 1, & n \geq 0, \\ 0, & \text{otherwise} \end{cases}$$

The sum

$$U(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} u[n]e^{-j\omega n} = \sum_{n=0}^{+\infty} e^{-j\omega n}$$

is not absolutely summable and thus its FT does not converge, while

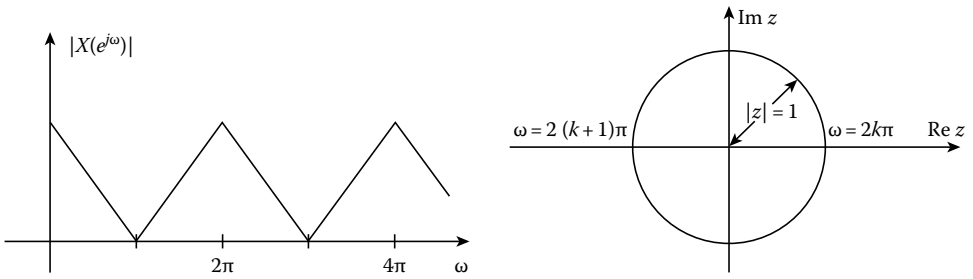


FIGURE 5.2 Warping of the linear frequency axis of the discrete-time FT into the unit circle of the z-transform. For example, the point (Re z, Im z) = (1, 0) on the unit circle corresponds to points $\omega = 2k\pi$ on the ω-axis.

$$U(z) = \sum_{n=-\infty}^{+\infty} u[n]z^{-n} = \sum_{n=0}^{+\infty} z^{-n}$$

converges for $|z| > 1$. As a consequence, if the ROC includes the unit circle, then the discrete-time FT of a given sequence will exist, otherwise it will not.

The unit circle captures the periodicity of the discrete-time FT. If we start evaluating the z-transform on the unit circle at the point (Re z, Im z) = (1, 0) corresponding to $\omega = 0$, going through (Re z, Im z) = (0, j), (Re z, Im z) = (−1, 0) which corresponds to $\omega = \pi$, and (Re z, Im z) = (0, $-j$), back to (Re z, Im z) = (1, 0) corresponding to $\omega = 2\pi$, we have evaluated one period of the FT and have returned to the same point. Thus, we are effectively warping the linear frequency axis into the unit circle (Figure 5.2).

We also mentioned that the z-transform is the discrete-time counterpart of the LT. Consider the function

$$x_{cs}(t) = \sum_{n=-\infty}^{+\infty} x_c(nT)\delta(t - nT) \tag{5.7}$$

or, the sampled version of the original continuous time function $x_c(t)$. Here, T is the sampling period, and $\delta(t)$ is the Dirac function. Taking the LT of $x_{cs}(t)$, we obtain

$$X_{cs}(s) = \sum_{n=-\infty}^{+\infty} x_c(nT)e^{-nTs} \tag{5.8}$$

If we now replace e^{sT} by z, we obtain the z-transform. Now, observe that $X_{cs}(s)$ in Equation 5.8 is periodic, because

$$\begin{aligned} X_{cs}\left(s + j\frac{2\pi}{T}\right) &= \sum_{n=-\infty}^{+\infty} x_c(nT)e^{-nT[s+j(2\pi)/T]} \\ &= \sum_{n=-\infty}^{+\infty} x_c(nT)e^{-nTs}e^{-j2\pi n} \\ &= X_{cs}(s) \end{aligned} \tag{5.9}$$

This means that $X_{cs}(s)$ is periodic along constant lines $\sigma = \sigma_{const}$ (parallel to the $j\omega$-axis). This further means that any line parallel to the $j\omega$-axis maps into a circle in the z-plane. It is easy to see that the $j\omega$-axis itself would map into the unit circle, while the left (or right) half-planes would map into the inside (or outside) of the unit circle, respectively.

Finally, let us say a few words about a very important class of signals, those whose z-transform is a rational function of z. They arise from systems that can be represented by linear constant-coefficient difference equations and are the signals with which we deal mostly in practice. If we represent such signals by

$$X(z) = \frac{N(z)}{D(z)} \tag{5.10}$$

then the zeroes of the numerator $N(z)$ are called zeroes of $X(z)$, while the zeroes of the denominator $D(z)$ are called poles of $X(z)$ (more precisely, a pole z_p will be a point at which $\lim_{z \to z_p} X(z)$ does not exist). How the poles can determine the region of convergence is the subject of a discussion later in this chapter.

5.3 Inverse *z*-Transform

We have seen that specifying the ROC when taking the z-transform of a sequence is an integral part of the process. For example, consider the following sequences: $x[n] = u[n]$ and $y[n] = -u[-n-1]$, where $u[n]$ is the unit-step function. Taking their z-transforms, we obtain

$$X(z) = \frac{1}{1 - z^{-1}}, \quad |z| > 1 \tag{5.11}$$

$$Y(z) = \frac{1}{1 - z^{-1}}, \quad |z| < 1 \tag{5.12}$$

They are identical except for their ROCs (Figure 5.3). This tells us that without the ROC, our z-transform is not complete, and that if we are given $1/(1 - z^{-1})$ as the z-transform of a sequence, the inverse is not uniquely specified unless we know the ROC.

We examine two ways of finding the inverse z-transform: first, and the formal one, by contour integration, and second, by partial fraction expansion. Note that the latter can be used only when $X(z)$ is a rational function of z; however, because most of the time we are dealing with rational functions of z, the partial fraction expansion method is used more often. We also mention two other informal techniques for determining the inverse z-transform.

5.3.1 Contour Integration

The formal inversion process for the z-transform is given by contour integration. It is obtained by taking the expression for the z-transform given by Equation 5.3 and multiplying both sides by $(1/2\pi j)z^{k-1}$. Then the result is integrated counterclockwise along a closed contour C in the z-plane containing the origin, leading to

$$\oint_C \frac{1}{2\pi j} X(z) z^{k-1} dz = \oint_C \frac{1}{2\pi j} z^{k-1} \sum_{n=-\infty}^{+\infty} x[n] z^{-n} dz \tag{5.13}$$

We choose the contour of integration to lie within the ROC, which will allow us to interchange the order of integration and summation on the right side of Equation 5.13. This leads to

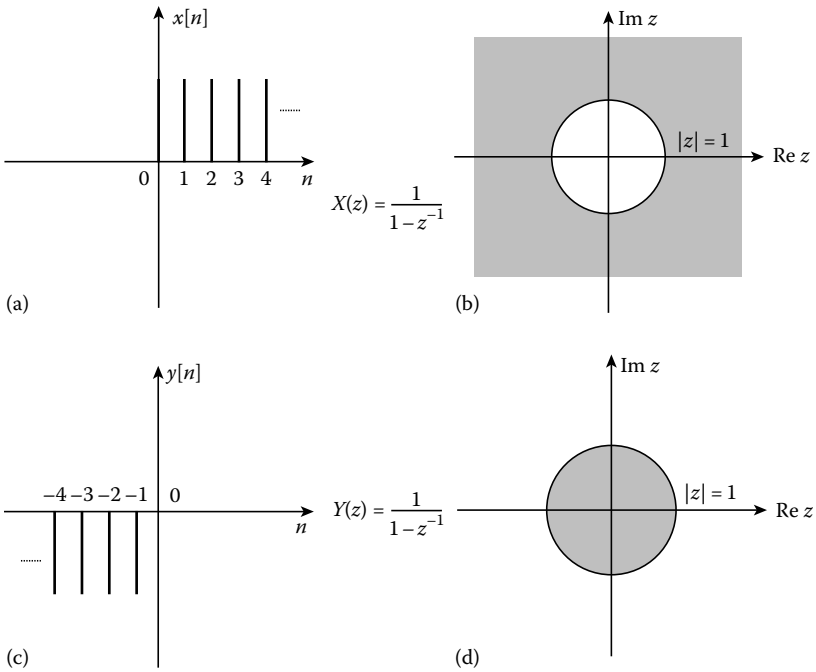


FIGURE 5.3 Two different sequences giving rise to the same z-transforms $1/(1-z^{-1})$ except for their ROCs. (a) Sequence $x[n]=u[n]$ and (b) its ROC $|z|>1$. (c) Sequence $y[n]=-u[-n-1]$ and (d) its ROC $|z|<1$.

$$\frac{1}{2\pi j}\oint_C X(z)z^{k-1}\mathrm{d}z=\sum_{n=-\infty}^{+\infty}x[n]\left(\frac{1}{2\pi j}\oint_C z^{-n+k-1}\mathrm{d}z\right) \tag{5.14}$$

The integral in parentheses on the right side of Equation 5.14 can be evaluated using Cauchy's integral formula, which states that if the contour of integration C contains the origin, and integration is performed counterclockwise, then

$$\frac{1}{2\pi j}\oint_C z^k\mathrm{d}z=\begin{cases}1, & k=-1,\\ 0, & \text{otherwise}\end{cases} \tag{5.15}$$

Substituting Equation 5.15 in Equation 5.14, we see that the integral is nonzero only for $n=k$, and thus Equation 5.14 can be rewritten as

$$\frac{1}{2\pi j}\oint_C X(z)z^{k-1}\mathrm{d}z=x[k] \tag{5.16}$$

Equation 5.16 is the inversion formula for the z-transform. To evaluate it for general functions can be quite difficult. However, in cases in which $X(z)$ is a rational function of z, we can make use of Cauchy's formula. It tells us that

$$\frac{1}{2\pi j}\oint_C \frac{F(z)}{z-z_\mathrm{p}}\mathrm{d}z=\begin{cases}F(z_\mathrm{p}), & z_\mathrm{p}\text{ inside }C,\\ 0, & \text{otherwise}\end{cases} \tag{5.17}$$

where C is a simple closed path and $F'(z)$, the derivative of $F(z)$, exists on and inside C. If we have a pole of multiplicity r enclosed in the contour C, and $F(z)$ and its $(r+1)$ derivatives exist in and on C, then

$$\frac{1}{2\pi j}\oint_C \frac{F(z)}{(z-z_p)^r}\mathrm{d}z = \begin{cases} \frac{1}{(r-1)!}\frac{\mathrm{d}^{r-1}}{\mathrm{d}z^{r-1}}F(z)|_{z=z_p}, & z_p \text{ inside } C, \\ 0, & \text{otherwise} \end{cases} \tag{5.18}$$

Equations 5.17 and 5.18 are called *residues*. Use Equations 5.17 and 5.18, and if we express

$$X(z)z^{k-1} = \frac{F(z)}{(z-z_p)^r} \tag{5.19}$$

where $F(z)$ has no poles at $z = z_p$, then Cauchy's residue theorem says that

$$x[k] = \frac{1}{2\pi j}\oint_C X(z)z^{k-1}\mathrm{d}z = \sum_i R_i \tag{5.20}$$

where R_i are residues of $X(z)z^{k-1}$ at the poles inside the contour C. The poles outside the contour do not contribute to the sum. If no poles are inside the contour of integration for a certain k, then $x[k]$ is zero for that k. Do not ignore the fact that the contour of integration C must lie within the ROC.

In some instances, it may be quite cumbersome to evaluate Equation 5.20, for example, when we have a multiple-order pole at $z = 0$, whose order depends on k. In that case we can rewrite Equation 5.20 as

$$x[k] = \sum_i R_i'$$

where

R_i' is the residue of $X(1/z)z{-}k{-}1$ at the poles inside the contour C'
C' is a circle of radius $1/s$ if C is a circle of radius s

For more details, see Ref. [7].

5.3.2 Partial Fraction Expansion

Another method of obtaining the inverse z-transform is by using partial fraction expansion. Note, however, that the partial fraction expansion method can be applied only to rational functions. Thus, suppose that $X(z)$ can be represented as in Equation 5.10. We can then rewrite it as

$$X(z) = \frac{N_0}{D_0}\cdot\frac{\prod_{i=1}^{N}(1-n_i z^{-1})}{\prod_{i=1}^{D}(1-d_i z^{-1})} \tag{5.21}$$

where

n_i are the nontrivial zeroes of $N(z)$
d_i are the zeroes of $D(z)$

The partial fraction expansion of $X(z)$ can be written as

$$X(z) = \sum_{i=0}^{N-D} A_i z^{-i} + \sum_{i=1}^{D_s}\frac{B_i}{(1-d_i z^{-1})} + \sum_{i=1}^{D_m}\sum_{m=1}^{p_i}\frac{C_{mi}}{(1-d_i z^{-1})^m} \tag{5.22}$$

Here, if $N \geq D$, A_i can be obtained by long division of the numerator by the denominator; otherwise, that first sum in Equation 5.22 disappears. In the second sum D_s denotes the number of single poles d_i of $X(z)$, and the coefficients B_i can be obtained as

$$B_i = \left(1 - d_i z^{-1}\right) X(z)\big|_{z=d_i} \tag{5.23}$$

The third sum (double sum) represents the part with multiple poles. D_m is the number of multiple poles d_i, and p_i are their respective multiplicities. The coefficients C_{mi} can be obtained as

$$C_{mi} = \frac{1}{(p_i - m)!(-d_i)^{pi-m}} \frac{\mathrm{d}^{pi-m}}{\mathrm{d}z^{pi-m}} [(1 - d_i z)^{pi} X(z^{-1})]\big|_{z=d_i^{-1}} \tag{5.24}$$

Once we have the expression (Equation 5.22), we can recognize each term as the z-transform of a known sequence. For example, $B_i/(1 - d_i z^{-1})$ will be the z-transform of either $B_i d_i^n u[n]$ or $-B_i d_i^n u[-n-1]$, depending on whether $|z| > |d_i|$, or $|z| < |d_i|$.

5.3.3 Other Methods for Obtaining the Inverse z-Transform

Although the two methods presented previously will work in most cases, sometimes it can be more convenient to use simpler techniques.

One of these is the inspection method [7]. It consists of learning to recognize some often-used z-transform pairs. For example, if we are given $3/(1 - z^{-1})$ with the ROC $|z| < 1$, from Equation 5.21 we can recognize it as the z-transform of $-3u[-n-1]$. In this process the tables of z-transform pairs are an invaluable tool. An extensive list of z-transform pairs can be found in Ref. [3].

Another technique can be used if we are given a z-transform in the form of a power series expansion:

$$X(z) = \cdots + x[-1]z + x[0] + x[1]z^{-1} + \cdots$$

Then, we can identify each term with the appropriate power of z. For example, the coefficient with z^{-k} will be $x[k]$.

5.4 Properties of the z-Transform

Although we can always obtain a z-transform of a sequence by directly applying its definition as given in Equation 5.3, it is useful to have a list of properties at hand to help calculate a particular z-transform or inverse z-transform more easily. We divide these properties into two categories: properties of the ROC, and properties of the z-transform itself. In what follows R_x will denote the ROC of the signal $x[n]$, while R_{x_-} and R_{x_+} will denote its lower/upper bounds, respectively (as given in Equation 5.5).

5.4.1 Region of Convergence

The ROC is an integral part of the z-transform of a sequence. Thus, this section goes into more detail on some of the points touched upon earlier. These properties and the order in which they are presented follow those in Ref. [7]; therefore, for more details, see Ref. [7].

First, we said that the ROC is an annular region in the z-plane, i.e., $0 \leq R_- < |z| < R_+ \leq \infty$. This follows from the fact that if the z-transform converges for $z = z_1$, it will converge for all z such that $|z| = |z_1|$, that is, for all z on the circle $|z| = |z_1|$. Then, if we put $|z| = 1$ in Equation 5.3, we obtain Equation 5.6, or the discrete-time FT. Therefore, it is obvious that the FT of $x[n]$ converges iff the z-transform of $x[n]$ converges for $|z| = 1$, that is, iff the ROC of the z-transform contains the unit circle.

Another useful property is that the ROC cannot contain any poles. This stems from the fact that if it did, the z-transform at the pole would be infinite and would not converge.

Consider now what happens if the sequence is of finite duration—it is zero except in a finite interval $-\infty < N_1 \leq n \leq N_2 < +\infty$. If all the values are finite, then the sequence is clearly absolutely summable and the z-transform will converge everywhere, except possibly at points $z = 0$ or $z = \infty$. Using the same type of arguments one can conclude that if the sequence is right-sided (it is zero for $n < N_1 < +\infty$), then the ROC will be the annular region outside of the finite pole of $X(z)$ of the largest magnitude. Similarly, if the sequence is left-sided (it is zero for $n > N_2 > -\infty$), then the ROC will be the annular region inside the finite pole of $X(z)$ of the smallest magnitude. As a result, if a sequence is neither left- nor right-sided, the ROC will be an annular region bounded on the interior and the exterior by a pole.

5.4.2 Properties of the Transform

The sequences $x[n]$, $y[n], \ldots$ will have associated z-transforms $X(z)$, $Y(z), \ldots$, with ROCs R_x, $R_y, \ldots$, in which each ROC will have its associated lower and upper bounds, as given in Equation 5.5.

5.4.2.1 Linearity

$$ax[n] + by[n] \leftrightarrow aX(z) + bY(z) \quad \text{ROC} \supset R_x \cap R_y \tag{5.25}$$

To prove this, apply the definition given in Equation 5.5. Note that the resulting ROC is at least as large as the intersection of the two starting ROCs. For example, if both $X(z)$ and $Y(z)$ are rational functions of z, and by adding $aX(z)$ to $bY(z)$ we introduce a zero that cancels one of the poles, the resulting ROC is larger than the intersection. If, on the other hand, no pole/zero cancelation exists, the resulting ROC is exactly the intersection.

5.4.2.2 Shift in Time

$$x[n - i] \leftrightarrow z^{-i}X(z) \quad \text{ROC} = R_x \tag{5.26}$$

The proof is straightforward and follows by the change of variables $k = n - i$ in Equation 5.5. Note that the resulting ROC could gain or lose a few poles at $z = 0$ or $z = \infty$.

5.4.2.3 Time Reversal

$$x[-n] \leftrightarrow X\left(\frac{1}{z}\right) \quad \frac{1}{R_{x_+}} < |z| < \frac{1}{R_{x_-}} \tag{5.27}$$

Again, the proof is straightforward and follows by the change of variables $k = -n$ in Equation 5.5.

5.4.2.4 Multiplication by an Exponential Sequence

One can easily show that the following holds:

$$a^n x[n] \leftrightarrow X\left(\frac{z}{a}\right) \quad |a|R_{x_-} < |z| < |a|R_{x_+} \tag{5.28}$$

Because z/a is the variable in the transform domain, we can see that all the poles of the original $X(z)$ have been scaled by a.

5.4.2.5 Multiplication by a Ramp

This property could also be called "differentiation of $X(z)$":

$$nx[n] \leftrightarrow -z\frac{dX(z)}{dz}, \quad \text{ROC} = R_x \tag{5.29}$$

To demonstrate this, differentiate both sides of Equation 5.3 with respect to z and then multiply by $-z$ to obtain Equation 5.29.

5.4.2.6 Convolution in Time

In transform domain convolution becomes simply the product of the two sequences. If we denote convolution by *

$$x[n]^*y[n] = \sum_{k=-\infty}^{+\infty} x[k]y[n-k] = \sum_{k=-\infty}^{+\infty} y[k]x[n-k] \tag{5.30}$$

then

$$x[n]^*y[n] \leftrightarrow X(z)\cdot Y(z), \quad \text{ROC} \supset R_x \cap R_y \tag{5.31}$$

Although the proof is not difficult, we write it here, because this is one of the most useful properties. Thus, take the convolution in Equation 5.30, multiply it by z^{-n} and sum over n

$$\begin{aligned}\sum_{n=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} x[k]y[n-k]z^{-n} &= \sum_{k=-\infty}^{+\infty} x[k]\sum_{n=-\infty}^{+\infty} y[n-k]z^{-n} \\ &= \sum_{k=-\infty}^{+\infty} x[k]\sum_{p=-\infty}^{+\infty} y[p]z^{-(p+k)} \\ &= \sum_{k=-\infty}^{+\infty} x[k]z^{-k}\sum_{p=-\infty}^{+\infty} y[p]z^{-p} \\ &= X(z)Y(z)\end{aligned}$$

where we have used change of variables $p = n - k$. As in the case of linearity, if it happens that a pole residing at the border of one of the ROCs is canceled by a zero from the other transform, then the resulting ROC will be larger than the intersection of the individual ROCs; otherwise, it will be exactly their intersection.

5.4.3 Convolution in z-Domain

Convolution in z-domain is given by (for more details, refer to Ref. [3] or [7])

$$x[n]\cdot y[n] \leftrightarrow \frac{1}{2\pi j}\oint_C X(\lambda)Y\left(\frac{z}{\lambda}\right)\lambda^{-1}d\lambda \quad R_{x_-}R_{y_-} < |z| < R_{x_+}R_{y_+} \tag{5.32}$$

where C is a closed contour in the intersection of the ROCs of $X(\lambda)$ and $Y(z/\lambda)$, and integration is performed counterclockwise. This property is the generalization of the periodic convolution property of the FT. Suppose that the contour C is the unit circle and $\lambda = e^{j\omega}$, $z = e^{j\theta}$. Also, observe that $d\lambda = je^{j\omega}\,d\omega$, and that if λ goes around the unit circle, ω ranges from $-\pi$ to π. Then,

$$\frac{1}{2\pi j}\oint_C X(\lambda)Y\left(\frac{z}{\lambda}\right)\lambda^{-1}d\lambda = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})Y[e^{j(\theta-\omega)}]d\omega$$

This equation states that the product of two sequences has as its FT the periodic convolution of their FTs.

5.4.3.1 Conjugation

If we are given a complex sequence $x^*[n]$, then its z-transform pair is

$$x^*[n] \leftrightarrow X^*(z^*) \quad \text{ROC} = R_x \tag{5.33}$$

5.4.3.2 Real Part

This property and the next one use the conjugation property given in Equation 5.33. Thus, if we express

$$\text{Re}\{x[n]\} = \frac{x[n] + x^*[n]}{2}$$

then by Equation 5.33 and the linearity of the z-transform

$$\text{Re}\{x[n]\} \leftrightarrow \frac{1}{2}[X(z) + X^*(z^*)] \quad \text{ROC} \supset R_x \tag{5.34}$$

5.4.3.3 Imaginary Part

Similarly to the previous property, if we express

$$\text{Im}\{x[n]\} = \frac{x[n] - x^*[n]}{2j}$$

then by Equation 5.33 and the linearity of the z-transform

$$\text{Im}\{x[n]\} \leftrightarrow \frac{1}{2j}[X(z) - X^*(z^*)] \quad \text{ROC} \supset R_x \tag{5.35}$$

5.4.3.4 Parseval's Relation

Parseval's relation is widely used in various transform domains, usually to find the energy of a signal. We start here with its more general formulation, and then reduce it to its usual form

$$\sum_{n=-\infty}^{+\infty} x[n]y^*[n] = \frac{1}{2\pi j}\oint_C X(\lambda)Y^*\left(\frac{1}{\lambda^*}\right)\lambda^{-1}d\lambda \tag{5.36}$$

where C is a closed contour in the intersection of the ROCs of $X(\lambda)$ and $Y^*(1/\lambda^*)$, and integration is performed counterclockwise. Again, for the proof, we refer the reader to Ref. [3] or [7].

Suppose now that $y[n] = x[n]$. Then Equation 5.36 reduces to

$$\sum_{n=-\infty}^{+\infty} |x[n]|^2 = \frac{1}{2\pi j}\oint_C X(\lambda)X^*\left(\frac{1}{\lambda^*}\right)\lambda^{-1}d\lambda \tag{5.37}$$

If both $x[n]$ and $y[n]$ converge on the unit circle, i.e., their FTs exist, then, we can choose $\lambda = e^{j\omega}$ and Equation 5.36 becomes

$$\sum_{n=-\infty}^{+\infty} x[n]y^{\star}[n] = \frac{1}{2\pi j}\oint_{C} X(e^{j\omega})Y^{\star}(e^{j\omega})e^{-j\omega}\mathrm{d}(e^{j\omega})$$
$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})Y^{\star}(e^{j\omega})\mathrm{d}\omega$$

which is the usual Parseval's relation in Fourier domain.

5.4.3.5 Initial Value Theorem

If $x[n]$ is causal (it is 0 for $n < 0$), then

$$x[0] = \lim_{z\to\infty} X(z) \tag{5.38}$$

5.4.3.6 Final Value Theorem

If the poles of $X(z)$ are all inside the unit circle, then

$$\lim_{n\to\infty} x[n] = \lim_{z\to 1}[(1 - z^{-1})X(z)] \tag{5.39}$$

5.5 Role of the *z*-Transform in Linear Time-Invariant Systems

The class of systems dealt with mostly are linear time-invariant systems. We discuss here the role of the z-transform in such systems. Recall that if we are given a system with input $x[n]$ and output $y[n]$, described by an operator $H[\cdot]$

$$y[n] = H[x[n]]$$

and $H[\cdot]$ is defined to be linear and time invariant, then

1. If inputs $x_1[n]$ and $x_2[n]$ produce outputs $y_1[n]$ and $y_2[n]$, then the input $ax_1[n]+bx_2[n]$ will produce the output $ay_1[n] + by_2[n]$.
2. If the input $x[n]$ produces output $y[n]$, then the input $x[n-k]$ will produce the output $y[n-k]$.

We recall a few properties of discrete linear time-invariant systems. For more details, refer to Refs. [4,7].

Note first that $x[n]$ can be written as a superposition of unit samples as

$$x[n] = \sum_{k=-\infty}^{+\infty} x[k]\delta[n-k] \tag{5.40}$$

If this is the input and we want to find the corresponding output, then

$$\begin{aligned} y[n] &= H[x[n]] \\ &= H\left[\sum_{k=-\infty}^{+\infty} x[k]\delta[n-k]\right] \\ &= \sum_{k=-\infty}^{+\infty} x[k]H[\delta[n-k]] \\ &= \sum_{k=-\infty}^{+\infty} x[k]h[n-k] \end{aligned} \tag{5.41}$$

Here, $h[n]$ is called the unit-sample response, defined as the response of the system to the unit sample $\delta[n]$. Equation 5.41 is a convolution sum, thus it can be written as

$$y[n] = x[n]^* h[n] \tag{5.42}$$

One of the most important properties of the z-transform is that the convolution in time has a product as its transform-domain pair. Therefore, using Equation 5.31, we can write Equation 5.42 in z-domain as

$$Y(z) = X(z)H(z) \tag{5.43}$$

Here, $H(z)$ is called the system transfer function

$$H(z) = \sum_{n=-\infty}^{+\infty} h[n]z^{-n} \tag{5.44}$$

We may also obtain this transfer function in another manner, if we assume that the systems we are dealing with are those that can be described by linear constant-coefficient difference equations (much in the same way as continuous-time systems that can be represented by linear constant-coefficient differential equations). Along the way, the use of z-transform greatly simplifies analysis of such systems and the solutions to these equations reduce to solutions of algebraic equations. Hence, suppose that our system can be described by the following linear constant-coefficient difference equation:

$$\sum_{k=0}^{D} b_k y[n-k] = \sum_{k=0}^{N} a_k x[n-k] \tag{5.45}$$

This equation has many solutions. We assume that the system is causal and moreover, we assume that the initial conditions are satisfied so that the system is linear and time invariant. With these factors in mind, we can write the expression for the output of the system as

$$y[n] = -\sum_{k=1}^{D} \frac{b_k}{b_0} y[n-k] + \sum_{k=0}^{N} \frac{a_k}{b_0} x[n-k] \tag{5.46}$$

This means that the output at time n depends on the outputs from D previous instants as well as from the input at time n and at N previous instants. Taking the z-transform of both sides of Equation 5.45 and using the linearity and shift in time properties of the z-transform, we obtain

$$Y(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{\sum_{k=0}^{D} b_k z^{-k} X(z)} \tag{5.47}$$

Finally, using Equation 5.43 we can identify $H(z)$ here as

$$H(z) = \frac{\sum_{k=0}^{N} a_k z^{-k}}{\sum_{k=0}^{D} b_k z^{-k}} \tag{5.48}$$

The function $H(z)$ is often referred to as a filter. If the denominator is a delay, i.e., $b_i\, z^{-i}$, such a filter will be a finite impulse response filter (FIR), as opposed to the infinite impulse response filter (IIR). Taking the inverse z-transform of Equation 5.48 we may obtain the unit-sample response of the system $h[n]$.

Finally, let us see how one might obtain the frequency response of the system. Evaluate Equation 5.44 on the unit circle (assuming that the ROC of $H(z)$ contains the unit circle)

$$H(z)|_{|z|=1} = H(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} h[n]e^{-j\omega n} \tag{5.49}$$

Therefore, the frequency response of the system is the system transfer function evaluated on the unit circle:

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}} \tag{5.50}$$

A consequence of this is that a linear time-invariant system is bounded-input–bounded-output stable (i.e., bounded input produces bounded output), iff the ROC of the transfer function $H(z)$ contains the unit circle.

We mentioned earlier that we are using the double-sided z-transform, while the one-sided z-transform is most useful when solving linear constant-coefficient difference equations with nonzero initial conditions. To transform such an equation into an algebraic equation, we used the linearity and shift in time properties of the double-sided z-transform. We would like to be able to do the same with the one-sided z-transform; however, we must re-derive the shift in time property in order to do that. Therefore, assume we are given a sequence $x[n-i]$ as in Equation 5.26.

Taking its one-sided z-transform, we obtain

$$\begin{aligned}\sum_{n=0}^{+\infty} x[n-i]z^{-n} &= \sum_{p=-i}^{+\infty} x[p]z^{-(p+i)} \\ &= \sum_{p=-i}^{-1} x[p]z^{-(p+i)} + z^{-i}\sum_{p=0}^{+\infty} x[p]z^{-p} \\ &= x[-i] + \cdots + x[-1]z^{-(i-1)} + z^{-i}X(z)\end{aligned} \tag{5.51}$$

To solve Equation 5.45, we can take the one-sided z-transform of both sides and use the linearity and shift in time properties to obtain

$$\sum_{k=0}^{D} b_k \left[\sum_{p=-k}^{-1} y[p]z^{-(p+k)} + z^{-k}Y(z)\right] = \sum_{k=0}^{N} a_k \left[\sum_{p=-k}^{-1} x[p]z^{-(p+k)} + z^{-k}X(z)\right] \tag{5.52}$$

The output $y[n]$ for $n \geq 0$ can be obtained as the inverse z-transform of

$$Y(z) = \left(\frac{\sum_{k=0}^{N} a_k z^{-k}}{\sum_{k=0}^{D} b_k z^{-k}}\right)X(z) + \left[\frac{\sum_{k=0}^{N} a_k \sum_{p=-k}^{-1} x[p]z^{-(p+k)} - \sum_{k=0}^{D} b_k \sum_{p=-k}^{-1} y[p]z^{-(p+k)}}{\sum_{k=0}^{D} b_k z^{-k}}\right] \tag{5.53}$$

To solve for $Y(z)$ we need to know initial conditions $y[n]$, for $n = -D, \ldots, -1$, and for $x[n]$ for $n = -N, \ldots, -1$. Note that if $x[n]$ is causal, and the initial conditions are all zero, the above solution is the same as the one when the double-sided z-transform is used; i.e., it reduces to the first term on the right side of Equation 5.53.

5.6 Variations on the *z*-Transform

5.6.1 Multidimensional *z*-Transform

Although the two-dimensional and multidimensional z-transforms are related to the one-dimensional z-transform, they are not straightforward generalizations of it. We give here a brief overview of the two-dimensional z-transform. For more details on it and the multidimensional z-transform, see Refs. [1,2].

The two-dimensional z-transform of a sequence $x[n_1, n_2]$ is given by

$$X(z_1, z_2) = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} x[n_1, n_2] z_1^{-n_1} z_2^{-n_2} \tag{5.54}$$

Here, z_1 and z_2 are complex variables and the region in the four-dimensional (z_1, z_2) space in which the previous double sum converges is called the ROC. In one dimension, the ROC is an annular region in the z-plane, while here it is called the Reinhardt domain. The analog of the unit circle in one dimension is the unit bicircle for $|z_1| = 1$ and $|z_2| = 1$. On the unit bicircle, the two-dimensional z-transform becomes the two-dimensional discrete-time FT.

Properties of the two-dimensional z-transform are similar to those of the one-dimensional z-transform, and they can be found in Refs. [1,2]. Here, we list just two. First, for separable signals, the following holds:

$$x[n_1]y[n_2] \leftrightarrow X(z_1)Y(z_2) \tag{5.55}$$

where (z_1, z_2) is the ROC, if z_1 is in R_x and z_2 is in R_y. The differentiation property is as follows:

$$n_1 n_2 x[n_1, n_2] \leftrightarrow z_1 z_2 \frac{\partial^2}{\partial z_1 \partial z_2} X[z_1, z_2], \quad \text{ROC} = R_x \tag{5.56}$$

In analyzing two-dimensional systems it is useful to identify singularities. Given a two-dimensional linear shift-invariant system with an associated constant-coefficient difference equation

$$\sum_{k_1=0}^{D_1} \sum_{k_2=0}^{D_2} b_{k_1 k_2}\ y[n_1 - k_1, n_2 - k_2] = \sum_{k_1=0}^{N_1} \sum_{k_2=0}^{N_2} a_{k_1 k_2}\ x[n_1 - k_1, n_2 - k_2] \tag{5.57}$$

we can find the equivalent transfer function as

$$H(z_1, z_2) = \left(\frac{\sum_{k_1=0}^{N_1} \sum_{k_2=0}^{N_2} a_{k_1 k_2} z_1^{-k_1} z_2^{-k_2}}{\sum_{k_1=0}^{D_1} \sum_{k_2=0}^{D_2} b_{k_1 k_2} z_1^{-k_1} z_2^{-k_2}} \right) = \frac{N(z_1, z_2)}{D(z_1, z_2)} \tag{5.58}$$

Then, the zero of $H(z_1, z_2)$ is a point at which $A(z_1, z_2) = 0$ and $B(z_1, z_2) \neq 0$, while a pole is a point at which $B(z_1, z_2) = 0$. Note, however, that both zeroes and poles are continuous surfaces rather than a discrete set of points, as in one dimension. Note, also, that unlike in one dimension, no fundamental theorem of algebra tells us how to factorize a multidimensional polynomial into its factors, and thus, it is not easy to isolate poles and zeroes.

5.6.2 Modified *z*-Transform

The original, one-sided *z*-transform was developed to deal only with the signal at is sampling instants, and discard the rest. In many systems, particularly in mixed analog-digital systems, it is important to conserve the information about the signal between the sampling instants. Jury, among others, used the modified ***z***-transform [3] to take care of this problem. The idea is to delay the continuous-time function $x_c(t)$ by a fictitious delay $(1-\Delta)T$, where Δ varies from 0 to 1 (Figure 5.4) in order to get all the values of $x_i(t)$ for $t=(n-1+\Delta)T$, $0 \le \Delta \le 1$, $n \in \mathbb{Z}$; T is the sampling period. Then, the modified *z*-transform is defined as follows:

$$X_c(z,\Delta) = \sum_{n=-\infty}^{+\infty} x_c[(n-1+\Delta)T]z^{-n}, \quad 0 \le \Delta \le 1 \tag{5.59}$$

or, using the change of variable $k=n-1$,

$$X_c(z,\Delta) = z^{-1} \sum_{k=-\infty}^{+\infty} x_c[(k+\Delta)T]z^{-k}, \quad 0 \le \Delta \le 1 \tag{5.60}$$

It is easy to see that the *z*-transform as defined in Equation 5.3 can be obtained as a special case of the modified *z*-transform as

$$X_c(z) = |X_c(z,\Delta)|_{\Delta=1} \tag{5.61}$$

Similar to the *z*-transform, the modified *z*-transform possesses a number of useful properties. For these, the reader is referred to Ref. [3].

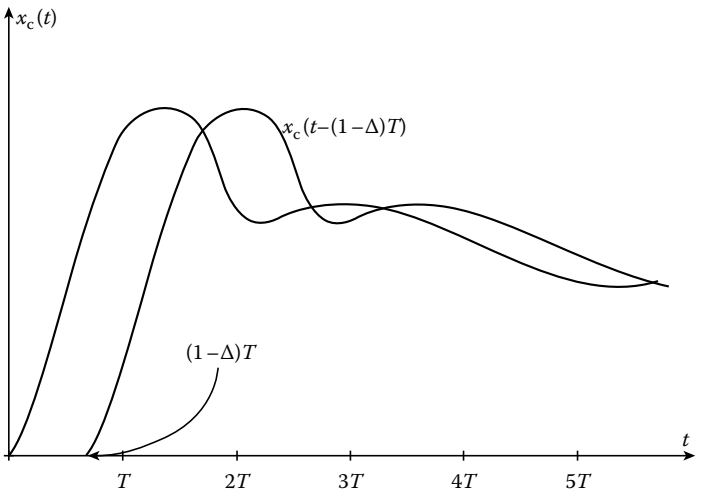


FIGURE 5.4 Modified *z*-transform takes into account values of the function between sampling instants by creating a fictitious delay $(1-\Delta)T$.

5.6.3 Chirp z-Transform Algorithm

This is a brief discussion of an algorithm for computing the z-transform of a finite-length sequence, much as was done in the case of the DFT. Suppose we wanted to compute the z-transform on a circle concentric to the unit circle. We could use the DFT algorithm, with some minor modifications. However, the chirp z-transform algorithm used in radar systems [9] can be more efficient and allows one to compute the z-transform of a finite-length sequence on a spiral contour in the z-plane. The samples we compute are equally spaced in angle over some portion of the spiral. The algorithm employs the DFT as well, and has the complexity of

$$(N + M - 1)\log_2(N + M - 1)$$

where

N is the number of nonzero values of the sequence $x[n]$, i.e., $x[0], \ldots, x[N-1]$
M is the number of points at which we evaluate the z-transform, z_k, $k = 1, \ldots, M$

5.7 Concluding Remarks

This chapter developed the basics of the z-transform. We demonstrated that it is a generalization of the discrete-time FT, and have explored the relationships between the two. We also discussed the connection to the LT. The inverse z-transform was presented, using both contour integration and partial fraction expansion. Although the partial fraction expansion method is easier, it can be used only for rational functions of z. We explored a number of properties of the ROC of the z-transform, as well as important properties of the transform itself. Finally, we showed how the z-transform is used in solving linear constant-coefficient difference equations. To conclude the section, we discussed the modified z-transform and the multidimensional z-transform. We also briefly mentioned the chirp z-transform algorithm, used for computing a few points of the z-transform of a finite-length sequence. For more details on these topics, refer to Refs. [1–3,5–7,9,10].

Acknowledgments

I thank Professor Eli Jury for his encouragement and technical insight while writing this manuscript, as well as an anonymous reviewer for his useful remarks.

References

1. N. K. Bose, *Applied Multidimensional System Theory*, New York: Van Nostrand Reinhold, 1982.
2. D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1984.
3. E. I. Jury, *Theory and Application of the z-Transform Method*, Malabar, FL: Robert E. Krieger, 1986.
4. T. Kailath, *Linear Systems*, Englewood Cliffs, NJ: Prentice Hall, 1980.
5. L. C. Ludeman, *Fundamentals of DSP*, New York: Harper & Row, 1986.
6. C. D. McGillem and G. R. Cooper, *Continuous and Discrete Signal and System Analysis*, New York: Holt, Rinehart & Winston, 1974.
7. A. V. Oppenheim and R.W. Shafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.
8. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 2nd edn., New York: McGraw-Hill, 1984.
9. L. R. Rabiner, R. W. Schafer, and C. M. Rader, The chirp z-transform algorithm, *IEEE Trans. Audio Electroacoust.*, 17, 86–92, 1969.
10. R. Vich, *z-Transform Theory and Applications*, Boston, MA: D. Reidel, 1987.

6

Wavelet Transforms

6.1 Introduction **6**-2
Scope and Outline • General Notations and Acronyms

6.2 Signal Representation Using Basis Functions **6**-3
Ideal Bandpass Wavelet • L^2 Spaces, Basis Functions, and Orthonormal Bases • Wavelet Transforms • Haar Wavelet Basis • Basic Properties of Wavelet Transforms • Filter Bank Interpretation and Time–Frequency Representation • Wavelet Basis and Fourier Basis • Most General Form of Wavelet Transformation

6.3 Short-Time Fourier Transform **6**-16
Filter Bank Interpretation • Wavelet Transform versus Short-Time Fourier Transform

6.4 Digital Filter Banks and Subband Coders **6**-22
Reconstruction from Subbands • Polyphase Representation • Paraunitary Perfect Reconstruction System • Parametrization of Paraunitary Filter Banks • Maximally Flat Solutions • Tree-Structured Filter Banks • Filter Banks and Basis Functions

6.5 Deeper Study of Wavelets, Filter Banks, and Short-Time Fourier Transforms **6**-30

6.6 Space of L^1 and L^2 Signals **6**-31
Lebesgue Integrals • L^p Signals • Fourier Transforms • Convolutions

6.7 Riesz Basis, Biorthogonality, and Other Fine Points **6**-37
Finite Dimensional Vector Spaces • Biorthogonal Systems, Riesz Bases, and Inner Products

6.8 Frames in Hilbert Spaces **6**-43
Definition of a Frame • Representing Arbitrary Vectors in Terms of Frame Elements • Exact Frames, Tight Frames, Riesz Bases, and Orthonormal Bases • Frame Operator, Dual Frame, and Biorthogonality

6.9 Short-Time Fourier Transform: Invertibility, Orthonormality, and Localization **6**-47
Time–Frequency Sampling Density for Frames and Orthonormal Bases

6.10 Wavelets and Multiresolution **6**-50
Idea of Multiresolution • Definition of Multiresolution Analysis • Relation between Multiresolution and Wavelets • Relation between Multiresolution Analysis and Paraunitary Filter Banks • Continuous Time Filter Banks and Multiresolution • Further Manifestations of Orthonormality • Generating Wavelet and Multiresolution Basis by Design of $\phi(t)$

P. P. Vaidyanathan
California Institute of Technology

Igor Djokovic
PairGain Technologies

6.11 Orthonormal Wavelet Basis from Paraunitary Filter Banks 6-63
Convergence of Infinite Products • Infinite Product Defining the Scaling Function • Orthonormal Wavelet Basis from Paraunitary Filter Bank • Wavelet Tight Frames
6.12 Compactly Supported Orthonormal Wavelets 6-71
6.13 Wavelet Regularity 6-72
Smoothness and Hölder Regularity Index • Frequency-Domain Decay and Time-Domain Smoothness • Time-Domain Decay and Time-Domain Regularity • Wavelets with Specified Regularity
6.14 Concluding Remarks 6-78
Why Wavelets? • Further Reading
Acknowledgments 6-79
References 6-79

6.1 Introduction

Transform techniques such as the Fourier and Laplace transforms and the z-transform have long been used in a wide variety of scientific and engineering disciplines [1,2]. In a number of applications in which we require a joint time–frequency picture, it is necessary to consider other types of transforms or time–frequency representations. Many such methods have evolved. The wavelet transform (WT) technique [3–5] in particular has some unique advantages over other kinds of time–frequency representations such as the short-time Fourier transform (STFT). For historical developments as well as many technical details and original material, see Ref. [5]. This chapter describes some of these representations and explains the advantages of the wavelet transform, as well as the reason for its recent popularity.

A subclass of wavelet transforms [6] has an intimate connection with the theory of digital filter banks [7–10]. Filter banks have been known to the signal processing community for over three decades (see Ref. [7] and references therein). It is this relation that makes it possible to construct in a systematic way a wide family of wavelets with several desirable properties such as compact support (i.e., finite duration), smoothness, good time–frequency localization, and basis orthonormality (all these terms will be explained later).

The connection between wavelets and filter banks finds beautiful mathematical expression in the theory of multiresolution [11]. This enables us to compute the wavelet transform coefficients using the so-called fast wavelet transform (FWT), which is essentially a tree-structured filter bank. In addition to the practical value, many deep results from several disciplines find a unified home in the theory and development of the wavelet transform. This includes signal processing, circuit theory, communications, and mathematics. Our emphasis here is on this unification, and the beautiful big picture that it provides. Other tutorials on wavelets with different choices of emphasis can be found in Refs. [7,12–14].

6.1.1 Scope and Outline

The literature on wavelets is enormous, and an attempt to do justice to everything would prove futile. Even a list of references that is fair to all contributors would be too long. We, therefore, restrict discussions to basic, core material. Sections 6.2 through 6.5 provide an overview, with the presentation given at a level that can be comprehended by most engineers. The more advanced results on wavelets, which brought them great attention in recent years, are presented in Sections 6.9 through 6.13. At the heart of these results lie several powerful mathematical tools, which are usually not familiar to engineers, and so we present a fairly extensive math review in three sections (Sections 6.6 through 6.8). We suggest that the reader go through this review material once and then use it primarily as a reference.

The advanced Sections 6.9 through 6.13 are organized such that the main points, summarized as theorems for convenience of reference, can be appreciated even without the mathematical background

material in Sections 6.6 through 6.8. The mathematical sections do, however, facilitate a deeper understanding. It is our hope that these sections will bring most readers to a point where they can pursue wavelet literature without difficulty.

Why Wavelets?
A commonly asked question is "Why wavelets?," that is, "what are the advantages offered by wavelets over other types of transform techniques such as, the Fourier transform?" The answer to this question is fairly sophisticated, and also depends on the level at which we address the question. Several discussions addressing this question are scattered throughout this chapter. A convenient listing of the locations of these discussions is given in Section 6.14 under "Why Wavelets?"

6.1.2 General Notations and Acronyms

1. Boldfaced quantities represent matrices and vectors.
2. Notations $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}^{\star}$, and $\mathbf{A}^{\dagger}$ represent, respectively, the transpose, conjugate, and transpose–conjugate of the matrix $\mathbf{A}$.
3. Accent tilde is defined as follows: $\tilde{\mathbf{H}}(z) = \mathbf{H}^{\dagger}(1/z^{\star})$; thus, if $\mathbf{H}(z) = \mathbf{\Sigma}_n \mathbf{h}(n) z^{-n}$, then $\tilde{\mathbf{H}}(z) = \sum_n \mathbf{h}^{\dagger}(-n) z^{-n}$. On the unit circle $\tilde{\mathbf{H}}(z) = \mathbf{H}^{\dagger}(z)$.
4. *Acronyms*. BIBO (bounded-input–bounded-output); FIR (finite impulse response); IIR (infinite impulse response); LTI (linear time invariant); PR (perfect reconstruction); STFT (short-time Fourier transform); WT (wavelet transform).
5. For LTI systems, "stability" stands for BIBO stability.
6. $\delta(n)$ denotes the unit pulse or discrete time impluse, defined such that $\delta(0) = 1$ and $\delta(n) = 0$ otherwise. This should be distinguished from the Dirac delta function [2], which is denoted as $\delta_a(t)$.
7. *Figures*. Sampled versions of continuous time signals are indicated with an arrow on the top (e.g., Figure 6.10a). The sampled versions are impulse trains of the form $\mathbf{\Sigma}_n c(n) \delta_a(t-n)$, and are functions of continuous t.

6.2 Signal Representation Using Basis Functions

The electrical engineer is very familiar with the Fourier transform (FT) and its role in the study of LTI systems or filters. For example, the frequency response of an LTI system is the FT of its impulse response. The FT is also used routinely in the design and analysis of circuits. As a reminder, the FT of a signal $x(t)$ is given by the familiar integral $X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} \mathrm{d}t$ and the inverse transform by*

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} \mathrm{d}\omega \tag{6.1}$$

From this equation, we can say that $x(t)$ has been expressed as a linear superposition (or linear combination) of an infinite number of functions $g_{\omega}(t) \stackrel{\Delta}{=} e^{j\omega t}$. Because the frequency ω is a continuous variable, uncountably many functions $g_{\omega}(t)$ are superimposed. Electrical engineers, in particular signal processors and communications engineers, are also familiar with two special classes of signals which can be regarded as a superposition of countably many functions:

$$x(t) = \sum_{n=-\infty}^{\infty} \alpha_n g_n(t) \tag{6.2}$$

* At the moment, it is not necessary to worry about the existence, invertibility, and the type (e.g., L^1 or L^2) of the FT. We return to the mathematical subtleties in Section 6.6.

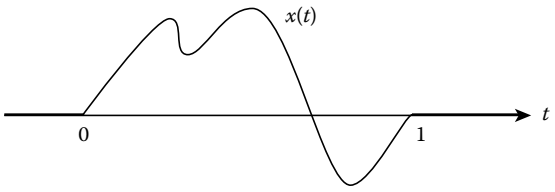


FIGURE 6.1 Finite duration signal, with support $0 \le t \le 1$.

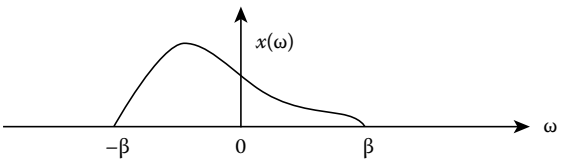


FIGURE 6.2 Fourier transform of a signal band-limited to $|\omega| < \beta$.

where α_n are scalars (possibly complex) uniquely determined by $x(t)$. These two examples are time-limited signals for which we can find a Fourier series (FS), and band-limited signals that can be reconstructed from uniformly spaced samples by weighting them with shifted sinc functions (see the following paragraph).

First consider a time-limited signal $x(t)$ with duration $0 \le t \le 1$ (Figure 6.1). Under some mild conditions such a signal can be represented in the form of Equation 6.2 with $g_n(t) = e^{j2\pi nt}$. The expression 6.2 is then the FS of $x(t)$, and α_n are the Fourier coefficients. (In contrast we say that Equation 6.1 is the Fourier integral of $x(t)$.) The transform domain signal $\{\alpha_n\}$ is a sequence, and the transform domain variable is discrete, namely, the frequencies $\omega_n \stackrel{\Delta}{=} 2\pi n$. Because $e^{j2\pi nt}$ is periodic in t with period one, the right-hand side of Equation 6.2 is periodic, and it represents $x(t)$ only in $0 \le t \le 1$. It is sometimes convenient to replace the complex functions $e^{j2\pi nt}$ with the set of real functions $1, \sqrt{2}\cos(2\pi nt), \sqrt{2}\sin(2\pi nt), n > 0$, especially in circuit analysis.

Next, consider a band-limited signal $x(t)$ with FT $X(\omega)$ as demonstrated in Figure 6.2. If we sample the signal at the Nyquist rate 2β rad/s (i.e., sampling period $T = \pi/\beta$), then multiple copies of the FT are generated [2], and we can recover $x(t)$ from the samples by use of an ideal low-pass filter $F((\omega)$ (Figure 6.3). The impulse response of the filter is the sinc function $f(t) = \sin \beta t/\beta t$ so that the reconstruction formula is

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) f(t - nT) = \sum_{n=-\infty}^{\infty} x(nT) \frac{\sin \beta(t - nT)}{\beta(t - nT)}, \quad T = \pi/\beta \tag{6.3}$$

Comparing to Equation 6.2 we see that the transform domain coefficients α_n can be regarded as the samples $x(nT)$, whereas the functions $g_n(t)$ are the shifted sinc functions.

If a signal is time limited or band limited, we can express it as a countable linear combination of a set of fundamental functions (called basis functions, in fact an orthonormal basis; see below). If the signal is more arbitrary, i.e., not limited in time or bandwidth, can we still obtain such a countable linear combination? Suppose we restrict $x(t)$ to be a finite energy signal (i.e., $\int |x(t)|^2 dt < \infty$; also called L^2 signals, see below). Then this is possible. In fact, we can find an unusual kind of basis called the wavelet basis, fundamentally different from the Fourier basis. Representation of $x(t)$ using this basis has,

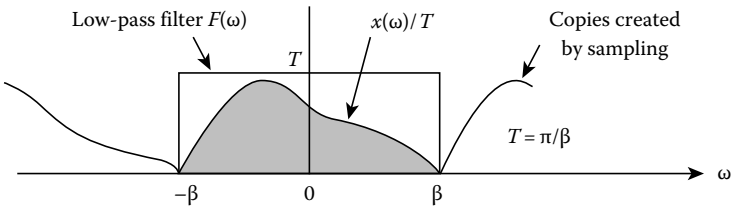


FIGURE 6.3 Use of low-pass filter $F(\omega)$ to recover $x(t)$ from its samples.

in some applications, some advantages over the Fourier representation or the short-time (windowed) Fourier representation. Wavelet bases also exist for many other classes of signals, but this discussion is limited to the L^2 class of signals.

The most common kind of wavelet representation takes the form

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{kn} \underbrace{2^{k/2}\psi(2^k t - n)}_{\psi_{kn}(t)} \tag{6.4}$$

The functions $\psi_{kn}(t)$ are typically (but not necessarily) linearly independent and form a basis for finite energy signals. The basis is very special in the sense that all the functions $\psi_{kn}(t)$ are derived from a single function $\psi(t)$ called the wavelet, by two operations: dilation ($t \rightarrow 2^k t$) and time shift ($t \rightarrow t - 2^{-k} n$). The advantage of such a basis is that it allows us to capture the details of a signal at various scales, while providing time-localization information for these ''scales.'' Examples in future sections clarify this idea.

Why Worry about Signal Representations?

A common feature of all the previous discussions is that we have taken a signal $x(t)$ and found an equivalent representation in terms of the transform domain quantity $\{\alpha_n\}$ in Equation 6.2, or $\{c_{kn}\}$ in Equation 6.4. If our only aim is to compute α_n from $x(t)$ and then recompute $x(t)$ from α_n, that would be a futile exercise. The motivation in practice is that the transform domain quantities are better suited in some sense. For example, in audio coding, decomposition of a signal into frequency components is motivated by the fact that the human ear perceives higher frequencies with less frequency resolution. We can use this information. We can also code the high-frequency components with relatively less precision, thereby enabling data compression. In this way, we can take into account perceptual information during compression. Also, we can account for the fact that the error allowed by the human ear (due to quantization of frequency components) depends on the frequency masking property of the ear, and perform optimum-bit allocation for a given bit rate. Other applications of signal representations using wavelets include numerical analysis, solution of differential equations, and many others [5,15,16].

The main point, in any case, is that we typically perform certain manipulations with the transform domain coefficients α_n (or c_{kn} in Equation 6.4) before we recombine them to form an approximation of $x(t)$. Therefore, we really only have

$$\hat{x}(t) = \sum_n \hat{\alpha}_n g_n(t) \tag{6.5}$$

where $\{\hat{\alpha}_n\}$ approximates $\{\alpha_n\}$. This discussion gives rise to many questions: how best to choose the basis functions $g_n(t)$ for a given application? How to choose the compressed signal $\{\hat{\alpha}_n\}$ so that for a given data rate the reconstruction error is minimized? What, indeed, is the best way to define the reconstruction error?

These questions are deep and complicated and will take us too far afield. Our goal is to point out the basic advantages (sometimes) offered by the WT over other kinds of transforms (e.g., the FT).

6.2.1 Ideal Bandpass Wavelet

Consider a bandpass signal $x(t)$ with FT as shown in Figure 6.4. Such signals arise in communication applications. The bandedges of the signal are ω_1 and ω_2 (and $-\omega_1$ and ((ω_2 on the negative side, which is natural if $x(t)$ is real). Viewed as a low-pass signal, the total bandwidth (counting negative frequencies also) is $2\omega_2$, but viewed as a bandpass signal, the total bandwidth is only 2β where $\beta = \omega_2 - \omega_1$. Does it mean that we can sample it at the rate 2β rad/s (which is the Nyquist rate for the low-pass case)?

In the low-pass case, sampling at Nyquist rate was enough to ensure that the copies of the spectrum created by the sampling did not overlap (Figure 6.3). In the bandpass case, we have two sets of such copies; one created by the positive half of the frequency $\omega_1 \leq \omega \leq \omega_2$ and the other by the negative

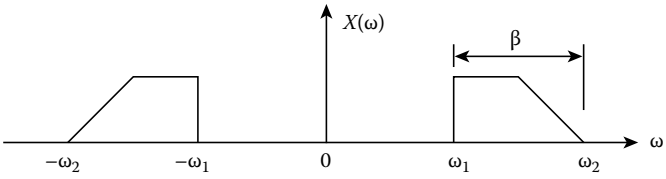


FIGURE 6.4 Fourier transform of a bandpass signal.

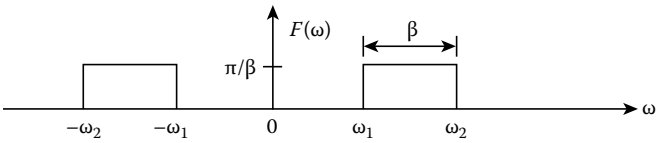


FIGURE 6.5 Bandpass filter to be used in the reconstruction of the bandpass signal from its samples.

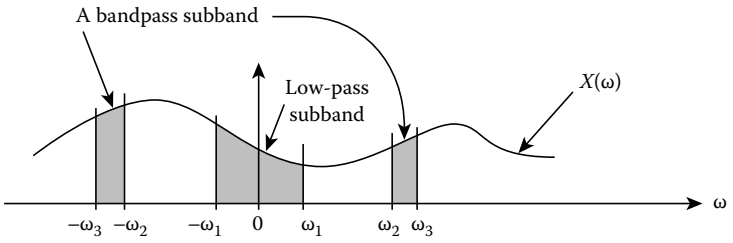


FIGURE 6.6 Splitting a signal into frequency subbands.

half $-\omega_2 \le \omega \le -\omega_1$. This makes the problem somewhat more complicated. It can be shown that, for sampling at the rate 2β, no overlap of images exists iff one of the edges, ω_1 or ω_2, is a multiple of 2β. This is called the bandpass sampling theorem. The reconstruction of $x(t)$ from the samples proceeds exactly as in the low-pass case, except that the reconstruction filter $F(\omega)$ is now a bandpass filter (Figure 6.5), occupying precisely the signal bandwidth. The first part of Equation 6.3, therefore, is still valid, i.e., $x(t) = \Sigma_n\, x(nT)\, f(t - nT)$, where $T = \pi/\beta$ again, but the sinc function is replaced with the bandpass impulse response $f(t)$.

Given a signal $x(t)$, imagine now that we have split its frequency axis into subbands in some manner (Figure 6.6). Letting $y_k(t)$ denote the kth subband signal, we can write $x(t) = \Sigma_k y_k(t)$. This can be visualized as passing $x(t)$ through a bank of filters $\{H_k(\omega)\}$ (Figure 6.7a), with responses as in Figure 6.7b. Note that each subband region is symmetric with respect to zero frequency, and therefore supports positive as well as negative frequencies. If the subband region $\omega_\kappa \le |\omega| < \omega_{k+1}$ satisfies the bandpass sampling condition, then the bandpass signal $y_k(t)$ can be expressed as a linear combination of its samples as before. Thus, $x(t) = \sum_k y_k(t) = \sum_k \sum_n y_k(nT_k) f_k(t - nT_k)$, where $T_k = \pi/\beta_k$. Here, $f_k(t)$ is the impulse response of the reconstruction filter (or synthesis filter) $F_k(\omega)$ shown in Figure 6.7c. Figure 6.7a also illustrates this reconstruction schematic.

Figure 6.8 shows the set of synthesis filters $\{F_k(\omega)\}$ for two examples of frequency splitting arrangement, namely uniform splitting and nonuniform (octave) splitting. We will see later that the uniform splitting arrangement gives an example of the STFT representation (Sections 6.3 and 6.9). In this section, we are interested in octave splitting. The bandedges of the filters here are $\omega_k = 2^k\pi$ ($k = \ldots -1, 0, 1, 2, \ldots$). The bandedges are such that $y_k(t)$ is a signal satisfying the bandpass sampling theorem. It has

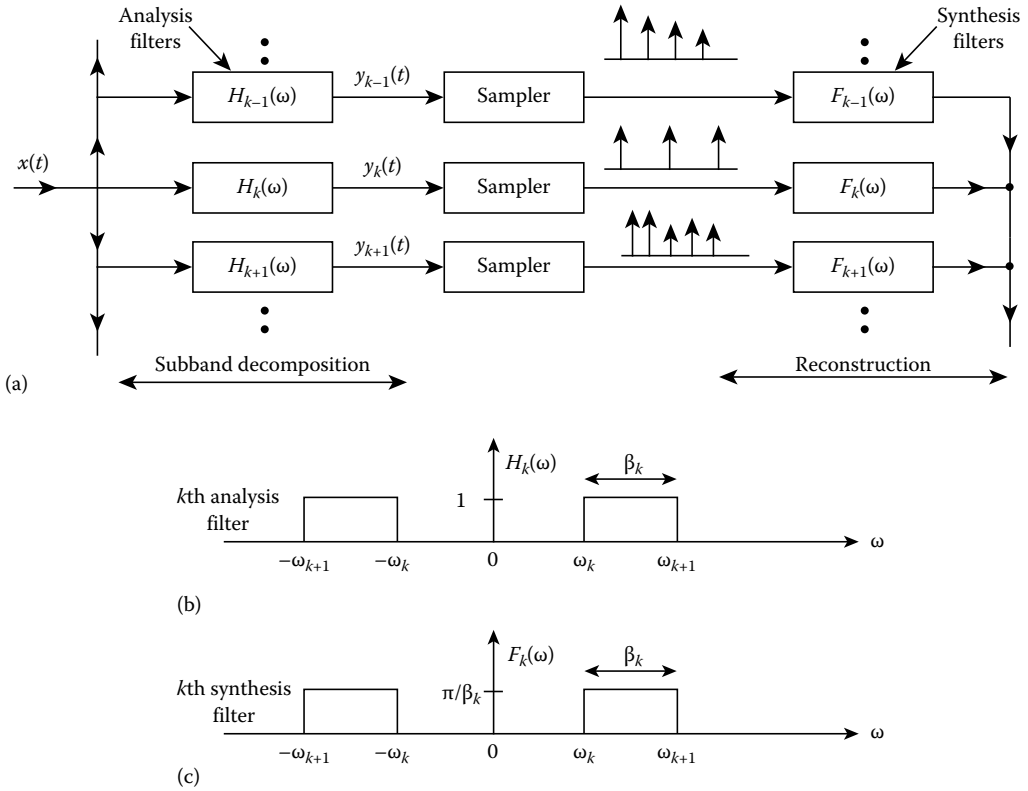


FIGURE 6.7 (a) Splitting a signal into subband signals, sampling, and then recombining; (b) response of the kth analysis filter; and (c) response of kth synthesis filter.

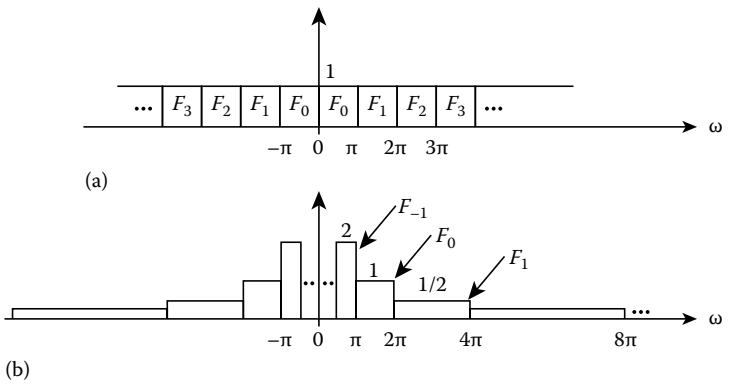


FIGURE 6.8 Two possible schemes to decompose a signal into frequency bands: (a) uniform splitting and (b) octave-band splitting. The responses shown are those of synthesis filters.

$\beta_\kappa = 2^k\pi$, according to the notation of Figure 6.7. It can be sampled at period $T_k = \pi/\beta_k = 2^{-k}$ without aliasing, and we can reconstruct it from samples as

$$y_k(t) = \sum_{n=-\infty}^{\infty} y_k(2^{-k}n) f_k(t - 2^{-k}n) \tag{6.6}$$

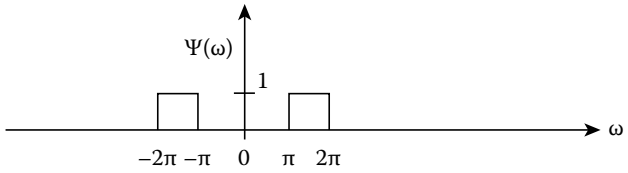


FIGURE 6.9 Fundamental bandpass function that generates a bandpass wavelet.

As k increases, the bandwidths of the filters increase so the sample spacing $T_k = 2^{-k}$ becomes finer. Because $x(t) = \Sigma_k y_k(t)$ we see that $x(t)$ can be expressed as

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} y_k(2^{-k}n) f_k(t - 2^{-k}n) \tag{6.7}$$

Our definition of the filters shows that the frequency responses are scaled versions of each other, i.e., $F_k(\omega) = 2^{-k}\ \psi(2^{-k}\omega)$, with $\psi(\omega)$ as in Figure 6.9. The impulse responses are therefore related as $f_k(t) = \psi(2^k t)$, and we can rewrite Equation 6.7 as

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} y_k(2^{-k}n) \psi(2^{-k}t - n) \tag{6.8}$$

We will write this as $x(t) = \Sigma_k \Sigma_n C_{kn}\ \psi_{kn}(t)$ by defining $c_{kn} = 2^{-k/2} y_k(2^{-k}n)$ and

$$\psi_{kn}(t) = 2^{k/2}\psi(2^k t - n) = 2^{k/2}\psi(2^k(t - 2^{-k}n)) \tag{6.9}$$

Then the functions $y_{kn}(t)$ will have the same energy $\int |\psi_{kn}(t)|^2\ \mathrm{d}t$ for all k, n. From the analysis/synthesis filter bank point of view (Figure 6.7) this is equivalent to making $H_k(\omega) = F_k(\omega)$ and rescaling, as shown in Figure 6.10. With filters so rescaled, the wavelet coefficients c_{kn} are just samples of the outputs of the analysis filters $H_k(\omega)$.

The function $\psi(2^k t)$ is a dilated version of $\psi(t)$ (squeezed version if $k > 0$ and stretched version if $k < 0$). The dilation factor 2^k is a power of 2, so this is said to be a dyadic dilation. The function $\psi(2^k(t - 2^{-k}n))$ is a shifted version of the dilated version. Thus, we have expressed $x(t)$ as a linear combination of shifted versions of (dyadic) dilated versions of a single function $\psi(t)$. The shifts $2^{-k}n$, are in integer multiples of 2^{-k}, where k governs the dilation. For completeness, note that the impulse response $\psi(t)$ corresponding to the function in Figure 6.9 is given by

$$\psi(t) = \frac{\sin(\pi t/2)}{\pi t/2} \cos(3\pi t/2) \quad \text{(ideal bandpass wavelet)} \tag{6.10}$$

This is plotted in Figure 6.11.

In Equation 6.8 we obtained a wavelet representation for $x(t)$ (compare to Equation 6.4). The function $\psi(t)$ is called the ideal bandpass wavelet, also known as the Littlewood–Paley wavelet. We now introduce some terminology for convenience and then return to more detailed definitions and discussions of the WT.

6.2.2 L^2 Spaces, Basis Functions, and Orthonormal Bases

Most of our discussions are restricted to the class of L^2 functions or square integrable functions, i.e., functions $x(t)$ for which $\int |x(t)|^2\ \mathrm{d}t$ exists and has a finite value. The norm, or L^2 norm, of such functions, denoted $\|x(t)\|_2$, is defined as $\|x(t)\|_2 = (\int |x(t)|^2 \mathrm{d}t)^{1/2}$. The notation $L^2[a, b]$ stands for L^2 functions that

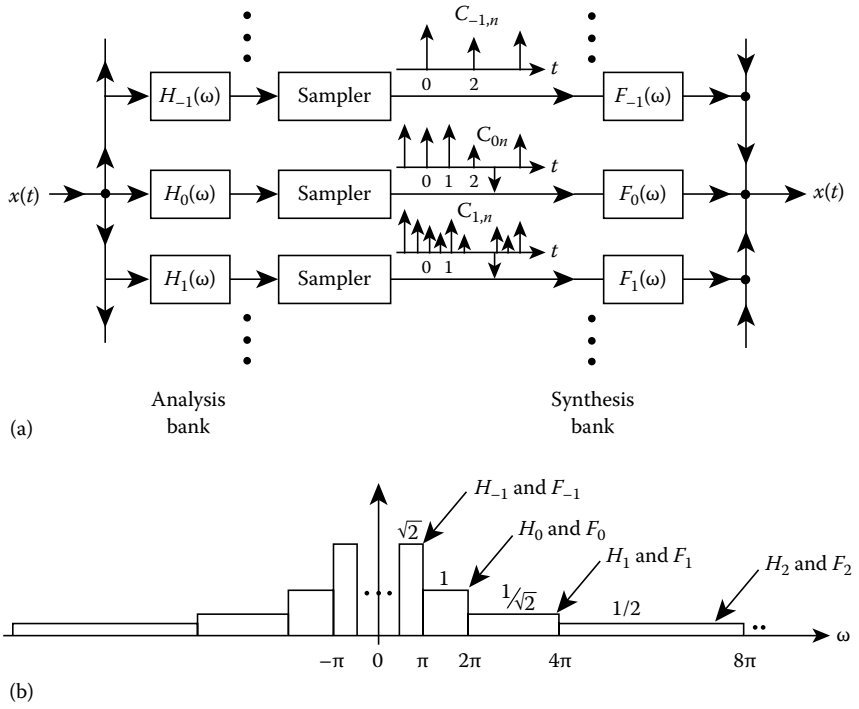


FIGURE 6.10 Octave-band splitting scheme. (a) The analysis bank, samplers, and synthesis bank and (b) the filter responses.

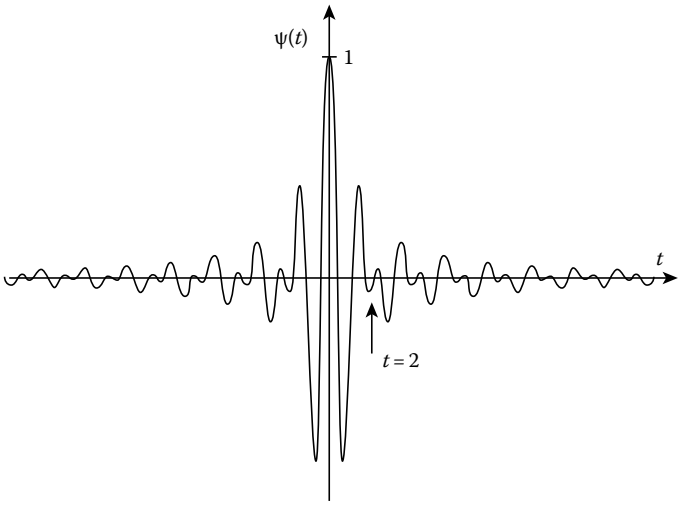


FIGURE 6.11 Ideal bandpass wavelet.

are zero outside the interval $a \leq t \leq b$. The set $L^2(R)$ is the class of L^2 functions supported on the real line $-\infty < t < \infty$. We often abbreviate $L^2(R)$ as L^2.

The class of L^2 functions forms a (normed) linear vector space, i.e., any finite linear combination of functions in L^2 is still in L^2. In fact, it forms a special linear space such that a countable basis exists. That is, a sequence of linearly independent functions $\{g_n(t)\}$ exists in L^2 such that any L^2 function $x(t)$ can be

expressed as $x(t) = \sum_n \alpha_n g_n(t)$, for a unique set of $\{\alpha_n\}$. We say that $g_n(t)$ are the basis functions. L^2 spaces have orthonormal bases. For such a basis, the basis functions satisfy

$$\langle g_k(t), g_m(t)\rangle = \delta(k-m) \tag{6.11}$$

where the notation $\langle f(t), g(t)\rangle \triangleq \int f(t)g^*(t)dt$ denotes the inner product between $f(t)$ and $g(t)$. For an orthonormal basis, the coefficients α_n in the expansion $x(t) = \sum_{n=-\infty}^{\infty} \alpha_n g_n(t)$ can thus be computed using the exceptionally simple relation

$$\alpha_n = \langle x(t), g_n(t)\rangle \tag{6.12}$$

Two examples of orthonormal basis were shown previously. The first is the FS expansion of a time-limited signal ($0 \le t \le 1$). Here, the basis functions $\{e^{j2\pi nt}\}$ are clearly orthonormal, with integrals going from 0 to 1. The second example is the expansion Equation 6.3 of a band-limited signal; it can be shown that the shifted versions $f(t - nT)$ of the sinc functions form an orthonormal basis for band-limited signals (integrals going from $-\infty$ to ∞).

Orthogonal Projections

Suppose we consider a subset $\{g_{n_k}(t)\}$ of the orthonormal basis $\{g_n(t)\}$. Let $\mathcal{S}$ denote the subspace generated by $\{g_{n_k}(t)\}$ (an accurate statement would be that $\mathcal{S}$ is the "closure of the span of $\{g_{n_k}(t)\}$"; see Section 6.7). Consider the linear combination $y(t) = \sum_k \alpha_{n_k} g_{n_k}(t)$ where the α_{n_k} are evaluated as above, i.e., $\alpha_{n_k} = \langle x(t), g_{n_k}(t)\rangle$ for some signal $x(t)$. Then, $y(t) \in \mathcal{S}$, and it can be shown that among all functions in $\mathcal{S}$, $y(t)$ is the unique signal closest to $x(t)$ (i.e, $\|x(t) - y(t)\|_2$ is the smallest). We say that $y(t)$ is the orthogonal projection of $x(t)$ onto the subspace $\mathcal{S}$, and write

$$y(t) = P_{\mathcal{S}}[x(t)] \tag{6.13}$$

6.2.3 Wavelet Transforms

If a signal $x(t)$ is in L^2, then its FT $X(\omega)$ exists in the L^2 sense (Section 6.6). We will see in Section 6.6 that the discussion which resulted in Equation 6.8 is applicable for any signal $x(t)$ in L^2. Equation 6.8 means that the signal can be expressed as a linear combination of the form

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{kn} \underbrace{2^{k/2}\psi(2^k t - n)}_{\psi_{kn}(t)} \tag{6.14}$$

where $\psi(t)$ is the impulse response (Figure 6.11) of the bandpass function $\psi(\omega)$ in Figure 6.9.* Because the frequency responses for two different values of k do not overlap, the functions $\psi_{kn}(t)$ and $\psi_{mi}(t)$ are orthogonal for $k \neq m$ (use Parseval's relation). For a given k, the functions $\psi_{kn}(t)$ are shifted versions of the impulse responses of the bandpass filter $F_k(\omega)$. From the ideal nature of this bandpass filter, we can show that $\psi_{kn}(t)$ and $\psi_{km}(t)$ are also orthonormal for $n \neq m$. Thus, the set of functions $\{\psi_{kn}(t)\}$, with k and n ranging over all integers, forms an orthonormal basis for the class of L^2 functions, i.e., any L^2 function can be expressed as in Equation 6.14 and furthermore,

$$\langle \psi_{kn}(t), \psi_{mi}(t)\rangle = \delta(k-m)\delta(n-i) \tag{6.15}$$

Because of this orthonormality, the coefficients c_{kn} are computed very easily as

$$c_{kn} = \langle x(t), \psi_{kn}(t)\rangle = \int_{-\infty}^{\infty} x(t)2^{k/2}\psi^*(2^k t - n)dt \tag{6.16}$$

* The above equality and the convergence of the summation should be interpreted in the L^2 sense; see Section 6.6.

Defining

$$\eta(t) = \psi^*(-t) \tag{6.17}$$

this takes the form

$$c_{kn} = \langle x(t), \eta^*_{kn}(-t)\rangle = \int_{-\infty}^{\infty} x(t)2^{k/2}\eta(n - 2^k t)\mathrm{d}t \tag{6.18}$$

resembling a convolution.

Wavelet Transform Definitions
A set of basis functions $\psi_{kn}(t)$ derived from a single function $\psi(t)$ by dilations and shifts of the form

$$\psi_{kn}(t) = 2^{k/2}\psi(2^k t - n) \tag{6.19}$$

is said to be a wavelet basis, and $\psi(t)$ is called the wavelet function. The coefficients c_{kn} are the wavelet transform coefficients. The formula (Equation 6.16) that performs the transformation from $x(t)$ to c_{kn} is the wavelet transform of the signal $x(t)$. Equation 6.14 is the wavelet representation or the inverse wavelet transform. While this is only a special case of more general wavelet decompositions outlined at the end of this section, it is perhaps the most popular and useful.

Note that the kth dilated version $\psi(2^k\, t)$ has the shifted versions $\psi(2^k\, t - n) = \psi(2^k(t - 2^{-k}n))$, so the amount of shift is in integer multiples of 2^{-k}. Thus, the stretched versions are shifted by larger amounts and squeezed versions by smaller amounts. Even though we developed these ideas based on an example, the above definitions still hold generally for any orthonormal wavelet basis. For the ideal bandpass wavelet, the function $\psi(t)$ is real and symmetric (Equation 6.10) so that $\eta(t) = \psi(t)$. For more general orthonormal wavelets, we have the relation $\eta(t) = \psi^*\,(-t)$. We say that $\eta(t)$ is the analyzing wavelet (because of Equation 6.18) and $\psi(t)$ the synthesis wavelet (because of Equation 6.14). For the nonorthonormal case we still have the transform and inverse transform equations as above, but the relation between $\psi(t)$ and $\eta(t)$ is not as simple as $\eta(t) = \psi^*\,(-t)$.

Before exploring the properties and usefulness of wavelets let us turn to a distinctly different example. This shows that unlike the Fourier basis functions $\{e^{j2\pi nt}\}$, the wavelet basis functions can be designed by the user. This makes them more flexible, interesting, and useful.

6.2.4 Haar Wavelet Basis

An orthonormal basis for L^2 functions was found by Haar [5] as early as 1910, which satisfies the definition of a wavelet basis given above. That is, the basis functions $\psi_{kn}(t)$ are derived from a single function $\psi(t)$ using dilations and shifts as in Equation 6.19. To explain this system, first, consider a signal $x(t) \in L^2[0, 1]$. The Haar basis is built from two functions called $\phi(t)$ and $\psi(t)$, as described in Figure 6.12. The basis function $\phi(t)$ is a constant in $[0, 1]$. The basis function $\psi(t)$ is constant on each half interval, and its integral is zero. After this, the remaining basis functions are obtained from $\psi(t)$ by dilations and shifts as indicated. It is clear from the figure that any two of these functions are mutually orthogonal. We have an orthonormal set, and it can be shown that this set of functions is an orthonormal basis for $L^2[0, 1]$. However, this is not exactly a wavelet basis yet because of the presence of $\phi(t)$.*

If we eliminate the requirement that $x(t)$ be supported or defined only on $[0, 1]$ and consider $L^2(R)$ functions then we can still obtain an orthonormal basis of the above form by including the shifted versions $\{\psi(2^k\, t - n)\}$ for all integer values of n, as well as the shifted versions $\{\phi(t - n)\}$.

An alternative to the use of $\{\phi(t - n)\}$ would be to use stretched (i.e., $\psi(2^k\, t)$, $k < 0$) as well as squeezed (i.e., $\psi(2^k\, t)$, $k > 0$) versions of $\psi(t)$. The set of functions can thus be written as in Equation 6.19, which

* We see in Section 6.10 that the function $\phi(t)$ arises naturally in the context of the fundamental idea of multiresolution.

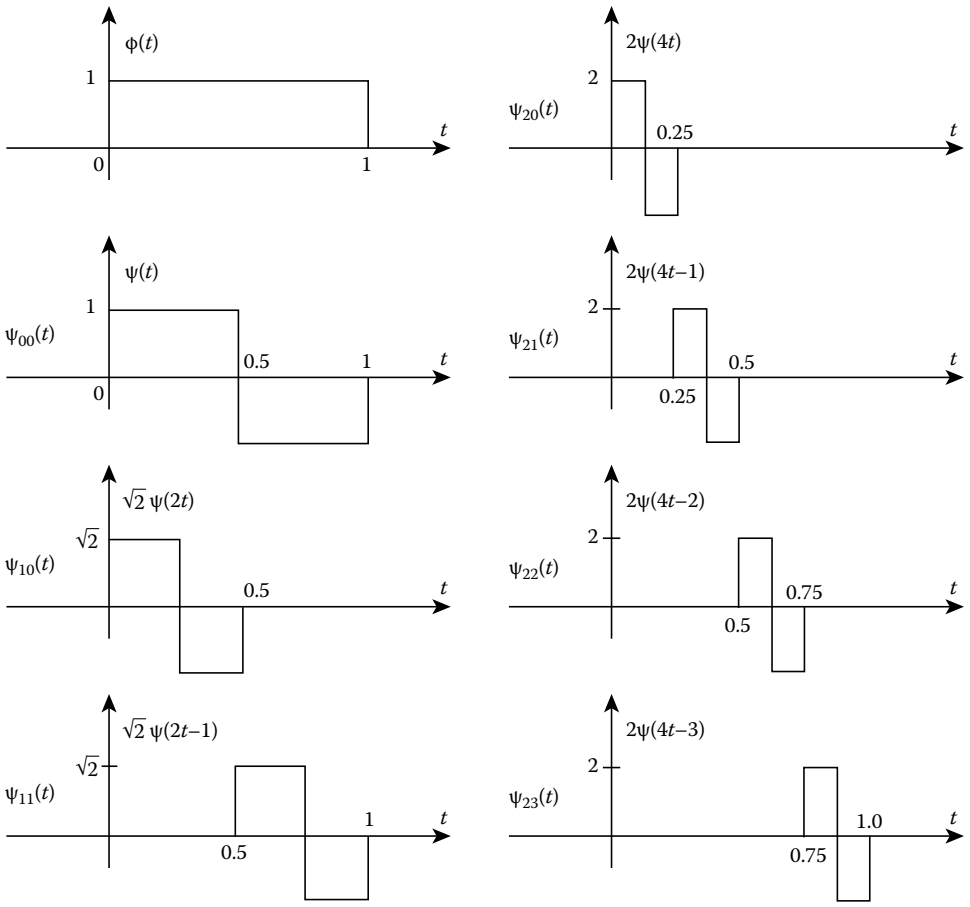


FIGURE 6.12 Examples of basis functions in the Haar basis for $L^2[0, 1]$.

has the form of a wavelet basis. It can be shown that this forms an orthonormal basis for $L^2(R)$. The FT of the Haar wavelet $\psi(t)$ is given by

$$\Psi(\omega) = je^{-j\omega/2}\frac{\sin^2(\omega/4)}{\omega/4} \quad \text{(Haar wavelet)} \tag{6.20}$$

The Haar wavelet has limited duration in time, whereas the ideal bandpass wavelet (Equation 6.10), being band limited, has infinite duration in time.

6.2.5 Basic Properties of Wavelet Transforms

Based on the definitions and examples provided so far, we can already draw some very interesting conclusions about wavelet transforms, and obtain a preliminary comparison to the FT.

1. Concept of scale: The functions $\psi_{kn}(t)$ are useful to represent increasingly finer "variations" in the signal $x(t)$ at various levels. For large k, the function $\psi_{kn}(t)$ looks like a "high frequency signal." This is especially clear from the plots of the Haar basis functions. (For the bandpass wavelets, see below.) Because these basis functions are not sinusoids, we do not use the term "frequency" but rather the term "scale." We say that the component $\psi_{kn}(t)$ represents a finer scale for larger k.

Accordingly k (sometimes $1/k$) is called the scale variable. Thus, the function $x(t)$ has been represented as a linear combination of component functions that represent variations at different scales. For instance, consider the Haar basis. If the signal expansion (Equation 6.14) has a relatively large value of $c_{4,2}$ this means that the component at scale $k = 4$ has large energy in the interval $[2/2^4, 3/2^4]$ (Figure 6.14).

2. Localized basis: The preceding comment shows that if a signal has energy at a particular scale concentrated in a slot in the time domain, then the corresponding c_{kn} has large value, i.e., $\psi_{kn}(t)$ contributes more to $x(t)$. The wavelet basis, therefore, provides localization information in time domain as well as in the scale domain. For example, if the signal is zero everywhere except in the interval $[2/2^4, 3/2^4]$ then the subset of the Haar basis functions which do not have their support in this interval are simply absent in this expansion.

Note that the Haar wavelet has compact support, that is, the function $\psi(t)$ is zero everywhere outside a closed bounded interval ($[0, 1]$ here). While the above discussions are motivated by the Haar basis, many of them are typically true, with some obvious modifications, for more general wavelets. Consider, for example, the ideal bandpass wavelet (Figure 6.11) obtained from the bandpass filter $\Psi(\omega)$ in Figure 6.9. In this case the basis functions do not have compact support, but are still locally concentrated around $t = 0$. Moreover, the basis functions for large k represent "fine" information, or the frequency component around the center frequency of the filter $F_k(\omega)$ (Figure 6.10). The Haar wavelet and the ideal bandpass wavelet are two extreme examples (one is time limited and the other band-limited). Many intermediate examples can be constructed.

6.2.6 Filter Bank Interpretation and Time–Frequency Representation

We know that the wavelet coefficients c_{kn} for the ideal bandpass wavelet can be viewed as the sampled version of the output of a bandpass filter (Figure 6.10a). The same is true for any kind of WT. For this recall, the Equation 6.18 for the wavelet coefficients. This can be interpreted as the set of sampled output sequences of a bank of filters $H_k(\omega)$, with impulse response $h_k(t) = 2^{k/2}\, h_o(2^k t)$, where $h_o(t) = \eta(t)$. Thus the wavelet transform can be interpreted as a nonuniform continuous time analysis filter bank, followed by samplers. The Haar basis and ideal bandpass wavelet basis are two examples of the choice of these bandpass filters.

The wavelet coefficients c_{kn} for a given scale k are therefore obtained by sampling the output $y_k(t)$ of the bandpass filter $H_k(\omega)$, as indicated in Figure 6.13a. The first subscript k (the scale variable) represents the filter number. As k increases by 1, the center frequency ω_k increases by a factor of 2. The wavelet coefficients c_{kn} at scale k are merely the samples $y_k(2^{-k}n)$. As k increases, the filter bandwidth increases, and thus the samples are spaced by a proportionately finer amount 2^{-k}. The quantity $c_{kn} = y_k(2^{-k}n)$ measures the "amount" of the "frequency component" around the center frequency ω_k of the analysis filter $H_k(\omega)$, localized in time around $2^{-k}n$.

In wavelet transformation, the transform domain is represented by the two integer variables k and n. This means that the transform domain is two dimensional (the time–frequency domain), and is discretized. We say that c_{kn} is a time–frequency representation of $x(t)$. Section 6.3 explains that this is an improvement over another time–frequency representation, the STFT, introduced many years ago in the signal-processing literature.

Synthesis Filter Bank and Reconstruction

The inner sum in Equation 6.14 can be interpreted as follows: For each k, convert the sequence c_{kn} into an impulse train* $\Sigma_n c_{kn}\delta_a(t - 2^{-k}n)$ and pass it through a bandpass filter $F_k(\omega) = 2^{-k/2}\Psi(2^{-k}\omega)$ with impulse response $f_k(t) = 2^{k/2}\Psi(2^k t)$. The outer sum merely adds the outputs of all these filters. Figure 6.13a shows this interpretation. Therefore, the reconstruction of the signal $x(t)$ from the wavelet

* $\delta_a(t)$ is the Dirac delta function [2]. It is used here only as a schematic. The true meaning is that the output of $f_k(t)$ is $\Sigma_n c_{kn} f_k(t - 2^{-k}n)$.

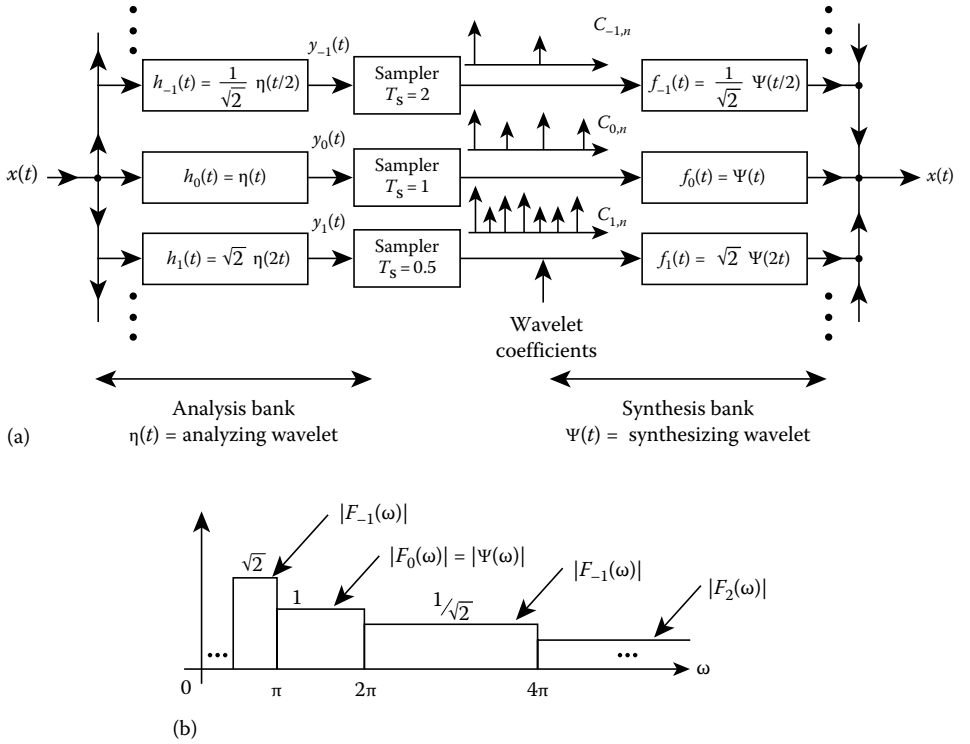


FIGURE 6.13 (a) Representing the dyadic WT as an analysis bank followed by samplers, and the inverse transform as a synthesis bank. For the orthonormal case, $\psi(t) = \eta^*(-t)$, and $f_k(t) = h_k{}^*(-t)$. (b) Filter responses for the example in which $\psi(t)$ is the ideal bandpass wavelet.

coefficients c_{kn} is equivalent to the implementation of a nonuniform continuous time synthesis filter bank, with synthesis filters $f_k(t) = 2^{k/2} f_0(2^k t)$ generated by dilations of a single filter $f_0(t) \stackrel{\Delta}{=} \psi(t)$.

As mentioned earlier, the analyzing wavelet $\eta(t)$ and the synthesis wavelet $\psi(t)$ are related by $\eta(t) = \psi^*(-t)$ in the orthonormal case. Thus, the analysis and synthesis filters are related as $h_k(t) = f_k^*(-t)$; i.e., $H_k(\omega) = F_k^*(\omega)$. For the special case of the ideal bandpass wavelet (Equation 6.10), $\psi(t)$ is real and symmetric so that $f_k(t) = f_k^*(-t)$; i.e., $h_k(t) = f_k(t)$. Figure 6.13 summarizes the relations described in the preceding paragraphs.

Design of Wavelet Functions

Because all the filters in the analysis and synthesis banks are derived from the wavelet function $\psi(t)$, the quality of the frequency responses depends directly on $\Psi(\omega)$. In the time domain the Haar basis has poor smoothness (it is not even continuous), but it is well localized (compactly supported). Its FT $\psi(\omega)$, given in Equation 6.20, decays only as $1/\omega$ for large ω. The ideal bandpass wavelet, on the other hand, is poorly localized in time, but has very smooth behavior. In fact, because it is band limited, $\psi(t)$ is infinitely differentiable, but it decays only as $1/t$ for large t. Thus, the Haar wavelet and the ideal bandpass wavelet represent two opposite extremes of the possible choices.

We could carefully design the wavelet $\psi(t)$ such that it is reasonably well localized in time domain, while at the same time sufficiently smooth or "regular." The term regularity is often used to quantify the degree of smoothness. For example, the number of times we can differentiate the wavelet $\psi(t)$ and the degree of continuity (so-called Hölder index) of the last derivative are taken as measures of regularity. We return to this in Sections 6.11 through 6.13, where we also present systematic procedures for design of the function $\psi(t)$. This can be designed in such a way that $\{2^{k/2}\psi(2^k t - n)\}$ forms an orthonormal basis with

prescribed decay and regularity properties. It is also possible to design $\psi(t)$ such that we obtain other kinds of structures rather than an orthonormal basis, e.g., a Riesz basis or a frame (Sections 6.7 and 6.8).

6.2.7 Wavelet Basis and Fourier Basis

Returning to the Fourier basis $g_k(t) = \{e^{j2\pi kt}\}$ for functions supported on [0, 1], we see that $g_k(t) = g_1(kt)$, so that all the functions are dilated versions (dilations being integers rather than powers of integers) of $g_1(t)$. However, *these do not have the localization property of wavelets*. To understand this, note that $e^{j2\pi kt}$ has unit magnitude everywhere, and sines and cosines are nonzero almost everywhere. Thus, if we have a function $x(t)$ that is identically zero in a certain time slot (e.g., Figure 6.14), then in order for the infinite series $\sum_n \alpha_n e^{j2\pi nt}$ to represent $x(t)$, extreme cancellation of terms must occur in that time slot. In contrast, if a compactly supported wavelet basis is used, it provides localization information as well as information about "frequency contents" in the form of scales. The "transform domain" in traditional FT is represented by a single continuous variable ω. In the wavelet transform, where the transform coefficients are c_{kn}, the transform domain is represented by two integers k and n.

It is also clear that WT provides a great deal of flexibility because we can choose $\psi(t)$. With FTs, on the other hand, the basis functions (sines and cosines) are pretty much fixed (see, however, Section 6.3 on STFT).

6.2.8 Most General Form of Wavelet Transformation

The most general form of wavelet transform is given by

$$X(a,b)\frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right)\mathrm{d}t \tag{6.21}$$

where a and b are real. This is called the continuous wavelet transform (CWT) because a and b are continuous variables. The transform domain is a two-dimensional domain (a, b). The restricted version of this, in which a and b take a discrete set of values $a = c^{-k}$ and $b = c^{-k}n$ (where k and n vary over the set of all integers), is called the discrete wavelet transform (DWT). The further special case, in which $c = 2$, i.e., $a = 2^{-k}$ and $b = 2^{-k}n$, is the WT discussed so far (Equation 6.16) and is called the dyadic DWT. Expansions of the form given in Equation 6.14 are also called wavelet series expansions by analogy with the FS expansion (a summation rather than an integral).

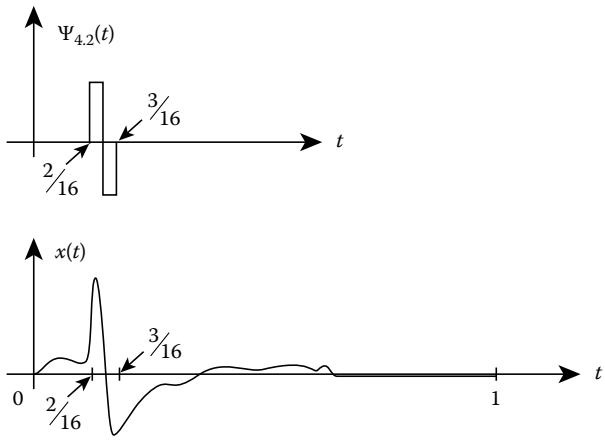


FIGURE 6.14 Example of an $L^2[0,1]$ signal $x(t)$ for which the Haar component $\psi_{4,2}(t)$ dominates.

For fixed a, Equation 6.21 is a convolution. Thus, if we apply the input signal $x(t)$ to a filter with impulse response $\psi(-t/a)/\sqrt{|a|}$, its output, evaluated at time b, will be $X(a, b)$. The filter has frequency response $\sqrt{|a|}\psi(-a\omega)$. If we imagine that $\psi(\omega)$ has a good bandpass response with center frequency ω_0, then the above filter is bandpass with center frequency $-a^{-1}\omega_0$; i.e., the wavelet transform $X(a, b)$, which is the output of the filter at time b, represents the "frequency content" of $x(t)$ around the frequency $-a^{-1}\omega_0$ "around" time b. Ignoring the minus sign (because $\psi(t)$ and $x(t)$ are typically real anyway), we see that the variable a^{-1} is analogous to frequency. In wavelet literature, the quantity $|a|$ is usually referred to as the scale rather than inverse frequency.

For reasons that cannot be explained with our limited exposure thus far, the wavelet function $\psi(t)$ is restricted to be such that $\int\psi(t)\mathrm{d}t = 0$. For the moment, notice that is equivalent to $\Psi(0) = 0$, which is consistent with the bandpass property of $\psi(t)$. In Section 6.10, where we generate wavelets systematically using multiresolution analysis, we see that this condition follows naturally from theoretical considerations.

6.3 Short-Time Fourier Transform

In many applications, we must accommodate the notion of frequency that evolves or changes with time. For example, audio signals are often regarded as signals with a time-varying spectrum, e.g., a sequence of short-lived pitch frequencies. This idea cannot be expressed with the traditional FT because $X(\omega)$ for each ω depends on $x(t)$ for all t.

The STFT was introduced as early as 1946 by Gabor [5] to provide such a time–frequency picture of the signal. Here, the signal $x(t)$ is multiplied with a window $v(t-\tau)$ centered or localized around time τ (Figure 6.15) and the FT of $x(t)\ v(t-\tau)$ computed as

$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t)v(t-\tau)e^{-j\omega t}\mathrm{d}t \tag{6.22}$$

This is then repeated for shifted locations of the window, i.e., for various values of τ. That is, we compute not just one FT, but infinitely many. The result is a function of both time τ and frequency ω. If this must be practical we must make two changes: compute the STFT only for discrete values of ω, and use only a discrete a number of window positions τ. In the traditional STFT both ω and τ are discretized on uniform grids:

$$\omega = k\omega_s, \quad \tau = nT_s \tag{6.23}$$

The STFT is thus defined as

$$X_{\mathrm{STFT}}(k\omega_s, nT_s) = \int_{-\infty}^{\infty} x(t)v(t-nT_s)e^{-jk\omega_s t}\mathrm{d}t, \tag{6.24}$$

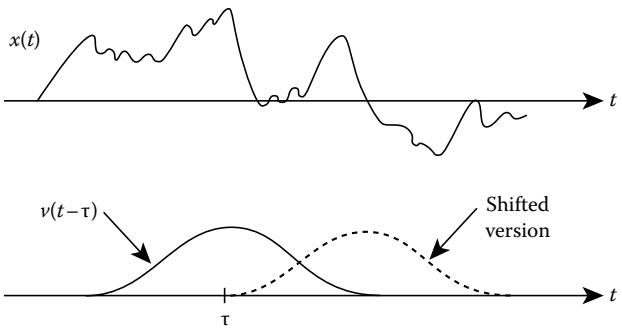


FIGURE 6.15 Signal $x(t)$ and the sliding window $v(t-\tau)$.

which we abbreviate as $X_{\text{STFT}}(k, n)$ when there is no confusion. Thus the time domain is mapped into the time–frequency domain. The quantity $X_{\text{STFT}}(k\omega_s, nT_s)$ represents the FT of $x(t)$ around time nT_s and around frequency $k\omega_s$. This, in essence, is similar to the WT: in both cases the transform domain is a two-dimensional discrete domain.

We compare wavelets and STFT on several grounds, giving a filter bank view and comparing time–frequency resolution and localization properties. Section 6.9 provides a comparison on deeper grounds: for example, when can we reconstruct a signal $x(t)$ from the STFT coefficients $X_{\text{STFT}}(k, n)$? Can we construct an orthonormal basis for L^2 signals based on the STFT? The advantage of WTs over the STFT will be clear after these discussions.

6.3.1 Filter Bank Interpretation

The STFT evaluated for some frequency ω_k can be rewritten as

$$X_{\text{STFT}}(\omega_k, \tau) = e^{-j\omega_k\tau} \int_{-\infty}^{\infty} x(t)v(t-\tau)e^{-j\omega_k(t-\tau)}\,dt \tag{6.25}$$

This integral looks like a convolution of $x(t)$ with the filter impulse response

$$h_k(t) \stackrel{\Delta}{=} v(-t)e^{j\omega_k t} \tag{6.26}$$

If $v(-t)$ has a FT looking like a low-pass filter then $h_k(t)$ looks like a bandpass filter with center frequency ω_k (Figure 6.16). Thus, $X_{\text{STFT}}(\omega_k, \tau)$ is the output of this bandpass filter at time τ, downshifted in frequency by ω_k. The result is a low-pass signal $y_k(t)$ whose output is sampled uniformly at time $\tau = nT_s$. For every frequency ω_k so analyzed, one such filter channel exists. With the frequencies uniformly located at $\omega_k = k\omega_s$, we get the analysis filter bank followed by downshifters and samplers as shown in Figure 6.17.

The STFT coefficients $X_{\text{STFT}}(k\omega_s, nT_s)$, therefore, can be regarded as the uniformly spaced samples of the outputs of a bank of bandpass filters $H_k(\omega)$, all derived from one filter $h_0(t)$ by modulation:

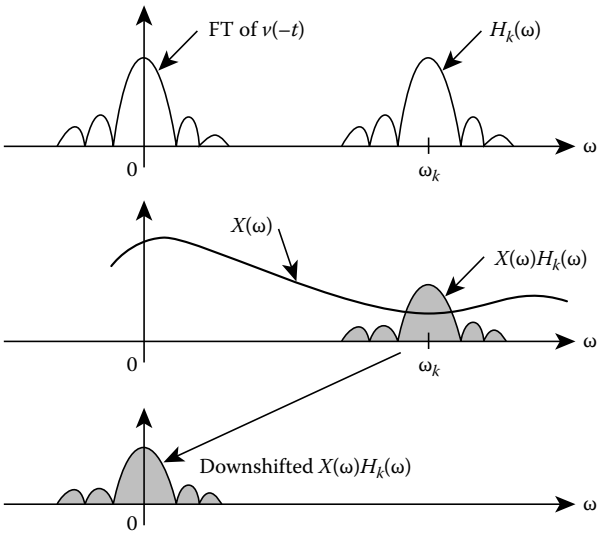


FIGURE 6.16 STFT viewed as a bandpass filter followed by a downshifter.

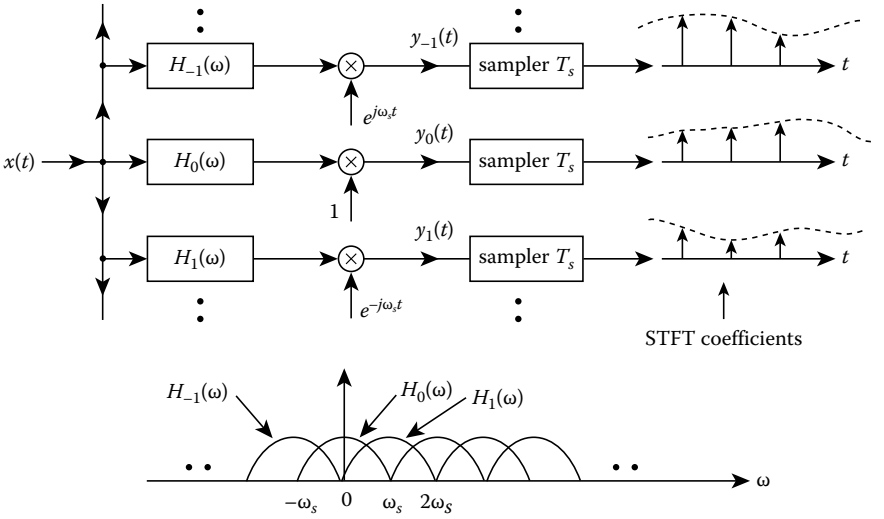


FIGURE 6.17 STFT viewed as an analysis bank of uniformly shifted filters.

$h_k(t) = e^{jk\omega st}\, h_0(t)$; i.e., $H_k(\omega) = H_0(\omega - k\omega_s)$. (The filters are one sided in frequency so they have complex coefficients in the time domain, but ignore these details for now.) The output of $H_k(\omega)$ represents a portion of the FT $X(\omega)$ around the frequency $k\omega_s$. The downshifted version $y_k(t)$ is therefore a low-pass signal. In other words, it is a slowly varying signal, whose evolution as a function of t represents the evolution of FT $X(\omega)$ around frequency $k\omega_s$. By sampling this slowly varying signal, we can therefore compress the transform domain information.

If the window is narrow in the time domain, then $H_k(\omega)$ has large bandwidth. That is, a good time resolution and poor frequency resolution are obtained. If the window is wide, the opposite is true. Thus, if we try to capture the local information in time by making a narrow window, we get a fuzzy picture in frequency. Conversely, in the limit, as the filter becomes extremely localized in frequency, the window is very broad and STFT approaches the ordinary FT. That is, the time–frequency information collapses to the all-frequency information of ordinary FT. We see that time–frequency representation is inherently a compromise between time and frequency resolutions (or localizations). This is related to the uncertainty principle: as windows get narrow in time they have to get broad in frequency, and vice versa.

Optimal Time–Frequency Resolution: The Gabor Window

What is the best frequency resolution one can obtain for a given time resolution? For a given duration of the window $v(t)$ how small can the duration of $V(\omega)$ be? If we define duration according to common sense, we are already in trouble because if $v(t)$ has finite duration then $V(\omega)$ has infinite duration. A more useful definition of duration is called the root mean square (RMS) duration. The RMS time duration D_t and the RMS frequency duration D_f for the window $v(t)$ are defined such that

$$D_t^2 = \frac{\int t^2 |v(t)|^2 \mathrm{d}t}{\int |v(t)|^2 \mathrm{d}t}, \quad D_f^2 = \frac{\int \omega^2 |V(\omega)|^2 \mathrm{d}\omega}{\int |V(\omega)|^2 \mathrm{d}\omega} \tag{6.27}$$

Intuitively, D_t cannot be arbitrarily small for a specified D_f. The uncertainty principle says that $D_t D_f \geq 0.5$. Equality holds iff $v(t)$ has the shape of a Gaussian, i.e., $v(t) = Ae^{-\alpha t^2}$, $\alpha > 0$. Thus, best joint time–frequency resolution is obtained by using the Gaussian window. This is also intuitively acceptable for the reason that the Gaussian is its own FT (except for scaling of variables and so forth). Gabor used the Gaussian window as early as 1946. Because it is of infinite duration, a truncated approximation is used in practice. The STFT based on the Gaussian is called the Gabor transform. A limitation of the

Gabor transform is that it does not give rise to an orthonormal signal representation; in fact, it cannot even provide a ''stable basis.'' (Sections 6.7 and 6.9 explain the meaning of this.)

6.3.2 Wavelet Transform versus Short-Time Fourier Transform

The STFT works with a fixed window $v(t)$. If a high-frequency signal is being analyzed, many cycles are captured by the window, and a good estimate of the FT is obtained. If a signal varies very slowly with respect to the window, however, then the window is not long enough to capture it fully. From a filter bank viewpoint, notice that all the filters have identical bandwidths (Figure 6.17). This means that the frequency resolution is uniform at all frequencies, i.e., the ''percentage resolution'' or accuracy is poor for low frequencies and becomes increasingly better at high frequencies. The STFT, therefore, does not provide uniform percentage accuracy for all frequencies; the computational resources are somehow poorly distributed.

Compare this with the WT, which is represented by a nonuniform filter bank (Figure 6.8b). Here, the frequency resolution gets poorer as the frequency increases, but the fractional resolution (i.e., the filter bandwidth $\Delta\omega_k$ divided by the center frequency ω_k) is constant for all k (the percentage accuracy is uniformly distributed in frequency). In the time domain this is roughly analogous to having a large library of windows; narrow windows are used to analyze high-frequency components and very broad windows are used to analyze low-frequency components. In electrical engineering language the filter bank representing WT is a constant Q filter bank, or an octave band filter bank. Consider, for example, the Haar wavelet basis. Here, the narrow basis functions $\psi_{2,n}(t)$ of Figure 6.12 are useful to represent the highly varying components of the input, and are correspondingly narrower (have shorter support than the functions $\psi_{1,n}(t)$).

A second difference between the STFT and the WTs is the sampling rates at the outputs of the bandpass filters. These are identical for the STFT filters (since all filters have the same bandwidth). For the wavelet filters, these are proportional to the filter bandwidths, hence nonuniform (Figure 6.10a). This is roughly analogous to the situation that the narrower windows move in smaller steps compared to the wider windows. Compare again with Figure 6.12 where $\psi_{2,n}(t)$ are moved in smaller steps as compared to $\psi_{1,n}(t)$ in the process of constructing the complete set of basis functions. The nonuniform (constant Q) filter stacking (Figure 6.8b) provided by wavelet filters is also naturally suited for analyzing audio signals and sometimes even as components in the modeling of the human hearing system.

Time–Frequency Tiling

The fact that the STFT performs uniform sampling of time and frequency whereas the WT performs nonuniform sampling is represented by the diagram shown in Figure 6.18. Here, the vertical lines represent time locations at which the analysis filter bank output is sampled, and the horizontal lines represent the center frequencies of the bandpass filters. The time–frequency tiling for the STFT is a simple rectangular grid, whereas for the WT it has a more complicated appearance.

Example 6.1

Consider the signal $x(t) = \cos(10\pi t) + 0.5\cos(5\pi t) + 0.5\cos(5\pi t) + 1.2\delta_a(t - 0.07) + 1.2\delta_a(t + 0.07)$. It has impulses at $t = \pm 0.07$ in the time domain. Two impulses (or lines) are found in the frequency domain, at $\omega_1 = 5\pi$ and $\omega_2 = 10\pi$. The function is illustrated in Figure 6.19 with impulses replaced by narrow pulses. The aim is to try to compute the STFT or WT such that the impulses in time as well as those in frequency are resolved. Figure 6.20a through c shows the STFT plot for three widths of the window $v(t)$ and Figure 6.20d shows the wavelet plot. The details of the window $v(t)$ and the wavelet $\psi(t)$ used for this example are described next, but first let us concentrate on the features of these plots.

The STFT plots are time–frequency plots, whereas the wavelet plots are (a^{-1}, b) plots, where a and b are defined by Equation 6.21. As explained in Section 6.2, the quantity a^{-1} is analogous to frequency in

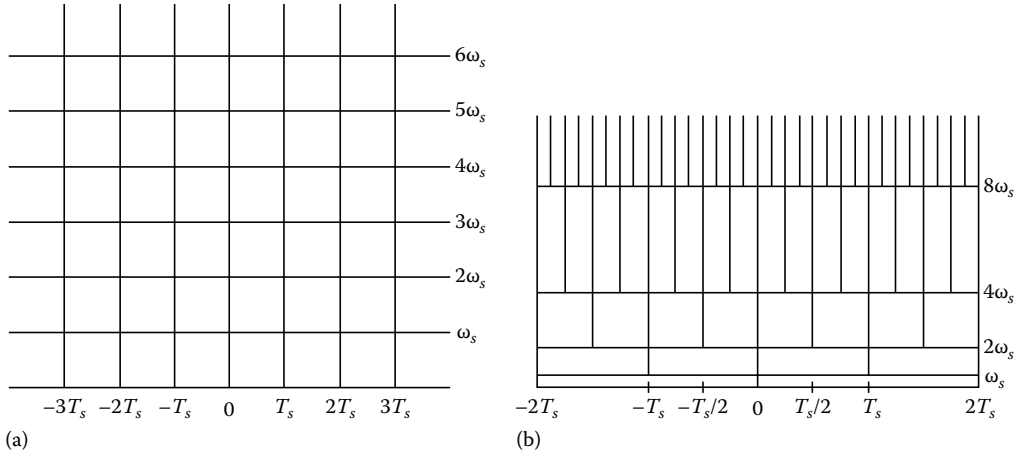


FIGURE 6.18 Time–frequency tiling schemes for (a) the STFT and (b) the WT.

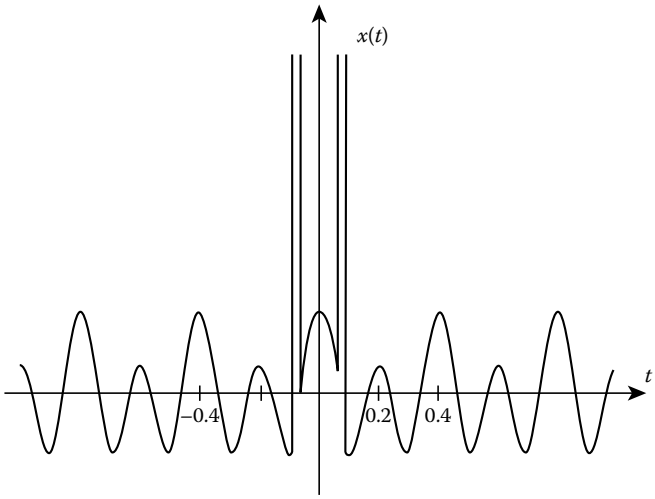


FIGURE 6.19 Signal to be analyzed by STFT and WT.

the STFT, and b is analogous to time in the STFT. The brightness of the plots in Figure 6.20 is proportional to the magnitude of the STFT or WT, so the transform is close to zero in the dark regions. We see that for a narrow window with width equal to 0.1, the STFT resolves the two impulses in time reasonably well, but the impulses in frequency are not resolved. For a wide window with width equal to 1.0, the STFT resolves the lines in frequency very well, but not the time domain impulses. For an intermediate window width equal to 0.3, the resolution is poor in both time and frequency. The wavelet transform plot (Figure 6.20d), on the other hand, simultaneously resolves both time and frequency very well. We can clearly see the locations of the two impulses in time, as well as the two lines in frequency.

The STFT for this example was computed using the Hamming window [2] defined as $v(t) = c[0.54 + 0.46\cos(\pi t/D)]$ for $-D \le t \le D$ and zero outside. The widths indicated in the figure correspond to $D = 0.1$, 1.0, and 0.3 (although the two-sided width is twice this). The wavelet transform was computed by using an example of the Morlet wavelet [5]. Specifically,

$$\psi(t) = e^{-t^2/16}(e^{j\pi t} - \alpha)$$

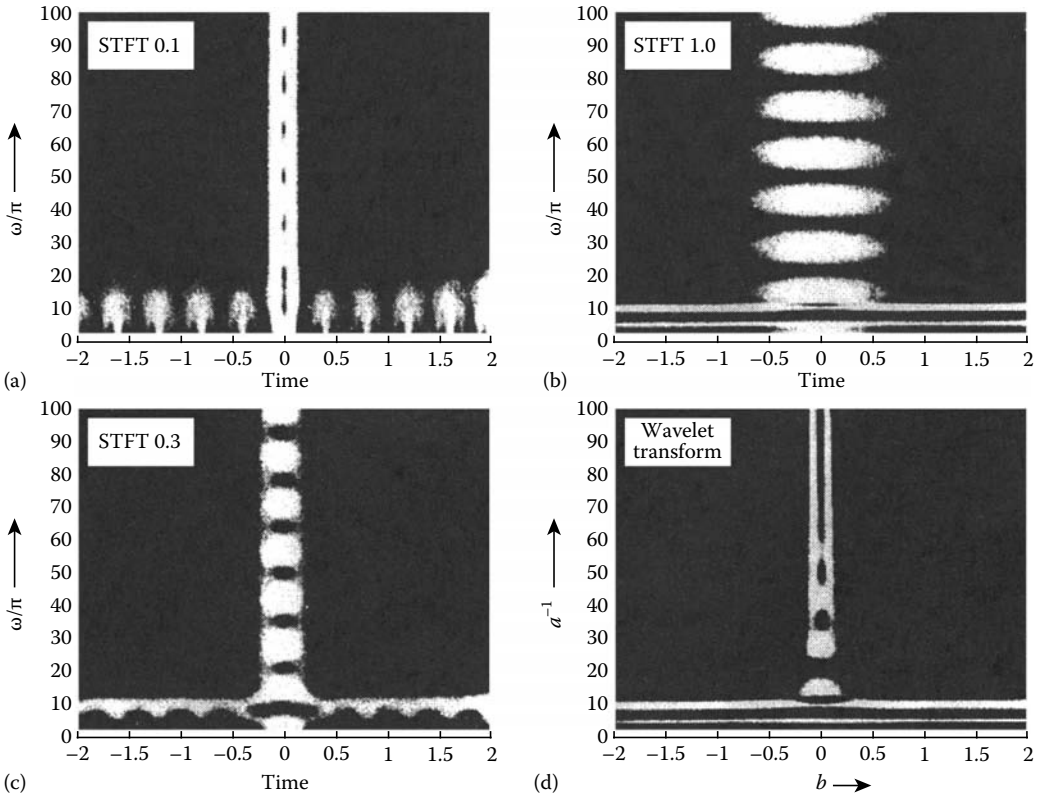


FIGURE 6.20 (a)–(c) STFT plots with window widths of 0.1, 1.0, and 0.3, respectively, and (d) WT plot.

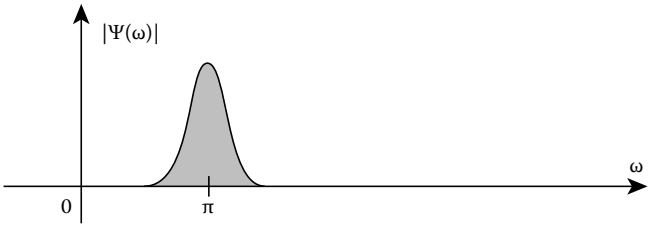


FIGURE 6.21 FT magnitude for the Morlet wavelet.

First, let us understand what this wavelet function is doing. The quantity $e^{-t^2/16}$ is the Gaussian (except for a constant scalar factor) with Fourier transform $4\sqrt{\pi}, e^{-4\omega^2}$ which is again Gaussian, concentrated near $\omega = 0$. Thus, $e^{-t^2/16}\ e^{j\pi t}$ has an FT concentrated around $\omega = \pi$. Ignoring the second term α in the expression for $\psi(t)$, we see that the wavelet is a narrowband bandpass filter concentrated around π (Figure 6.21).* If we set $a = 1$ in Equation 6.21, then $X(1, b)$ represents the frequency contents around π. Thus, the frequencies $\omega_1 = 5\pi$ and $\omega_2 = 10\pi$ in the given signal $x(t)$ show up around points $a^{-1} = 5$ and $a^{-1} = 10$ in the WT plot, as seen from Figure 6.20d. In the STFT plots, we have shown the frequency axis

* The quantity α in the expression of $\psi(t)$ ensures that $\int \psi(t)\ dt = 0$ (Section 6.2). Because α is very small, it does not significantly affect the plots in Figure 6.20.

as ω/π so that the frequencies ω_1 and ω_2 show up at 5 and 10, making it easy to compare the STFT plots with the wavelet plot.

Mathematical Issues to Be Addressed

While the filter bank viewpoint places wavelets and STFT on unified ground, several mathematical issues remain unaddressed. It is this deeper study that brings forth further subtle differences, giving wavelets a definite advantage over the STFT.

Suppose we begin from a signal $x(t) \in L^2$ and compute the STFT coefficients $X(k\omega_s, nT_s)$. How should we choose the sampling periods T_s and ω_s of the time and frequency grids so that we can reconstruct $x(t)$ from the STFT coefficients? (Remember that we are not talking about band-limited signals, and no sampling theorem is at work.) If the filters $H_k(\omega)$are ideal one-sided bandpass filters with bandwidth ω_s, the downshifted low-pass outputs $y_k(t)$ (Figure 6.16) can be sampled separately at the Nyquist rate ω_s or higher. This then tells us that $T_s \leq 2\pi/\omega_s$, that is,

$$\omega_s T_s \leq 2\pi \tag{6.28}$$

However, the use of ideal filters implies an impractical window $v(n)$.

If we use a practical window (e.g., one of finite duration), how should we choose T_s in relation to ω_s so that we can reconstruct $x(t)$ from the STFT coefficients $X(k\omega_s, nT_s)$? Is this a stable reconstruction? That is, if we make a small error in some STFT coefficient does it affect the reconstructed signal in an unbounded manner? Finally, does the STFT provide an orthonormal basis for L^2? These questions are deep and interesting, and require more careful treatment. We return to these in Section 6.9.

6.4 Digital Filter Banks and Subband Coders

Figure 6.22a shows a two-channel filter bank with input sequence $x(n)$ (a discrete-time signal). $G_a(z)$ and $H_a(z)$ are two digital filters, typically low-pass and high-pass. $x(n)$ is split into two subband signals, $x_0(n)$ and $x_1(n)$, which are then downsampled or decimated (see the following paragraphs for definitions). The total subband data rate, counting both subbands, is equal to the number of samples per unit time in the original signal $x(n)$. Digital filter banks provide a time–frequency representation for discrete time signals, similar to the STFT and WT for continuous time signals. The most common engineering application of the digital filter bank is in subband coding, which is used in audio, image, and video compression.

Neither subband coding nor such a time–frequency representation is the main point of our discussion here. We are motivated by the fact that a deep mathematical connection exists between this digital filter

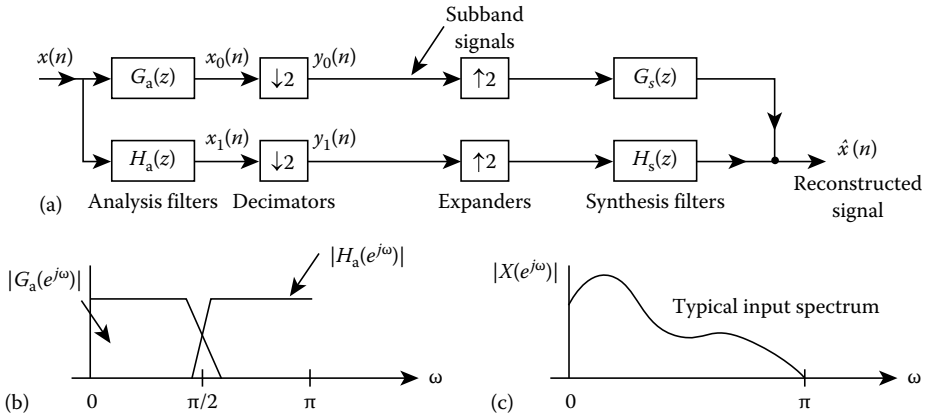


FIGURE 6.22 (a) The two-channel digital filter bank, (b) typical filter responses, and (c) typical input spectrum.

bank and the continuous time WT. This fundamental relation, discovered by Daubechies [6], is fully elaborated in Sections 6.10 through 6.13. This relation is what makes the WT so easy to design and attractive to implement in practice. Several detailed references on the topic of multirate systems and digital filter banks are available [7], and a detailed treatment can be found in Chapter 24 of Passive, *Active, and Digital Filters*, so we will be brief.

The Multirate Signal Processing Building Blocks: The building blocks in the digital filter bank of Figure 6.22a are digital filters, decimators, and expanders. The M-fold decimator or downsampler (denoted $\downarrow M$) is defined by the input–output relation $y(n) = x(Mn)$. The corresponding z-domain relation is $Y(z) = (1/M)\sum_{k=0}^{M-1} X(z^{1/M}e^{-j2\pi k/M})$. This relation is sometimes abbreviated by the notation $Y(z) = X(z)|_{\downarrow M}$ or $Y(e^{j\omega}) = X(e^{j\omega})|_{\downarrow M}$. The M-fold expander or upsampler (denoted $\uparrow M$) is defined by

$$y(n) = \begin{cases} x(n/M), & n = \text{multiple of } M, \\ 0, & \text{otherwise} \end{cases} \tag{6.29}$$

The transform domain relation for the expander is $Y(z) = X(z^M)$, i.e., $Y(e^{j\omega}) = X(e^{jM\omega})$.

6.4.1 Reconstruction from Subbands

In many applications, it is desirable to reconstruct $x(n)$ from the decimated subband signals $y_k(n)$ (possibly after quantization). For this, we pass $y_k(n)$ through expanders and combine them with the synthesis filters $G_s(z)$ and $H_s(z)$. The system is said to have the perfect reconstruction (PR) property if $\hat{x}(n) = cx(n - n_0)$ for some $c \neq 0$ and some integer n_0. The PR property is not satisfied for several reasons. First, subband quantization and bit allocation are present, which are the keys to data compression using subband techniques. However, because our interest here lies in the connection between filter banks and wavelets, we will not be concerned with subband quantization. Second, because the filters $G_a(z)$ and $H_a(z)$ are not ideal, aliasing occurs due to decimation. Using the above equations for the decimator and expander building blocks, we can obtain the following expression for the reconstructed signal: $\hat{X}(z) = T(z)X(z) + A(z)X(-z)$, where $T(z) = 0.5[G_a(z)G_s(z) + H_a(z)H_s(z)]$ and $A(z) = 0.5[G_a(-z)G_s(z) + H_a(-z)H_s(z)]$. The second term having $X(-z)$ is the aliasing term due to decimation. It can be eliminated if the filters satisfy

$$G_a(-z)G_s(z) + H_a(-z)H_s(z) = 0 \quad \text{(alias cancellation)} \tag{6.30}$$

We can then obtain PR [$\hat{x}(z) = 0.5X(z)$] by setting

$$G_a(z)G_s(z) + H_a(z)H_s(z) = 1 \tag{6.31}$$

A number of authors have developed techniques to satisfy the PR conditions. In this chapter we are interested in a particular technique to satisfy Equations 6.30 and 6.31, the conjugate quadrature filter (CQF) method, which was independently reported by Smith and Barnwell in 1984 [18] and by Mintzer in 1985 [19]. Vaidyanathan [20] showed that these constructions are examples of a general class of M channel filter banks satisfying a property called orthonormality or paraunitariness. More references can be found in Ref. [7]. The two-channel CQF solution was later rediscovered in the totally different contexts of multiresolution analysis [11] and compactly supported orthonormal wavelet construction [6]. These are discussed in subsequent sections.

Conjugate Quadrature Filter Solution

Suppose the low-pass filter $G_a(z)$ is chosen such that it satisfies the condition

$$\tilde{G}_a(z)G_a(z) + \tilde{G}_a(-z)G_a(-z) = 1 \quad \text{for all } z \tag{6.32}$$

If we now choose the high-pass filter $H_a(z)$ and the two synthesis filters as

$$H_a(z) = z^{-1}\tilde{G}_a(-z), \quad G_s(z) = \tilde{G}_a(z), \quad H_s(z) = \tilde{H}_a(z) \tag{6.33}$$

then Equations 6.30 and 6.31 are satisfied, and $\hat{x}(n) = 0.5x(n)$. In the time domain the above equations become

$$h_a(n) = -(-1)^n g_a^*(-n+1), \quad g_s(n) = g_a^*(-n), \quad h_s(n) = h_a^*(-n) \tag{6.34}$$

The synthesis filters are time-reversed conjugates of the analysis filters. If we design a filter $G_a(z)$ satisfying the single condition (Equation 6.32) and determine the remaining three filters as described previously, then the system has the PR property. A filter $G_a(z)$ satisfying Equation 6.32 is said to be power symmetric. Readers familiar with half-band filters will notice that the condition (Equation 6.32) says simply that $\tilde{G}_a(z)G_a(z)$ is half-band. To design a PR CQF system, we first design a low-pass half-band filter $G(z)$ with $G(e^{j\omega}) \geq 0$, and then extract a spectral factor $G_a(z)$. That is, find $G_a(z)$ such that $G(z) = \tilde{G}_a(z)G_a(z)$. The other filters can be found from Equation 6.33.

6.4.2 Polyphase Representation

The polyphase representation of a filter bank provides a convenient platform for studying theoretical questions and also helps in the design and implementation of PR filter banks. According to this representation the filter bank of Figure 6.22a can always be redrawn as in Figure 6.23a, which in turn can be redrawn as Figure 6.23b using standard multirate identities. Here, $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are the "polyphase matrices," determined uniquely by the analysis and synthesis filters, respectively.

If we impose the condition $\mathbf{R}(z)\mathbf{E}(z) = \mathbf{I}$, that is

$$\mathbf{R}(z) = \mathbf{E}^{-1}(z) \tag{6.35}$$

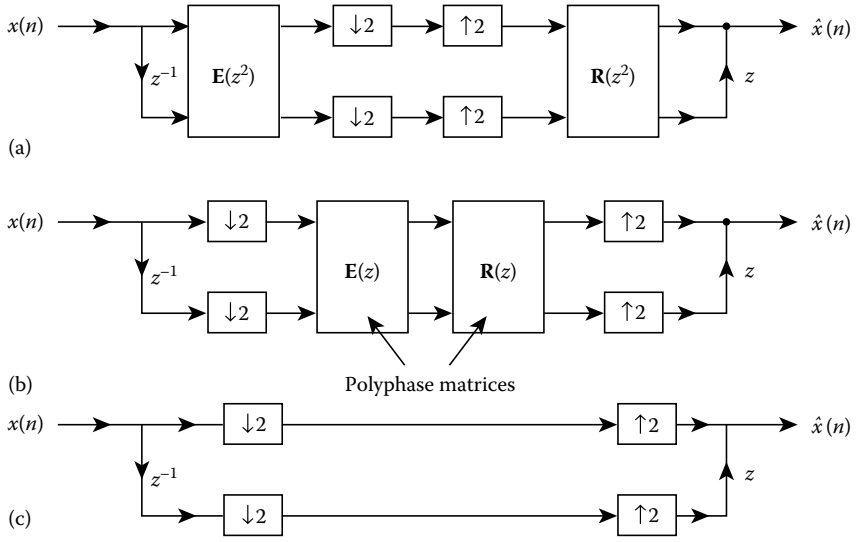


FIGURE 6.23 (a) The polyphase form of the filter bank, (b) further simplification, and (c) equivalent structure when $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$.

the system reduces to Figure 6.23c, which is a perfect reconstruction system with $\hat{x}(n) = x(n)$. Equation 6.35 will be called the PR condition. Notice that insertion of arbitrary scale factors and delays to obtain $\mathbf{R}(z) = cz^{-K}\mathbf{E}^{-1}(z)$ does not affect the PR property.

6.4.3 Paraunitary Perfect Reconstruction System

A transfer matrix* $\mathbf{H}(z)$ is said to be paraunitary if $\mathbf{H}(e^{j\omega})$ is unitary; that is, $\mathbf{H}^{\dagger}(e^{j\omega})\mathbf{H}(e^{j\omega}) = \mathbf{I}$ (more generally $\mathbf{H}^{\dagger}(e^{j\omega})\mathbf{H}(e^{j\omega}) = c\mathbf{I}$, $c > 0$, for all ω). In all practical designs, the filters are rational transfer functions so that the paraunitary condition implies $\tilde{\mathbf{H}}(z)\mathbf{H}(z) = \mathbf{I}$ for all z, where the notation $\tilde{\mathbf{H}}(z)$ was explained in Section 6.1. Note that $\tilde{\mathbf{H}}(z)$ reduces to transpose conjugation $\mathbf{H}^{\dagger}(e^{j\omega})$ on the unit circle. A filter bank in which $\mathbf{E}(z)$ is paraunitary and $\mathbf{R}(z) = \tilde{\mathbf{E}}(z)$ enjoys the PR property $\hat{x}(n) = cx(n)$, $c \neq 0$. We often say that the analysis filter pair $\{G_a(z), H_a(z)\}$ is paraunitary instead of saying that the corresponding polyphase matrix is paraunitary.

The paraunitary property has played a fundamental role in electrical network theory [1,21], and has a rich history (see references in Chapters 6 and 14 of Ref. [7]). Essentially, the scattering matrices of lossless (LC) multiports are paraunitary, i.e., unitary on the imaginary axis of the s-plane.

Properties of Paraunitary Filter Banks

Define the matrices $\mathbf{G}_a(z)$ and $\mathbf{G}_s(z)$ as follows:

$$\mathbf{G}_a(z) = \begin{bmatrix} G_a(z) & G_a(-z) \\ H_a(z) & H_a(-z) \end{bmatrix}, \quad \mathbf{G}_s(z) = \begin{bmatrix} G_s(z) & H_s(z) \\ G_s(-z) & H_s(-z) \end{bmatrix} \tag{6.36}$$

Notice that these two matrices are fully determined by the analysis filters and synthesis filters, respectively. If $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are paraunitary with $\tilde{\mathbf{E}}(z)\ \mathbf{E}(z) = 0.5\mathbf{I}$ and $\tilde{\mathbf{R}}(z)\ \mathbf{R}(z) = 0.5\mathbf{I}$, it can be shown that

$$\tilde{\mathbf{G}}_a(z)\mathbf{G}_a(z) = \mathbf{I}, \quad \tilde{\mathbf{G}}_s(z)\mathbf{G}_s(z) = \mathbf{I} \tag{6.37}$$

In other words, the matrices $\mathbf{G}_a(z)$ and $\mathbf{G}_s(z)$ defined previously are paraunitary as well.

Half-Band Property and Power Symmetry Property: The paraunitary property $\mathbf{G}_a(z)\tilde{\mathbf{G}}_a(z) = \mathbf{I}$ is also equivalent to $\mathbf{G}_a(z)\tilde{\mathbf{G}}_a(z) = \mathbf{I}$, which implies

$$\tilde{G}_a(z)G_a(z) + \tilde{G}_a(-z)G_a(-z) = 1 \tag{6.38}$$

In other words, $G_a(z)$ is a power symmetric filter. A transfer function $G(z)$ satisfying $G(z) + G(-z) = 1$ is called a half-band filter. The impulse response of such $G(z)$ satisfies $g(2n) = 0$ for all $n \neq 0$ and $g(0) = 0.5$. We see that the power symmetry property of $G_a(z)$ says that $\tilde{\mathbf{G}}_a(z)G_1(z)$ is a half-band filter. In terms of frequency response the power symmetry property of $G_a(z)$ is equivalent to

$$|G_a(e^{j\omega})|^2 + |G_a(-e^{j\omega})|^2 = 1 \tag{6.39}$$

Imagine that $G_a(z)$ is a real-coefficient low-pass filter so that $|G_a(e^{j\omega})|^2$ has symmetry with respect to zero frequency. Then $|G_a(e^{j\omega})|^2$ is as demonstrated in Figure 6.24, and the power symmetry property means that the two plots in the figure add up to unity. In this figure ω_p and ω_s are the bandedges, and δ_1 and δ_2 are the peak passband ripples of $G_a(e^{j\omega})$ (for definitions of filter specifications see Refs. [2] or [7]). Notice

* Transfer matrices are essentially transfer functions of multi-input multi-output systems. A review is found in Chapter 13 of Ref. [7].

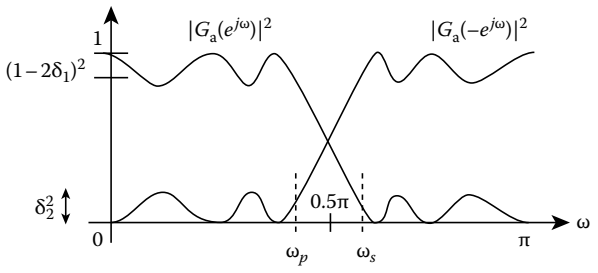


FIGURE 6.24 Magnitude responses $|G_a(e^{j\omega})|^2$ and $|G_a(e^{j\omega})|^2$ for a real-coefficient power symmetric filter $G_a(z)$.

in particular that power symmetry of $G_a(z)$ implies that a symmetry relation exists between the passband and stopband specifications of $G_a(e^{j\omega})$. This is given by $\omega_s = \pi - \omega_p$, $\delta_2{}^2 = 1 - (1 - 2\delta_1)^2$.

Relation between Analysis Filters: The property $\tilde{\mathbf{G}}_a(z)\mathbf{G}_a(z) = \mathbf{I}$ implies a relation between $G_a(z)$ and $H_a(z)$, namely $H_a(z) = e^{j\theta}z^N \tilde{G}_a(-z)$, where θ is arbitrary and N is an arbitrary odd integer. Let $N = -1$ and $\theta = 0$ for future simplicity. Then

$$H_a(z) = z^{-1}\tilde{G}_a(-z) \tag{6.40}$$

In particular, we have $|H_a(e^{j\omega})| = |G_a(-e^{j\omega})$. Combining with the power symmetry property (Equation 6.39), we see that the two analysis filters are power complementary:

$$|G_a(e^{j\omega})|^2 + |H_a(e^{j\omega})|^2 = 1 \tag{6.41}$$

for all ω. With $G_a(z) = \Sigma_n g_a(n)z^{-n}$ and $H_a(z) = \Sigma_n h_a(n)z^{-n}$ we can rewrite Equation 6.40 as

$$h_a(n) = -(-1)^n g_a^\star(-n+1) \tag{6.42}$$

Relation between Analysis and Synthesis Filters: If we use the condition $\mathbf{R}(z) = \tilde{\mathbf{E}}(z)$ in the definitions of $\mathbf{G}_s(z)$ and $\mathbf{G}_a(z)$ we obtain $\mathbf{G}_s(z) = \tilde{\mathbf{G}}_a(z)$, from which we conclude that the synthesis filters are given by $G_s(z) = \tilde{G}_a(z)$ and $H_s(z) = \tilde{H}_a(z)$. We can also rewrite these in the time domain; summarizing all this, we have

$$G_s(z) = \tilde{G}_a(z), \quad H_s(z) = \tilde{H}_a(z), \quad g_s(n) = g_a^\star(-n), \quad h_s(n) = h_a^\star(-n) \tag{6.43}$$

The synthesis filter coefficients are time-reversed conjugates of the analysis filters. Their frequency responses are conjugates of the analysis filter responses. In particular, $|G_s(e^{j\omega})| = |G_a(e^{j\omega})|$ and $|H_s(e^{j\omega})| = |H_a(e^{j\omega})|$. In view of the preceding relations, the synthesis filters have all the properties of the analysis filters. For example, $G_s(e^{j\omega})$ is power symmetric, and the pair $\{G_s(e^{j\omega}), H_s(e^{j\omega})\}$ is power complementary. Finally, $H_s(z) = z\tilde{G}_s(-z)$, instead of Equation 6.40.

Relation to Conjugate Quadrature Filter Design: The preceding discussions indicate that in a paraunitary filter bank the filter $G_a(z)$ is power symmetric, and the remaining filters are derived from $G_a(z)$ as in Equations 6.40 and 6.43. This is precisely the CQF solution for PR, stated at the beginning of this section.

Summary of Filter Relations in a Paraunitary Filter Bank

If the filter bank of Figure 6.22a is paraunitary, then the polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$ (Figure 6.23) satisfy $\tilde{\mathbf{E}}(z)\ \mathbf{E}(z) = 0.5\mathbf{I}$ and $\tilde{\mathbf{R}}(z)\ \mathbf{R}(z) = 0.5\mathbf{I}$ Equivalently, the filter matrices $\mathbf{G}_a(z)$ and $\mathbf{G}_s(z)$ and satisfy $\tilde{\mathbf{G}}_a(z)\mathbf{G}_a(z) = \mathbf{I}$ and $\tilde{\mathbf{G}}_s(z)\mathbf{G}_s(z) = \mathbf{I}$. A number of properties follow from these:

1. All four filters, $G_a(z)$, $H_a(z)$, $G_s(z)$, and $H_s(z)$ are power symmetric. This property is defined, for example, by the relation given in Equation 6.38. This means that the filters are spectral factors of half-band filters; for example, $\tilde{G}_s(z)G_s(z)$ is half band.
2. The two analysis filters are related as in Equation 6.40, so the magnitude responses are related as $|H_a(e^{j\omega})| = |G_a(-e^{j\omega})|$. The synthesis filters are time-reversed conjugates of the analysis filters as shown by Equation 6.43. In particular, $G_s(e^{j\omega}) = G_a^\star(e^{j\omega})$ and $H_s(e^{j\omega}) = H_a^\star(e^{j\omega})$.
3. The analysis filters form a power complementary pair, i.e., Equation 6.41 holds. The same is true for the synthesis filters.

4. Any two-channel paraunitary system satisfies the CQF equations (Equations 6.32 and 6.33) (except for delays, constant scale factors, etc.). Conversely, any CQF design is a paraunitary filter bank.
5. The design procedure for two-channel paraunitary (i.e., CQF) filter banks is as follows: design a zero-phase, low-pass, half-band filter $G(z)$ with $G(e^{j\omega}) \geq 0$ and then extract a spectral factor $G_a(z)$. That is, find $G_a(z)$ such that $G(z) = \tilde{G}_a(z)G_a(z)$. Then choose the remaining three filters as in Equation 6.33, or equivalently, as in Equation 6.34.

6.4.4 Parametrization of Paraunitary Filter Banks

Factorization theorems exist for matrices, which allow the expression of paraunitary matrices as a cascade of elementary paraunitary blocks. For example, let $\mathbf{H}(z) = \sum_{n=0}^{L} \mathbf{h}(n)z^{-n}$ be a 2×2 real causal FIR transfer matrix (thus, $\mathbf{h}(n)$ are 2×2 matrices with real elements). This is paraunitary iff it can be expressed as $\mathbf{H}(z) = \mathbf{R}_N\, \Lambda(z)\mathbf{R}_{N-1} \ldots \mathbf{R}_1(z)\Lambda z\mathbf{R}_0\mathbf{H}_0$ where

$$\mathbf{R}_m = \begin{bmatrix} \cos\theta_m & \sin\theta_m \\ -\sin\theta_m & \cos\theta_m \end{bmatrix}, \quad \mathbf{\Lambda}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}, \quad \mathbf{H}_0 = \begin{bmatrix} \alpha & 0 \\ 0 & \pm\alpha \end{bmatrix} \tag{6.44}$$

where α and θ_m are real. For a proof see Ref. [7]. The unitary matrix $\mathbf{R}_m$ is called a rotation operator or the Givens rotation. The factorization gives rise to a cascaded lattice structure that guarantees the paraunitary property structurally. This is useful in the design as well as the implementation of filter banks, as explained in Ref. [7]. Thus, if the polyphase matrix is computed using the cascaded structure, $G_a(z)$ is guaranteed to be power symmetric, and the relation $H_a(z) = z^{-1}\tilde{G}_a(-z)$ between the analysis filters automatically holds. Further results on factorizations are described in Chapter 24 of *Passive, Active, and Digital Filters*.

6.4.5 Maximally Flat Solutions

The half-band filter $G(z) \triangleq \tilde{G}_a(z)G_a(z)$ can be designed in many ways. One can choose to have equiripple designs or maximally flat designs [2]. An early technique for designing FIR maximally flat filters was proposed by Herrmann in 1971 [7]. This method gives closed form expressions for the filter coefficients and can be adapted easily for the special case of half-band filters. Moreover, the design automatically guarantees the condition $G(e^{j\omega}) \geq 0$, which in particular implies zero phase.

The family of maximally flat half-band filters designed by Herrmann is demonstrated in Figure 6.25. The transfer function has the form

$$G(z) = z^K \left(\frac{1+z^{-1}}{2}\right)^{2K} \sum_{n=0}^{K-1} (-z)^n \binom{K+n-1}{n} \left(\frac{1-z^{-1}}{2}\right)^{2n} \tag{6.45}$$

The filter has order $4K-2$. On the unit circle we find $2K$ zeroes, and all of these zeroes are concentrated at the point $z = -1$ (i.e., at $\omega = \pi$). The remaining $2K-2$ zeroes are located in the z-plane such that $G(z)$ has the half-band property described earlier (i.e., $G(z) + G(-z) = 1$).

Section 6.13 explains that if the CQF bank is designed by starting from Herrmann's maximally flat half-band filter, then it can be used to design continuous time wavelets with arbitrary regularity (i.e., smoothness) properties.

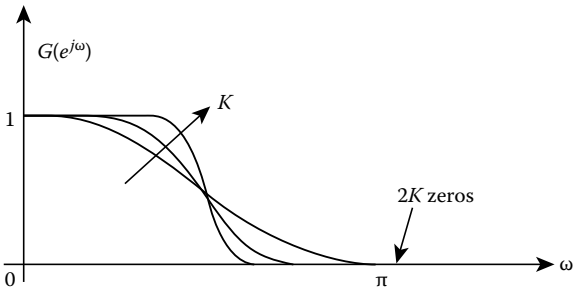


FIGURE 6.25 Maximally flat half-band filter response with $2K$ zeroes at π.

6.4.6 Tree-Structured Filter Banks

The idea of splitting a signal $x(n)$ into two subbands can be extended by splitting a subband signal further, as demonstrated in Figure 6.26a. In this example the low-pass subband is split repeatedly. This is called a tree-structured filter bank. Each node of the tree is a two-channel analysis filter bank.

The synthesis bank corresponding to Figure 6.26a is illustrated in Figure 6.26b. We combine the signals in pairs in the same manner that we split them. It can be demonstrated that if $\{G_a(z), H_a(z), G_s(z), H_s(z)\}$ is a PR system (i.e., satisfies $\hat{x}(n) = x(n)$ when connected in the form Figure 6.22a), then the tree-structured analysis/synthesis system of Figure 6.26 has PR $\hat{x}(n) = x(n)$.

The tree-structured system can be redrawn in the form shown in Figure 6.27. For example, if we have a tree structure similar to Figure 6.26 with three levels, we have $M = 4$, $n_0 = 2$, $n_1 = 4$, $n_2 = 8$, and $n_3 = 8$. If we assume that the responses of the analysis filters $G_a(e^{j\omega})$ and $H_a(e^{j\omega})$ are as in Figure 6.28a, the

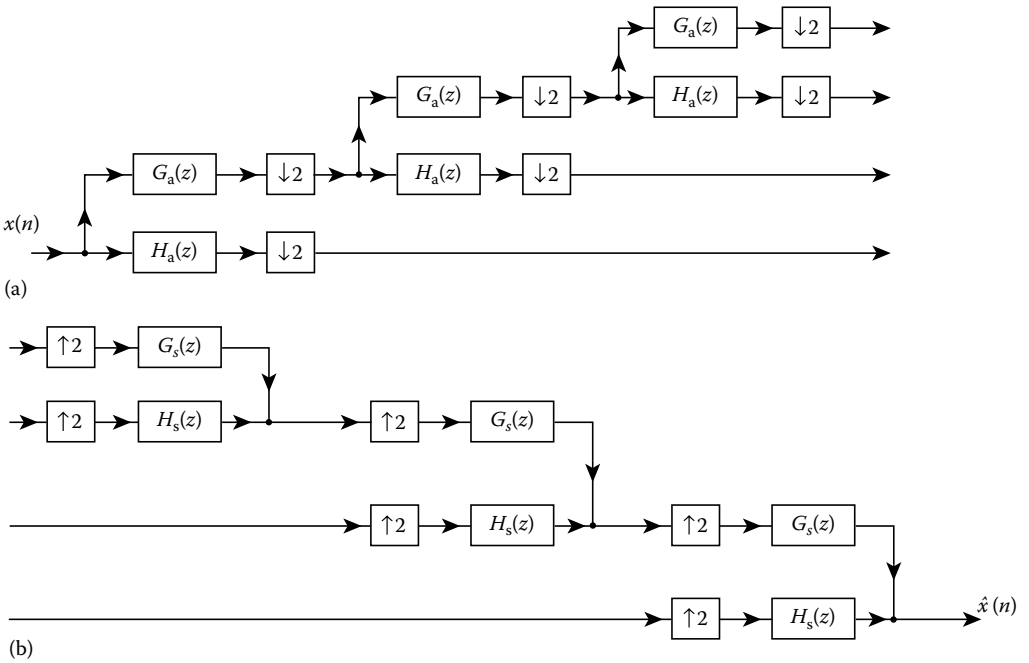


FIGURE 6.26 Tree-structured filter banks: (a) analysis bank and (b) synthesis bank.

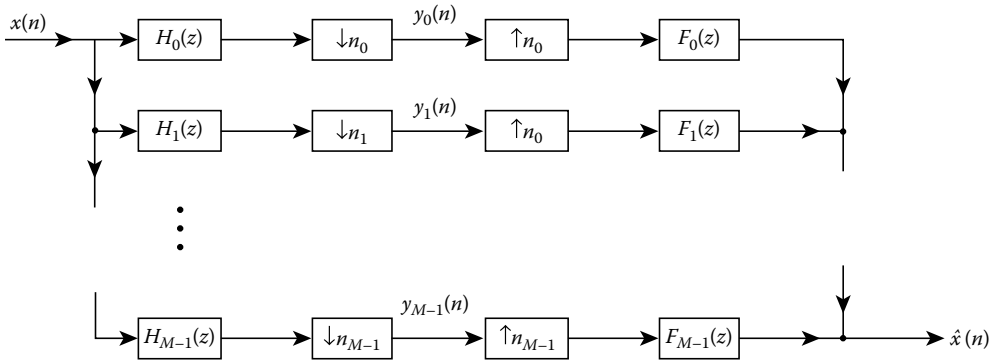


FIGURE 6.27 General nonuniform digital filter bank.

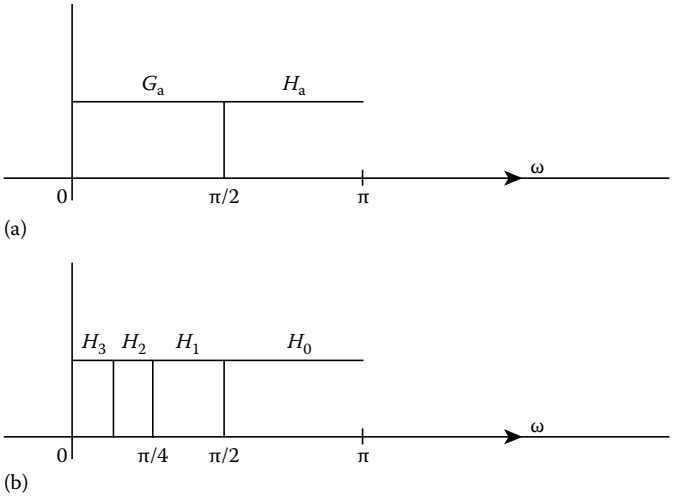


FIGURE 6.28 Example of responses: (a) $G_a(z)$ and $H_a(z)$ and (b) tree-structured analysis bank.

responses of the analysis filters $H_k(e^{j\omega})$ are as shown in Figure 6.28b. Note that this resembles the wavelet transform (Figure 6.8b). The outputs of different filters are subsampled at different rates exactly as for wavelets. Thus, the tree-structured filter bank bears a close relationship to the WT. Sections 6.10 through 6.13 present the precise mathematical connection between the two, and the relation to multiresolution analysis.

6.4.7 Filter Banks and Basis Functions

Assuming PR $\hat{x}(n) = x(n)$, we can express $x(n)$ as

$$x(n) = \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} y_k(m) \underbrace{f_k(n - n_k m)}_{\eta_{km}(n)} \tag{6.46}$$

where

$y_k(n)$ are the decimated subband signals
$f_k(n)$ are the impulse response of $F_k(z)$

Thus, the system is analogous to the filter bank systems which represented the continuous time STFT and WT in Sections 6.2 and 6.3. The collection of subband signals $y_k(m)$ can be regarded as a time–frequency representation for the sequence $x(n)$. As before, k denotes the frequency index and m the time index in the transform domain. If we have a PR filter bank, we can recover $x(n)$ from this time–frequency representation using Equation 6.46. The doubly indexed family of discrete time sequences $\{\eta_{km}(n)\}$ can be regarded as "basis functions" for the representation of $x(n)$.

To make things mathematically accurate, let $x(n) \in l^2$ (i.e., $\boldsymbol{\Sigma}_n |x(n)|^2$ is finite). If the two-channel filter bank $\{G_a(z), H_a(z), G_s(z), H_s(z)\}$, which makes up the tree structure of Figure 6.26 is paraunitary, it can be shown that $\eta_{km}(n)$ is an orthonormal basis for l^2. Orthonormality means

$$\sum_{n=-\infty}^{\infty} \eta_{k_1 m_1}(n) \eta^*_{k_2 m_2}(n) = \delta(k_1 - k_2)\delta(m_1 - m_2) \tag{6.47}$$

Notice that the basis functions (sequences) are not derived from a single function. Instead, they are derived from a finite number of filters $\{f_k(n)\}$ by time shifts of a specific form. The wavelet basis $\{2^{k/2}\,\psi\,(2^k t - n)\}$, on the other hand, is derived from a *single* wavelet function $\psi(t)$. We say that $\{\eta_{km}(n)\}$ is a *filter bank* type of basis for the space of l^2 sequences.

6.5 Deeper Study of Wavelets, Filter Banks, and Short-Time Fourier Transforms

We already know what the WT is and how it compares with the STFT, at least qualitatively. We are also familiar with time–frequency representations and digital filter banks. It is now time to fill in several important details, and generally be more quantitative. For example, we would like to mention some major technical limitations of the STFT which are not obvious from its definition, and explain that wavelets do not have this limitation.

For example, if the STFT is used to obtain an orthonormal basis for L^2 signals, the time–frequency RMS durations of the window $v(t)$ should satisfy $D_t\,D_f = \infty$. That is, either the time or the frequency resolution is very poor (Theorem 6.5). Also, if we have an STFT system in which the time–frequency sampling product $\omega_s T_s$ is small enough to admit redundancy (i.e., the vectors are not linearly independent as they would be in an orthonormal basis), the previous difficulty can be eliminated (Section 6.9).

The Gabor transform, while admittedly a tempting candidate because of the optimal time–frequency resolution property ($D_t D_f$ minimized), has a disadvantage. For example, if we want to recover the signal $x(t)$ from the STFT coefficients, the reconstruction is *unstable* in the so-called critically sampled case (Section 6.9). That is, a small error in the STFT coefficients can lead to a large error in reconstruction.

The WT does not suffer from the above limitations of the STFT. Sections 6.11 through 6.13 show how to construct orthonormal wavelet bases with good time and frequency resolutions. We also show that we can start from a paraunitary digital filter bank and construct orthonormal wavelet bases for $L^2(R)$ very systematically (Theorem 6.13). Moreover, this can be done in such a way that many desired properties (e.g., compact support, orthonormality, good time–frequency resolution, smoothness, and so forth) can be incorporated during the construction (Section 6.13). Such a construction is placed in evidence by the theory of multiresolution, which gives a unified platform for wavelet construction and filter banks (Theorems 6.6. and 6.7).

At this point, the reader may want to preview the above-mentioned theorems in order to get a flavor of things to come. However, to explain these results quantitatively, it is very convenient to review a number of mathematical tools. The need for advanced tools arises because of the intricacies associated with basis functions for infinite dimensional spaces, i.e., spaces in which the set of basis functions is an infinite set. (For finite dimensional spaces an understanding of elementary matrix theory would have been sufficient.) For example, a representation of the form $x(t) = \Sigma c_n f_n(t)$ in an infinite dimensional space could be *unstable* in the sense that a small error in the transform domain $\{c_n\}$ could be amplified in an unbounded manner during reconstruction. A special type of basis called the *Riesz basis* does not have this problem (orthonormal bases are special cases of these). Also, the so-called frames (Section 6.8) share many good properties of the Riesz bases but may have redundant vectors (i.e., not a linearly independent set of vectors). For example, the concept of frames arises in the comparison of wavelets and the STFT. General STFT frames have an advantage over STFT bases. Frames also come into consideration when the connection between wavelets and paraunitary digital filter banks is explained in Section 6.11. When describing the connection between wavelets and nonunitary filter banks, one again encounters Riesz bases and the idea of biorthogonality.

Because it is difficult to find all the mathematical background material in one place, we review a carefully selected set of topics in the next few sections. These are very useful for a deeper understanding of wavelets and STFT. The material in Section 6.6 is fairly standard (Lebesgue integrals, L^p spaces, L^1 and L^2 FTs). The material in Sections 6.7 and 6.8 (Riesz bases and frames) are less commonly known among

engineers, but play a significant role in wavelet theory. The reader may want to go through these review sections (admittedly dense), once during first reading and then use them as a reference. Following this review, we return to our discussions of wavelets, STFT, and filter banks.

6.6 Space of L^1 and L^2 Signals

We developed the wavelet representation in Section 6.2 based on the framework of a bank of bandpass filters. To make everything mathematically meaningful it becomes necessary to carefully specify the types of signals, types of FTs, etc. For example, the concept of ideal bandpass filtering is appealing to engineers, but a difficulty arises. An ideal bandpass filter $H(\omega)$ is not stable, that is, $\int |h(t)| \mathrm{d}t$ does not exist [2]. In other words $h(t)$ does not belong to the space L^1 (see below).

Why should this matter if we are discussing theory? The frequency domain developments based on Figure 6.2, which finally give rise to the time domain expression (Equation 6.8), implicitly rely on the convolution theorem (convolution in time implies multiplication in frequency). However, the convolution theorem is typically proved only for L^1 signals and bounded L^2 signals; it is not valid for arbitrary signals. Therefore, care must be exercised when using these familiar engineering notions in a mathematical discussion.

6.6.1 Lebesgue Integrals

In most engineering discussions, we think of the integrals as Riemann integrals, but in order to handle several convergence questions in the development of Fourier series, convolution theorems, and wavelet transforms, it is necessary to use Lebesgue integration. Lebesgue integration theory has many beautiful results which are not true for the Riemann integral under comparable assumptions about signals. This includes theorems that allow us to interchange limits, integrals, and infinite sums freely.

All integrals in this chapter are Lebesgue integrals. A review of Lebesgue integration is beyond the scope of this chapter, although many excellent references, for example, [22], are available. A few elementary comparisons between Riemann and Lebesgue integrals are given next.

1. If $x(t)$ is Riemann integrable on a bounded interval $[a, b]$ then it is also Lebesgue integrable on $[a, b]$. The converse is not true, however. For example, if we define $x(t) = -1$ for all rationals and $x(t) = 1$ for all irrationals in $[0, 1]$ then $x(t)$ is not Riemann integrable in $[0, 1]$, but it is Lebesgue integrable, and $\int_0^1 x(t)\mathrm{d}t = 1$.
2. A similar statement is not true for the unbounded interval $(-\infty, \infty)$. For the unbounded interval $(-\infty, \infty)$ the Riemann integral is defined only as a limit called the improper integral.* Consider the sinc function defined as: $s(t) = \sin t/t$ for $t \neq 0$, and $s(0) = 1$. This has an improper Riemann integral $= \pi$, but is not Lebesgue integrable.
3. If $x(t)$ is Lebesgue integrable, so is $|x(t)|$. The same is not true for Riemann integrals, as demonstrated by the sinc function $s(t)$ of the preceding paragraph.
4. If $|x(t)|$ is Lebesgue integrable, so is $x(t)$ as long as it is measurable.† This, however, is not true for Riemann integrals. If we define $x(t) = -1$ for all rationals and 1 for all irrationals in $[0, 1]$, it is not Riemann integrable in $[0, 1]$ although $|x(t)|$ is.
5. If $x(t)$ is (measurable and) bounded by nonnegative Lebesgue integrable function $g(t)$ (i.e., $|x(t)| \leq g(t)$), then $x(t)$ is Lebesgue integrable.

---

* Essentially, we consider $\int_{-a}^{b} x(t)\mathrm{d}t$ and let a and b go to ∞ *separately*. This limit, the improper Riemann integral, should not be confused with Cauchy principal value, which is the limit of $\int_{-a}^{a} x(t)\mathrm{d}t$ as $a \to \infty$. The function $x(t) = t$ has Cauchy principal value $= 0$, but the improper Riemann integral does not exist.

† The notion of measurable function is very subtle. Any continuous function is measurable, and any Lebesgue integrable function is measurable. In fact, examples of nonmeasurable functions are so rare and so hard to construct that practically no danger exists that we will run into one. We take measurability for granted and never mention it.

Sets of Measures Zero

A subset $\mathcal{S}$ of real numbers is said to have measure zero if, given $\varepsilon > 0$, we can find a countable union $\cup_i I_i$ of open intervals I_i (intervals of the form (a_i, b_i), i.e., $a_i < x < b_i$) such that $\mathcal{S} \subset \cup_i I_i$ and the total length of the intervals is less than ε. For example, the set of all integers (in fact, any countable set of real numbers, e.g., rationals) has measure zero. Uncountable sets of real numbers exist that have measure zero, a famous example being the Cantor set [22].

When something is said to be true "almost everywhere" (abbreviated a.e.) or "for almost all t," it means that the statement holds everywhere, except possibly on a set of measure zero. For example, if $x(t) = y(t)$ everywhere except for integer values of t, then $x(t) = y(t)$ a.e. An important fact in Lebesgue integration theory is that if two Lebesgue integrable functions are equal a.e., then their integrals are equal. In particular, if $x(t) = 0$ a.e., the Lebesgue integral $\int x(t)\mathrm{d}t$ exists and is equal to zero.

Convergence Theorems

What makes the Lebsegue integral so convenient is the existence of some powerful theorems which allow us to interchange limits with integrals and summations under very mild conditions. These theorems have been at the center of many beautiful results in Fourier and wavelet transform theory.

Let $\{g_k(t)\}$, $1 \le k \le \infty$ be a sequence of Lebesgue integrable functions. In general, this sequence may not have a limit, and even if it did, the limit may not be integrable. Under some further mild postulates, we can talk about limits and their integrals. In what follows we often say "$g(t)$ is a pointwise limit a.e., of the sequence $\{g_k(t)\}$," or "$g_k(t)$ converges to $g(t)$ a.e." This means that for any chosen value of t (except possibly in a set of measure zero), we have $g_k(t) \to g(t)$ as $k \to \infty$.

Monotone Convergence Theorem: *Suppose* $\{g_k(t)\}$ *is* nondecreasing a.e., (*i.e., for almost all values of t,* $g_k(t)$ *is nondecreasing in k) and* $\int g_k(t)\mathrm{d}t$ *is a* bounded sequence. *Then* $\{g_k(t)\}$ *converges* a.e., *to a Lebesgue integrable function* $g(t)$ *and* $\lim_k \int g_k(t)\mathrm{d}t = \int \lim_k g_k(t)\mathrm{d}t$, i.e., $\lim_k \int g_k(t)\mathrm{d}t = \int g(t)\mathrm{d}t$. *That is, we can* interchange the limit with the integral.

Dominated Convergence Theorem: *Suppose* $\{g_k(t)\}$ *is dominated by a nonnegative Lebesgue integrable function* $f(t)$, *i.e.,* $|g_k(t)| \le f(t)$ a.e., *and* $\{g_k(t)\}$ *converges to a limit* $g(t)$ a.e. *Then the limit* $g(t)$ *is Lebesgue integrable and* $\lim_k \int g_k(t)\mathrm{d}t = \int \lim_k g_k(t)\mathrm{d}t$, i.e., $\lim_k \int g_k(t)\mathrm{d}t = \int g(t)\mathrm{d}t$. *That is, we can interchange the limit with the integral.*

Levi's Theorem: *Suppose* $\int \sum_{k=1}^{m} |g_k(t)|\mathrm{d}t$ *is a bounded sequence in* m. Then $\int \sum_{k=1}^{\infty} g_k(t)\mathrm{d}t = \sum_{k=1}^{\infty} \int g_k(t)\mathrm{d}t$. *This means, in particular, that* $\sum_{k=1}^{\infty} g_k(t)$ *converges* a.e., *to a Lebesgue integrable function. This theorem permits us to interchange infinite sums with the integrals.*

Fatou's Lemma: *Let* (a) $g_k(t) \ge 0$ a.e., (b) $g_k(t) \to g(t)$ a.e., and (c) $\int g_k(t)\,\mathrm{d}t \le A$ *for some* $0 < A < \infty$. *Then the limit* $g(t)$ *is Lebesgue integrable and* $\int g(t) \le A$. (Stronger versions of this result exist [23], but we shall not require them here.)

6.6.2 L^p Signals

Let p be an integer such that $1 \le p < \infty$. A signal $x(t)$ is said to be an L^p signal if it is measurable, and if $\int |x(t)|^p\,\mathrm{d}t$ exists. We define the L^p norm of $x(t)$ as $\|x(t)\|_p = \left[\int |x(t)|^p \mathrm{d}t\right]^{1/p}$. For fixed p, the set of L^p signals forms a vector space. It is a normed linear vector space, with norm defined previously. The term "linear" means that if $x(t)$ and $y(t)$ are in L^p, then $\alpha x(t) + \beta y(t)$ is also in L^p for any complex α and β. Because any two signals $x(t)$ and $y(t)$ that are equal a.e., cannot be distinguished (i.e., $\|x(t) - y(t)\|_p = 0$), each element in L^p is in reality "a set of functions that are equal a.e." Each such set becomes an "equivalence class" in mathematical language. For $p = 2$ the quantity $\|x(t)\|_p^2$ is equal to the energy of $x(t)$, as defined in signal processing texts. Thus, an L^2 signal is a finite-energy (or square-integrable) signal. For $p = \infty$ the above definitions do not make sense, and we simply define L^∞ to be the space of essentially bounded signals. A signal $x(t)$ is said to be essentially bounded if there exists a number $B < \infty$ such that $|x(t)| \le B$ a.e. We often omit the term "essential" for simplicity; it arises because of the a.e. in

the inequality. The norm $\|x(t)\|_\infty$ is taken as essential supremum of $|x(t)|$ over all t. That is, $\|x(t)\|_\infty$ is the smallest number such that $|x(t)| \le \|x(t)\|_\infty$ a.e.

L^1, L^2, and L^∞ functions are particularly interesting for engineers. Note that neither L^1 nor L^2 contains the other. However, bounded L^1 functions are in L^2, and L^2 functions on bounded intervals are in L^1. That is

$$L^1 \cap L^\infty \subset L^2 \quad \text{and} \quad L^2[a,b] \subset L^1[a,b] \tag{6.48}$$

Thus, L^2 is already bigger than bounded L^1 functions. Moreover,

$$x(t) \in L^1 \cap L^\infty \Rightarrow x(t) \in L^p \quad \text{for all } p > 1$$

This follows because $|x(t)|^p \le |x(t)|\ \|x(t)\|_\infty^{p-1}$.Thus, $|x(t)|^p$ is (measurable and) bounded by a Lebesgue integrable function (because $|x(t)|$ is integrable), and is therefore integrable.

Orthonormal Signals in L^2

The inner product $\langle x(t), y(t)\rangle = \int x(t)y^*(t)\mathrm{d}t$ always exists for any $x(t)$ and $y(t)$ in L^2. Thus, the product of two L^2 functions is an L^1 function. If $\langle x(t), y(t)\rangle = 0$ we say that $x(t)$ and $y(t)$ are orthogonal. Clearly, $\|x(t)\|_2^2 = \langle x(t), x(t)\rangle$. Consider a sequence $\{g_n(t)\}$ of signals such that any pair of these are orthogonal, and $\|g_n(t)\|_2 = 1$ for all n. This is said to be an orthonormal sequence. The following two results are fundamental.

THEOREM 6.1

Let $\{g_n(t)\}$, $1 \le n \le \infty$ be an orthonormal sequence in L^2. Define $c_n = \langle x(t), g_n(t)\rangle$ for some $x(t) \in L^2$. Then the sum $\boldsymbol{\Sigma}_n|c_n|2$ converges, and $\boldsymbol{\Sigma}_n|c_n|^2 \le \|x(t)\|^2$.

THEOREM 6.2: (Riesz–Fischer Theorem)

Let $\{g_n(t)\}$, $1 \le n \le \infty$ be an orthonormal sequence in L^2 and let $\{c_n\}$ be a sequence of complex numbers such that $\boldsymbol{\Sigma}_n|c_n|^2$ converges. Then there exists $x(t) \in L^2$ such that $c_n = \langle x(t), g_n(t)\rangle$, and $x(t) = \boldsymbol{\Sigma}_n c_n g_n(t)$ (with equality interpreted in the L^2 sense; see below).

The space L^2 is more convenient to work with than L^1. For example, inner product and the concept of orthonormality are undefined in L^1. Moreover (see following section), the FT in L^2 has more time–frequency symmetry than in L^1. In Section 6.7, we will define unconditional bases, which have the property that any rearrangement continues to be a basis. It turns out that any orthonormal basis in L^2 is unconditional, whereas the L^1 space does not even have an unconditional basis.

Equality and Convergence in L^p Sense

Let $x(t)$ and $y(t)$ be L_P functions ($p < \infty$). Then $\|x(t) - y(t)\|_p = 0$ iff $x(t) = y(t)$ a.e. For example, if $x(t)$ and $y(t)$ differ only for every rational t we still have $\|x(t) - y(t)\|_p = 0$. Whenever $\|x(t) - y(t)\|_p = 0$, we say that $x(t) = y(t)$ in L^p sense. Now consider a statement of the form

$$x(t) = \sum_{n=1}^{\infty} c_n g_n(t) \tag{6.49}$$

for $p < \infty$, where $g_n(t)$ and $x(t)$ are in L^p. This means that the sum converges to $x(t)$ in the L^p sense; that is, $\|x(t) - \Sigma_{n=1}^{N} c_n g_n(t)\|_p$ goes to zero as $N \to \infty$. If we modify the limit $x(t)$ by adding some number to $x(t)$ for all rational t, the result is still a limit of $\Sigma_{n=1}^{N} c_n g_n(t)$ in the L^p sense. L^p limits are unique only the a.e. sense. We omit the phrase "in the L^p sense" whenever it is clear from the context.

l^p Spaces

Let p be an integer with $1 \le p \le \infty$. The collection of all sequences $x(n)$ such that $\Sigma_n |x(n)|^p$ converges to a finite value is denoted l^p. This is a linear space with norm $\|x(n)\|_p$ defined so that $\|x(n)\|_p = (\Sigma_n |x(n)|^p)^{1/p}$. Unlike L^p spaces, l^p spaces satisfy the following inclusion rule:

$$l^1 \subset l^2 \subset l^3 \ldots l^\infty \tag{6.50}$$

The spaces l^1 and l^2 are especially interesting in circuits and signal processing. If $h(n) \in l^1$, $\Sigma_n |h(n)| < \infty$. This is precisely the condition for the BIBO stability of a linear time invariant system with impulse response $h(n)$ [2].

Continuity of Inner Products

If $\{x_n(t)\}$ is a sequence in L^2 and has an L^2 limit $x(t)$, then, for any $y(t) \in L^2$,

$$\lim_{n\to\infty} \langle x_n(t), y(t)\rangle = \left\langle \lim_{n\to\infty} x_n(t), y(t)\right\rangle = \langle x(t), y(t)\rangle \tag{6.51}$$

with the second limit interpreted in the L^2 sense. Thus, limits can be interchanged with inner product signs. Similarly, infinite summation signs can be interchanged with the inner product sign, that is, $\sum_{n=1}^{\infty} \langle \alpha_n x_n(t), y(t)\rangle = \left\langle \sum_{N=1}^{\infty} \alpha_n x_n(t), y(t)\right\rangle$, provided the second summation is regarded as an L^2 limit. These follow from the fundamental property that *inner products are continuous* [23].

Next, suppose $\{x_n(t)\}$ is a sequence of functions L^p for some integer $p \ge 1$, and suppose $x_n(t) \to x(t)$ in the L^p sense. Then $\|x_n(t)\|_p \to \|x(t)\|_p$ as well. We can rephrase this as

$$\lim_{n\to\infty} \|x_n(t)\|_p = \left\| \lim_{n\to\infty} x_n(t)\right\|_p = \|x(t)\|_p \tag{6.52}$$

Thus, the limit sign can be interchanged with the norm sign, where the limit in the second expression is in the L^p sense. This follows because

$$\left| \|x_n(t)\|_p - \|x(t)\|_p \right| \le \|x_n(t) - x(t)\|_p \to 0 \text{ as } n \to \infty.$$

6.6.3 Fourier Transforms

The Fourier transform is defined for L^1 and L^2 signals in different ways. The properties of these two types of FT are significantly different. In the signal processing literature, in which we ultimately seek engineering solutions (such as filter approximation with rational transfer functions), this distinction often is not necessary. However, when we try to establish that a certain set of signals is a basis for a certain class, we must be careful, especially if we use tools such as the FT, convolution theorem, etc. (as we implicitly did in Section 6.2). Detailed references for this section include Refs. [15,22,23].

L^1 Fourier Transform

Given a signal $x(t) \in L^1$, its FT $X(\omega)$ (the L^1 FT) is defined in a manner that is familiar to engineers:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{6.53}$$

The existence of this integral is assured by the fact that $x(t)$ is in L^1.* In fact, the preceding integral exists iff $x(t) \in L^1$. The L^1 FT has the following properties:

1. $X(\omega)$ is a continuous function of ω.
2. $X(\omega) \to 0$ as $|\omega| \to \infty$. This is called the Riemann–Lebesgue lemma.
3. $X(\omega)$ is bounded, and $\|x(\omega)\| \leq \|x(t)\|_1$.

In engineering applications, we often draw the ideal low-pass filter response ($F(\omega)$ in Figure 6.3) and consider it to be the FT of the impulse response $f(t)$, but this frequency response is discontinuous and already violates property 1. This is because $f(t)$ is not in L^1 and $F(\omega)$ is not the L^1-FT of $f(t)$. That $f(t)$ is not in L^1 is consistent with the fact that the ideal filter is not BIBO stable (i.e., a bounded input may not produce bounded output because $\int|f(t)|\,dt$ is not finite).

Inverse Fourier Transform: The FT $X(\omega)$ of an L^1 signal generally is not in L^1. For example, if $x(t)$ is the rectangular pulse, then $X(\omega)$ is the sinc function which is not absolutely integrable. Thus, the familiar inverse transform formula

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega \tag{6.54}$$

does not make sense in general. However, because $X(\omega)$ is continuous and bounded, it is integrable on any bounded interval, so $\int_{-c}^{c} X(\omega)e^{j\omega t}d\omega/2\pi$ exists for any finite c. This quantity may even have a limit as $c \to \infty$, even if the Lebesgue integral or improper Rieman integral, does not exist. Such a limit (the Cauchy principal value) does represent the original function $x(t)$ under some conditions. Two such cases are outlined next.

Case 1: Thus, suppose $x(t) \in L^1$ and suppose that is of bounded variation in an interval $[a, b]$; that is, it can be expressed as the difference of two nondecreasing functions [22]. Then, we can show that the above Cauchy principal value exists, and

$$\frac{x(t^+) + x(t^-)}{2} = \lim_{c\to\infty} \frac{1}{2\pi} \int_{-c}^{c} X(\omega) e^{j\omega t} d\omega \tag{6.55}$$

for every $t \in (a, b)$. The notations $x(t^-)$ and $x(t^+)$ are the left-hand limit and the right-hand limit respectively, of $x(\cdot)$ at t; for functions of bounded variation, these limits can be shown to exist. If $x(\cdot)$ is continuous at t, then $x(t^-) = x(t^+) = x(t)$, and above reduces to the familiar inversion formula.

Case 2: Suppose now that $x(t) \in L^1$ and $X(\omega) \in L^1$ as well. Then, the integral $y(t) \triangleq \int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega/2\pi$ exists as a Lebesgue integral, and $y(t) = x(t)$ a.e. [23]. In particular, if $x(\cdot)$ is continuous at t, $x(t) = \int_{-c}^{c} X(\omega)e^{j\omega t}d\omega/2\pi$.

If $x(t)$ and $X(\omega)$ are both in L^1 they are both in L^2 as well. This is shown as follows: because $x(t) \in L^1$ implies that $X(\omega)$ is bounded. We see that $X(\omega) \in L^1 \cap L^\infty$. So $X(\omega) \in L^p$ for all integer p (see Section 6.6.2). In particular, $X(\omega) \in L^2$, so $x(t) \in L^2$ as well (by Parseval's relation; see below).

L^2 Fourier Transform

The L^1 Fourier transform lacks the convenient property of time–frequency symmetry. For example, even though $x(t)$ is in L^1, $X(\omega)$ may not in L^1. Also, even though $x(t)$ may not be continuous, $X(\omega)$ is necessarily continuous. The space L^2 is much easier to work with. Not only can we talk about inner products and orthonormal bases, perfect symmetry also exists between time and frequency domains. We must define L^2-FT differently because the usual definition (Equation 6.53) is meaningful only for L^1

* Because $x(t)$ is Lebesgue integrable (hence, measurable), the product $x(t)e^{-j\omega t}$ is measurable, and it is bounded by the integrable function $|x(t)|$. Thus, $x(t)e^{-j\omega t}$ is integrable.

signals. Suppose $x(t) \in L^2$ and we truncate it to the interval $[-n, n]$. This truncated version is in L^1 because of Equation 6.48, and its L^1 FT exists:

$$X_n(\omega) = \int_{-n}^{n} x(t)e^{-j\omega t}\,dt \tag{6.56}$$

It can be shown that $X_n(\omega)$ is in L^2 and that the sequence $\{X_n(\omega)\}$ has a limit in L^2. That is, there exists an L^2 function $X(\omega)$ such that

$$\lim_{n\to\infty} \|X_n(\omega) - X(\omega)\|_2 = 0 \tag{6.57}$$

This limit $X(\omega)$ is defined to be the L^2 FT of $x(t)$. Some of the properties are listed next:

1. $X(\omega)$ is in L^2, and we can compute $x(t)$ from $X(\omega)$ in an entirely analogous manner, namely the L^2 limit of $\int_{-n}^{n} X(\omega)e^{j\omega t}d\omega/2\pi$.
2. If $x(t)$ is in L^1 and L^2, then the above computation gives the same answer as the L^1-FT Equation 6.53 a.e. For example, consider the rectangular pulse $x(t) = 1$ in $[-1, 1]$ and zero otherwise. This is in L^1 and L^2, and the FT using either definition is $X(\omega) = 2\sin\omega/\omega$. This answer is in L^2, but not in L^1. The inverse L^2-FT of $X(\omega)$ is the original $x(t)$.
3. If $x(\omega) \in L^2$ and $x(\omega) \in L^1$ then the Lebesgue integral $\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega/2\pi$ exists, and equals $x(t)$ a.e.
4. Parseval's relation holds, i.e., $\sqrt{2\pi}\|x(t)\|_2 = \|X(\omega)\|_2$. Thus, the FT is a linear transformation from L^2 to L^2, which preserves norms except the scale factor $\sqrt{2\pi}$. (Note that this would not make sense if $x(t)$ were only in L^1.) In particular, it is a bounded transformation because the norm $\|X(\omega)\|_2$ in the transform domain is bounded by the norm $\|x(t)\|_2$ in the original domain.
5. Unlike the L^1-FT, the L^2-FT $X(\omega)$ need not be continuous. For example, the impulse response of an ideal low-pass filter (sinc function) is in L^2 and its FT is not continuous.
6. Let $\{f_n(t)\}$ be a sequence in L^2 and let $x(t) = \Sigma_n c_n f_n(t)$ be a convergent summation (in the L^2 sense). With upper case letters denoting the L^2-FTs, $x(\omega) = \Sigma_n c_n F_n(\omega)$. This result is obvious for finite summations because of the linearity of the FT. For infinite summations, this follows from the property that the L^2-FT is a continuous mapping from L^2 to L^2. (This in turn follows from the result that it is a bounded linear transformation). The continuity allows us to move the FT operation inside the infinite summation.

Thus, complete symmetry exists between the time and frequency domains. The L^2-FT is a one-to-one mapping from L^2 onto L^2. Moreover, because $\sqrt{2\pi}\|x(t)\|_2 = \|X(\omega)\|_2$, it is a norm preserving mapping—one says that the L^2-FT is an isometry from L^2 to L^2.

l^1 Fourier Transform

If a sequence $x(n) \in l^1$ its discrete-time FT $X(e^{j\omega}) = \Sigma_n x(n)e^{-j\omega n}$ exists, and is the l^1-FT of $x(n)$. It can be demonstrated that $X(e^{j\omega})$ is a continuous function of ω and that $|X(e^{j\omega})|$ is bounded.

6.6.4 Convolutions

Suppose $h(t) \in L^1$ and $h(t) \in L^p$ for some p in $1 \le p \le \infty$. The familiar convolution integral defined by $(x^* h)(t) = \int x(\tau)h(t-\tau)d\tau$ exists for almost all t [23]. If we define a function $y(t)$ to be $x \cdot h$ where it exists and to be zero elsewhere, the result is, in fact, an L^p function. We simply say that the convolution of an L^1, function with an L^p function gives an L^p function. By recalling that an LTI system is stable (i.e., BIBO stable), iff its impulse response is in L^1, we have the following examples:

1. If an L^1 signal is input to a stable LTI system, the output is in L^1. Because the convolution of two L^1 signals is in L^1, the cascade of two stable LTI systems is stable, a readily accepted fact in engineering.
2. If an L^2 signal (finite energy input) is input to a stable LTI system, the output is in L^2.
3. If an L^∞ signal is input to a stable LTI system, the output is in L^∞ (i.e., bounded inputs produce bounded outputs).

If $x(t)$ and $h(t)$ are both in L^1, their convolution $y(t)$ is in L^1, and all three signals have L^1-FT. The convolution theorem [23] says that theses three are related as $Y(\omega) = H(\omega)\, X(\omega)$. When signals are not necessarily in L^1 we cannot in general write this, even if convolution might itself be well defined.

Convolution Theorems for L^2 Signals

For all our discussions in the preceding sections, the signals were restricted to be in L^2, but not necessarily in L^1. In fact, even the filters are often only in L^2. For example, ideal bandpass filters (Figure 6.8) are unstable, and therefore only in L^2. For arbitrary L^2 signals $x(t)$ and $h(t)$, the convolution theorem does not hold. We therefore need to better understand L^2 convolution.

Assume that $x(t)$ and $h(t)$ are both in L^2. Their convolution $y(t) = \int x(\tau)h(t-\tau)\mathrm{d}\tau$ exists for all t, as the integral is only an inner product in L^2. Using Schwartz inequality [23], we also have $|y(t)| \leq \|x(t)\|_2 \, \|h(t)\|_2$, that is, $y(t) \in L^\infty$. Suppose the filter $h(t)$ has the further property that the frequency response $H(\omega)$ is bounded, i.e., $|H(\omega)| \leq B$ a.e., for some $B < \infty$. Then we can show that $y(t) \in L^2$, and that the convolution theorem holds ($Y(\omega) = H(\omega)X(\omega)$). To prove this, note that

$$y(t) = \int x(\tau)h(t-\tau)\mathrm{d}\tau = \frac{1}{2\pi}\int X(\omega)H(\omega)e^{j\omega t}\mathrm{d}\omega \tag{6.58}$$

from Parseval's relation which holds for L^2 signals [23]. If $|H(\omega)| \leq B$, then $|X(\omega)H(\omega)|^2 \leq B^2|X(\omega)|^2$. Therefore, $|X(\omega)H(\omega)|^2$ is bounded by the integrable function $|X(\omega)|^2$, and is therefore integrable. Thus, $X(\omega)H(\omega) \in L^2$, and the preceding equation establishes that $y(t) \in L^2$. The equation also shows that $y(t)$ and $H(\omega)X(\omega)$ form an L^2-FT pair, so $Y(\omega) = H(\omega)X(\omega)$.

Bounded L^2 Filters

Filters for which $h(t) \in L^2$ and $H(\omega)$ bounded are called bounded L^2 filters. The preceding discussion shows that bounded L^2 filters admit the convolution theorem, although arbitrary L^2 filters do not. Another advantage of bounded L^2 filters is that a cascade of two bounded L^2 filters, $h_1(\mathrm{t})$ and $h_2(\mathrm{t})$, is a bounded L^2 filter, just as a cascade of two stable filters would be stable. To see this, note that the cascaded impulse response is the convolution $h(t) = (h_1 * h_2)(t)$. By the preceding discussion, $h(t) \in L^2$, and moreover, $H(\omega) = H_1(\omega)H_2(\omega)$. Clearly, $H(\omega)$ is still bounded. Bounded L^2 filters are therefore very convenient to work with. Fortunately, all filters in the discussion of wavelets and filter banks are bounded L^2 filters, even though they may not be BIBO stable (as are the ideal bandpass filters in Figure 6.8). We summarize the preceding discussions as follows.

THEOREM 6.3: (Convolution of L^2 functions)

We say that $h(t)$ is a bounded L^2 filter if $h(t) \in L^2$ and $|H(\omega)| \leq B < \infty$ a.e.

1. *Let $x(t) \in L^2$ and let $h(t)$ be a bounded L^2 filter. Then $y(t) = (x * h)(t)$ exists for all t and $y(t) \in L^2$. Moreover, $Y(\omega) = H(\omega)X(\omega)$.*
2. *If $h_1(t)$ and $h_2(t)$ are bounded L^2 filters, then their cascade $h(t) = (h_1 * h_2)(t)$ is a bounded L^2 filter and $H(\omega) = H_1(\omega)H_2(\omega)$.*

6.7 Riesz Basis, Biorthogonality, and Other Fine Points

In a finite dimensional space, such as the space of all N-component Euclidean vectors, the ideas of basis and orthonormal basis are easy to appreciate. When we extend these ideas to infinite dimensional spaces (i.e., where the basis $\{g_n(t)\}$ has infinite number of functions), a number of complications and subtleties arise. Our aim is to point these out. References for this section include Refs. [5,15,24].

Readers familiar Hilbert spaces will note that the L^2 space is a Hilbert space; all our developments here are valid for any Hilbert space $\mathcal{H}$. Elements in $\mathcal{H}$ (vectors) are typically denoted *x, y,* etc. When we deal with Hilbert space L^2, the vectors are functions and are denoted as $x(t)$, $y(t)$, etc., for clarity. Similarly, for the special case of Euclidean vectors we use boldface, e.g., $\mathbf{x}$, $\mathbf{y}$, etc. The reader not familiar with Hilbert spaces can assume that all discussions are in L^2 and that x is merely a simplification of the notation $x(t)$.

6.7.1 Finite Dimensional Vector Spaces

We first look at the finite dimensional case and then proceed to the infinite dimensional case. Consider an $N \times N$ matrix $\mathbf{F} = [\mathbf{f}_1\ \mathbf{f}_2 \ldots \mathbf{f}_N]$. We assume that this is nonsingular, that is, the columns $\mathbf{f}_n$ are linearly independent. These column vectors form a basis for the N-dimensional Euclidean space $\mathcal{C}^N$ of complex N-component vectors. This space is an example of a finite dimensional Hilbert space, with inner product defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^\dagger = \sum_{n=1}^{N} x_n x_n^\star$. The norm $\|\mathbf{x}\|$ induced by this inner product is defined as $\|\mathbf{x}\| \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. Thus, $\|\mathrm{x}\|^2 = \mathbf{x}^\dagger \mathbf{x} = \sum_{n=1}^{N} |x_n|^2$.

Any vector $\mathrm{x} \in \mathcal{C}^N$ can be expressed as $\mathbf{x} = \mathbf{\Sigma}_{n=1}^{N} c_n \mathbf{f}_n$ for some uniquely determined set of scalars c_n. We can abbreviate this as $\mathbf{x} = \mathbf{Fc}$, where $\mathbf{c} = [c_1\ c_2 \ldots c_N]^{\mathrm{T}}$. The matrix $\mathbf{F}$ can be regarded as a linear transformation from $\mathcal{C}^N$ to $\mathcal{C}^N$. The nonsingularity of $\mathbf{F}$ means that for every $\mathbf{x} \in \mathcal{C}^N$ we can find a unique $\mathbf{c}$ such that $\mathbf{x} = \mathbf{Fc}$.

6.7.1.1 Boundedness of F and Its Inverse

In practice, we have further requirement that if the norm $\|\mathbf{c}\|$ is "small" then $\|\mathbf{x}\|$ should also be "small," and vice versa. This requirement implies, for example, that if a small error occurs in the transmission or estimate of the vector $\mathbf{c}$, the corresponding error in x is also small. From the relation $\mathbf{x} = \mathbf{Fc}$ we obtain

$$\|\mathbf{x}\|^2 = \mathbf{x}^\dagger \mathbf{x} = \mathbf{c}^\dagger \mathbf{F}^\dagger \mathbf{F} \mathbf{c} \tag{6.59}$$

Letting λ_{M} and λ_{m} denote the maximum and minimum eigenvalues of $\mathbf{F}^\dagger \mathbf{F}$, it then follows that $\|\mathbf{x}\|^2 \geq \lambda_{\mathrm{m}} \|\mathbf{c}\|^2$ and that $\|\mathbf{x}\|^2 \leq \lambda_{\mathrm{M}} \|\mathbf{c}\|^2$. That is

$$\lambda_{\mathrm{m}} \|\mathbf{c}\|^2 \leq \|\mathbf{x}\|^2 \leq \lambda_{\mathrm{M}} \|\mathbf{c}\|^2 \tag{6.60}$$

with $0 < \lambda_{\mathrm{m}} \leq \lambda_{\mathrm{M}} < \infty$, where $0 < \lambda_{\mathrm{m}}$ follows from the nonsingularity of $\mathbf{F}$. Thus, the transformation $\mathbf{F}$, which converts $\mathbf{c}$ into $\mathbf{x}$, has an amplification factor bounded by λ_{M} in the sense that $\|\mathbf{x}\|^2 \leq \lambda_{\mathrm{M}} \|\mathbf{c}\|^2$. Similarly, the inverse transformation $\mathbf{G} = \mathbf{F}^{-1}$, which converts $\mathbf{x}$ into $\mathbf{c}$, has amplification bounded by $1/\lambda_{\mathrm{m}}$. Because λ_{M} is finite, we say that F is a bounded linear transformation, and because $\lambda_{\mathrm{m}} \neq 0$, we see that the inverse transformation is also bounded.

Using $\mathbf{x} = \mathbf{\Sigma}_n c_n \mathbf{f}_n$ and $\|\mathrm{c}\|^2 = \mathbf{\Sigma}_n |c_n|^2$ we can rewrite the preceding inequality as

$$A \sum_n |c_n|^2 \leq \left\| \sum_n c_n \mathbf{f}_n \right\|^2 \leq B \sum_n |c_n|^2 \tag{6.61}$$

where

$A = \lambda_m > 0$

$B = \lambda_M < \infty$, and all summations are for $1 \leq n \leq N$

Readers familiar with the idea of a Riesz basis in infinite dimensional Hilbert spaces will notice that the above is in the form that agrees with that definition. We will return to this issue later.

Biorthogonality

With $\mathbf{F}^{-1}$ denoted as $\mathbf{G}$, let $\mathbf{g}_n^\dagger$ denote the rows of $\mathbf{G}$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^\dagger \\ \mathbf{g}_2^\dagger \\ \vdots \\ \mathbf{g}_N^\dagger \end{bmatrix}, \quad \mathbf{F} = [\mathbf{f}_1\ \mathbf{f}_2 \ldots \mathbf{f}_N] \tag{6.62}$$

The property $\mathbf{GF} = \mathbf{I}$ implies $\mathbf{g}_k^\dagger \mathbf{f}_n = \delta(k-n)$:

$$\langle \mathbf{f}_n, \mathbf{g}_k \rangle = \delta(k-n) \tag{6.63}$$

for $1 \le k, n \le N$. Equivalently, $\langle \mathbf{g}_k\ \mathbf{f}_n \rangle = \delta(k-n)$.

Two sets of vectors, $\{\mathbf{f}_n\}$ and $\{\mathbf{g}_k\}$, satisfying Equation 6.63 are said to be **biorthogonal**. Because $\mathbf{c} = \mathbf{F}^{-1}\ \mathbf{x} = \mathbf{Gx}$ we can write the elements of $\mathbf{c}$ as $c_n = \mathbf{g}_n^\dagger \mathbf{x} = \langle \mathbf{x}, \mathbf{g}_n \rangle$. Then $\mathbf{x} = \sum_n c_n \mathbf{f}_n = \sum_n \langle \mathbf{x}, \mathbf{g}_n \rangle \mathbf{f}_n$. Next, $\mathbf{G}^\dagger$ is a nonsingular matrix, therefore, we can use its columns $\mathbf{g}_n$, instead of the columns of $\mathbf{F}$, to obtain a similar development, and express the arbitrary vector $\mathbf{x} \in \mathcal{C}^N$ as $\mathbf{x} = \boldsymbol{\Sigma}_n \langle \mathbf{x}, \mathbf{f}_n \rangle \mathbf{g}_n$. Summarizing, we have

$$\mathbf{x} = \sum_n \langle \mathbf{x}, \mathbf{g}_n \rangle \mathbf{f}_n = \sum_n \langle \mathbf{x}, \mathbf{f}_n \rangle \mathbf{g}_n \tag{6.64}$$

where the summations are for $1 \le n \le N$. By using the expressions $c_n = \langle \mathbf{x}, \mathbf{g}_n \rangle$ and $\mathbf{x} = \boldsymbol{\Sigma}_n c_n \mathbf{f}_n$, we can rearrange the inequality Equation 6.61 into $B^{-1}\ \|\mathbf{x}\|^2 \le \boldsymbol{\Sigma}_n\ |\langle \mathbf{x}, \mathbf{g}_n \rangle|^2 \le A^{-1} \|\mathbf{x}\|^2$. With the columns $\mathbf{g}_n$ of $\mathbf{G}^\dagger$, instead of the columns of $\mathbf{F}$, used as the basis for $\mathcal{C}^N$, we obtain similarly

$$A\|\mathbf{x}\|^2 \le \sum_n |\langle \mathbf{x}, \mathbf{f}_n \rangle|^2 \le B\|\mathbf{x}\|^2 \tag{6.65}$$

where $1 \le n \le N$, and $A = \lambda_m$, $B = \lambda_M$ again. Readers familiar with the idea of a frame in an infinite dimensional Hilbert space will recognize that the above inequality defines a frame $\{\mathbf{f}_n\}$.

Orthonormality

The basis $\mathbf{f}_n$ is said to be orthonormal if $\langle \mathbf{f}_k, \mathbf{f}_n \rangle = \delta(k-n)$, i.e., $\mathbf{f}_n^\dagger \mathbf{f}_k = \delta(k-n)$. Equivalently, $\mathbf{F}$ is unitary, i.e., $\mathbf{F}^\dagger\ \mathbf{F} = \mathrm{I}$. In this case the rows of the inverse matrix G are the quantities $\mathbf{f}_n{}^\dagger$. But $\mathbf{F}^\dagger\ \mathbf{F} = \mathbf{I}$, and we have $\lambda_m = \lambda_M = 1$ or $A = B = 1$. With this, Equation 6.60 becomes $\|\mathbf{c}\| = \|\mathbf{x}\|$, or $\boldsymbol{\Sigma}_n |c_n|^2 = \|\boldsymbol{\Sigma}_n c_n \mathbf{f}_n\|^2$. This shows that Equation 6.61 is a generalization of the orthonormal situation. Similarly, biorthogonality (Equation 6.63) is a generalization of orthonormality.

Basis in Infinite Dimensional Spaces

When the simple idea of a basis in a finite dimensional space (e.g., the Euclidean space $\mathcal{C}^N$) is extended to infinite dimensions, several new issues arise which make the problem nontrivial. Consider the sequence of functions $\{f_n\}$, $1 \le n \le \infty$ in a Hilbert space $\mathcal{H}$. Because of the infinite range of n, linear combinations of the form $\sum_{n=1}^{\infty} c_n f_n$ must now be considered. The problem that immediately arises is one of convergence. For arbitrary sequences c_n this sum does not converge, so the statement "all linear combinations" must be replaced with something else.*

* Our review uses $1 \le n \le \infty$ to be consistent with standard math texts, but all the crucial results hold for doubly infinite sequences and summations, i.e., for the case $-\infty \le n \le \infty$. This is what we need in the case of Fourier and wavelet bases; see, for example, Equations 6.3 and 6.4.

Closure of Span

First define the set of all finite linear of the form $\sum_{n=1}^{\infty} c_n f_n$, where N varies over all integers ≥ 1. This is called the span of $\{f_n\}$. Now suppose $\mathbf{x} \in \mathcal{H}$ is a vector not necessarily in the span of $\{f_n\}$, but can be approximated as closely as we wish by vectors in the span. In other words, given an $\varepsilon > 0$ we can find N and the sequence of constants c_{nN} such that

$$\left\| x - \sum_{n=1}^{N} c_{nN} f_n \right\| < \varepsilon \tag{6.66}$$

where $\|x\|$ is the norm defined as $\|x\| = \sqrt{\langle x, x \rangle}$. If we append all such vectors x to the span of $\{f_n\}$ we get the closure of the span of $\{f_n\}$.* Note that c_{nN}, in general, depends on ε because N depends on ε.

Completeness

A sequence of vectors $\{f_n\}$ is said to be complete in $\mathcal{H}$ if the closure of the linear span of $\{f_n\}$ equals $\mathcal{H}$. Therefore, any $x \in \mathcal{H}$ can be approximated, as closely as we wish, by finite linear combinations of f_n in the sense of Equation 6.66. This is also expressed by saying that the linear span of $\{f_n\}$ is dense in $\mathcal{H}$. Completeness of $\{f_n\}$ in a Hilbert space is equivalent to the statement that the only vector orthogonal to all f_n is the zero vector.

Infinite Summations

When we write $x = \sum_{n=1}^{\infty} c_n f_n$, we mean that the infinite summation converges to x in the norm of $\mathcal{H}$. In other words, given $\varepsilon > 0$, there exists n_0 such that

$$\left\| x - \sum_{n=1}^{N} c_n f_n \right\| < \varepsilon \quad \text{for all } N \geq n_0 \tag{6.67}$$

This statement is stronger than saying that x is in the closure of the linear span of $\{f_n\}$. The latter statement only requires Equation 6.66, where N and hence c_{nN} depends on ε. In Equation 6.67, $\{c_n\}$ is a fixed sequence.

Linear Independence

Let $\{f_n\}$, $n = 1, 2, \ldots$ be a sequence of vectors in an infinite dimensional Hilbert space $\mathcal{H}$. Unlike in a finite dimensional space, one must distinguish between several types of linear independence.

Type 1: $\{f_n\}$ has *finite linear independence if* $\sum_{n=1}^{N} c_n f_n = 0$, for any finite N implies $c_n = 0$, $1 \leq n \leq N$.

Type 2: $\{f_n\}$ is ω-*independent if* $\sum_{n=1}^{\infty} c_n f_n = 0$ implies $c_n = 0$ for all n (where the infinite sum is interpreted as explained above).

Type 3: $\{f_n\}$ is *minimal* if none of the f_m is in the closure of the span of the remaining set of f_n.

Type 3 independence implies type 2, which in turn implies type 1. Thus, type 3 is the strongest kind of linear independence. The reason that it is stronger than type 2 is that type 2 implies we cannot have $f_m = \mathbf{\Sigma}_{n \neq m} c_n f_n$. However, for a type 2 independent sequence $\{f_n\}$, it is possible that we can make

$$\left\| f_m - \sum_{\substack{n=1 \\ n \neq m}}^{N} c_{nN} f_n \right\| < \varepsilon \tag{6.68}$$

* The term "closure" has its origin from the theory of metric spaces, more generally, topological vector spaces. We do not require the deeper, more general meaning here.

for any given $\varepsilon > 0$ by choosing N and c_{nN} properly.* Type 3 linear independence prohibits even this. The comments following Example 6.3 make this distinction more clear (see p. 6–42 and 6–43).

Basis or Schauder Basis

A sequence of vectors $\{f_n\}$ in $\mathcal{H}$ is a Schauder basis for $\mathcal{H}$ if any $x \in \mathcal{H}$ can be expressed as $x = \sum_{n=1}^{\infty} c_n f_n$, and the sequence of scalars $\{c_n\}$ is unique for a given x. The second condition can be replaced with the statement that $\{f_n\}$ is ω-independent. A subtle result for Hilbert spaces [24] is that a Schauder basis automatically satisfies minimality (i.e., type 3 independence).

A Schauder basis is ω-independent and complete in the sense defined above. Conversely, ω-independence and completeness do not imply that $\{f_n\}$ is a Schauder basis; completeness only means that we may approximate any vector as closely as we wish in the sense of Equation 6.66, where c_{kN} depend on N. In this chapter "independence" (or linear independence) stands for ω-independence. Similarly "basis" stands for Schauder basis unless qualified otherwise.

Riesz Basis

Any basis $\{\mathbf{f}_n\}$ in a finite dimensional space satisfies Equation 6.61, which in turn ensures that the transformation from x to $\{c_n\}$ and that from $\{c_n\}$ to $\mathbf{x}$ are stable. For a basis in an infinite dimensional space, Equation 6.61 is not automatically guaranteed, as shown by the following example.

Example 6.2

Let $\{e_n\}$, $1 \le n \le \infty$ be an orthonormal basis in a Hilbert space $\mathcal{H}$, and define the sequence $\{f_n\}$ by $f_n = e_n/n$. Then we can show that f_n is still a basis, i.e., it satisfies the definition of a Schauder basis. Suppose we pick $x = \varepsilon e_k$ for some k. Then, $x = \sum_n c_n f_n$, with $c_k = \varepsilon k$, and $c_n = 0$ for all other n. Thus, $\sum_n |c_n|^2 = \varepsilon^2 k^2$ and grows as k increases, although $\|x\| = \varepsilon$ for all k. That is, a "small" error in x can become amplified in an unbounded manner. Recall that this could never happen in the finite dimensional case because $A > 0$ in Equation 6.61. For our basis $\{f_n\}$, we can indeed show that no $A > 0$ satisfies Equation 6.61. To see this let $c_n = 0$ for all n, except that $c_k = 1$. Then, $\sum_n c_n f_n = f_k = e_k/k$ and has norm $1/k$. So Equation 6.61 reads $A \le 1/k^2 \le B$ for all $k \ge 1$. This is not possible with $A > 0$.

If $\{e_n\}$, $1 \le n \le \infty$ is an orthonormal basis in an infinite dimensional Hilbert space $\mathcal{H}$, then any vector $x \in \mathcal{H}$ can be expressed uniquely as $x = \sum_{n=1}^{\infty} c_n e_n$ where

$$\|x\|^2 = \sum_{n=1}^{\infty} |c_n|^2$$

This property automatically ensures the stability of the transformations from x to $\{c_n\}$ and vice versa. The Riesz basis is defined such that this property is made more general.†

Definition of a Riesz Basis: A sequence $\{f_n\}$, $1 \le n \le \infty$ in a Hilbert space $\mathcal{H}$ is a Riesz basis if it is *complete* and constants A and B exist such that $0 < A \le B < \infty$ and

$$A \sum_{n=1}^{\infty} |c_n|^2 \le \left\| \sum_{n=1}^{\infty} c_n f_n \right\|^2 \le B \sum_{n=1}^{\infty} |c_n|^2 \tag{6.69}$$

for all choice of c_n satisfying $\sum_n |c_n|^2 < \infty$.

---

* As we make ε increasingly smaller, we may need to change N and *all coefficients* c_{kN}. Therefore, this does not imply $f_m = \Sigma_{n \ne m} c_n f_n$ for fixed $\{c_n\}$.

† For readers familiar with bounded linear transformations in Hilbert spaces, we state that a basis is a Riesz basis iff it is related on an orthonormal basis via a bounded linear transformation with bounded inverse.

In a finite dimensional Hilbert space, A and B come from the extreme eigenvalues of a nonsingular matrix $\mathbf{F}^{\dagger}\mathbf{F}$, so automatically $A > 0$ and $B < \infty$. In other words, any basis in a finite dimensional space is a Riesz basis. As Example 6.2 shows, this may not be the case in infinite dimensions.

Unconditional Basis

It can be shown that the Riesz basis is an unconditional basis, that is, any reordering of $\{f_n\}$ is also a basis (and the new c_n are the correspondingly reordered versions). This is a nontrivial statement; an arbitrary (Schauder) basis is not necessarily unconditional. In fact, the space of L^1 functions (which is a Banach space, not a Hilbert space) does not have an unconditional basis.

Role of the Constants A and B

1. *Strongest linear independence*: The condition $A > 0$ means, in particular, that $\mathbf{\Sigma}_n c_n f_n \neq 0$ unless c_n is zero for all n. This is just ω-independence. Actually the condition $A > 0$ means that the vectors $\{f_n\}$ are independent in the strongest sense (type 3), that is, $\{f_n\}$ is minimal. To see this, assume this is not the case by supposing some vector f_m is in the closure of the span of the others. Then, given arbitrary $\varepsilon > 0$ we can find N and c_{nN} satisfying Equation 6.66 with $x = f_m$. Defining $c_n = -c_{nN}$ for $n \neq m$ and $c_m = 1$, Equation 6.69 implies $A(1 + \mathbf{\Sigma}_{n \neq m} |c_{nN}|^2) \leq \varepsilon^2$. Because ε is arbitrary, this is not possible for $A > 0$.
2. *Distance between vectors:* The condition $A > 0$ also implies that no two vectors in $\{f_n\}$ can become "arbitrarily close." To see this, choose $c_k = -c_m = 1$ for some k, m and $c_n = 0$ for all other n. Then Equation 6.69 gives $2A \leq \|f_k - f_m\|^2 \leq 2B$. Thus, the distance between any two vectors is at least $\sqrt{2A}$, at most $\sqrt{2B}$.
3. *Bounded basis:* A Riesz basis is a bounded basis in the sense that $\|f_n\|$ cannot get arbitrarily large. In fact, by choosing $c_n = 0$ for all but one value of n, we can see that $0 < A \leq \|f_n\|^2 \leq B < \infty$. That is, the norms of the vectors in the basis cannot become arbitrarily small or large. Note that the basis in Example 6.2 violates this, because $\|f_n\| = 1/n$. Therefore, Example 6.2 is only a Schauder basis and not a Riesz basis.
4. *Stability of basis*. The condition $A > 0$ yields $\mathbf{\Sigma}_n |c_n|^2 \leq A^{-1} \|x\|^2$, where $x = \mathbf{\Sigma}_n c_n f_n$. This means that the transformation from the vector x to the sequence $\{c_n\}$ is bounded, so a small error in x is not amplified in an unbounded manner. Similarly, the inequality $\|x\|^2 \leq B \mathbf{\Sigma}_n |c_n|^2$ shows that role of B is to ensure that the inverse transformation from c_n to x is bounded. Summarizing, the transformation from x to $\{c_n\}$ is numerically stable (i.e., small errors not severely amplified) because $A > 0$, and the reconstruction of x from $\{c_n\}$ is numerically stable because $B < \infty$.
5. *Orthonormality:* For a Riesz basis with $A = B = 1$ the condition (Equation 6.69) reduces to $\mathbf{\Sigma}_n |c_n|^2 = \|\mathbf{\Sigma}_n c_n f_n\|^2$. It can be shown that such a Riesz is as simply an orthonormal basis. The properties listed above show that the Riesz basis is as good as orthonormal basis in most applications. Any Riesz basis can be obtained from an orthonormal basis by means of a bounded linear transformation with a bounded linear inverse.

Example 6.3: Mishaps with a System That Is Not a Riesz Basis

Let us modify Example 6.2 to $f_n = (e_n/n) + e_1$, $n \geq 1$, where $\{e_n\}$ is an orthonormal basis. As $n \to \infty$ the vectors f_n move arbitrarily closer together (although $\|f_n\|$ approaches unity from above). Formally, $f_n - f_m = (e_n/n) - (e_m/m)$, so $\|f_n - f_m\|^2 = (1/n^2) + (1/m^2)$, which goes to zero as n, $m \to \infty$. Thus there does not exist $A > 0$ satisfying Equation 6.69 (because of comment 2 above). This, then, is not a Riesz basis; in fact, this is not even a Schauder basis (see below). This example also has $B = \infty$. To see this let $c_n = 1/n$, then $\mathbf{\Sigma}_n |c_n|^2$ converges, but $\|\mathbf{\Sigma}_{n=1}^{N} c_n f_n\|^2$ does not converge as $N \to \infty$ (as we can verify), so Equation 6.69 is not satisfied for finite B. Such mishaps cannot occur with a Riesz basis.

In this example, $\{f_n\}$ is not minimal (which is type 3 independence). Note that $|f_1 - f_n|$ gets arbitrarily small as n increases to infinity, therefore, f_1 is in the closure of the span of $\{f_n\}$, $n \neq 1$. However, $\{f_n\}$ is ω-independent; no sequence $\{c_n\}$ exists such that $\|\mathbf{\Sigma}_{n=1}^{N} c_n f_n\| \to 0$ as $N \to \infty$. In

any case, the fact that $\{f_n\}$ is not minimal (i.e., not independent in the strongest sense) shows that it is not even a Schauder basis.

6.7.2 Biorthogonal Systems, Riesz Bases, and Inner Products

When discussing finite dimensional Hilbert spaces, we found that given a basis $\mathbf{f}_n$ (columns of a nonsingular matrix) we can express any vector $\mathbf{x}$ as a linear combination $\mathbf{x} = \boldsymbol{\Sigma}_n \langle \mathbf{x}, \mathbf{g}_n \rangle \mathbf{f}_n$, where $\mathbf{g}_n$ is such that the biorthogonality property $\langle \mathbf{f}_m, \mathbf{g}_n \rangle = \delta(m-n)$ holds. A similar result is true for infinite dimensional Hilbert spaces.

THEOREM 6.4: Biorthogonality and Riesz Basis

Let $\{f_n\}$ be a basis in a Hilbert space $\mathcal{H}$. Then, there exists a unique sequence $\{g_n\}$ biorthogonal to $\{f_n\}$, that is,

$$\langle f_m, g_n \rangle = \delta(m-n) \quad \text{(biorthogonality)} \tag{6.70}$$

Moreover, the unique expansion of any $x \in \mathcal{H}$ in terms of the basis $\{f_n\}$ is given by

$$x = \sum_{n=1}^{\infty} \langle x, g_n \rangle f_n \tag{6.71}$$

It is also true that the biorthogonal sequence $\{g_n\}$ is a basis and that $x = \boldsymbol{\Sigma}_{n=1}^{\infty} \langle x, f_n \rangle g_n$. Moreover, if $\{f_n\}$ is a Riesz basis, then $\boldsymbol{\Sigma}_n |\langle x, g_n \rangle|^2$ and $\boldsymbol{\Sigma}_n |\langle x, f_n \rangle|^2$ are finite, and we have

$$A\|x\|^2 \leq \sum_{n=1}^{\infty} |\langle x, f_n \rangle|^2 \leq B\|x\|^2 \tag{6.72}$$

where A and B are the same constants as in the definition (Equation 6.69) of a Riesz basis.

This beautiful result resembles the finite dimensional version, where f_n corresponds to the column of a matrix and g_n corresponds to the rows (conjugated) of the inverse matrix. In this sense we can regard the biorthogonal pair of sequences $\{f_n\}$, $\{g_n\}$ as inverses of each other. Both are bases for $\mathcal{H}$. A proof of the above result can be obtained by combining the ideas on pages 28–32 of Ref. [24]. The theorem implies, in particular, that if $\{f_n\}$ is a Riesz basis, then any vector in the space can be written in the form $\boldsymbol{\Sigma}_{n=1}^{\infty} c_n f_n$, where $c_n \in l^2$.

Summary of Riesz Basis

The Riesz basis $\{f_n\}$ in a Hilbert space is a complete set of vectors, linearly independent in the strongest sense (i.e., type 3 or minimal). It is a bounded basis with bounded inverse. Any two vectors are separated by at least $\sqrt{2A}$, that is, $\|f_n - f_m\|^2 \geq 2A$. The norm of each basis vector is bounded as $\|f_n\| \leq \sqrt{B}$. In the expression $x = \sum_n c_n f_n$ the computation of x from c_n as well as the computation of c_n from x are numerically stable because $B < \infty$ and $A > 0$, respectively. A Riesz basis with $A = B = 1$ is an orthonormal basis. In fact, any Riesz basis can be obtained from an orthonormal basis via a bounded linear transformation with a bounded inverse. Given any basis $\{f_n\}$ in a Hilbert space, a unique biorthogonal sequence $\{g_n\}$ exists such that we can express any $x \in \mathcal{H}$ as $x = \boldsymbol{\Sigma}_{n=1}^{\infty} \langle x, g_n \rangle f_n$ as well as $x = \boldsymbol{\Sigma}_{n=1}^{\infty} \langle x, f_n \rangle g_n$; if this basis is also a Riesz basis, then $\boldsymbol{\Sigma}_n |\langle x, f_n \rangle|^2$ and $\boldsymbol{\Sigma}_n |\langle x, g_n \rangle|^2$ are finite. If $\{f_n\}$ is a Riesz basis, then any vector $x \in \mathcal{H}$ can be written in the form $x = \boldsymbol{\Sigma}_{n=1}^{\infty} c_n f_n$, where $c_n \in l^2$.

6.8 Frames in Hilbert Spaces

A frame in a Hilbert space $\mathcal{H}$ is a sequence of vectors $\{f_n\}$ with certain special properties. While a frame is not necessarily a basis, it shares some properties of a basis. For example, we can express any vector $x \in \mathcal{H}$

as a linear combination of the frame elements, i.e., $x = \Sigma_n c_n f_n$. However, frames generally have redundancy—the frame vectors are not necessarily linearly independent, even in the weakest sense defined in Section 6.7. The Riesz basis (hence, any orthonormal basis) is a special case of frames. The concept of a frame is useful when discussing the relation between wavelets, STFTs, and filter banks. Frames were introduced by Duffin and Schaeffer [25], and used in the context of wavelets and STFT by Daubechies [5]. Excellent tutorials can be found in Refs. [12,24].

6.8.1 Definition of a Frame

A sequence of vectors $\{f_n\}$ in a (possibly infinite dimensional) Hilbert space $x \in \mathcal{H}$ is a frame if there exist constants A and B with $0 < A \leq B < \infty$ such that for any $x \in \mathcal{H}$ we have

$$A\|x\|^2 \leq \sum_{n=1}^{\infty} |\langle x, f_n \rangle|^2 \leq B\|x\|^2 \tag{6.73}$$

The constants A and B are called frame bounds.

In Section 6.7, we saw that a Riesz basis, which by definition satisfies Equation 6.69 and also satisfies Equation 6.72, which is precisely the frame definition. A Riesz basis is, therefore, also a frame, but it is a special case of a frame, where the set of vectors is minimal.

Any frame is complete. That is, if a vector $x \in \mathcal{H}$ is orthogonal to all elements in $\{f_n\}$, then $x = 0$, otherwise $A > 0$ is violated. Thus, any $x \in \mathcal{H}$ is in the closure of the span of the frame. In fact, we will see that more is true; for example, we can express $x = \Sigma c_n f_n$, although $\{c_n\}$ may not be unique. The frame elements are not necessarily linearly independent, as demonstrated by examples below. A frame, then, is not necessarily a basis. Compare Equation 6.73 with the Riesz basis definition (Equation 6.69), where the left inequality forced the vectors f_n to be linearly independent (in fact, minimal). The left inequality for a frame only ensures completeness, not linear independence.

6.8.2 Representing Arbitrary Vectors in Terms of Frame Elements

We will see that, given a frame $\{f_n\}$ we can associate with it another sequence $\{g_n\}$ called the dual frame, such that any element $x \in \mathcal{H}$ can be represented as $x = \Sigma_{n=1}^{\infty} \langle x, g_n \rangle f_n$. We also can write $x = \Sigma_{n=1}^{\infty} \langle x, f_n \rangle g_n$. This representation in terms of $\{f_n\}$ and $\{g_n\}$ resembles the biorthogonal system discussed in Section 6.7, but some differences are pointed out later.

Stability of Computations

To obtain the representation $x = \sum_{n=1}^{\infty} \langle x, f_n \rangle g_n$ we compute (at least conceptually) the coefficients $\langle x, f_n \rangle$ for all n. This computation is a linear transformation from $\mathcal{H}$ to the space of sequences. The inverse transform computes x from this sequence by using the formula $x = \sum_{n=1}^{\infty} \langle x, f_n \rangle g_n$. The condition $B < \infty$ in the frame definition ensures that the transformation from x to $\langle x, f_n \rangle$ is bounded. Similarly, the condition $A > 0$ ensures that the inverse transformation, from $\langle x, f_n \rangle$ to x is bounded. The conditions $A > 0$ and $B < \infty$, therefore, ensure stability; small errors in one domain are not arbitrarily amplified in the other domain. A similar advantage was pointed out earlier for the Riesz basis—for arbitrary bases in infinite dimensional spaces such an advantage cannot be claimed (Example 6.4).

If we wish to use the dual representation $x = \sum_{n=1}^{\infty} \langle x, g_n \rangle f_n$ instead of $x = \sum_{n=1}^{\infty} \langle x, f_n \rangle g_n$, we must compute $\langle x, g_n \rangle$, etc.; the roles of A and B are taken up by $1/B$ and $1/A$, respectively, and similar discussions hold. This is summarized in Figure 6.29.

6.8.3 Exact Frames, Tight Frames, Riesz Bases, and Orthonormal Bases

The resemblance between a Riesz basis and a frame is striking. Compare Equation 6.69 with Equation 6.73. One might wonder what the precise relation is. Thus far, we know that a Riesz basis is a frame.

To go deeper, we need a definition: a frame $\{f_n\}$ which ceases to be a frame if any element f_k is deleted is said to be an exact frame. Such a frame has no redundancy. A frame with $A = B$ is said to be a tight frame. The defining property reduces to $\|x\|^2 = A^{-1}\,\mathbf{\Sigma}_n |\langle x, f_n\rangle|^2$, resembling Parseval's theorem for an orthonormal basis. A frame is *normalized* if $\|f_n\| = 1$ for all n. The following facts concerning exact frames and tight frames are fundamental:

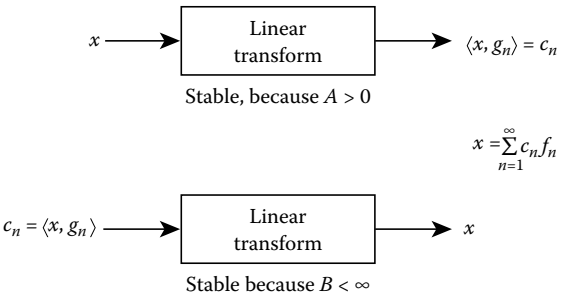


FIGURE 6.29 Representation of x using frame elements $\{f_n\}$. The transformation from x to $\{c_n\}$ and vice versa are stable.

1. A tight frame with $A = B = 1$ and $\|f_n\| = 1$ for all n (i.e., a normalized tight frame with frame bound = 1) is an orthonormal basis [5].
2. $\{f_n\}$ is an exact frame iff it is a Riesz basis [24]. Moreover, if a frame is not exact then it cannot be a basis [12]. *Thus, if a frame is a basis, it is certainly a Riesz basis.*
3. Because an orthonormal basis is a Riesz basis, a normalized tight frame with frame bound equal to 1 is automatically an exact frame.

Some examples follow that serve to clarify the preceding concepts and definitions. In these examples, the sequence $\{e_n\}$, $n \geq 1$ is an orthonormal basis for $\mathcal{H}$. Thus, $\{e_n\}$ is a tight frame with $A = B = 1$, and $\|e_n\| = 1$.

Example 6.4

Let $f_n = e_n/n$ as in Example 6.2. Then $\{f_n\}$ is still a (Schauder) basis for $\mathcal{H}$, but it is not a frame. In fact, this satisfies Equation 6.73 only with $A = 0$; i.e., the inverse transformation (reconstruction) from $\langle x, f_n\rangle$ to x is not bounded. To see why $A = 0$, note that if we let $x = e_k$ for some $k > 0$ then $\|x\| = 1$, whereas $\mathbf{\Sigma}_n |\langle x, f_n\rangle|^2 = 1/k^2$. The first inequality in the frame definition becomes $A \leq 1/k^2$, which cannot be satisfied for all k unless $A = 0$. In this example a finite B works because $|\langle x, f_n \rangle| = |\langle x, e_n \rangle|/n$ for each n. Therefore, $\mathbf{\Sigma}_n |\langle x, f_n\rangle|^2 \leq \mathbf{\Sigma}_n |\langle x, e_n\rangle|^2 = \|x\|^2$.

Example 6.5

Suppose we modify the above example as follows: define $f_n = (e_n/n) + e_1$. We know that this is no longer a basis (Example 3). We now have $B = \infty$ in the frame definition, so this is not a frame. To verify this, let $x = e_1$ so $\|x\| = 1$. Then $\langle x, f_n\rangle = 1$ for all $n > 1$, so $\mathbf{\Sigma}_n |\langle x, f_n \rangle|^2$ does not converge to a finite value.

Example 6.6

Consider the sequence of vectors $\{e_1, e_1, e_2, e_2, \ldots\}$. This is a tight frame with frame bounds $A = B = 2$. Note that, even though the vectors are normalized and the frame is tight, this is not an orthonormal basis. This has a redundancy of two in the sense that each vector is repeated twice. This frame is not even a basis, therefore, it is not a Riesz basis.

Example 6.7

Consider the sequence of vectors $\{e_1, (e_2/\sqrt{2}), (e_2/\sqrt{2}), (e_3/\sqrt{3}), (e_3/\sqrt{3}), (e_3/\sqrt{3}), \ldots\}$. Again, redundancy occurs, so it is not a basis. It is a tight frame with $A = B = 1$, but not an exact frame, and clearly not a basis. It has redundancy (repeated vectors).

Frame Bounds and Redundancy

For a tight frame with unit norm vectors f_n, the frame bound measures the redundancy. In Example 6.6 the redundancy is two (every vector repeated twice), and indeed $A = B = 2$. In Example 6.7, where we still have redundancy, the frame bound $A = B = 1$ does not indicate it. The frame bound of a tight frame measures redundancy only if the vectors f_n have unit norm as in Example 6.6.

6.8.4 Frame Operator, Dual Frame, and Biorthogonality

The frame operator $\mathcal{F}$ associated with a frame $\{f_n\}$ in a Hilbert space $\mathcal{H}$ is a linear operator defined as

$$\mathcal{F}x = \sum_{n=1}^{\infty} \langle x, f_n \rangle f_n \tag{6.74}$$

The summation can be shown to be convergent by using the definition of the frame. The frame operator $\mathcal{F}$ takes a vector $x \in \mathcal{H}$ and produces another vector in $\mathcal{H}$. The norm of $\mathcal{F}$ x is bounded as follows:

$$A\|x\| \leq \|\mathcal{F}x\| \leq B\|x\| \tag{6.75}$$

The frame operator is a bounded linear operator (because $B < \infty$; hence, it is a continuous operator [12]. Its inverse is also a bounded linear operator because $A > 0$).

From Equation 6.74 we obtain $\langle \mathcal{F}x, x \rangle = \Sigma_n |\langle x, f_n \rangle|^2$ by interchanging the inner product with the infinite summation. This is permitted by the continuity of the operator $\mathcal{F}$ and the continuity of inner products (Section 6.6). Because $\{f_n\}$ is complete, the right-hand side is positive for $x \neq 0$. Thus, $\langle \mathcal{F}x, x \rangle > 0$ unless $x = 0$, that is, $\mathcal{F}$ is a positive definite operator. The realness of $\langle \mathcal{F}x, x \rangle$ also means that $\mathcal{F}$ is self-adjoint, or $\langle \mathcal{F}x, y \rangle = \langle x, \mathcal{F}y \rangle$ for any $x, y \in \mathcal{H}$.

The importance of the frame operator arises from the fact that if we define $g_n = \mathcal{F}^1 f_n$, any $x \in \mathcal{H}$ can be expressed as

$$x = \sum_{n=1}^{\infty} \langle x, g_n \rangle f_n = \sum_{n=1}^{\infty} \langle x, f_n \rangle g_n \tag{6.76}$$

The sequence $\{g_n\}$ is itself a frame in $\mathcal{H}$ called the dual frame. It has frame bounds B^{-1} and A^{-1}. Among all representations of the form $x = \Sigma_n c_n f_n$, the representation $x = \Sigma_n \langle x, g_n \rangle f_n$ possesses the special property that the energy of the coefficients is minimized, i.e., $\Sigma_n |\langle x, g_n \rangle|^2 \leq \Sigma_n |c_n|^2$ with equality iff $c_n = \langle x, g_n \rangle$ for all n [12]. As argued earlier, the computation of $\langle x, f_n \rangle$ from x and the inverse computation of x from $\langle x, f_n \rangle$ are numerically stable operations because $B < \infty$ and $A > 0$, respectively.

For the special case of a tight frame ($A = B$), the frame operator is particularly simple. We have $\mathcal{F}x = Ax$, and so $g_n = \mathcal{F}^{-1} f_n = f_n / A$. Any vector $x \in \mathcal{H}$ can be expressed as

$$x = \frac{1}{A} \sum_{n=1}^{\infty} \langle x, f_n \rangle f_n \quad \text{(tight frames)} \tag{6.77}$$

Notice also that Equation 6.73 gives

$$\sum_{n=1}^{\infty} |\langle x, f_n \rangle|^2 = A\|x\|^2 \quad \text{(tight frames)} \tag{6.78}$$

For a tight frame with $A = 1$, these equations resemble the representation of x using an orthonormal basis, even though such a tight frame is not necessarily a basis because of possible redundancy (Example 6.7).

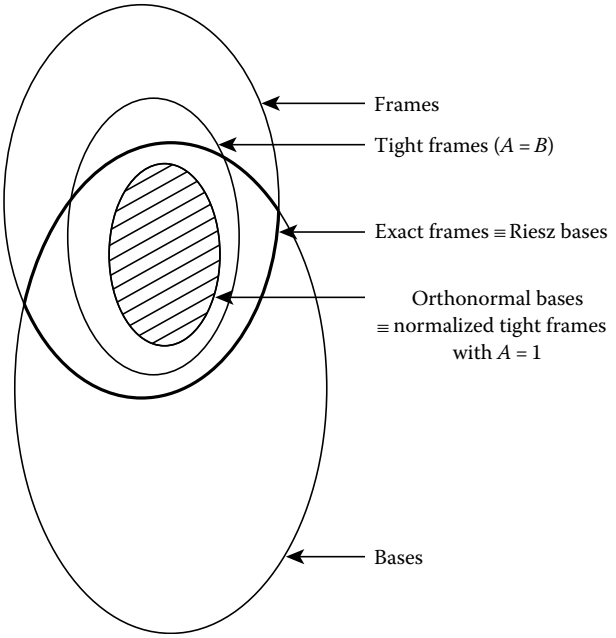


FIGURE 6.30 Venn diagram showing the relation between frames and bases in a Hilbert space.

Exact Frames and Biorthogonality

For the special case of an exact frame (i.e., a Riesz basis) the sequence $\{f_n\}$ is minimal, and it is biorthogonal to the dual frame sequence $\{g_n\}$. This is consistent with our observation at the end of Section 6.7.

Summary of Frames

A sequence of vectors $\{f_n\}$ in a Hilbert space $\mathcal{H}$ is a frame if there exist constants $A > 0$ and $B < \infty$ such that Equation 6.73 holds for every vector $x \in \mathcal{H}$. Frames are complete (because $A > 0$), but not necessarily linearly independent. The constants A and B are called the frame bounds. A frame is *tight* if $A = B$. A tight frame with $A = B = 1$ and with normalized vectors ($\|f_n\| = 1$) is an orthonormal basis. For a tight frame with $\|f_n\| = 1$, the frame bound A measures redundancy. Any vector $x \in \mathcal{H}$ can be expressed in either of the two ways shown in Equation 6.76. Here, $g_n = \mathcal{F}^{-1} f_n$, where $\mathcal{F}$ is the frame *operator* defined in Equation 6.74. The frame operator is a bounded linear operator and is self-adjoint (in fact, positive). The sequence $\{g_n\}$ is the dual frame and has frame bounds B^{-1} and A^{-1}. For a tight frame the frame representation reduces to Equation 6.77. A frame is *exact* if deletion of any vector f_m destroys the frame property. A sequence $\{f_n\}$ is an exact frame iff it is a Riesz basis. An exact frame $\{f_n\}$ is biorthogonal to the dual frame $\{g_n\}$.

Figure 6.30 is a Venn diagram, which shows the classification of frames and bases and the relationship between these.

6.9 Short-Time Fourier Transform: Invertibility, Orthonormality, and Localization

In Section 6.8, we saw that a vector x in an infinite dimensional Hilbert space (e.g., a function $x(t)$ in L^2) can be expanded in terms of a sequence of vectors $\{f_n\}$ called a frame, that is $x = \sum_{n=1}^{\infty} \langle x, g_n \rangle f_n$. One of the most important features of frames is that the construction of the expansion coefficients $\langle x, g_n \rangle$ from x as well as the reconstruction of x from these coefficients are numerically stable operations

because $A > 0$ and $B < \infty$ (Section 6.8). Riesz and orthonormal bases, which are special cases of a frame (Figure 6.30), also share this numerical stability.

In Section 6.3, we attempted to represent an L^2 function in terms of the STFT. The STFT coefficients are constructed using the integral (Equation 6.24). Denote for simplicity

$$g_{kn}(t) = v^*(t - nT_s)e^{jk\omega_s t} \tag{6.79}$$

The computation of the STFT coefficients can be written as

$$X_{\text{STFT}}(k\omega_s, nT_s) = \langle x(t), g_{kn}(t)\rangle \tag{6.80}$$

This is a linear transformation which converts $x(t)$ into a two-dimensional sequence because k and n are integers. Our hope is to be able to reconstruct $x(t)$ using an inverse linear transformation (inverse STFT) of the form

$$x(t) = \sum_{k=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} X_{\text{STFT}}(k\omega_s, nT_s)f_{kn}(t) \tag{6.81}$$

We know that this can be done in a numerically stable manner if $\{g_{kn}(t)\}$ is a frame in L^2 and $\{f_{kn}(t)\}$ is the dual frame. The fundamental questions, then, are under what conditions does $\{g_{kn}(t)\}$ constitute a frame? Under what further conditions does this become a Riesz basis, better still, an orthonormal basis? With such conditions, what are the time–frequency localization properties of the resulting STFT? The answers depend on the window $v(t)$ and the sample spacings ω_s and T_s.

We first construct a very simple example which shows the existence of orthonormal STFT bases, and indicate a fundamental disadvantage in the example. We then state the answers to the above general questions without proof. Details can be found in Refs. [5,12,16].

Example 6.8: Orthonormal Short-Time Fourier Transform Basis

Suppose $v(t)$ is the rectangular window shown in Figure 6.31, applied to an L^2 function $x(t)$. The product $x(t)v(t)$ therefore has finite duration. If we sample its FT at the rate $\omega_s = 2\pi$ we can recover $x(t)v(t)$ from these samples (this is like a Fourier series of the finite duration waveform $x(t)v(t)$). Shifting the window by successive integers (i.e., $T_s = 1$), we can in this way recover successive pieces of $x(t)$ from the STFT, with sample spacing $\omega_s = 2\pi$ in the frequency domain. Thus, the choice $T_s = 1$ and $\omega_s = 2\pi$ (so, $\omega_s T_s = 2\pi$) leads to an STFT $X_{\text{STFT}}(k\omega_s, nT_s)$, from which we can reconstruct $x(t)$ for all t. The quantity $g_{kn}(t)$ becomes

$$g_{kn}(t) = v(t - n)e^{jk\omega_s t} = v(t - n)e^{j2\pi kt} \tag{6.82}$$

Because the successive shifts of the window do not overlap, the functions $g_{kn}(t)$ are orthonormal for different values of n. The functions are also orthonormal for different values of k. Summarizing, the rectangular window of Figure 6.31, with the time–frequency sampling durations $T_s = 1$ and $\omega_s = 2\pi$, produces an orthonormal STFT basis for L^2 functions.

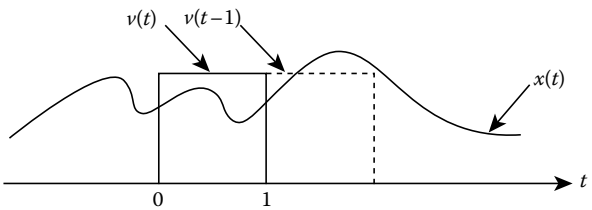


FIGURE 6.31 Rectangular window in STFT.

This example is reminiscent of the Nyquist sampling theorem in the sense that we can reconstruct $x(t)$ from (time–frequency) samples, but the differences are that $x(t)$ is an L^2 signal, not necessarily

band-limited. Note that T_s and ω_s cannot be arbitrarily interchanged (even if $\omega_s T_s = 2\pi$ is preserved). Thus, if we had chosen $T_s = 2$ and $\omega_s = \pi$ (preserving the product $\omega_s T_s$) we would not have obtained a basis because two successive positions of the window would be spaced too far apart and we would miss 50% of the signal $x(t)$.

6.9.1 Time–Frequency Sampling Density for Frames and Orthonormal Bases

Let us assume that $v(t)$ is normalized to have unit energy, i.e., $\int |v(t)|^2 \, dt = 1$ so that $\| g_{kn}(t) \| = 1$ for all k, n. If we impose the condition that $g_{kn}(t)$ be a frame, then it can be shown that the frame bounds satisfy the condition

$$A \leq \frac{2\pi}{\omega_s T_s} \leq B \tag{6.83}$$

regardless of how $v(t)$ is chosen. As an orthonormal basis is a tight frame with $A = B = 1$, an STFT orthonormal basis must therefore have $\omega_s T_s = 2\pi$.

It can further be shown that if $\omega_s T_s > 2\pi$, $\{g_{kn}(t)\}$ cannot be a frame. For $\omega_s T_s < 2\pi$, we can find frames (but not orthonormal basis) by appropriate choice of window $v(t)$. The critical time–frequency sampling density is $(\omega_s T_s)^{-1} = (2\pi)^{-1}$. If the density is smaller we cannot have frames, and if it is larger we cannot have orthonormal basis, only frames.

6.9.1.1 Orthonormal Short-Time Fourier Transform Bases Have Poor Time–Frequency Localization

If we wish to have an orthonormal STFT basis, the time–frequency density is constrained so that $\omega_s T_s = 2\pi$. Under this condition suppose we choose $v(t)$ appropriately to design such a basis. The time–frequency localization properties of this system can be judged by computing the mean square durations D_t^2 and D_f^2 defined in Equation 6.27 It has been shown by Balian and Low [5,16] that one of these is necessarily infinite no matter how $v(t)$ is designed. Thus, an orthonormal STFT basis always satisfies $D_t D_f = \infty$. That is, either the time localization or the frequency resolution is very poor. This is summarized in the following theorem.

THEOREM 6.5

Let the window $v(t)$ be such that $\{g_{kn}(t)\}$ in Equation 6.79 is an orthonormal basis for L^2 (which means, in particular, that $\omega_s T_s = 2\pi$). Define the RMS durations D_t and D_f for the window $v(t)$ as usual (Equation 6.27). Then, either $D_t = \infty$ or $D_f = \infty$.

Return now to Example 6.8, where we constructed an orthonormal STFT basis using the rectangular window of Figure 6.31. Here, $T_s = 1$ and $\omega_s = 2\pi$ so that $\omega_s T_s = 2\pi$. The window $v(t)$ has finite mean square duration D_t^2. Its FT $V(\omega)$ has magnitude $|V(\omega)| = |\sin(\omega/2)/(\omega/2)|$ so that $\int \omega^2 |V(\omega)|^2 \, d\omega$ is not finite. This demonstrates the result of Theorem 6.5. One can try to replace the window $v(t)$ with something for which $D_t D_f$ is finite, but this cannot be done without violating orthonormality.

6.9.1.2 Instability of the Gabor Transform

Gabor constructed the STFT using the Gaussian window $v(t) = ce^{-t^2/2}$. In this case the sequence of functions $\{g_{kn}(t)\}$ can be shown to be complete in L^2 (in the sense defined in Section 6.7) as long as $\omega_s T_s \leq 2\pi$. However, if $\omega_s T_s = 2\pi$ the system is not a frame because it can be shown that $A = 0$ in Equation 6.73. Thus, the reconstruction of $x(t)$ from $X_{STFT}(k\omega_s, nT_s)$ is unstable if $\omega_s T_s = 2\pi$ (Section 6.8), even though $\{g_{kn}(t)\}$ is complete. Although the Gabor transform has the ideal time–frequency

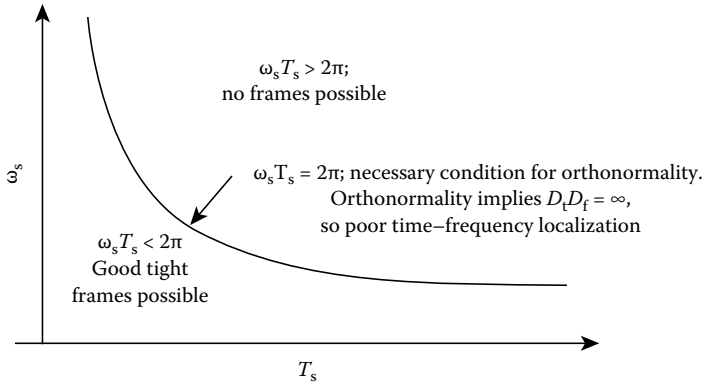


FIGURE 6.32 Behavior of STFT representation for various regions of time–frequency sampling product $\omega_s T_s$. The curve $\omega_s T_s = 2\pi$ is critical; see text.

localization (minimum D_t D_f), it cannot provide a *stable basis*; hence, it is certainly not an orthonormal basis, whenever $\omega_s T_s = 2\pi$.

Because orthonormal STFT basis is not possible if $\omega_s T_s \neq 2\pi$, this shows that an orthonormal basis can never be achieved with the Gabor transform (Gaussian windowed STFT), no matter how we choose ω_s and T_s. The Gabor example also demonstrates the fact that even if we successfully construct a complete set of functions (not necessarily a basis) to represent $x(t)$, it may not be useful because of the instability of reconstruction. If we construct Riesz bases (e.g., orthonormal bases) or more generally frames, this disadvantage disappears. For example, with the Gabor transform if we let $\omega_s T_s < 2\pi$ then all is well. We obtain a frame (so $A > 0$ and $B < \infty$ in Equation 6.73); we have stable reconstruction and good time–frequency localization, but not orthonormality. Figure 6.32 summarizes these results pertaining to the time–frequency product $\omega_s T_s$ in the STFT.

A major advantage of the WT over the STFT is that it is free from the above difficulties. For example, we can obtain an orthonormal basis for L^2 with excellent time–frequency localization (finite, controllable $D_t D_f$). We will also see how to constrain such a wavelet $\psi(t)$ to have the additional property of regularity or smoothness. Regularity is a property which is measured by the continuity and differentiability of $\psi(t)$. More precisely, it is quantified by the Hölder index (defined in Section 6.13). In the next few sections where we construct wavelets based on paraunitary filter banks, we will see how to achieve all this systematically.

6.10 Wavelets and Multiresolution

Sections 6.11 through 6.13 show how to construct compactly supported wavelets systematically to obtain orthonormal bases for L^2. The construction is such that excellent time–frequency localization is possible. Moreover, the smoothness or regularity of the wavelets can be controlled. The construction is based on the two-channel paraunitary filter bank described in Section 6.4. In that section, we denoted the synthesis filters as $G_s(z)$ and $H_s(z)$, with impulse response $g_s(n)$ and $h_s(n)$, respectively.

All constructions are based on obtaining the wavelet $\psi(t)$ and an auxiliary function $\phi(t)$, called the scaling function, from the impulse response sequences $g_s(n)$ and $h_s(n)$. We do this by using time domain recursions of the form

$$\phi(t) = 2\sum_{n=-\infty}^{\infty} g_s(n)\phi(2t-n), \quad \psi(t) = 2\sum_{n=-\infty}^{\infty} h_s(n)\phi(2t-n) \tag{6.84}$$

called dilation equations. Equivalently, in the frequency domain

$$\Phi(\omega) = G_s(e^{j\omega/2})\Phi(\omega/2), \quad \Psi(\omega) = H_s(e^{j\omega/2})\Phi(\omega/2) \tag{6.85}$$

If $\{G_s(z), H_s(z)\}$ is a paraunitary pair with further mild conditions (e.g., that the low-pass filter $G_s(e^{j\omega})$ has a zero at π and no zeroes in $[0, \pi/3]$) the preceding recursions can be solved to obtain $\psi(t)$, which gives rise to an orthonormal wavelet basis $\{2^{k/2}\psi(2^k t - n)\}$ for L^2. By constraining $G_s(e^{j\omega})$ to have a sufficient number of zeroes at π we can further control the Hölder index (or regularity) of $\psi(t)$ (Section 6.13).

Our immediate aim is to explain the occurrence of the function $\phi(t)$, and the curious recursions (Equation 6.84) called the dilation equations or two-scale equations. These have origin in the beautiful theory of multiresolution for L^2 spaces [4,11]. Because multiresolution theory lays the foundation for the construction of the most practical wavelets to date, we give a brief description of it here.

6.10.1 Idea of Multiresolution

Return to Figure 6.13a, where we interpreted the wavelet transformation as a bank of continuous time analysis filters followed by samples, and the inverse transformation as a bank of synthesis filters. Assume for simplicity the filters are ideal bandpass. Figure 6.13b is a sketch of the frequency responses. The bandpass filters $F_k(\omega) = 2^{-k/2}\psi(\omega/2^k)$ become increasingly narrow as k decreases (i.e., as k becomes more and more negative). Instead of letting k be negative, suppose we keep only $k \geq 0$ and include a low-pass filter $\Phi(\omega)$ to cover the low frequency region. Then, we get the picture of Figure 6.33. This is analogous to Figure 6.12, where we used the pulse function $\phi(t)$ instead of using negative k in $\psi(2^k t - n)$.

Imagine for a moment that $\Phi(\omega)$ is an ideal low-pass filter with cutoff $\pm\pi$. Then we can represent any L^2 function $F(\omega)$ with support restricted to $\pm\pi$ in the form $F(\omega) = \sum_{n=-\infty}^{\infty} a_n \Phi(\omega)e^{-j\omega n}$. This is simply the FS expansion of $F(\omega)$ in $[-\pi, \pi]$, and it follows that $\sum_n |a_n|^2 < \infty$ (Theorem 6.1). In the time domain, this means

$$f(t) = \sum_{n=-\infty}^{\infty} a_n \phi(t - n) \tag{6.86}$$

Let us denote by V_0 the closure of the span of $\{\phi(t-n)\}$. Thus, V_0 is the class of L^2 signals that are band-limited to $[-\pi, \pi]$. Because $\phi(t)$ is the sinc function, the shifted functions $\{\phi(t-n)\}$ form an orthonormal basis for V_0.

Consider now the subspace $W_0 \subset L^2$ of bandpass functions band-limited to $\pi < |\omega| \leq 2\pi$. The bandpass sampling theorem (Section 6.2) allows us to reconstruct such a bandpass signal $g(t)$ from its samples $g(n)$ by using the ideal filter $\Psi(\omega)$. Denoting the impulse response of $\Psi(\omega)$ by $\psi(t)$ we see

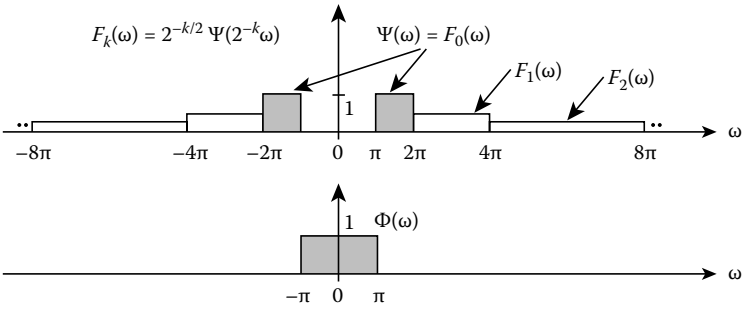


FIGURE 6.33 Low-pass function $\Phi(\omega)$, bandpass function $\Psi(\omega)$, and stretched bandpass filters $F_k(\omega)$.

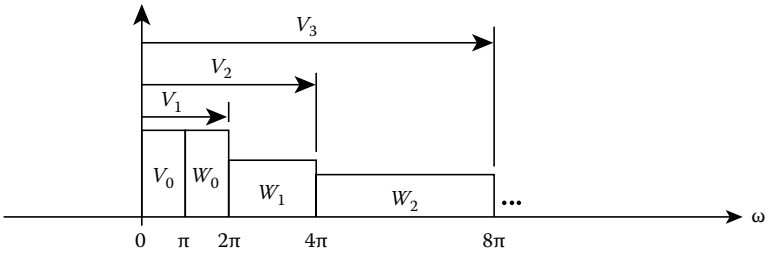


FIGURE 6.34 Toward multiresolution analysis. The spaces $\{V_k\}$ and $\{W_k\}$ spanned by various filter responses.

that $\{\psi(t-n)\}$ spans W_0. It can be verified that $\{\psi(t-n)\}$ is an orthonormal basis for W_0. Moreover, as $\Psi(\omega)$ and $\Phi(\omega)$ do not overlap, it follows from Parseval's theorem that W_0 is orthogonal to V_0.

Next, consider the space of all signals of the form $f(t)+g(t)$, where $f(t) \in V_0$ and $g(t) \in W_0$. This space is called the direct sum (in this case, orthogonal sum) of V_0 and W_0, and is denoted as $V_1 = V_0 \oplus W_0$. It is the space of all L^2 signals band limited to $[-2\pi, 2\pi]$. We can continue in this manner and define the spaces V_k and W_k for all k. Then, V_k is the space of all L^2 signals band limited to $[-2^k\pi, 2^k\pi]$, and W_k is the space of L^2 functions band limited to $2^k\pi < |\omega| \leq 2^{k+1}\pi$. The general recursive relation is $V_{k+1} = V_k \oplus W_k$. Figure 6.34 demonstrates this for the case in which the filters are ideal bandpass. Only the positive half of the frequency axis is shown for simplicity.

It is clear that we could imagine V_0 itself to be composed of subspaces V_{-1} and W_{-1}. Thus, $V_0 = V_{-1} \oplus W_{-1}$, $V_{-1} = V_{-2} \oplus W_{-2}$, and so forth. In this way, we have defined a sequence of spaces $\{V_k\}$ and $\{W_k\}$ for all integers k such that the following conditions are true:

$$V_{k+1} = V_k \oplus W_k, \quad W_k \perp W_m, \quad k \neq m \tag{6.87}$$

where $\perp$ means "orthogonal," i.e., the functions in W_k are orthogonal to those in W_m. It is clear that $V_k \subset V_{k+1}$.

We will see later that even if the ideal filters $\Phi(\omega)$ and $\Psi(\omega)$ *are replaced with nonideal approximations*, we can sometimes define sequences of subspaces V_k and W_k satisfying the above conditions. The importance of this observation is that *whenever* $\Psi(\omega)$ *and* $\Phi(\omega)$ are such that we can construct such a subspace structure, the impulse response $\psi(t)$ of the filter $\Psi(\omega)$ can be used to generate an orthonormal wavelet basis. Although this might appear to be too complicated and convoluted, we will see that the construction of the function $\phi(t)$ is quite simple and elegant, and simplifies the construction of orthonormal wavelet bases. A realization of these ideas based on paraunitary filter banks is presented in Section 6.11. It is now time to be more precise with definitions as well as statements of the results.

6.10.2 Definition of Multiresolution Analysis

Consider a sequence of closed subspaces $\{V_k\}$ in L^2, satisfying the following six properties:

1. *Ladder property*: $\cdots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \cdots$.
2. $\bigcap_{k=-\infty}^{\infty} V_k = \{0\}$.
3. Closure of $\bigcup_{k=-\infty}^{\infty} V_k$ is equal to L^2.
4. *Scaling property*: $x(t) \in V_k$ iff $x(2t) \in V_{k+1}$. Because this implies "x(t) $\in V_0$ iff $x(2^k t) \in V_k$," all the spaces V_k are scaled versions of the space V_0. For $k > 0$, V_k is a *finer space* than V_0.
5. *Translation invariance*: If $x(t) \in V_0$, then $x(t-n) \in V_0$; that is, the space V_0 is invariant to translations by integers. By the previous property, this means that V_k is invariant to translations by $2^{-k}n$.

6. *Special orthonormal basis*: A function $\phi(t) \in V_0$ exists such that the integer shifted versions $\{\phi(t-n)\}$ form an orthonormal basis for V_0. Employing property 4, this means that $\{2^{k/2}\phi(2^k t-n)\}$ is an orthonormal basis for V_k. The function $\phi(t)$ is called the scaling function of multiresolution analysis.

Comments on the Definition

Notice that the scaling function $\phi(t)$ determines V_0, hence all V_k. We say that $\phi(t)$ generates the entire multiresolution analysis $\{V_k\}$. The sequence $\{V_k\}$ is said to be a ladder of subspaces because of the inclusion property $V_k \subset V_{k+1}$. The technical terms closed and closure, which originate from metric space theory, have simple meanings in our context because L^2 is a Hilbert space. Thus, the subspace V_k is "closed" if the following is true: whenever a sequence of functions $\{f_n(t)\} \in V_k$ converges to a limit $f(t) \in L^2$ (i.e., $\|f(t)-f_n(t)\| \to 0$ as $n \to \infty$), the limit $f(t)$ is in V_k itself. In general, an infinite union of closed sets is not closed, which is why we need to take "closure" in the third property on page 187. The third property simply means that any element $x(t)L^2$ can be approximated arbitrary closely (in the L^2 norm sense) by an element in $\bigcup_{k=-\infty}^{\infty} V_k$.

General Meaning of W_k

In the general setting of the previous definition, the subspace W_k is defined as the orthogonal complement of V_k with respect to V_{k+1}. Thus, the relation $V_{k+1} = V_k \oplus W_k$, which was valid in the ideal bandpass case (Figure 6.34), continues to hold.

Haar Multiresolution

A simple example of multiresolution in which $\Phi(\omega)$ is not ideal low-pass is the Haar multiresolution, generated by the function $\phi(t)$ in Figure 6.35a. Here, V_0 is the space of all functions that are piecewise constants on intervals of the form $[n, n+1]$. We will see later that the function $\psi(t)$ associated with this example is as in Figure 6.35b—the space W_0 is spanned by $\{\psi(t-n)\}$. The space V_k contains functions that are constants in $[2^{-k}\, n,\ 2^{-k}\,(n+1)]$. Figure 6.35c and d shows examples of functions belonging to V_0 and V_1. For this example, the six properties in the definition of multiresolution are particularly clear (except perhaps property 3, which also can be proved).

The multiresolution analysis generated by the ideal bandpass filters (Figures 6.33 and 6.34) is another simple example, in which $\phi(t)$ is the sinc function. We see that the two elementary orthonormal wavelet examples (Haar wavelet and the ideal bandpass wavelet) also generate a corresponding multiresolution analysis. The connection between wavelets and multiresolution is deeper than this, and is elaborated in the following section.

Derivation of the Dilation Equation

Because $\{\sqrt{2}\phi(2t-n)\}$ is an orthonormal basis for V_1 (see p. 6-53), and because $\phi(t) \in V_0 \subset V_1, \phi(t)$ can be expressed as a linear combination of the functions $\{\sqrt{2}\phi(2t-n)\}$:

$$\phi(t) = 2\sum_{n=-\infty}^{\infty} g_s(n)\phi(2t-n) \qquad (6.88)$$

(dilation equation)

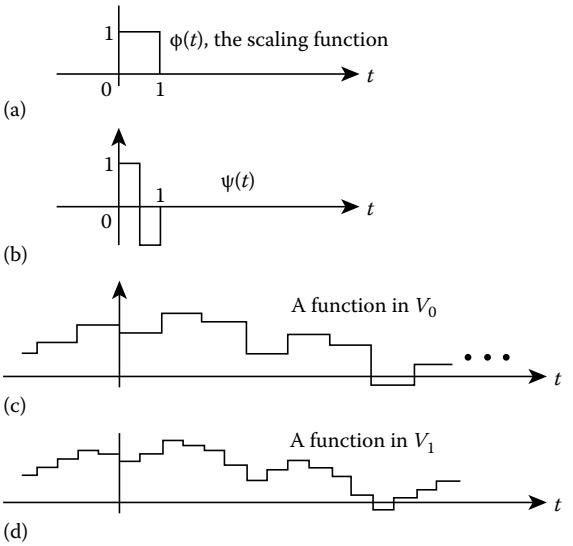


FIGURE 6.35 Harr multiresolution example. (a) The scaling function $\phi(t)$ that generates multiresolution, (b) the function $\psi(t)$ that generates W_0, (c) example of a member of V_0, and (d) example of a member of V_1.

Thus, the dilation equation arises naturally out of the multiresolution condition. For example, the Haar scaling function $\phi(t)$ satisfies the dilation equation

$$\phi(t) = \phi(2t) + \phi(2t-1) \tag{6.89}$$

The notation $g_s(n)$ and the factor 2 in the dilation equation might appear arbitrary now, but are convenient for future use. Orthonormality of $\{\phi(t-n)\}$ implies that $\|\phi(t)\| = 1$, and that $\{\sqrt{2}\phi(2t-n)\}$ are orthonormal. Therefore, $\Sigma_n |g_s(n)|^2 = 0.5$ from Equation 6.88.

Example 6.9: Nonorthonormal Multiresolution

Consider the triangular function shown in Figure 6.36a. This has $\|\phi(t)\| = 1$ and satisfies the dilation equation

$$\phi(t) = \phi(2t) + 0.5\phi(2t-1) + 0.5\phi(2t+1) \tag{6.90}$$

as demonstrated in Figure 6.36b. With V_k denoting the closure of the span of $\{2^{k/2}\phi(2^k t - n)\}$ it can be shown that the spaces $\{V_k\}$ satisfy all the conditions in the multiresolution definition, except one. Namely, $\{\phi(t-n)\}$ does not form an orthonormal basis (for example, compare $\phi(t)$ and $\phi(t-1)$). We will see later (Example 6.10) that it does form a Riesz basis and that it can be converted into an orthonormal basis by orthonormalization. This example is a special case of a family of scaling functions called spline functions [15].

We will see below that starting from an orthonormal multiresolution system (in particular from the function $\phi(t)$) one can generate an orthonormal wavelet basis for L^2. The wavelet bases generated from splines $\phi(t)$ after orthonormalization are called spline wavelets [15]. These are also called the Battle–Lemarié family of wavelets. The link between multiresolution analysis and wavelets is explained quantitatively in Section 6.10.3.

Multiresolution Approximation of L^2 Functions

Given a multiresolution analysis, we know that $\bigcap_{k=-\infty}^{\infty} V_k = \{0\}$ and that the closure of $\bigcup_{k=-\infty}^{\infty} V_k = L^2$. From this it can be shown that the W_k make up the entire L^2 space, that is

$$L^2 = \bigoplus_{k=-\infty}^{\infty} W_k \tag{6.91a}$$

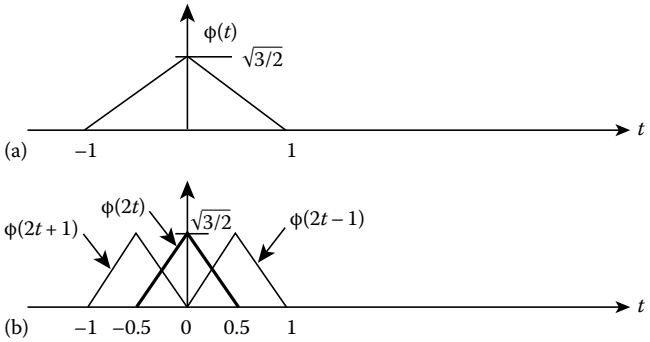


FIGURE 6.36 Example of a scaling function $\phi(t)$ generating nonorthogonal multiresolution. (a) The scaling function and (b) demonstrating the dilation equation.

We can approximate an arbitrary L^2 function $x(t)$ to a certain degree of accuracy by projecting it onto V_k for appropriate k. Thus, let $x_k(t)$ be this orthogonal projection (Section 6.2). Suppose we increase k to $k+1$. Because $V_{k+1} = V_k \oplus W_k$ and W_k is orthogonal to V_k, we see that the new approximation $x_{k+1}(t)$ (projection onto the finer space V_{k+1}) is given by $x_{k+1}(t) = x_k(t) + y_k(t)$, where $y_k(t)$ is in W_k.

Thus, when we go from scale k to scale $k+1$ we go to a larger space $V_{k+1} \supset V_k$, which permits a finer approximation. This is nicely demonstrated in the two extreme examples mentioned previously. For the example with ideal filters (Figures 6.33 and 6.34), the process of passing from scale k to $k+1$ is like admitting higher frequency components, which are orthogonal to the existing low-pass components. For the Haar example (Figure 6.35) where $\psi(t)$ and $\phi(t)$ are square pulses, when we pass from k to $k+1$ we permit finer pulses (i.e., highly localized finer variations in the time domain). For this example, Figure 6.35c and d demonstrates the projections $x_k(t)$ and $x_{k+1}(t)$ at two successive resolutions. The projections are piecewise-constant approximations of an L^2 signal $x(t)$.

By repeated application of $V_{k+1} = V_k \oplus W_k$, we can express V_0 as

$$V_0 = \bigoplus_{k=-\infty}^{-1} W_k \tag{6.91b}$$

which, together with Equation 6.91a, yields

$$L^2 = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots \tag{6.91c}$$

This has a nice interpretation based on Figure 6.34. The L^2 signal $x(t)$ has been decomposed into orthogonal components belonging to V_0 (low-pass component), W_0 (bandpass component), W_1 (bandpass with higher bandwidth and center frequency), etc.

We can find an infinite number of multiresolution examples by choosing $\phi(t)$ appropriately. It is more important now to obtain systematic techniques for constructing such examples. The quality of the example is governed by the quality of $\psi(t)$ and $\phi(t)$—the time localization and frequency resolution they can provide, the smoothness (regularity) of these functions, and the ease with which we can implement these approximations.

6.10.3 Relation between Multiresolution and Wavelets

Suppose $\phi(t) \in L^2$ generates an orthonormal multiresolution $\{V_k\}$, as defined in Section 6.10.2. We know $\phi(t) \in V_0$ and that $\{\phi(t-n)\}$ is an orthonormal basis for V_0. Moreover, $\phi(t)$ satisfies the dilation Equation 6.88, and the sequence $\{g_s(n)\} \in \ell^2$ defines the filter $G_s(e^{j\omega})$.

Now consider the finer space $V_1 = V_0 \oplus W_0$, where W_0 is orthonormal to V_0. If $f(t) \in W_0$ then $f(t) \in V_1$, so it is a linear combination of $\sqrt{2}\phi(2t-n)$ (property 6; see definitions). Using this and the fact that W_0 is orthogonal to V_0, we can show that $F(\omega)$ (the L^2-*FT* of $f(t)$) has a special form. This is given by

$$F(\omega) = e^{j\omega/2} G_s^\star(-e^{j\omega/2}) \Phi(\omega/2) H(e^{j\omega})$$

where $H(e^{j\omega})$ is 2π-periodic. The special case of this with $H(e^{j\omega}) = 1$ is denoted $\Psi(\omega)$; that is

$$\Psi(\omega) = e^{j\omega/2} G_s^\star(-e^{j\omega/2}) \Phi(\omega/2) \tag{6.92}$$

The above definition of $\Psi(\omega)$ is equivalent to

$$\psi(t) = 2 \sum_{n=-\infty}^{\infty} (-1)^{n+1} g_s^\star(-n-1) \phi(2t-n) \quad \text{[dilation equation for } \psi(t)\text{]} \tag{6.93}$$

The function $\psi(t)$ satisfying this equation has some useful properties. First, it is in L^2. This follows from Theorem 6.2 (Riesz–Fisher Theorem), because $\sum_n |g_s(n)|^2$ is finite. It can be shown that $\psi(t-n) \in W_0$ and that $\{\psi(t-n)\}$ is an orthonormal basis for W_0. This implies that $\{2^{k/2}\psi(2^k\, t-n)\}$ is an orthonormal basis for W_k because $f(t) \in W_0$ iff $f\,(2^k t) \in W_k$, which is a property induced by the scaling property (property 4 in the definition of multiresolution). In view of Equation 6.91 we conclude that the sequence $\{2^{k/t}\,\psi(2^k\, t-n)\}$, with k and n varying over all integers, forms a basis for L^2. Summarizing we have the following result:

THEOREM 6.6: (Multiresolution and Wavelets)

Let $\phi(t) \in L^2$ generate an orthonormal multiresolution, i.e., a ladder of spaces $\{V_k\}$ satisfying the six properties in the definition of multiresolution; $\{\phi(t-n)\}$ is an orthonormal basis for V_0. Then, $\phi(t)$ satisfies the dilation Equation 6.88 for some $g_s(n)$ with $\Sigma_n\,|g_s(n)\,|^2 = 0.5$. Define the function $\psi(t)$ according to the dilation Equation 6.93. Then, $\psi(t) \in W_0 \subset L^2$, and $\{\psi(t-n)\}$ is an orthonormal basis for W_0. Therefore, $\{2^{k/2}\,\psi(2^k\, t-n)\}$ is an orthonormal basis for W_k, just as $\{2^{k/2}\,\phi(2^k\, t-n)\}$ is an orthonormal basis for V_k (for fixed k). Moreover, with k and n varying over all integers, the doubly indexed sequence $\{2^{k/2}\,\psi(2^k\, t-n)\}$ is an orthonormal wavelet basis for L^2.

Thus to construct a wavelet basis for L^2 we have only to construct an orthonormal basis $\{\phi(t-n)\}$ for V_0. Everything else follows from that. All proofs can be found in Refs. [5,11,15].

6.10.4 Relation between Multiresolution Analysis and Paraunitary Filter Banks

Denoting

$$h_s(n) = (-1)^{n+1} g_s^*(-1-n), \quad \text{i.e., } H_s(e^{j\omega}) = e^{j\omega} G_s^*(-e^{j\omega})$$

we see that $\phi(t)$ and $\psi(t)$ satisfy the two dilation equations in Equation 6.84. By construction, $\psi(t) \in W_0$ and $\phi(t) \in V_0$. The fact that W_0 and V_0 are mutually orthogonal subspaces can be used to show that $H_s(e^{j\omega})$ and $G_x(e^{j\omega})$ satisfy

$$G_s^*(e^{j\omega})H_s(e^{j\omega}) + !G_s^*(-e^{j\omega})H_s(-e^{j\omega}) = 0 \tag{6.94}$$

Also, orthonormality of $\{\phi(t-n)\}$ leads to the power complementary property

$$\left|G_s\left(e^{j\omega}\right)\right|^2 + \left|G_s\left(-e^{j\omega}\right)\right|^2 = 1 \tag{6.95}$$

In other words, $G_s(e^{j\omega})$ is a power symmetric filter. That is, the filter $|G_s(e^{j\omega})|^2$ is a half-band filter. Using $H_s(e^{j\omega}) = e^{j\omega}G_s(-e^{j\omega})$, we also have

$$\left|H_s\left(e^{j\omega}\right)\right|^2 + \left|H_s\left(-e^{j\omega}\right)\right|^2 = 1 \tag{6.96}$$

A compact way to express the above three equations is by defining the matrix

$$\mathbf{G}_s(e^{j\omega}) = \begin{bmatrix} G_s(e^{j\omega}) & H_s(e^{j\omega}) \\ G_s(-e^{j\omega}) & H_s(-e^{j\omega}) \end{bmatrix}$$

The three properties in Equations 6.94 through 6.96 are equivalent to $\mathbf{G}_s^{\dagger}(e^{\ j\omega})\mathbf{G}_s(e^{j\omega}) = \mathbf{I}$; i.e., the matrix $\mathbf{G}_s(e^{j\omega})$ is unitary for all ω. This matrix was defined in Section 6.4 in the context of paraunitary digital

filter banks. Thus, the filters $G_s(e^{j\omega})$ and $H_s(e^{j\omega})$ constructed from a multiresolution setup constitute a paraunitary (CQF) synthesis bank.

Thus, orthonormal multiresolution automatically gives rise to paraunitary filter banks. Starting from a multiresolution analysis we obtained two functions $\phi(t)$ and $\psi(t)$. These functions generate orthonormal bases $\{\phi(t-n)\}$ and $\{\psi(t-n)\}$ for the orthogonal subspaces V_0 and W_0. The functions $\phi(t)$ and $\psi(t)$ generated in this way satisfy the dilation equations (Equation 6.84). Defining the filters $G_s(z)$ and $H_s(z)$ from the coefficients $g_s(n)$ and $h_s(n)$ in an obvious way, we find that these filters form a paraunitary pair.

This raises the following fundamental question: If we start from a paraunitary pair $\{G_s(z), H_s(z)\}$ and define the functions $\phi(t)$ and $\psi(t)$ by (successfully) solving the dilation equations, do we obtain an orthonormal basis $\{\phi(t-n)\}$ for multiresolution, and a wavelet basis $\{2^{k/2}\,\psi(2^k\,t-n)\}$ for the space of L^2 functions? The answer, fortunately, is in the affirmative, subject to some minor requirements which can be trivially satisfied in practice.

6.10.4.1 Generating Wavelet and Multiresolution Coefficients from Paraunitary Filter Banks

Recall that the subspaces V_0 and W_0 have the orthonormal bases $\{\phi(t-n)\}$ and $\{\psi(t-n)\}$, respectively. By the scaling property, the subspace V_k has the orthonormal basis $\{\phi_{kn}(t)\}$, and similarly the subspace W_k has the orthonormal basis $\{\psi_{kn}(t)\}$, where, as usual, $\phi_{kn}(t) = 2^{k/2}\phi\ (2^k\ t-n)$ and $\psi_{kn}(t) = 2^{k/2}\ \psi(2^k t-n)$. The orthogonal projections of a signal $x(t) \in L^2$ onto V_k and W_k are given, respectively, by

$$\begin{aligned} P_k[x(t)] &= \sum_{n=-\infty}^{\infty} \langle x(t), \phi_{kn}(t)\rangle \phi_{kn}(t) \quad \text{and} \\ Q_k[x(t)] &= \sum_{n=-\infty}^{\infty} \langle x(t), \psi_{kn}(t)\rangle \psi_{kn}(t) \end{aligned} \tag{6.97}$$

(Section 6.2). Denote the scale-k projection coefficients as $d_k(n) = \langle x(t), \phi_{kn}(t)\rangle$ and $c_k(n) = \langle x(t), \psi_{kn}(t)\rangle$ for simplicity. (The notation c_{kn} was used in earlier sections, but $c_k(n)$ is convenient for the present discussion.) We say that $d_k(n)$ are the multiresolution coefficients at scale k and $c_k(n)$ are the wavelet coefficients at scale k.

Assume that the projection coefficients $d_k(n)$ are known for some scale, e.g., $k=0$. We will show that $d_k(n)$ and $c_k(n)$ for the coarser scales, i.e., $k=-1, -2, \ldots$ can be generated by using a paraunitary analysis filter bank $\{G_a(e^{j\omega}), H_a(e^{j\omega})\}$, corresponding to the synthesis bank $\{G_s(e^{j\omega}), H_s(e^{j\omega})\}$ (Section 6.4). We know $\phi(t)$ and $\psi(t)$ satisfy the dilation equations (Equation 6.84). By substituting the dilation equations into the right-hand sides of $\phi_{kn}(t) = 2^{k/2}\ \phi(2^k\ t-n)$ and $\psi_{kn}(t) = 2^{k/2}\ \psi(2^k\ t-n)$, we obtain

$$\begin{aligned} \phi_{kn}(t) &= \sqrt{2}\ \sum_{m=-\infty}^{\infty} g_s(m-2n)\phi_{k+1,\,m}(t) \quad \text{and} \\ \psi_{kn}(t) &= \sqrt{2}\ \sum_{m=-\infty}^{\infty} h_s(m-2n)\phi_{k+1,\,m}(t) \end{aligned} \tag{6.98}$$

A computation of the inner products $d_k(n) = \langle x(t), \phi_{kn}(t)\rangle$ and $c_k(n) = \langle x(t), \psi_{kn}(t)\rangle$ yields

$$\begin{aligned} d_k(n) &= \sum_{m=-\infty}^{\infty} \sqrt{2} g_a(2n-m) d_{k+1}(m) \\ c_k(n) &= \sum_{m=-\infty}^{\infty} \sqrt{2} h_a(2n-m) d_{k+1}(m) \end{aligned} \tag{6.99}$$

where $g_a(n) = g_s^*(-n)$ and $h_a(n) = h_s^*(-n)$ are the analysis filters in the paraunitary filter bank.

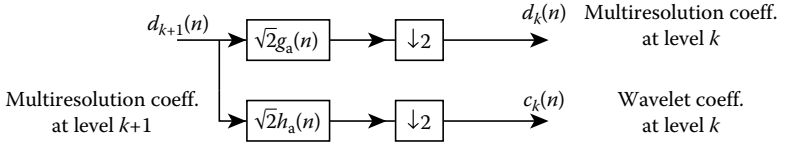


FIGURE 6.37 Generating the wavelet and multiresolution coefficients at level k from level $k+1$.

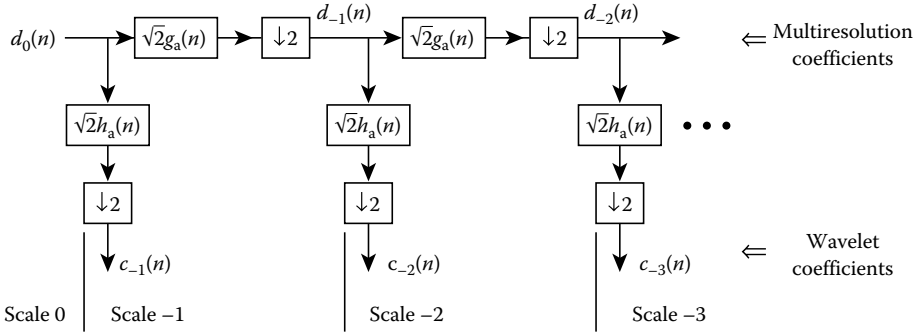


FIGURE 6.38 Tree-structured analysis bank generating wavelet coefficients $c_k(n)$ and multiresolution coefficients $d_k(n)$ recursively.

The beauty of these equations is that they look like discrete-time convolutions. Thus, if $d_{k+1}(n)$ is convolved with the impulse response $\sqrt{2}g_a(n)$ and the output decimated by 2, the result is the sequence $d_k(n)$. A similar statement follows for $c_k(n)$. The above computation can therefore be interpreted in filter bank form as in Figure 6.37. Because of the PR property of the two-channel system (Figure 6.22), it follows that we can reconstruct the projection coefficients $d_{k+1}(n)$ from the projection coefficients $d_k(n)$ and $c_k(n)$.

6.10.4.2 Fast Wavelet Transform

Repeated application of this idea results in Figure 6.38, which is a tree-structured paraunitary filter bank (Section 6.4) with analysis filters $\sqrt{2}g_a(n)$and$\sqrt{2}h_a(n)$ at each stage. Thus, given the projection coefficients $d_0(n)$ for V_0, we can compute the projection coefficients $d_k(n)$ and $c_k(n)$ for the coarser spaces V_{-1}, W_{-1}, V_{-2}, W_{-2},.... This scheme is sometimes referred to as the fast wavelet transform (FWT). Figure 6.39 shows a schematic of the computation. In this figure each node (heavy dot) represents a decimated paraunitary analysis bank $\{\sqrt{2}g_a(n), \sqrt{2}h_a(n)\}$. The subspaces W_m and V_m are indicated in the figure rather than the projection coefficients.

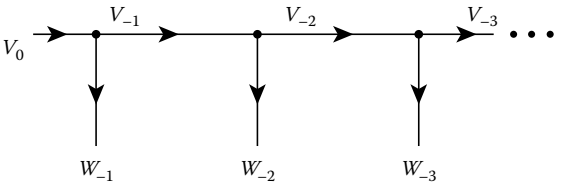


FIGURE 6.39 Schematic of the tree-structured filter bank, which generates the coefficients of the projections onto V_k and W_k.

Computation of the Initial Projection Coefficient: Everything depends on the computation of $d_0(n)$. Note that $d_0(n) = \langle x(t), \phi(t-n)\rangle$, which can be written as the integral $d_0(n) = \int x(t)\phi^*(t-n)\mathrm{d}t$. An elaborate computation of this integral is avoided in practice. If the scale $k=0$ is fine enough—if $x(t)$ does not change much within the duration where $\phi(t)$ is significant—we can approximate this integral with the sample value $x(n)$; i.e., $d_0(n) \approx x(n)$. Improved approximations of $d_0(n)$ have been suggested by other authors, but are not reviewed here.

6.10.5 Continuous Time Filter Banks and Multiresolution

The preceding discussions show the deep connection between orthonormal multiresolution analysis and discrete-time paraunitary filter banks. As shown by Equation 6.91c, any L^2 signal $x(t)$ can be written as a sum of its projections onto the mutually orthogonal spaces V_0, W_0, W_1, etc.:

$$x(t) = \sum_n d_0(n)\phi(t-n) + \sum_{k=0}^{\infty}\sum_n c_k(n)2^{k/2}\psi(2^k t - n)$$

This decomposition itself can be given a simple filter bank interpretation, with continuous time filters and samplers. For this, first note that the V_0 component $\sum_n d_0(n)\phi(t-n)$ can be regarded as the output of a filter with impulse response $\phi(t)$, with the input chosen as the impulse train $\sum_n d_0(n)\delta_a(t-n)$. Similarly, the W_k component $\sum_n c_k(n)2^{k/2}\psi(2^k t - n)$ is the output of a filter with impulse response $f_k(t) = 2^{k/2}\psi(2^k t)$, in response to the input $\sum_n c_k(n)\delta_a(t - 2^{-k}n)$. This interpretation is shown by the synthesis bank of Figure 6.40a.

The projection coefficients $d_0(n)$ and $c_k(n)$ can also be interpreted nicely. For example, we have $d_0(n) = \langle x(t), \phi(t-n)\rangle$ by orthonormality. This inner product can be explicitly written out as

$$d_0(n) = \int x(t)\phi^\star(t-n)\mathrm{d}t$$

The integral can be interpreted as a convolution of $x(t)$ with $\phi^\star(-t)$. Consider the output of the filter with impulse response $\phi^\star(-t)$, with the input chosen as $x(t)$. This output, sampled at time n, gives $d_0(n)$.

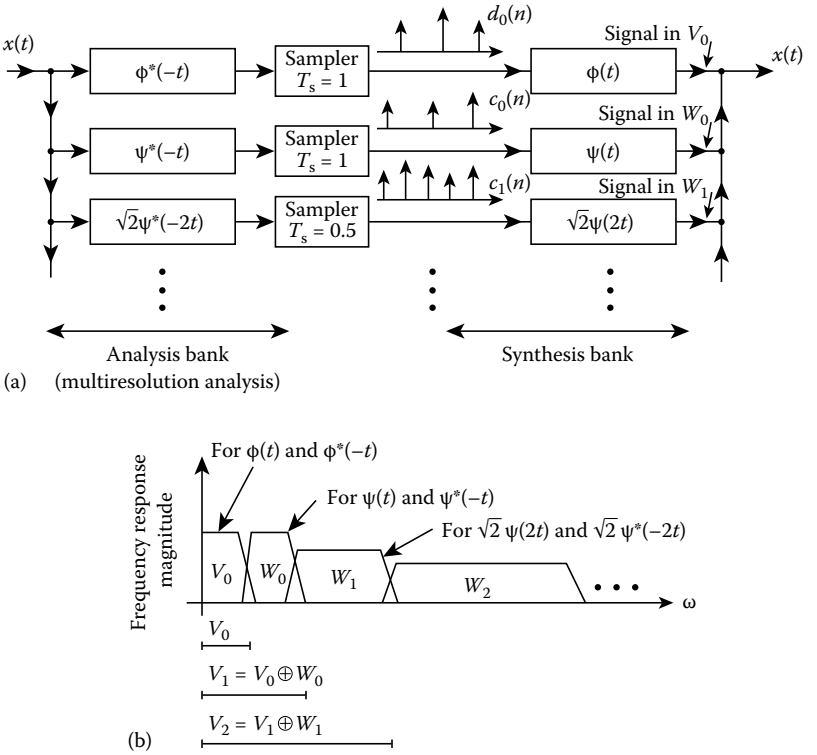


FIGURE 6.40 (a) Multiresolution analysis and resynthesis in filter bank form and (b) typical frequency responses.

Similarly, $c_k(n)$ can be interpreted as the output of the filter $h_k(t) = 2^{k/2}\psi^*(-2^k\ t)$, sampled at the time $2^{-k}n$. The analysis bank of Figure 6.40a illustrates this interpretation. Thus, the projection coefficients $d_0(n)$ and $c_k(n)$ are the sampled versions of the outputs of an analysis filter bank.

Notice that all the filters in the filter bank are determined by the scaling function $\phi(t)$ and the wavelet function $\psi(t)$. Every synthesis filter $f_k(t)$ is the time-reversed conjugate of the corresponding analysis filter $h_k(t)$, that is, $f_k(t) = h^*{}_k(-t)$ (a consequence of orthonormality). In terms of frequency responses this means $F_k(\omega) = H_k{}^*(\omega)$. For completeness of the picture, Figure 6.40b shows typical frequency response magnitudes of these filters.

6.10.6 Further Manifestations of Orthonormality

The orthonormality of the basis functions $\{\phi(t-n)\}$ and $\{\psi(t-n)\}$ have further consequences, summarized below. A knowledge of these will be useful when we generate the scaling function $\phi(t)$ and the wavelet function $\psi(t)$ systematically in Section 6.11 from paraunitary filter banks.

6.10.6.1 Nyquist Property and Orthonormality

With $\phi(t) \in L^2$, the autocorrelation function $R(t) = \int \phi(t)\phi^*(t-\tau)dt$ exists for all τ because this is simply an inner product of two elements in L^2. Clearly, $R(0) = |\phi(t)|^2 = 1$. Further, the orthonormality property $\langle \phi(t), \phi(t-n)\rangle = \delta(n)$ can be rewritten as $R(n) = \delta(n)$. Thus, $R(\tau)$ has periodic zero crossings at nonzero integer values of τ (Figure 6.41). This is precisely the *Nyquist property* familiar to communication engineers. The autocorrelation of the scaling function $\phi(t)$ is a Nyquist function. The same holds for the wavelet function $\psi(t)$.

Next, using Parseval's identity for L^2-FTs, we obtain $\langle \phi(t), \phi(t-n)\rangle = \int \Phi(\omega)\Phi(\omega)e^{j\omega n}d\omega/2\pi = \delta(n)$. If we decompose the integral into a sum of integrals over intervals of length 2π and use the 2π-periodicity of $e^{j\omega n}$, we obtain, after some simplification:

$$\sum_{k=-\infty}^{\infty} |\Phi(\omega + 2\pi k)|^2 = 1 \quad \text{a.e.} \tag{6.100}$$

This is the Nyquist condition, now expressed in the frequency domain. The term a.e., almost everywhere, arises from the fact that we have drawn a conclusion about an integrand from the value of the integral. Thus, $\{\phi(t-n)\}$ is orthonormal iff the preceding equation holds. A similar result follows for $\Psi(\omega)$, so orthonormality of $\{\psi(t-n)\}$ is equivalent to

$$\sum_{k=-\infty}^{\infty} |\Psi(\omega + 2\pi k)|^2 = 1 \quad \text{a.e.} \tag{6.101}$$

6.10.6.2 Case in Which Equalities Hold Pointwise

If we assume that all FTs are continuous, then equalities in the Fourier domain actually hold pointwise. This is the most common situation; in all examples here, the following are true: the filters $G_s(e^{j\omega})$ and

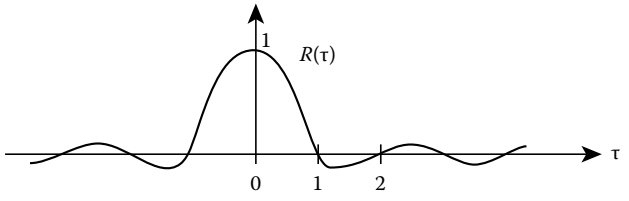


FIGURE 6.41 Example of an autocorrelation of the scaling function $\phi(t)$.

$H_s(e^{j\omega})$ are rational (FIR or IIR), so the frequency responses are continuous functions of ω, and $\phi(t)$ and $\psi(t)$ are not only in L^2, but also in L^1; i.e., $\phi(t)$, $\psi(t) \in L^1 \cap L^2$. Thus, $\Phi(\omega)$ and $\Psi(\omega)$ are continuous functions (Section 6.6).

With the dilation equation $\Phi(\omega) = G_s(e^{j\omega/2})\Phi(\omega/2)$ holding pointwise, we have $\Phi(0) = G_s(e^{j0})\Phi(0)$. In all our applications $\Phi(0) \neq 0$ (it is a low-pass filter), so $G_s(e^{j0}) = 1$. The power symmetry property

$$\left|G_s(e^{j\omega})\right|^2 + \left|G_s(-e^{j\omega})\right|^2 = 1$$

then implies $G_s(e^{j\pi}) = 0$. Because the high-pass synthesis filter is $H_s(e^{j\omega}) = e^{j\omega}G_s^*(-e^{j\omega})$ we conclude $H_s(e^{j0}) = 0$ and $H_s(e^{j\pi}) = -1$. Thus,

$$G_s(e^{j0}) = 1, \quad G_s(e^{j\pi}) = 0, \quad H_s(e^{j0}) = 0, \quad H_s(e^{j\pi}) = -1 \tag{6.102}$$

In particular, the low-pass impulse response $g_s(n)$ satisfies $\Sigma_n g_s(n) = 1$. Because we already have $\Sigma_n |g_s(n)|^2 = 0.5$ (Theorem 6.6), we have both of the following:

$$\sum_{n=-\infty}^{\infty} g_s(n) = 1 \quad \text{and} \quad \sum_{n=-\infty}^{\infty} |g_s(n)|^2 = 0.5 \tag{6.103}$$

From the dilation equation $\Phi(\omega) = G_s(e^{j\omega/2})\Phi(\omega/2)$, we obtain $\Phi(2\pi k) = G_s(e^{j\pi k})\Phi(\pi k)$. By using the fact that $G_s(e^{j\pi}) = 0$, and after elementary manipulations, we can show that

$$\Phi(2\pi k) = 0, \quad k \neq 0 \tag{6.104}$$

In other words, $\Phi(\omega)$ is itself a Nyquist function of ω. If Equation 6.100 is *assumed to hold pointwise*, the above implies that $|\Phi(0)| = 1$. Without loss of generality we will let $\Phi(0) = 1$, i.e., $\int(\phi(t)dt = 1$. The dilation equation for the wavelet function $\Psi(\omega)$ in Equation 6.85 shows that $\Psi(0) = 0$ (because $H_s(e^{j0}) = 0$ by Equation 6.102). That is, $\int\psi(t)dt = 0$. Summarizing, the scaling and wavelet functions satisfy

$$\begin{aligned} &\int_{-\infty}^{\infty} \phi(t)dt = 1, \quad \int_{-\infty}^{\infty} \psi(t)dt = 0 \quad \text{and} \\ &\int_{-\infty}^{\infty} |\phi(t)|^2 dt = \int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1 \end{aligned} \tag{6.105}$$

where property 3 follows from orthonormality. These integrals make sense because of the assumption $\phi(t) \in L^1 \cap L^2$. Another result that follows from $\Phi(2\pi k) = \delta(k)$ is that

$$\sum_{n=-\infty}^{\infty} \phi(t-n) = 1 \quad \text{a.e.} \tag{6.106}$$

Thus, the basis functions of the subspace V_0 add up to unity. Return to the Haar basis and notice how beautifully everything fits together.

6.10.7 Generating Wavelet and Multiresolution Basis by Design of $\phi(t)$

Most of the well-known wavelet basis families of recent times were generated by first finding a scaling function $\phi(t)$ such that it is a valid generator of multiresolution, and then generating $\psi(t)$ from $\phi(t)$. The first step, therefore, is to identify the conditions under which a function $\phi(t)$ will be a valid scaling function (i.e., it will generate a multiresolution). Once this is done and we successfully identify the coefficients $g_s(n)$ in the dilation equation for $\phi(t)$, we can identify the wavelet function $\psi(t)$ using the

second dilation equation in Equation 6.84. From Theorem 6.6, we know that if $\psi(t)$ is computed in this way, then $\{2^{k/2}\,\psi(2^k t - n)\}$ is an orthonormal wavelet basis for L^2. The following results can be deduced from the many detailed results presented in Ref. [5].

THEOREM 6.7: (Orthonormal Multiresolution)

Let $\phi(t)$ *satisfy the following conditions:* $\phi(t) \in L^1 \bigcap L^2$, $\int \phi(t)\,dt \neq 0$ (*i.e.*, $\Phi(0) \neq 0$), $\phi(t) = 2\boldsymbol{\Sigma}_n g_s(n)\,\phi(2t - n)$ *for some* $\{g_s(n)\}$, *and* $\{\phi(t - n)\}$ *is an orthonormal sequence. Then the following are true*:

1. $\phi(t)$ *generates a multiresolution. That is, if we define the space* V_k *to be the closure of the span of* $\{2^{k/2}\,\phi(2^k t - n)\}$, *then the set of spaces* $\{V_k\}$ *satisfies the six conditions in the definition of multiresolution.*
2. *Define* $\psi(t) = 2\boldsymbol{\Sigma}_n (-1)^{n+1}\, g_s^*(-n-1)\,\phi\,(2t - n)$. *Then,* $\psi(t)$ *generates an orthonormal wavelet basis for* L^2 *; that is,* $\{2^{k/2}\psi(2^k t - n)\}$, *with k and n varying over all integers, is an orthonormal basis for* L^2. *In fact, for fixed k, the functions* $\{2^{k/2}\psi(2^k t - n)\}$ *from an orthonormal basis for the subspace* W_k *(defined following the definition of multiresolution analysis).*

Comments: In many examples, $\phi(t) \in L^2$, and it is compactly supported. Then it is naturally in L^1 as well, so the assumption $\phi(t) \in L^2 \cap L^1$ is not too restrictive. Because $L^1 \cap L^2$ is dense in L^2, the previous construction still gives a wavelet basis for L^2. Notice also that the orthonormality of $\{\phi(t - n)\}$ implies orthonormality of $\{\sqrt{2}\phi(2t - n)\}$. The recursion $\phi(t) = 2\Sigma_n g_s(n)\phi(2t - n)$, therefore, is a Fourier series for $\phi(t)$ in L^2. Thus the condition $\boldsymbol{\Sigma}_n\, |g_s(n)|^2 = 0.5$ is automatically implied. This is not explicitly stated as part of the conditions in the theorem.

Orthonormalization

We know that orthonormality of $\{\phi(t - n)\}$ is equivalent to

$$\sum_{k=-\infty}^{\infty} |\Phi(\omega + 2\pi k)|^2 = 1 \tag{6.107}$$

Suppose now that this is not satisfied, but the weaker condition

$$a \leq \sum_{k=-\infty}^{\infty} |\Phi(\omega + 2\pi k)|^2 \leq b \tag{6.108}$$

holds for some $a > 0$ and $b < \infty$. Then, it can be shown that, we can at least obtain a Riesz basis (Section 6.7) of the form $\{\phi(t - n)\}$ for V_0. We can also normalize it to obtain an orthonormal sequence $\{\hat{\phi}(t - n)\}$ from which an orthonormal wavelet basis can be generated in the usual way. The following theorem summarizes the main results.

THEOREM 6.8

Let $\phi(t) \in L^1 \cap L^2$, $\int \phi(t)dt \neq 0$ (*i.e.*, $\Phi(0) \neq 0$), *and* $\phi(t) = 2\Sigma_n g_s(n)\phi(2t - n)$ *with* $\Sigma_n |g_s(n)|^2 < \infty$. *Instead of the orthonormality condition* (*Equation* 6.107), *let Equation* 6.108 *hold for some* $a > 0$ *and* $b < \infty$. *Then the following are true:*

1. $\{\phi(t - n)\}$ *is a Riesz basis for the closure* V_0 *of its span.*
2. $\phi(t)$ *generates a multiresolution. That is, if we define the space* V_k *to be the closure of the span of* $\{2^{k/2}\phi(2^k t - n)\}$ *the set of spaces* $\{V_k\}$ *satisfies the six conditions in the definition of multiresolution.*

If we define a new function $\hat{\phi}(t)$ in terms of its FT as

$$\hat{\Phi}(\omega) = \frac{\Phi(\omega)}{\left(\sum_k |\Phi(\omega + 2\pi k)|^2\right)^{0.5}} \tag{6.109}$$

then $\hat{\phi}(t)$ generates an orthonormal multiresolution, and satisfies a dilation equation similar to Equation 6.84. Using this we can define a corresponding wavelet function $\hat{\psi}(t)$ in the usual way. That is, if $\hat{\phi}(t) = 2\boldsymbol{\Sigma}_n g_s(n)\hat{\phi}(2t-n)$, choose $\hat{\psi}(t) = 2\boldsymbol{\Sigma}_n h_s(n)\hat{\phi}(2t-n)$, where $h_s(n) = (-1)^{n+1} g_s^*(-n-1)$. This wavelet $\hat{\psi}(t)$ generates an orthonormal wavelet basis for L^2. Note that the basis is not necessarily compactly supported if we start with compactly supported $\phi(t)$. An example is given in Figure 6.46b later.

Example 6.10: Battle–Lemarié Orthonormal Wavelets from Splines

In Example 6.9, we considered a triangular $\phi(t)$ (Figure 6.36), which generates a nonorthonormal multiresolution. In this example, we have

$$\Phi(\omega) = \sqrt{\frac{3}{2}}\left(\frac{\sin(\omega/2)}{(\omega/2)}\right)^2 \tag{6.110}$$

and it can be shown that

$$\sum_{k=-\infty}^{\infty} |\Phi(\omega + 2\pi k)|^2 = \frac{2 + \cos\omega}{2} \tag{6.111}$$

The inequality Equation 6.108 is satisfied with $a = 1/2$ and $b = 3/2$. Thus, we have a Riesz basis $\{\phi(t-n)\}$ for V_0. From this scaling function, we can obtain the normalized function $\hat{\Phi}(\omega)$ as above and then generate the wavelet function $\hat{\psi}(t)$ as explained earlier. This gives an orthonormal wavelet basis for L^2. $\hat{\phi}(t)$ does not, however, have compact support (unlike $\phi(t)$). Thus, the wavelet function $\hat{\psi}(t)$ generating the orthonormal wavelet basis is not compactly supported either.

6.11 Orthonormal Wavelet Basis from Paraunitary Filter Banks

The wisdom gained from the multiresolution viewpoint (Section 6.10) tells us a close connection exists between wavelet bases and two-channel digital filter banks. In fact, we obtained the equations of a paraunitary filter bank just by imposing the orthonormality condition on the multiresolution basis functions $\{\phi(t-n)\}$. This section presents the complete story. Suppose we start from a two-channel digital filter bank with the paraunitary property. Can we derive an orthonormal wavelet basis from this? To be more specific, return to the dilation equations (Equation 6.84) or equivalently Equation 6.85. Here, $g_s(n)$ and $h_s(n)$ are the impulse response coefficients of the two synthesis filters $G_s(e^{j\omega})$ and $H_s(e^{j\omega})$ in the digital filter bank. Given these two filters, can we "solve" for $\phi(t)$ and $\psi(t)$? If so, does this $\psi(t)$ generate an orthonormal basis for L^2 space? This section answers some of these questions. Unlike any other section, we also indicate a sketch of the proof for each major result, in view of the importance of these in modern signal processing theory.

Recall first that under some mild conditions (Section 6.10) we can prove that the filters must satisfy Equations 6.102 and 6.103, if we need to generate wavelet and multiresolution bases successfully. We impose these at the outset. By repeated application of the dilation equation, we obtain $\Phi(\omega) = G_s(e^{j\omega/2})$ $G_s(e^{j\omega/4})\Phi(\omega/4)$. Further indefinite repetition yields an infinite product. Using the condition $\Phi(0) = 1$, which we justified earlier, we obtain the infinite products

$$\Phi(\omega) = \prod_{k=1}^{\infty} G_s(e^{j\omega/2^k}) = G_s(e^{j\omega/2}) \prod_{k=2}^{\infty} G_s(e^{j\omega/2^k}) \tag{6.112a}$$

$$\Psi(\omega) = H_s(e^{j\omega/2}) \prod_{k=2}^{\infty} G_s(e^{j\omega/2^k}) \tag{6.112b}$$

The first issue to be addressed is the convergence of the infinite products above. For this we need to review some preliminaries on infinite products [22,23].

Ideal Bandpass Wavelet Rederived from the Digital Filter Bank: Before we address the mathematical details, let us consider a simple example. Suppose the pair of filters $G_s(e^{j\omega})$ and $H_s(e^{j\omega})$ are ideal brickwall low-pass and high-pass filters in Figure 6.28a. Then we can verify, by making simple sketches of a few terms in Equation 6.112, that the above infinite products yield the functions $\Phi(\omega)$ and $\psi(\omega)$ shown in Figure 6.33. That is, the ideal bandpass wavelet is indeed related to the ideal paraunitary filter bank by means of the above infinite product.

6.11.1 Convergence of Infinite Products

To define convergence of a product of the form $\prod_{k=1}^{\infty} a_k$, consider the sequence $\{p_n\}$ of partial products $p_n = \prod_{k=1}^{\infty} a_k$. If this converges to a (complex) number A with $0 < |A| < \infty$, we say that the infinite product converges to A. Convergence to zero should be defined more carefully to avoid degenerate situations (e.g., if $a_1 = 0$, then $p_n = 0$ for all n regardless of the remaining terms a_k, $k > 1$). We use definition in Ref. [22]. The infinite product is said to converge to zero iff $a_k = 0$ for a finite nonzero number of values of k, and if the product with these a_k deleted converges to a nonzero value.

6.11.1.1 Useful Facts about Infinite Products

1. Whenever $\prod_{k=1}^{\infty} a_k$ converges, it can be shown $a_k \to 1$ as $k \to \infty$. For this reason it is convenient to write $a_k = 1 + b_k$.
2. We say that $\prod_{k=1}^{\infty}(1 + b_k)$ converges absolutely if $\prod_{k=1}^{\infty}(1 + |b_k|)$ converges. Absolute convergence of $\prod_{k=1}^{\infty}(1 + b_k)$ implies its convergence.
3. It can be shown that the product $\prod_{k=1}^{\infty}(1 + |b_k|)$ converges iff the sum $\sum_{k=1}^{\infty}|b_k|$ converges. That is, $\prod_{k=1}^{\infty}(1 + b_k)$ converges absolutely iff $\sum_{k=1}^{\infty} b_k$ converges absolutely.

Example 6.11

The product $\prod_{k=1}^{\infty}(1 + k^{-2})$ converges because $\sum_{k=1}^{\infty} 1/k^2$ converges. Similarly, $\prod_{k=1}^{\infty}(1 - k^{-2})$ converges because it converges absolutely, by the preceding example. The product $\prod_{k=1}^{\infty}(1 + k^{-1})$ does not converge because $\sum_{k=1}^{\infty} 1/k$ diverges. Products such as $\prod_{k=1}^{\infty}(1/k^2)$ do not converge because the terms do not approach unity as $k \to \infty$.

6.11.1.2 Uniform Convergence

A sequence $\{p_n(z)\}$ of functions of the complex variable z converges *uniformly* to a function $p(z)$ on a set $\mathcal{S}$ in the complex plane if the convergence rate is the same everywhere in $\mathcal{S}$. More precisely, if we are given $\varepsilon > 0$ we can find N such that $|p_n(z) - p(z)| < \varepsilon$ for every $z \in \mathcal{S}$, as long as $n \geq N$. The crucial thing is that N depends only on ε and not on z, as long as $z \in \mathcal{S}$. A similar definition applies for functions of real variables.

We say that an infinite product of functions $\prod_{k=1}^{\infty} a_k(z)$ converges at a point z if the sequence of partial products $p_n(z) = \prod_{k=1}^{n} a_k(z)$ converges as described previously. If this convergence of $p_n(z)$ is uniform in a set $\mathcal{S}$, we say that infinite product converges uniformly on $\mathcal{S}$. Uniform convergence has similar advantages, as in the case of infinite summations. For example, if each of the functions $a_k(\omega)$ is continuous on the real interval $[\omega_1, \omega_2]$, then uniform convergence of the infinite product $A(\omega) = \prod_{k=1}^{\infty} a_k(\omega)$ on $[\omega_1, \omega_2]$ implies that the limit $A(\omega)$ is continuous on $[\omega_1, \omega_2]$. We saw above that convergence of infinite products can be related to that of infinite summations. The following theorem [23] makes the connection between uniform convergence of summations and uniform convergence of products.

THEOREM 6.9

Let $b_k(z)$, $k \geq 1$ be a sequence of bounded functions of the complex variable z, such that $\sum_{k=1}^{\infty} |b_k(z)|$ converges uniformly on a compact set $\mathcal{S}$ in the complex z plane. Then, the infinite product $\prod_{k=1}^{\infty}(1 + b_k(z))$ converges uniformly on $\mathcal{S}$. This product is zero for some z_0 iff $1 + b_k(z_0) = 0$ for some k.*

Uniform convergence and analyticity: We know that if a sequence of continuous functions converges uniformly to a function, then the limit is also continuous. A similar result is true for analytic functions. If a sequence $\{f_n(s)\}$ of analytic functions converges uniformly to a function $f(s)$, then $f(s)$ is analytic as well. For a more precise statement of this result see Theorem 10.28 in Ref. [23].

6.11.2 Infinite Product Defining the Scaling Function

Return now to the infinite product (Equation 6.112a). As justified in Section 6.10, assume $G_s(e^{j\omega})$ to be continuous, $G_s(e^{j0}) = 1$, and $\Phi(0) \neq 0$. Note that $G_s(e^{j0}) = 1$ is necessary for the infinite product to converge (because convergence of $\prod_k a_k$ implies that $a_k \to 1$; apply this for $\omega = 0$). The following convergence result is fundamental.

THEOREM 6.10: Convergence of the Infinite Product

Let $G_s(e^{j\omega}) = \sum_{n=-\infty}^{\infty} g_s(n)e^{-j\omega n}$. Assume that $G_s(e^{j0}) = 1$ and $\Sigma_n|ng_s(n)| < \infty$. Then

1. *The infinite product* (*Equation* 6.112a) *converges pointwise for all* ω. *In fact, it converges absolutely for all* ω, *and uniformly on compact sets (i.e., closed bounded sets, such as sets of the form* $[\omega_1,\omega_2]$).
2. *The quantity $G_s(e^{j\omega})$ as well as the limit $\Phi(\omega)$ of the infinite product* (*Equation* 6.112a) *are continuous functions of* ω.
3. *$G_s(e^{j\omega})$ is in L^2.*

Because the condition $\Sigma_n|ng_s(n)| < \infty$ implies $\Sigma_n|g_s(n)| < \infty$, the filter $G_s(e^{j\omega})$ is restricted to be stable, but the above result holds whether $g_s(n)$ is FIR or IIR.

***Sketch of Proof*:** Theorem 6.9 allows us to reduce the convergence of the product to the convergence of an infinite sum. For this we must write $G_s(e^{j\omega})$ in the form $1 - F(e^{j\omega})$ and consider the summation $\sum_{k=1}^{\infty} |F(e^{j\omega/2^{\kappa}})|$. Because, $G_s(e^{j0}) = 1 = \Sigma_n g_s(n)$, we can write $G_s(e^{j\omega}) = 1 - (1 - G_s(e^{j\omega})) = 1 - \Sigma_n g_s(n)(1 - e^{-j\omega n})$. However, $|\Sigma_n g_s(n)(1 - e^{-j\omega n})| \leq 2\Sigma_n|g_s(n) \sin(\omega n/2)| \leq |\omega|\Sigma_n| ng_s(n)|$(use $|\sin x/x| \leq 1$). $\Sigma_n|ng_s(n)|$ is assumed to converge, thus we have $|\Sigma_n g_s(n)(1 - e^{-j\omega n})| \leq c|\omega|$. Using this and the fact that $\sum_{k=1}^{\infty} 2^{-k}$ converges, we can complete the proof of part 1 (by applying Theorem 6.9). $\Sigma_n|ng_s(n)| < \infty$ implies in particular that $g_s(n) \in \ell^1$ therefore, its $\ell^1 - \text{FT}$ $G_s(e^{j\omega})$ is continuous (Section 6.6). The continuity $G_s(e^{j\omega})$, together with uniform convergence of the infinite product, implies that the pointwise limit $\Phi(\omega)$ is also continuous. Finally, because $\ell^1 \subset \ell^2$ (Section 6.6), we have $g_s(n) \in \ell^2$, that is, $G_s(e^{j\omega}) \in L^2[0, 2\pi]$ as well.

6.11.3 Orthonormal Wavelet Basis from Paraunitary Filter Bank

We now consider the behavior of the infinite product $\prod_{k=1}^{\infty} G_s(e^{j\omega/2^k})$, when $G_s(e^{j\omega})$ comes from a paraunitary filter bank. The paraunitary property implies that $G_s(e^{j\omega})$ is power symmetric. If we impose some further mild conditions on $G_s(e^{j\omega})$, the scaling function $\phi(t)$ generates an orthonormal

* For us, a compact set means any closed bounded set in the complex plane or on the real line. Examples are all points on and inside a circle in the complex plane, and the closed interval [a, b] on the real line.

multiresolution basis $\{\phi(t-n)\}$. We can then obtain an orthonormal wavelet basis $\{\psi_{kn}(t)\}$ (Theorems 6.6 and 6.7). The main results are given in Theorems 6.11 through 6.15.

First, we define the truncated partial products $P_n(\omega)$. Because $G_s(e^{j\omega})$ has period 2π, the term $G_s(e^{j\omega/2k})$ has period $2^{k+1}\,\pi$. For this reason the partial product $\prod_{k=1}^{n} G_s(e^{j\omega/2^k})$ has period $2^{n+1}\pi$, and we can regard the region $[-2^n\pi, 2^n\pi]$ to be the fundamental period. Let us truncate the partial product to this region, and define

$$P_n(\omega) = \begin{cases} \prod_{k=1}^{n} G_s(e^{j\omega/2^k}), & \text{for } -2^n\pi \le \omega \le 2^n\pi, \\ 0, & \text{otherwise} \end{cases} \tag{6.113}$$

This quantity will be useful later. We will see that this is in $L^2(R)$, and we can discuss $p_n(t)$, its inverse L^2-FT.

THEOREM 6.11

Let $G_s(e^{j\omega})$ be as in Theorem 6.10. In addition let it be power symmetric; in other words, $|G_s(e^{j\omega})|^2 + G_s(-e^{j\omega})|^2 = 1$. Notice in particular that this implies $G_s(e^{j\pi}) = 0$, because $G_s(e^{j0}) = 1$. Then the following are true:

1. $\int_0^{2\pi} |G_s(e^{j\omega})|^2 d\omega/2\pi = 0.5$.
2. *The truncated partial product $P_n(\omega)$ is in L^2, and $\int_{-\infty}^{\infty} |P_n(\omega)|^2 d\omega/2\pi = 1$ for all n. Further, the inverse L^2-FT, denoted as $p_n(t)$, gives to an orthonormal sequence $\{p_n(t-k)\}$, i.e., $\langle p_n(t-k), p_n(t-i)\rangle = \delta(k-i)$ for any $n \ge 1$.*
3. *The limit $\Phi(\omega)$ of the infinite product (Equation 6.112a) is in L^2; hence, it has an inverse L^2-FT, $\phi(t) \in L^2$. Moreover, $\|\phi(t)\|_2 \le 1$.*

Sketch of Proof: Part 1 follows by integrating both sides of $|G_s(e^{j\omega})|^2 + |G_s(-e^{j\omega})|^2 = 1$. The integral in part 2 is $\int_0^{2^{n+1}\pi} \prod_{k=1}^{n} |G_s(e^{j\omega/2^k})|^2 d\omega/2\pi$, which we can split into two terms such as $\int_0^{2^n\pi} + \int_{2_n\pi}^{2^{n+1}\pi}$. Using the 2π periodicity and the power symmetric property of $G_s(e^{j\omega})$, we obtain $\int |P_n|^2\, d\omega = \int |P_{n-1}|^2 d\omega$. Repeated application of this, together with part 1, yields $\int_{-\infty}^{\infty} |P_n(\omega)|^2 d\omega/2\pi = 1$. The proof of orthonormality of $\{p_n(t-k)\}$ follows essentially similarly by working with the modified integral $\int_{-\infty}^{\infty} |P_n(\omega)|^2 e^{j\omega(k-i)} d\omega/2\pi$, and using the half-band property of $|G_s(e^{j\omega})|^2$.

The third part is the most subtle, and uses Fatou's lemma for Lebesgue integrals (Section 6.6). For this, define $g_n(\omega) = |P_n(\omega)|^2$. Then, $\{g_n(\omega)\}$ is a sequence of nonnegative integrable functions such that $g_n(\omega) \to |\Phi(\omega)|^2$ pointwise for each ω. Because $\int g_n(\omega) d\omega = 2\pi$ (from part 2), Fatou's lemma assures us that $|\Phi(\omega)|^2$ is integrable with integral $\le 2\pi$. This proves part 3.

It is most interesting that the truncated partial products $P_n(\omega)$ give rise to orthonormal sequences $\{p_n(t-k)\}$. This orthonormality is induced by the paraunitary property, more precisely the power symmetry property of $G_s(e^{j\omega})$. This is consistent with the fact that the filter bank type of basis introduced in Section 6.4 is an orthonormal basis for ℓ^2 whenever the filter bank is paraunitary.

As the scaling function $\Phi(\omega)$ is the pointwise limit of $\{P_n(\omega)\}$ as $n \to \infty$, this leads to the hope that $\{\phi(t-k)\}$ is also an orthonormal sequence, so that we can generate a multiresolution and then a wavelet basis as in Theorems 6.6 and 6.7. This, however, is not always true. The crux of the reason is that $\Phi(\omega)$ is only the pointwise limit of $\{P_n(\omega)\}$, and not necessarily the L^2 limit. The distinction is subtle (see below). The pointwise limit property means that for any fixed ω, the function $P_n(\omega)$ approaches $\Phi(\omega)$. The L^2 limit property means that $\int |P_n(\omega) - \Phi(\omega)|^2 d\omega \to 0$. Neither of these limit properties implies the other; neither is stronger than the other. It can be shown that it is the L^2 limit that propagates the orthonormality property, and this is what we want.

THEOREM 6.12

Let $\{p_n\ (t-k)\}$ *be an orthonormal sequence for each n. That is* $\langle p_n(t-k), p_n(t-i)\rangle = \delta\ (k-i)$. *Suppose* $p_n(t) \to \phi(t)$ *in the* L^2 *sense. Then* $\{\phi(t-k)\}$ *is an orthonormal sequence.*

Proof: If we take limits as $n \to \infty$, we can write

$$\lim_{n\to\infty} \langle p_n(t-k),\, p_n(t-i)\rangle = \left\langle \lim_{n\to\infty} p_n(t-k),\ \lim_{n\to\infty} p_n(t-i)\right\rangle \tag{6.114}$$

This movement of the "limit" sign past the inner product sign is allowed (by continuity of inner products, Section 6.6), provided the limits in the second expression are L^2 limits. By the conditions of the theorem, the left side of the above equation is $\delta(k-i)$, whereas the right side is $\langle \phi(t-k), \phi(t-i)\rangle$. So the result follows.

6.11.3.1 L^2 Convergence versus Pointwise Convergence

The fact that L^2 limits are not necessarily pointwise limits is obvious from the fact that differences at a countable set of points do not affect integrals. The fact that pointwise limits are not necessarily L^2 limits is demonstrated by the sequence of L^2 functions $\{f_n(t)\}$, with $f_n(t)$ as in Figure 6.42. Note that $f_n(t) \to 0$ pointwise for each t, that is, the pointwise limit is $f(t) \equiv 0$. Hence, $\| f_n(t) - f(t) \| = \| f_n(t)\| = 1$ for all n, so $\| f_n(t) - f(t)\|$ does not go to zero as $n \to \infty$ and thus, $f(t)$ is not the L^2 limit of $f_n(t)$. Notice in this example that $1 = \lim_{n\to\infty} \int |f_n(t)|^2\, dt \neq \int \lim_{n\to\infty} |f_n(t)|^2\, dt = 0$. This is consistent with the fact that the Lebesgue dominated convergence theorem cannot be applied here—no integrable function dominates $|f_n(t)|^2$ for all n. In this example, the sequence $\{f_n(t)\}$ does not converge in the L^2 sense. In fact, $\|f_n(t) - f_m(t)\|^2 = 2$ for $n \neq m$. Thus, $\{f_n\}$ is not a Cauchy sequence [22] in L^2.

Some facts pertaining to pointwise and L^2 convergences: It can be shown that if $f_n(t) \to f(t)$ in L^2 sense and $f_n(t) \to g(t) \in L^2$ pointwise as well, then $f(t) = g(t)$ a.e. In particular, $\|f(t) - g(t)\| = 0$ and $\|f(t)\| = \|g(t)\|$. It also can be shown that if $f_n(t) \to f(t)$ in L^2 sense, then $\|f_n(t)\| \to \|f(t)\|$. Finally, if $f_n(t) \to f(t) \in L^2$ pointwise a.e., and $\|f_n(t)\| \to \|f(t)\|$ then $f_n(t) \to f\,(t)$ in L^2 sense as well [23].

THEOREM 6.13: Orthonormal Wavelet Basis

Let the filter $G_s(e^{j\omega}) = \sum_{n=-\infty}^{\infty} g_s(n) e^{-j\omega n}$ *satisfy the following properties*:

1. $G_s(e^{j0}) = 1$
2. $\Sigma_n |n g_s(n)| < \infty$
3. $|G_s(e^{j\omega})|^2 + |G_s(-e^{j\omega})|^2 = 1$ (power symmetry)
4. $G_s(e^{j\omega}) \neq 0$ for $\omega \in [-0.5\pi, 0.5\pi]$

Then the infinite product (*Equation* 6.112a) *converges to a limit* $\Phi(\omega) \in L^2$, *and its inverse FT* $\phi(t)$ *is such that* $\{\phi(t-n)\}$ *is an orthonormal sequence. Defining the wavelet function* $\psi(t)$ *as usual, i.e., as in Equation* 6.93, *the sequence* $\{2^{k/2}\, \psi(2^k\, t - n)\}$ (*with k and n varying over all integers*) *forms an orthonormal wavelet basis for* L^2.

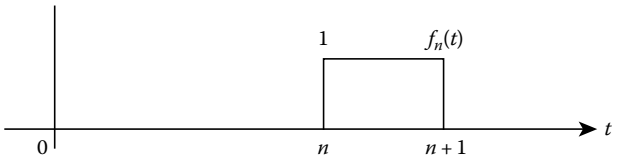


FIGURE 6.42 Sequence $\{f_n(t)\}$ with a pointwise limit is not a limit in the L^2 sense.

Sketch of Proof: We will show that the sequence $\{P_n(\omega)\}$ of partial products converges to $\Phi(\omega)$ in the L^2 sense, i.e., $\int|P_n(\omega)-\Phi(\omega)|^2 d\omega \to 0$, so that $p_n(t) \to \phi(t)$ in L^2 sense. The desired result then follows in view of Theorems 6.11 and 6.12. The key tool in the proof is the dominated convergence theorem for Lebesgue integrals (Section 6.6). First, the condition $G(e^{j\omega}) \neq 0$ in $[-0.5\pi, 0.5\pi]$ implies that $\Phi(\omega) \neq 0$ in $[-\pi, \pi]$. Because $|\Phi(\omega)|^2$ is continuous (Theorem 6.10) it has a minimum value $c^2 > 0$ in $[-\pi, \pi]$. Now the truncated partial product $P_n(\omega)$ can always be written as $P_n(\omega) = \Phi(\omega)/\Phi(\omega/2^n)$ in its region of support. Because $|\Phi(\omega/2^n)|^2 \geq c^2$ in $[-2^n\pi, 2^n\pi]$, we have $|P_n(\omega)|^2 \leq |\Phi(\omega)|^2/c^2$ for all ω. Define $Q_n(\omega) = |P_n(\omega) - \Phi(\omega)|^2$. Then using $|P_n(\omega)|^2 \leq |\Phi(\omega)|^2/c^2$ we can show that $Q_n(\omega) \leq \alpha|\Phi(\omega)|^2$ for some constant α. Because the right-hand side is integrable, and because $Q_n(\omega) \to 0$ pointwise (Theorem 6.10) we can use the dominated convergence theorem (Section 6.6) to conclude that $\lim_n \int Q_n(\omega)d\omega = \int \lim_n Q_n(\omega)\, d\omega = 0$. This completes the proof.

6.11.3.2 Computing the Scaling and Wavelet Functions

Given the coefficients $g_s(n)$ of the filter $G(e^{j\omega})$, how do we compute the scaling function $\Phi(t)$ and the wavelet function $\psi(t)$? Because we can compute $\psi(t)$ using $\psi(t) = 2\sum_{n=-\infty}^{\infty}(-1)^{n+1}g_s^*(-n-1)\phi(2t-n)$, the key issue is the computation of $\phi(t)$. In the preceding theorems $\phi(t)$ was defined only as an inverse L^2–FT of the infinite product $\Phi(\omega)$ given in Equation 6.112a. Because an L^2 function is determined only in the a.e. sense, this way of defining $\phi(t)$ itself does not fully determine $\phi(t)$. Recall, however, that the infinite product for $\Phi(\omega)$ was only a consequence of the more fundamental equation, the dilation equation $\phi(t) = 2\sum_{n=-\infty}^{\infty} g_s(n)\phi(2t-n)$. In practice $\phi(t)$ is computed using this equation, which is often a finite sum (Section 6.12). The procedure is recursive; we assume an initial solution for the function $\phi(t)$, substitute it into the right-hand side of the dilation equation, thereby recomputing $\phi(t)$, and then repeat the process. Details of this and discussions on convergence of this procedure can be found in Refs. [5,15,26].

6.11.3.3 Lawton's Eigenfunction Condition for Orthonormality [5]

Equation 6.100 is equivalent to the orthonormality of $\{\phi(t-n)\}$. Let $S(e^{j\omega})$ denote the left-hand side of Equation 6.100, which evidently has period 2π in ω. Using the frequency domain version of the dilation equation (Equation 6.85), it can be shown that the scaling function $\phi(t)$ generated from $G_S(e^{j\omega})$ is such that

$$|G_s(e^{j\omega})|^2 S(e^{j\omega})\big|_{\downarrow 2} = 0.5S(e^{j\omega}) \tag{6.115}$$

where the notation $\downarrow 2$ indicates decimation (Section 6.4). Thus, the function $S(e^{j\omega})$ can be regarded as an eigenfunction (with eigenvalue = 0.5) of the operator $\mathcal{F}$, which performs filtering by $|G_s(e^{j\omega})|^2$ followed by decimation.

Now consider the case in which the digital filter bank is paraunitary, so that $G_S(e^{j\omega})$ is power symmetric (i.e., satisfies Equation 6.95). The power symmetric condition can be rewritten in the form $|G_s(e^{j\omega})|^2\big|_{\downarrow 2} = 0.5$. Thus, in the power symmetric case the identity function is an eigenfunction of the operator F. If the only eigenfunction of the operation F is the identity function, it then follows that $S(e^{j\omega}) = 1$; i.e., Equation 6.100 holds and $\{\phi(t-n)\}$ is orthonormal.

The FIR Case: Section 6.12 shows that restricting $G_s(z)$ to be FIR ensures that $\phi(t)$ has finite duration. For the FIR case, Lawton and Cohen independently showed that the previous eigenfunction condition also works in the other direction. That is, if $\{\phi(t-n)\}$ has to be orthonormal, then the trignometric polynomial $S(e^{j\omega})$ satisfying Equation 6.115 must be unique up to a scale factor.* Details can be found in Ref. [5].

* A finite sum of the form $\sum_{n=N_1}^{N_2} p_n e^{j\omega n}$ is said to be a trignometric polynomial. If $G_s(e^{j\omega})$ is FIR, it can be demonstrated that the left-hand side of Equation 6.100 is not only periodic in ω, but is in fact a trignometric polynomial.

6.11.3.4 Examples and Counter-Examples

We already indicated after the introduction of Equation 6.112 that the example of the ideal bandpass wavelet can be generated formally by starting from the ideal brickwall paraunitary filter bank. We now discuss some other examples.

Example 6.12: Haar Basis from Filter Banks

A filter bank of the form Figure 6.22a with filters

$$G_a(z) = \frac{1+z^{-1}}{2} \quad H_a(z) = \frac{z^{-1}-1}{2} \quad G_s(z) = \frac{1+z^{-1}}{2} \quad H_s(z) = \frac{1-z^{-1}}{2}$$

is paraunitary. The magnitude responses of the synthesis filters, $|G_s(e^{j\boldsymbol{\omega}})| = |\cos(\boldsymbol{\omega}/2)|$ and $|H_s(e^{j\boldsymbol{\omega}})| = |\sin(\boldsymbol{\omega}/2)|$, are shown in Figure 6.43a. $G_s(z)$ satisfies all the conditions of Theorem 6.13. In this case we can evaluate the infinite products for $\boldsymbol{\Phi}(\boldsymbol{\omega})$ and $\boldsymbol{\psi}(\boldsymbol{\omega})$ explicitly by using the identity $\prod_{m=1}^{\infty} \cos(2^{-m}\boldsymbol{\omega}) = \sin \boldsymbol{\omega}/\boldsymbol{\omega}$. The resulting $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ are as shown in Figure 6.43b and c. These are precisely the functions that generate the Haar orthonormal basis.

Example 6.13: Paraunitary Filter Bank That Does Not Give Orthonormal Wavelets

Consider the filter bank with analysis filters $G_a(z) = (1+z^{-3})/2$, $H_a(z) = -(1-z^{-3})/2$, and synthesis filters $G_s(z) = (1+z^{-3})/2$, $H_s(z) = (1-z^{-3})/2$. Because this is obtained from the preceding example by the substitution $z \rightarrow z^3$, it remains paraunitary and satisfies the PR property. $G_s(z)$ satisfies all the properties of Theorem 6.13, except the fourth condition. With $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ obtained from $G_s(e^{j\boldsymbol{\omega}})$ using the usual dilation equations, the functions $\{\boldsymbol{\phi}(t-n)\}$ are *not orthonormal*. In addition, the wavelet functions $\{2^{k/2}\,\boldsymbol{\psi}(2^k\,t-n)\}$ do not form an orthonormal basis. These statements can be verified from the sketches of the functions $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ shown in Figure 6.44. Clearly, $\boldsymbol{\phi}(t)$ and $\boldsymbol{\phi}(t-1)$ are not orthogonal, and $\boldsymbol{\psi}(t)$ and $\boldsymbol{\psi}(t-2)$ are not orthogonal. In this example, $\|P_n(\boldsymbol{\omega})\| = 1$ for all n, whereas $\|\boldsymbol{\Phi}(\boldsymbol{\omega})\| = 1/\sqrt{3}$. The limit of $\|P_n(\boldsymbol{\omega})\|$ does not agree with $\|\boldsymbol{\Phi}(\boldsymbol{\omega})\|$, and our conclusion is that $\boldsymbol{\Phi}(\boldsymbol{\omega})$ is not the L^2 limit of $P_n(\boldsymbol{\omega})$. The L_2 limit of $P_n(\boldsymbol{\omega})$ does not exist in this example.

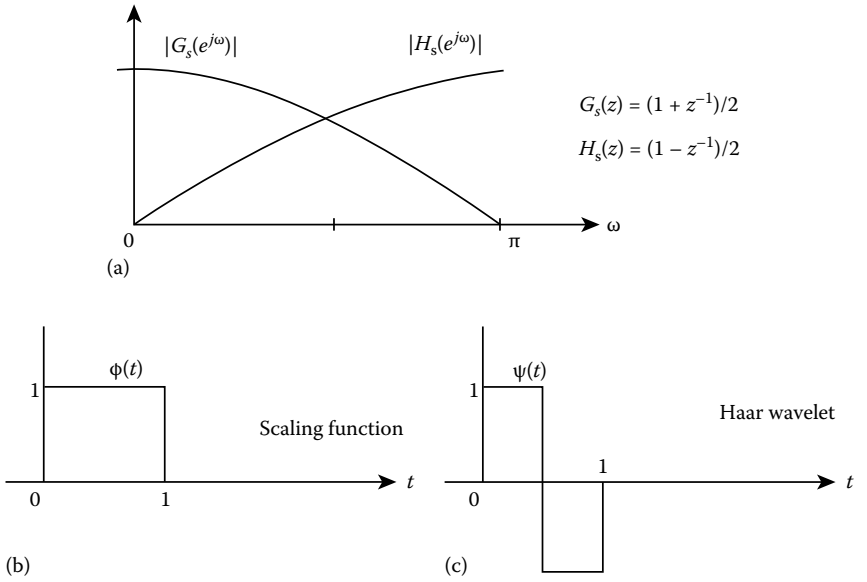


FIGURE 6.43 Haar basis generated from a paraunitary filter bank. (a) The synthesis filters in the paraunitary filter bank, (b) the scaling function, and (c) the wavelet function generated using dilation equations.

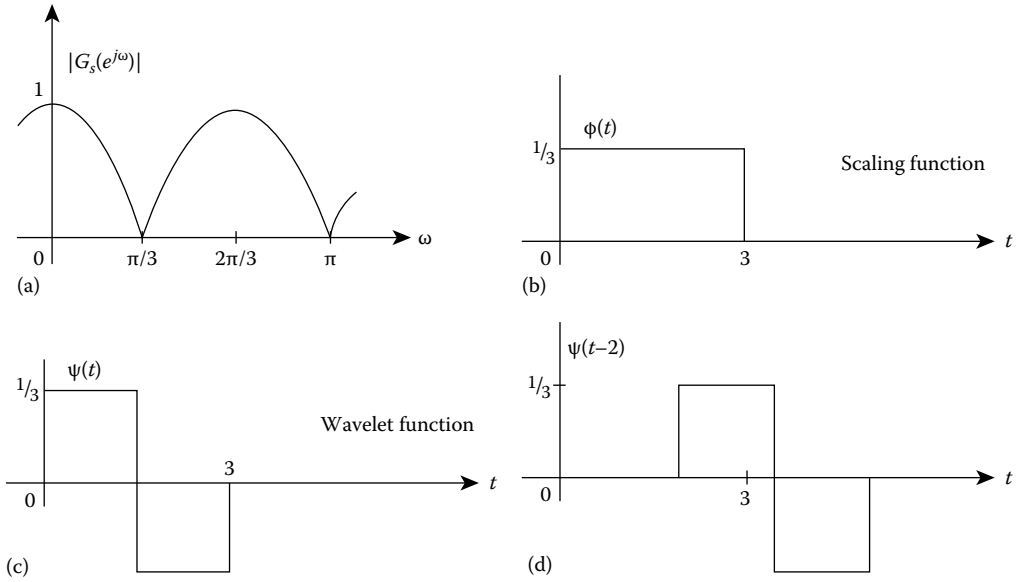


FIGURE 6.44 Paraunitary filter bank generating nonorthonormal $\{\phi(t-n)\}$. (a) The synthesis filter response, (b) the scaling function, (c) the wavelet function, and (d) a shifted version.

Thus, a paraunitary filter bank may not generate an orthonormal wavelet basis if the fourth condition in Theorem 6.13 is violated. However, this is hardly of concern in practice, because any reasonable low-pass filter designed for a two-channel filter bank will be free from zeroes in the region $[-0.5\pi, 0.5\pi]$. In fact, a stronger result was proved by Cohen, who derived necessary and sufficient conditions for an FIR paraunitary filter bank to generate an orthonormal wavelet basis. One outcome of Cohen's analysis is that the fourth condition in Theorem 6.13 can be replaced by the even milder condition that $G_s(e^{j\omega})$ not be zero in $[-\pi/3, \pi/3]$. In this sense, the condition for obtaining an orthonormal wavelet basis is trivially satisfied in practice. The case in which the fourth condition fails is primarily of theoretical interest; an attractive result in this context is Lawton's tight frame theorem.

6.11.4 Wavelet Tight Frames

Although the wavelet functions $\{2^{k/2}\psi(2^k t - n)\}$ generated from a paraunitary filter bank may not form an orthonormal basis when the fourth condition of Theorem 6.13 is violated, the functions always form a tight frame for L^2. Thus, any L^2 function can be expressed as an infinite linear combination of the functions $\{2^{k/2}\psi(2^k t - n)\}$. More precisely, we have the following result due to Lawton [5].

THEOREM 6.14: Tight Frames from Paraunitary Filter Banks

Let $G_s(e^{j\omega}) = \sum_{n=0}^{N} g_s(n)e^{-j\omega n}$ be a filter satisfying the following properties:

1. $G_s(e^{j0}) = 1$
2. $|G_s(e^{j\omega})|^2 + |G_s(-e^{j\omega})|^2 = 1$ (*power symmetry*)

Then, $\phi(t) \in L^2$. *Defining the wavelet function* $\psi(t)$ *as in Equation* 6.93, *the sequence* $\{2^{k/2}\psi(2^k t - n)\}$ (*with k and n varying over all integers*) *forms a tight frame for* L^2, *with frame bound unity* (i.e., $A = B = 1$; *see Section* 6.8).

Thus, the functions $\psi_{kn}(t)$ in Example 6.13 constitute a tight frame for L^2. From Section 6.8, we know that this tight frame property means that any $x(t) \in L^2$ can be expressed as

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \langle x(t), \psi_{kn}(t) \rangle \psi_{kn}(t) \tag{6.116}$$

where $\psi_{kn}(t) = 2^{k/2}\ \psi(2^k\ t - n)$. This expression is pretty much like an expansion into an orthonormal basis. We can find the wavelet coefficients $c_{kn} = \langle x(t), \psi_{kn}(t) \rangle$ exactly as in the orthonormal case. We also know that frames offer stability of reconstruction. Thus, in every respect this resembles an orthonormal basis, the only difference being that the functions are not linearly independent (redundancy exists in the wavelet tight frame $\{\psi_{kn}(t)\}$).

6.12 Compactly Supported Orthonormal Wavelets

Section 6.11 showed how to construct an orthonormal wavelet basis for L^2 space by starting from a paraunitary filter bank. Essentially, we defined two infinite products $\Phi(\omega)$ and $\Psi(\omega)$ starting from the digital low-pass filter $G_s(e^{j\omega})$. Under some mild conditions on $G_s(e^{j\omega})$, the products converge (Theorem 6.10). Under the further condition that $G_s(e^{j\omega})$ be power symmetric and nonzero in $[-0.5, 0.5]$, we saw that $\{\phi(t-k)\}$ forms an orthonormal set, and the corresponding $\{2^{k/2}\ \psi(2^k\ t-n)\}$ forms an orthonormal wavelet basis for L^2 (Theorem 6.13). If we further constrain $G_s(e^{j\omega})$ to be FIR, that is, $G_s(z) = \sum_{n=0}^{N} g_s(n) z^{-n}$, then the scaling function $\phi(t)$ and the wavelet function $\psi(t)$ have finite duration [5,6].

THEOREM 6.15

Let $G_s(z) = \sum_{n=0}^{N} g_s(n) z^{-n}$, with $G_s(e^{j0}) = 1$ and $H_s(e^{j\omega}) = e^{j\omega}\ G_s^{\star}(-e^{j\omega})$. Define the infinite products as in Equation 6.112a and b, and assume that the limits $\Phi(\omega)$ and $\Psi(\omega)$ are L^2 functions, for example, by imposing power symmetry condition on $G_s(z)$ as in Theorem 6.11. Then $\phi(t)$ and $\psi(t)$ (the inverse L^2-FTs) are compactly supported, with support in $[0, N]$.

The time decay of the wavelet $\psi(t)$ is therefore excellent. In particular, all the basis functions $\{2^{k/2}\ \psi(2^k\ t-n)\}$ are compactly supported. By further restricting the low-pass filter $G_s(z)$ to have a sufficient number of zeroes at $\omega = \pi$, we also ensure (Section 6.13) that the FT $\Psi(\omega)$ has excellent decay (equivalently $\psi(t)$ is regular or smooth in the sense to be quantified in Section 6.13).

The rest of this section is devoted to the technical details of the above result. The reader not interested in these details can move to Section 6.13 without loss of continuity. The theorem might seem ''obvious'' at first sight, and indeed a simple engineering argument based on Dirac delta functions can be given (p. 521 of Ref. [7]). However, the correct mathematical justification relies on a number of deep results in function theory. One of these is the celebrated Paley–Wiener theorem for band-limited functions.

Paley–Wiener Theorem: A beautiful result in the theory of signals is that if an L^2 function $f(t)$ is band-limited, that is, $F(\omega) = 0$, $|\omega| \geq \sigma$, then $f(t)$ is the ''real-axis restriction of an entire function.'' We say that a function $f(s)$ of the complex variable s is entire if it is analytic for all s. Examples are polynomials in s, exponentials such as e^s, and simple combinations of these. The function $f(t)$ obtained from $f(s)$ for real values of $s (s = t)$ is the real-axis restriction of $f(s)$.

Thus, if $f(t)$ is a band-limited signal then an entire function $f(s)$ exists such that its real-axis restriction is $f(t)$. In particular, therefore, a band-limited function $f(t)$ is continuous and infinitely differentiable with respect to the time variable t. The entire function $f(s)$ associated with band-limited function has the further

property $|f(s)| \leq ce^{\sigma|s|}$ for some c > 0. We express this by saying that $f(s)$ is exponentially bounded or of the exponential type. What is even more interesting is that the converse of this result is true: if $f(s)$ is an entire function of the exponential type, and the real-axis restriction $f(t)$ is in L^2, then $f(t)$ is band-limited. By interchanging the time and frequency variables, we can obtain similar conclusions for time-limited signals; this is what we need in the discussion of the time-limited (compactly supported) wavelets.

THEOREM 6.16: (Paley–Wiener)

Let $W(s)$ be an entire function such that for all s, we have $|W(s)| \leq c \exp(A|s|)$ for some c, $A > 0$, and the real-axis restriction $W(\omega)$ is in L^2. Then, there exists a function $w(t)$ in L^2 such that $W(s) = \int_{-A}^{A} w(t)e^{-jts}dt$.

A proof can be found in Ref. [23]. Thus, $w(t)$ can be regarded as a compactly supported function support in $[-A, A]$. Recall Equation 6.48 that $L^2[-A, A] \subset L^1[-A, A]$ so $w(t)$ is in $L^1[-A, A]$ and $L^2[-A, A]$. Therefore, $W(\omega)$ is the L^1-FT of $w(t)$, and agrees with the L^2-FT a.e.

Our aim is to show that the infinite product for $\Phi(\omega)$ satisfies the conditions of the Paley–Wiener theorem, and therefore that $\phi(t)$ is compactly supported. A modified version of the previous result is more convenient for this. The modification allows the support to be more general, namely $[-A_1, A_2]$, and permits us to work with the imaginary part of s instead of the absolute value.

THEOREM 6.17: (Paley–Wiener, Modified)

Let $W(s)$ be an entire function such that

$$|W(s)| \leq \begin{cases} c_1 \exp(A_1|\text{Im } s|), & \text{Im } s \geq 0 \\ c_2 \exp(A_2|\text{Im } s|), & \text{Im } s \leq 0 \end{cases} \tag{6.117}$$

for some $c_1, c_2, A_1, A_2 > 0$, and such that the real-axis restriction $W(\omega)$ is in L^2. Then a function $w(t)$ exists in L^2 such that $W(s) = \int_{-A_2}^{A_1} w(t)e^{-jts}dt$. We can regard $W(\omega)$ as the FT of the function $w(t)$ supported in $[-A_2, A_1]$.

This result can be made more general; the condition given in Equation 6.117 can be replaced with one in which the right-hand sides have the form $P_i(s) \exp(A_i|\text{Im } s|)$, where $P_i(s)$ are polynomials. We are now ready to sketch the proof that $\phi(t)$ and $\Psi(t)$ have the compact support $[0, N]$.

1. Using the fact that $G_s(z)$ is FIR and that $G_s(e^{j0}) = 1$, show that the product $\prod_{k=1}^{\infty} G_s(e^{js/2^k})$ converges uniformly on any compact set of the complex s-plane. (For real s, namely $s = \omega$, this holds even for the IIR case as long as $\sum_n |ng_s(n)|$ converges. This was shown in Theorem 6.10.)
2. Uniformity of convergence of the product guarantees that is limit $\Phi(s)$ is an entire function of the complex variable s (Theorem 10.28 [23]).
3. The FIR nature of $G_s(z)$ allows us to establish the exponential bound Equation 6.117 for $\Phi(s)$ with $A_2 = 0$ and $A_1 = N$. This shows that $\phi(t)$ is compactly supported in $[0, N]$. Because $\Psi(t)$ is obtained from the dilation Equation 6.93, the same result follows for $\Psi(t)$ as well.

6.13 Wavelet Regularity

From the preceding section, we know that if we construct the power symmetric FIR filter $G_s(z)$ properly, we can get an orthonormal multiresolution basis $\{\phi(t-n)\}$, and an orthonormal wavelet basis

$\{2^{k/2}\psi(2^k t - n)\}$ for L^2. Both of these bases are compactly supported. These are solutions to the two-scale dilation equations

$$\phi(t) = 2\sum_{n=0}^{N} g_s(n)\phi(2t - n) \tag{6.118}$$

$$\psi(t) = 2\sum_{n=0}^{N} h_s(n)\phi(2t - n) \tag{6.119}$$

where $h_s(n) = (-1)^{n+1} g_s^*(-n-1)$. In the frequency domain we have the explicit infinite product expressions (Equation 6.112) connecting the filters $G_s(z)$ and $H_s(z)$ to the L^2-FTs $\Phi(\omega)$ and $\Psi(\omega)$.

Figure 6.45a shows two cases of a ninth-order FIR filter $G_s(e^{j\omega})$ used to generate the compactly supported wavelet. The resulting wavelets are shown in Figure 6.45b and c. In both cases all conditions of Theorem 6.13 are satisfied so we obtain orthonormal wavelet bases for L^2. The filter $G_s(e^{j\omega})$ has more zeroes at π for case 2 than for case 1. The corresponding wavelet looks much smoother or "regular"; this is an example of a Daubechies wavelet. By designing $G_s(z)$ to have a sufficient number of zeroes at π, we can make the wavelet "as regular as we please." A quantitative discussion of the connection between the number of zeroes at π and the smoothness of $\psi(t)$ is given in the following discussions.

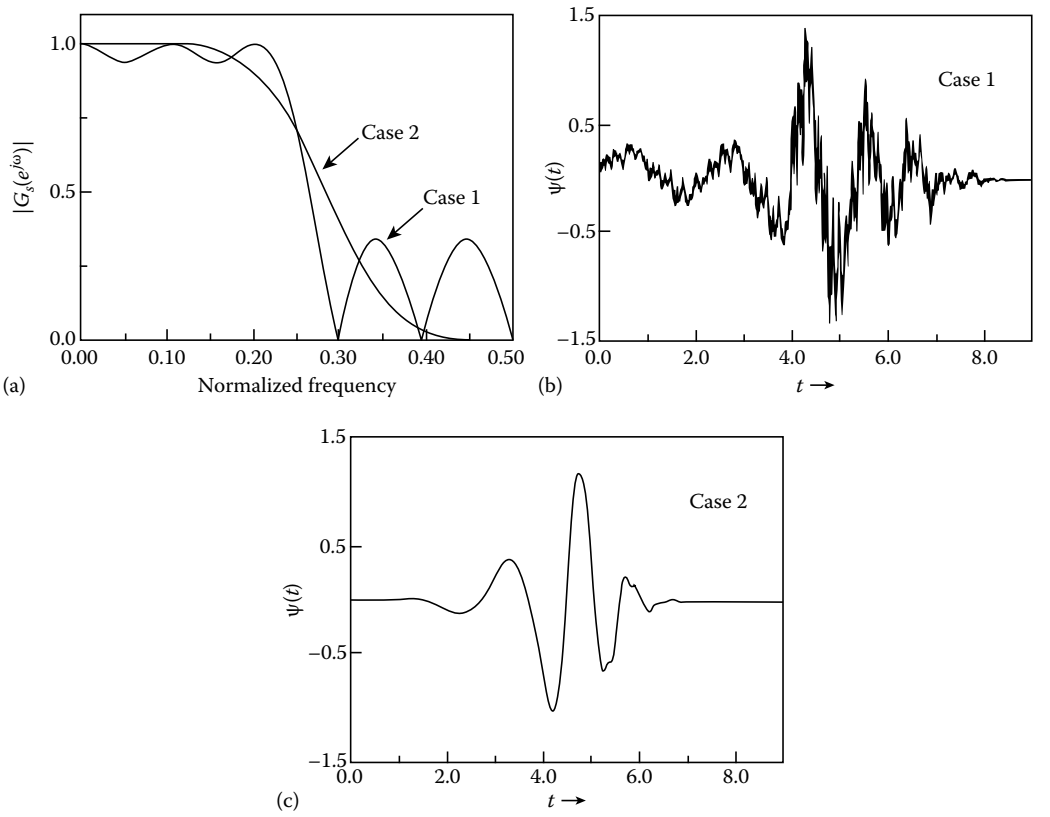


FIGURE 6.45 Demonstrating the importance of zeroes at π. (a) The response of the FIR filter $G_s(z)$ for two cases, and (b) and (c) the corresponding wavelet functions.

Quantitatively, the idea is that if $G_s(e^{j\omega})$ has a large number of zeroes at π, the function $\Phi(\omega)$ given by the infinite product (Equation 6.112a) decays "fast," as $\omega \to \infty$. This fast asymptotic decay in the frequency domain implies that the time function $\phi(t)$ is "smooth." Because $\psi(t)$ is derived from $\phi(t)$ using a finite sum (Equation 6.119), the smoothness of $\phi(t)$ is transmitted to $\psi(t)$. We will the make ideas more quantitative in the next few sections.

Why Regularity?

The point made above was that if we design an FIR paraunitary filter bank with the additional constraint that the low-pass filter $G_s(e^{j\omega})$ should have a sufficient number of zeroes at π, the wavelet basis functions $\Psi_{kn}(t)$ are sufficiently smooth. The smoothness requirement is perhaps the main new component brought into the filter bank theory from the wavelet theory. Its importance can be understood in a number of ways.

Consider the expansion $x(t) = \Sigma_{k,n}\, c_{k,n}\, \Psi_{kn}(t)$. Suppose we truncate this to a finite number of terms, as is often done in practice. If the basis functions are not smooth, the error can produce perceptually annoying effects in applications such as audio and image coding, even though the L^2 norm of the error may be small.

Next, consider a tree-structured filter bank. An example is shown in Figure 6.26. In the syntheses bank, the first path can be regarded as an effective interpolation filter, or an expander (e.g., $\uparrow 8$ in Figure 6.26b) followed by a filter of the form $G_s(e^{j\omega})G_s\,(e^{2j\omega})G_s\,(e^{4j\omega})\cdots G_s\,(e^{2^L\, j\omega})$. The same finite product can be obtained by truncating to $L+1$ terms the infinite product defining $\Phi(\omega)$ (Equation 6.112), and making a change of variables. Similarly, the remaining paths can be related to interpolation filters which are various truncated versions of the infinite product defining $\Psi(\omega)$ in Equation 6.112. Imagine we use the tree-structured system in subband coding. The quantization error in each subband is filtered through an interpolation filter. If the impulse response of the interpolation filter is not smooth enough (e.g., if it resembles Figure 6.45b), the filtered noise tends to show severe perceptual effects, for example, in image reconstruction. This explains, qualitatively, the importance of having "smooth impulse responses" for the synthesis filters.

6.13.1 Smoothness and Hölder Regularity Index

We are familiar with the notion of continuous functions. We say that $f(t)$ is continuous at t_0 if, for any $\varepsilon > 0$, we can find a $\delta > 0$ such that $|f(t) - f(t_0)| < \varepsilon$ for all t satisfying $|t - t_0| < \delta$. A stronger type of continuity, called Hölder continuity, is defined as follows: $f(t)$ is Hölder continuous in a region $\mathcal{S}$ if $|f(t_0) - f(t_1)| \le c|t_0 - t_1|^{\beta}$ for some $c, \beta > 0$, for all $t_0, t_1 \in \mathcal{S}$. This implies, in particular, continuity in the ordinary sense. If $\beta > 1$ the above would imply that $f(t)$ is constant on $\mathcal{S}$. For this reason, we have the restriction $0 < \beta \le 1$. As β increases from 0 to 1, the function becomes increasingly "smoother." The constant β is called the **Lipschitz constant** of the function $f(t)$.

Suppose the function $f(t)$ is n times differentiable in some region $\mathcal{S}$ and the nth derivative $f^{(n)}(t)$ is Hölder continuous with Lipschitz constant β. Define $\alpha = n + \beta$. We say that $f(t)$ belongs to the class C^{α}. The coefficient α is called the Hölder regularity index of $f(t)$. For example, $C^{3,4}$ is the class of functions that are three times differentiable and the third derivatives are Hölder continuous with Lipschitz constant equal to 0.4.

The Hölder regularity index α is taken as a quantitative measure of regularity or smoothness of the function $\psi(t)$. We sometimes say $\psi(t)$ has regularity α. Qualitatively speaking, a function with a large Hölder index is regarded as more "smooth" or "well-behaved." Because the dilation equations in the FIR case are finite summations, the Hölder indices of $\phi(t)$ and $\psi(t)$ are identical.

Some functions are differentiable an infinite number of times. That is, they belong to C^{∞}. Examples are e^t, sin t, and polynomials. C^{∞} functions even exist that are compactly supported (i.e., have finite duration; they will not be discussed here).

6.13.2 Frequency-Domain Decay and Time-Domain Smoothness

We can obtain time-domain smoothness of a certain degree by imposing certain conditions on the FT $\Psi(\omega)$. This is made possible by the fact that the rate of decay of $\Psi(\omega)$ as $\omega \to \infty$ (i.e., the asymptotic decay) governs the Hölder regularity index α of $\psi(t)$. Suppose $\Psi(\omega)$ decays faster than $(1+|\omega|)^{-(1+\alpha)}$:

$$|\Psi(\omega)| \leq \frac{c}{(1+|\omega|)^{1+\alpha+\varepsilon}} \quad \text{for all } \omega \tag{6.120}$$

for some $c > 0$, $\varepsilon > 0$. Then, $\Psi(\omega)(1+|\omega|)^{\alpha}$ is bounded by the integrable function $c/(1+|\omega|)^{1+\varepsilon}$, and is therefore (Lebesgue) integrable. Using standard Fourier theory it can be shown that this implies $\psi(t) \in C^{\alpha}$. In the wavelet construction of Section 6.11, which begins with a digital filter bank, the above decay of $\psi(\omega)$ can be accomplished by designing the digital filter $G_s(e^{j\omega})$ such that it has a sufficient number of zeroes at $\omega = \pi$.

Thus, the decay in the frequency domain translates into regularity in the time domain. Similarly, one can regard time-domain decay as an indication of smoothness in frequency. When comparing two kinds of wavelets, we can usually compare them in terms of time-domain regularity (frequency domain decay) and time-domain decay (frequency domain smoothness). An extreme example is one in which $\psi(t)$ is band limited. This means that $\Psi(\omega)$ is zero outside the passband, and so the "decay" is the best possible. Correspondingly, the smoothness of $\psi(t)$ is excellent; in fact, $\psi(t) \in C^{\infty}$. However, the decay of $\psi(t)$ may not be excellent (certainly it cannot be time limited if it is band limited).

Return to the two familiar wavelet examples, the Haar wavelet (Figure 6.12) and the bandpass wavelet (Figures 6.9 and 6.11). We see that the Haar wavelet has poor decay in the frequency domain because $\Psi(\omega)$ decays only as ω^{-1}. Correspondingly, the time-domain signal $\psi(t)$ is not even continuous, hence, not differentiable.* The bandpass wavelet, on the other hand, is band limited, so the decay in frequency is excellent. Thus, $\psi(t) \in C^{\infty}$, but it decays slowly, behaving similarly to t^{-1} for large t. These two examples represent two extremes of orthonormal wavelet bases for L^2.

The game, therefore, is to construct wavelets that have good decay in time as well as good regularity in time. An extreme hope is where $\psi(t) \in C^{\infty}$, and has compact support as well. It can be shown that such $\psi(t)$ can never give rise to an orthonormal basis, so we must strike a compromise between regularity in time and decay in time.

6.13.2.1 Regularity and Decay in Early Wavelet Constructions

In 1982, Stromberg showed how to construct wavelets in such a way that $\psi(t)$ has exponential decay, and at the same time has arbitrary regularity (i.e., $\psi(t) \in C^k$ for any chosen integer k). In 1985, Meyer constructed wavelets with band-limited $\psi(t)$ (so $\psi(t) \in C^{\infty}$ as for the bandpass wavelet), but he also showed how to design this $\psi(t)$ to decay faster than any chosen inverse polynomial, as $t \to \infty$. Figure 6.46a shows an example of a Meyer wavelet; a detailed description of this wavelet can be found in Ref. [5]. In both of the above constructions the wavelets gave rise to orthonormal bases for L^2.

In 1987 and 1988, Battle and Lemarié independently constructed wavelets with similar properties as Stromberg's wavelets, namely, $\psi(t) \in C^k$ for arbitrary k, and $\psi(t)$ decays exponentially. Their construction is based on spline functions and an orthonormalization step, as described in Section 6.10. The resulting wavelets, while not compactly supported, decay exponentially and generate orthonormal bases. Figure 6.46b shows an example of the Battle–Lemarié wavelet.

Table 6.1 gives a summary of the main features of these early wavelet constructions (first three entries). When these examples were constructed, the relation between wavelets and digital filter banks was not known. The constructions were not systematic or unified by a central theory. Moreover, it was not clear whether one could get a compactly supported (i.e., finite duration) wavelet $\psi(t)$, which at the same time

* It is true that $\psi(t)$ is differentiable almost everywhere, but the discontinuities at the points $t = 0$, 0.5, 1.0 will be very noticeable if we take linear combinations such as $\Sigma_{k,n} c_{kn}\, \psi_{kn}(t)$.

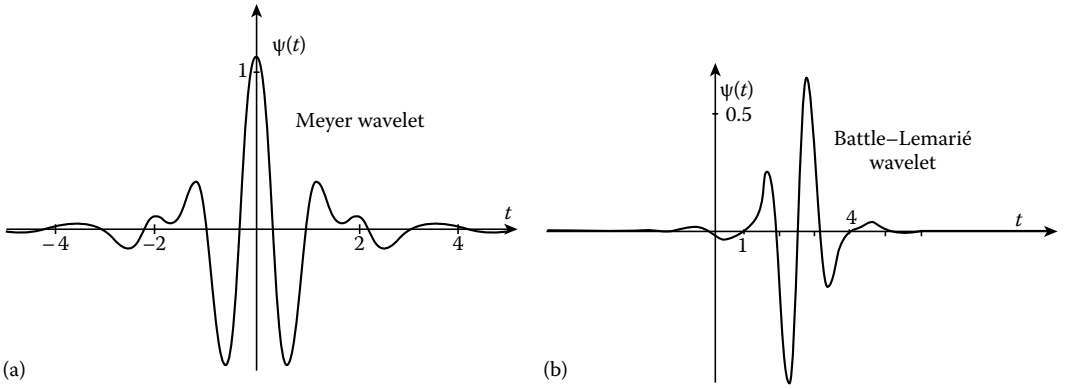


FIGURE 6.46 (a) An example of Meyer wavelet, and (b) an example of Battle–Lemarié wavelet.

TABLE 6.1 Summary of Several Types of Wavelet Bases for $L^2(R)$

Type of Wavelet	Decay of $\psi(t)$ in Time	Regularity of $\psi(t)$ in Time	Type of Wavelet Basis
Stromberg (1982)	Exponential	$\psi(t) \in C^k$; k can be chosen arbitrarily large	Orthonormal
Meyer (1985)	Faster than any chosen inverse polynomial	$\psi(t) \in C^\infty$ (band limited)	Orthonormal
Battle–Lemarié (1987, 1988) (splines)	Exponential	$\psi(t) \in C^k$; k can be chosen arbitrarily large	Orthonormal
Daubechies (1988)	Compactly supported	$\psi(t) \in C^\alpha$;. α can be chosen as large as we please	Orthonormal

had arbitrary regularity (i.e., $\psi(t) \in C^k$ for any chosen k), and generated an orthonormal wavelet basis. This was made possible for the first time when the relation between wavelets and digital filter banks was observed by Daubechies in Ref. [6]. Simultaneously and independently, Mallat invented the multiresolution framework and observed the relation between his framework, wavelets, and paraunitary digital filter banks (the CQF bank, Section 6.4). These discoveries have made the wavelet construction easy and systematic, as described in Sections 6.11 and 6.12. The way to obtain arbitrary wavelet regularity with this scheme is described next.

6.13.3 Time-Domain Decay and Time-Domain Regularity

We now state a fundamental limitation which arises when trying to impose regularity and decay simultaneously [5].

THEOREM 6.18: Vanishing Moments

Let $\{2^{k/2}\,\psi(2^k t - n)\}$, $-\infty \le k, n \le \infty$ *be an orthonormal set in* L^2. *Suppose the wavelet* $\psi(t)$ *satisfies the following properties:*

1. $|\psi(t)| \le c(1 + |t|)^{-(m+1+\varepsilon)}$ *for some integer* m *and some* $\varepsilon > 0$; *that is, the wavelet decays faster than* $(1 + |t|)^{-(m+1)}$.
2. $\psi(t) \in C^m$ (*i.e.,* $\psi(t)$ *differentiable* m *times*), *and the* m *derivatives are bounded.*

Then, the first m *moments of* $\psi(t)$ *are zero, that is,* $\int t^i \psi(t) \mathrm{d}t = 0 \quad \text{for} \quad 0 \le i \le m$.

Impossibility of Compact Support, Infinite Differentiability, and Orthonormality: Suppose we have an orthonormal wavelet basis such that $\psi(t)$ is compactly supported, and infinitely differentiable (i.e., $\psi(t) \in C\infty$). Then all the conditions of Theorem 6.18 are satisfied. So the moments of $\psi(t)$ are zero, and therefore $\psi(t) = 0$ for all t violating the unit-norm property of $\psi(t)$. Thus, we cannot design compactly supported orthonormal wavelets which are infinitely differentiable; only a finite Hölder index can be accomplished. A similar observation can be made even when $\psi(t)$ is not compactly supported as long as it decays faster than any inverse polynomial (e.g., exponential decay).

The vanishing moment condition $\int t^i \, ^{\psi}(t)dt = 0$, $0 \le i \le m$ implies that the L^2-FT $\Psi(\omega)$ has $m+1$ zeroes at $\omega = 0$. This follows by using standard theorems on the L^1-FT [23].* Thus, the first m derivatives of $\Psi(\omega)$ vanish at $\omega = 0$. This implies a certain degree of flatness at $\omega = 0$. Summarizing, we have the following result.

THEOREM 6.19: Flatness in Frequency and Regularity in Time

Suppose we have a compactly supported $\psi(t)$ generating an orthonormal wavelet basis $\{2^{k/2}\ \Psi(2^k t - n)\}$, and let $\psi(t) \in C^m$, with m derivatives bounded. Then, $\psi(\omega)$ has $m+1$ zeroes at $\omega = 0$.

Return now to the wavelet construction technique described in Section 6.11. We started from a paraunitary FIR filter bank (Figure 6.22a) and obtained the scaling function $\phi(t)$ and wavelet function $\psi(t)$ as in Equations 6.118 and 6.119. The FIR nature implies that $\psi(t)$ has compact support (Section 6.12). With the mild conditions of Theorem 6.13 satisfied, we have an orthonormal wavelet basis for L^2. We see that if the wavelet $\psi(t)$ has Hölder index α, it satisfies all the conditions of Theorem 6.19, where m is the integer part of α. Thus, $\psi(\omega)$ has $m+1$ zeroes at $\omega = 0$, but because $\Phi(0) \neq 0$ (Section 6.10), we conclude from the dilation equation $\Psi(\omega) = H_s(e^{j\omega/2})\Phi(\omega/2)$ that the high-pass FIR filter $H_s(z)$ has $m+1$ zeroes at $\omega = 0$ (i.e., at $z = 1$). Using the relation $H_s(e^{j\omega}) = e^{j\omega}\ G_s.\ (-e^{j\omega})$, we conclude that $G_s(e^{j\omega})$ has $m+1$ zeroes at $\omega = \pi$; that is, the low-pass FIR filter $G_s(z)$ has the form $G_s(z) = (1+z^{-1})^{m+1}\ F(z)$, where $F(z)$ is FIR. Summarizing, we have the theorem below.

THEOREM 6.20: Zeroes at π and Regularity

Suppose we wish to design a compactly supported orthonormal wavelet basis for L^2 by designing an FIR filter $G_s(z)$ satisfying the conditions of Theorem 6.13. *If $\psi(t)$ must have the Hölder regularity index α then it is necessary that $G_s(z)$ have the form $G_s(z) = (1+z^{-1})^{m+1}\ F(z)$, where $F(z)$ is FIR, and m is the integer part of α.*

One zero at π is essential. From Theorem 6.10, we know that we must have $G_s(e^{j0}) = 1$, for the infinite product (Equation 6.112a) to converge. Theorem 6.13 imposes further conditions that enable us to obtain an orthonormal wavelet basis for L^2. One of these conditions is the power symmetric property $|G_s(e^{j\omega})|^2 + |G_s(-e^{j\omega})|^2 = 1$. Together with $G_s(e^{j0}) = 1$, this implies $G_s(e^{j\pi}) = 0$. Thus, it is necessary to have at least one zero for $G_s(e^{j\omega})$ at π. The filter that generates the Haar basis (Example 6.12) has exactly one zero at π, but the Haar wavelet $\psi(t)$ is not even continuous. If we desire increased regularity (continuity, differentiability, etc.), we need to put additional zeroes at π, as the above theorem shows.

Design techniques for paraunitary filter banks do not automatically yield filters which have zeroes at π. This condition must be incorporated separately. The maximally flat filter bank solution (Section 6.4) does satisfy this property, and in fact even allows us to specify the number of zeroes at π.

* Because $\psi(t) \in L^2$ and has compact support, $\psi(t) \in L^1$ as well.

6.13.4 Wavelets with Specified Regularity

The fundamental connection between digital filter banks and continuous time wavelets, elaborated in the preceding sections, allows us to construct the scaling function $\phi(t)$ and the wavelet function $\psi(t)$ with specified regularity index α. If $G_s(z)$ has a certain number of zeroes at π, this translates into the Hölder regularity index α. What really matters is not only the number of zeroes at π, but also the order of the FIR filter $G_s(z)$.

For a given order N of the filter $G_s(z)$, suppose we wish to put as many of its zeroes as possible at π. Let this number be K. What is the largest possible K? Not all N zeroes can be at π because we have imposed the power symmetric condition on $G_s(z)$. The best we can do is to put all the unit circle zeroes at π. The power symmetric condition says that $G(z) \triangleq G_s(z)G_s(z)$ is a half-band filter. This filter has order $2N$, with $2K$ zeroes at π. Because we wish to maximize K for fixed N, the solution for $G(z)$ is the maximally flat FIR filter (Figure 6.25), given in Equation 6.45. As the filter in Equation 6.45 has $2K$ zeroes at π and order $2N = 4K - 2$, we conclude that $K = (N+1)/2$. For example, if $G_s(z)$ is a fifth-order power symmetric filter it can have at most three zeroes at π.

The 20% Regularity Rule

Suppose $G_s(z)$ has been designed to be FIR power symmetric of order N, with the number K of zeroes at π adjusted to be maximum (i.e., $K = (N+1)/2$). It can be shown that the corresponding scaling and wavelet functions have a Hölder regularity index $\alpha \approx 0.2\,K$. This estimate is poor for small K, but improves as K grows. Thus, every additional zero at π contributes to $\approx$20% improvement in regularity.

For $K = 4$ (i.e., seventh-order $G_s(z)$), we have $\alpha = 1.275$, which means that the wavelet $\psi(t)$ is once differentiable and the derivative is Hölder continuous with Lipschitz constant 0.275. For $K = 10$ (19th-order $G_s(z)$) we have $\alpha = 2.9$, so the wavelet $\psi(t)$ is twice differentiable and the second derivative has Hölder regularity index 0.9.

Design Procedure: The design procedure is therefore very simple. For a specified regularity index α, we can estimate K and hence $N = 2K - 1$. For this K, we compute the coefficients of the FIR half-band maximally flat filter $G(z)$ using Equation 6.45. From this, we compute a spectral factor $G_s(z)$ of the filter $G(z)$. Tables of the filter coefficients $g_s(n)$ for various values of N can be found in Ref. [5]. From the coefficients $g_s(n)$ of the FIR filter $G_s(z)$, the compactly supported scaling and wavelet functions are fully determined via the dilation equations. These wavelets are called Daubechies wavelets and were first generated in Ref. [6]. Figure 6.45c is an example, generated with a ninth-order FIR filter $G_s(z)$, where response is shown as case 2 in Figure 6.45a.

The above regularity estimates, based on frequency domain behavior, give a single number α, which represents the regularity of $\psi(t)$ for all t. It is also possible to define pointwise or local regularity of the function $\psi(t)$ so that its smoothness can be estimated as a function of time t. These estimation methods, based on time domain iterations, are more sophisticated, but give a detailed view of the behavior of $\psi(t)$. Detailed discussions on obtaining various kinds of estimates for regularity can be found in Refs. [5,26].

6.14 Concluding Remarks

We introduced the WT and studied its connection to filter banks and STFTs. A number of mathematical concepts such as frames and Riesz bases were reviewed and used later for a more careful study of wavelets. We introduced the idea of multiresolution analysis and explained the connections both to filter banks and wavelets. This connection was then used to generate orthonormal wavelet bases from paraunitary filter banks. Such wavelets have compact support when the filter bank is FIR. The regularity or smoothness of the wavelet was quantified in terms of the Hölder exponent. We showed that we can achieve any specified Hölder exponent for compactly supported wavelets by restricting the low-pass filter of the FIR paraunitary filter bank to be a maximally flat power symmetric filter, with a sufficient number of zeroes at π.

6.14.1 Why Wavelets?

Discussions comparing wavelets with other types of time–frequency transforms appear at several places in this chapter. Here is a list of these discussions:

1. Section 6.2 discusses basic properties of wavelets and gives an elementary comparison of wavelet basis with the Fourier basis.
2. Section 6.3 compares the WT with the STFT and shows the time–frequency tilings for both cases (Figures 6.18 and 6.20).
3. Section 6.9 gives a deeper comparison with the STFT in terms of stability properties of the inverse, existence of frames, etc.
4. Section 6.13 presents a comparison to the traditional filter bank design approach. In traditional designs, the appearance of zero(es) at π is not considered important. At the beginning of Section 6.13.1, we discuss the importance of these zeroes in wavelets as well as in tree-structured filter banks.

6.14.2 Further Reading

The literature on wavelet theory and applications is enormous. This chapter is only a brief introduction, concentrating on one-dimensional orthonormal wavelets. Many results can be found on the topics of multidimensional wavelets, biorthogonal wavelets, and wavelets based on IIR filter banks. Two special issues of the *IEEE Transactions* have appeared on the topic thus far [27,28]. Multidimensional wavelets are treated by several authors in the edited volume of Ref. [15], and the filter bank perspective can be found in the work by Kovačevi´cand Vetterli [27]. Advanced results on multidimensional wavelets can be found in Ref. [29]. Advanced results on wavelets constructed from *M*-channel filter banks can be found in the chapter by Gopinath and Burrus in the edited volume of Ref. [15], and in the work by Steffen et al. [28]. The reader can also refer to the collections of chapters in Refs. [15,16], and the many references therein.

Acknowledgments

The authors are grateful to Dr. Ingrid Daubechies, Princeton University, Princeton, New Jersey, for many useful e-mail discussions on wavelets. This work was supported in part by Office of Naval Research grant N00014-93-1-0231, Rockwell International, and Tektronix, Inc.

References

1. M. E. Van Valkenburg, *Introduction to Modern Network Synthesis,* New York: John Wiley & Sons, 1960.
2. A.V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1989.
3. A. Grossman and J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, *SIAM Journal of Mathematical Analysis,* 15, 723–736, 1984.
4. Y. Meyer, *Wavelets and Operators,* Cambridge: Cambridge University Press, 1992.
5. I. Daubechies, *Ten Lectures on Wavelets,* SIAM, CBMS Series, April 1992.
6. I. Daubechies, Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics,* 41, 909–996, November 1988.
7. P. P. Vaidyanathan, *Multirate Systems and Filter Banks,* Englewood Cliffs, NJ: Prentice Hall, 1993.
8. M. Vetterli, A theory of multirate filter banks, *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-35, 356–372, March 1987.

9. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets,* Orlando, FL: Academic Press, 1992.
10. H. S. Malvar, *Signal Processing with Lapped Transforms,* Norwood, MA: Artech House, 1992.
11. S. Mallat, Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$, *Transactions of the American Mathematical Society,* 315, 69–87, September 1989.
12. C. E. Heil and D. F. Walnut, Continuous and discrete wavelet transforms, *SIAM Review,* 31, 628–666, December 1989.
13. M. Vetterli and C. Herley, Wavelets and filter banks, *IEEE Transactions on Signal Processing,* SP-40, 1992.
14. R. A. Gopinath and C. S. Burrus, A tutorial overview of filter banks, wavelets, and interrelations, *Proceedings of IEEE International Symposium on Circuits and Systems,* Chicago, IL, pp. 104–107, May 1993.
15. C. K. Chui, Vol. 1, *An Introduction to Wavelets,* and Vol. 2 (edited), *Wavelets: A Tutorial in Theory and Applications,* Orlando, FL: Academic Press, 1992.
16. J. J. Benedetto and M. W. Frazier, *Wavelets: Mathematics and Applications,* Boca Raton, FL: CRC Press, 1994.
17. J. B. Allen and L. R. Rabiner, A unified theory of short-time spectrum analysis and synthesis, *Proceedings of the IEEE,* 65, 1558–1564, November 1977.
18. M. J. T. Smith and T. P. Barnwell, III, A procedure for designing exact reconstruction filter banks for tree structured subband coders, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* 27.1.1–27.1.4, San Diego, CA, March 1984.
19. F. Mintzer, Filters for distortion-free two-band multirate filter banks, *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-33, 626–630, June 1985.
20. P. P. Vaidyanathan, Theory and design of *M*-channel maximally decimated quadrature mirror filters with arbitrary *M*, having perfect reconstruction property, *IEEE Transactions on Acoustics, Speech, and Signal Processing,* ASSP-35, 476–492, April 1987.
21. V. Belevitch, *Classical Network Theory,* San Francisco: Holden Day, 1968.
22. T. M. Apostol, *Mathematical Analysis,* Reading, MA: Addison-Wesley, 1974.
23. W. Rudin, *Real and Complex Analysis,* New York: McGraw-Hill, 1966.
24. R. M. Young, *An Introduction to Nonharmonic Fourier Series,* New York: Academic Press, 1980.
25. R. J. Duffin and A. C. Schaeffer, A class of nonharmonic Fourier series, *Transactions of the American Mathematical Society,* 72, 341–366, 1952.
26. O. Rioul, Sample regularity criteria for subdivision schemes, *SIAM Journal on Mathematical Analysis,* 23, 1544–1576, November 1992.
27. Special issue on wavelet transforms and multiresolution signal analysis, *IEEE Transactions on Information Theory,* 38, March 1992.
28. Special issue on wavelets and signal processing, *IEEE Transactions on Signal Processing,* 41, December 1993.
29. A. Cohen and I. Daubechies, Non-separable bidimensional wavelet bases, *Revista Matemática IberoAmericana,* 9, 51–137, 1993.

7

Graph Theory

7.1 Introduction 7-1
7.2 Basic Concepts 7-1
7.3 Cuts, Circuits, and Orthogonality 7-7
7.4 Incidence, Circuit, and Cut Matrices of a Graph 7-9
Incidence Matrix • Cut Matrix • Circuit Matrix
7.5 Orthogonality Relation and Ranks of Circuit and Cut Matrices 7-12
7.6 Spanning Tree Enumeration 7-15
7.7 Graphs and Electrical Networks 7-18
7.8 Tellegen's Theorem and Network Sensitivity Computation 7-21
7.9 Arc Coloring Theorem and the No-Gain Property 7-25
References 7-28

Krishnaiyan Thulasiraman
University of Oklahoma

7.1 Introduction

Graph theory had its beginning in Euler's solution of what is known as the Konigsberg Bridge problem. Kirchhoff developed the theory of trees in 1847 as a tool in the study of electrical networks. This was the first application of graph theory to a problem in physical science. Electrical network theorists have since played a major role in the phenomenal advances of graph theory that have taken place. A comprehensive treatment of these developments may be found in Ref. [1]. In this chapter, we develop most of those results which form the foundation of graph theoretic study of electrical networks.

Our development of graph theory is self-contained, except for the definitions of standard set-theoretic operations and elementary results from matrix theory. We wish to note that the ring sum of two sets S_1 and S_2 refers to the set consisting of all those elements which are in S_1 or in S_2 but not in both S_1 and S_2.

7.2 Basic Concepts

A graph $G = (V, E)$ consists of two sets: a finite set $V = (v_1, v_2, \ldots, v_n)$ of elements called vertices and a finite set $E = (e_1, e_2, \ldots, e_m)$ of elements called edges. Each edge is identified with a pair of vertices. If the edges of G are identified with ordered pairs of vertices, then G is called a directed or an oriented graph. Otherwise G is called an undirected or a nonoriented graph. Graphs are amenable for pictorial representations. In a pictorial representation each vertex is represented by a dot and each edge is represented by a line segment joining the dots associated with the edge. In directed graphs, we assign an orientation or direction to each edge. If the edge is associated with the ordered pair (v_i, v_j), then this edge is oriented from v_i to v_j. If an edge e connects vertices v_i and v_j, then it is denoted by $e = (v_i, v_j)$. In a directed graph, (v_i, v_j) refers to an edge directed from v_i to v_j. An undirected graph and a directed graph are shown in Figure 7.1. Unless explicitly stated, the term "graph" may refer to an undirected graph or to a directed graph.

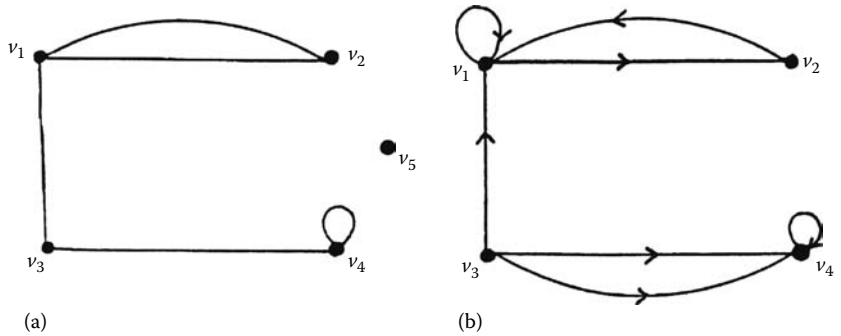


FIGURE 7.1 (a) An undirected graph; (b) a directed graph.

The vertices v_i and v_j associated with an edge are called the end vertices of the edge. All edges having the same pair of end vertices are called parallel edges. In a directed graph, parallel edges refer to edges connecting the same pair of vertices v_i and v_j oriented in the same direction from v_i to v_j or from v_j to v_i. For instance, in the graph of Figure 7.1a, the edges connecting v_1 and v_2 are parallel edges. In the directed graph of Figure 7.1b the edges connecting v_3 and v_4 are parallel edges. However, the edges connecting v_1 and v_2 are not parallel edges because they are not oriented in the same direction. If the end vertices of an edge are not distinct, then the edge is called a self-loop. The graph of Figure 7.1a has one self-loop and the graph of Figure 7.1b has two self-loops.

An edge is said to be incident on its end vertices. In a directed graph the edge (v_i, v_j) is said to be incident out of v_i and is said to be incident into v_j. Vertices v_i and v_j are adjacent if an edge connects v_i and v_j.

The number of edges incident on a vertex v_i is called the degree of v_i and is denoted by $d(v_i)$. In a directed graph $d_{\text{in}}(v_i)$ refers to the number of edges incident into vertex v_i, and it is called the in-degree of v_i. $d_{\text{out}}(v_i)$ refers to the number of edges incident out of vertex v_i, and it is called the out-degree of v_i. If $d(v_i) = 0$, then v_i is called an isolated vertex. If $d(v_i) = 1$, then v_i is called a pendant vertex. A self-loop at a vertex v_i is counted twice while computing $d(v_i)$. As an example, in the graph of Figure 7.1a, $d(v_1) = 3$, $d(v_4) = 3$, and v_5 is an isolated vertex. In the directed graph of Figure 7.1b $d_{\text{in}}(v_1) = 3$, $d_{\text{out}}(v_1) = 2$.

Note that in a directed graph, for every vertex v_i,

$$d(v_i) = d_{\text{in}}(v_i) + d_{\text{out}}(v_i)$$

THEOREM 7.1

1. *The sum of the degrees of the vertices of a graph G is equal to 2m, where m is the number of edges of G.*
2. *In a directed graph with m edges, the sum of the in-degrees and the sum of the out-degrees are both equal to m.*

Proof

1. Because each edge is incident on two vertices, it contributes 2 to the sum of the degrees of G. Hence, all edges together contribute $2m$ to the sum of the degrees.
2. Proof follows if we note that each edge is incident out of exactly one vertex and incident into exactly one vertex.

THEOREM 7.2

The number of vertices of odd degree in any graph is even.

Proof: By Theorem 7.1, the sum of the degrees of the vertices is even. Thus, the sum of the odd degrees must be even. This is possible only if the number of vertices of odd degree is even.

Consider a graph $G = (V, E)$. The graph $G' = (V', E')$ is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. If every vertex in V' is an end vertex of an edge in E', then G' is called the induced subgraph of G on E'. As an example, a graph G and two subgraphs of G are shown in Figure 7.2.

In a graph G, a path P connecting vertices v_i and v_j is an alternating sequence of vertices and edges starting at v_i and ending at v_j with all vertices except v_i and v_j being distinct. In a directed graph, a path P connecting vertices v_i and v_j is called a directed path from v_i to v_j if all the edges in P are oriented in the same direction as we traverse P from v_i toward v_j. If a path starts and ends at the same vertex, it is called a circuit.* In a directed graph, a circuit in which all the edges are oriented in the same direction is called a directed circuit. It is often convenient to represent paths and circuits by the sequence of edges representing them.

For example, in the undirected graph of Figure 7.3a P: e_1, e_2, e_3, e_4 is a path connecting v_1 and v_5 and C: e_1, e_2, e_3, e_4, e_5, e_6 is a circuit. In the directed graph of Figure 7.3b P: e_1, e_2, e_7, e_5 is a directed path and C: e_1, e_2, e_7, e_6 is a directed circuit. Note that e_7, e_5, e_4, e_1, e_2 is a circuit in this directed graph, although it is not a directed circuit.

Two vertices v_i and v_j are said to be connected in a graph G if a path in G connects v_i and v_j. A graph G is connected if every pair of vertices in G is connected; otherwise, it is a disconnected graph. For example, the graph G in Figure 7.4a is connected, but the graph in Figure 7.4b is not connected.

A connected subgraph $G' = (V', E')$ of a graph $G = (V, E)$ is a component of G if adding to G' an edge $e \in E - E'$ results in a disconnected graph. Thus, a connected graph has exactly one component. For example, the graph in Figure 7.4b is not connected and has two components.

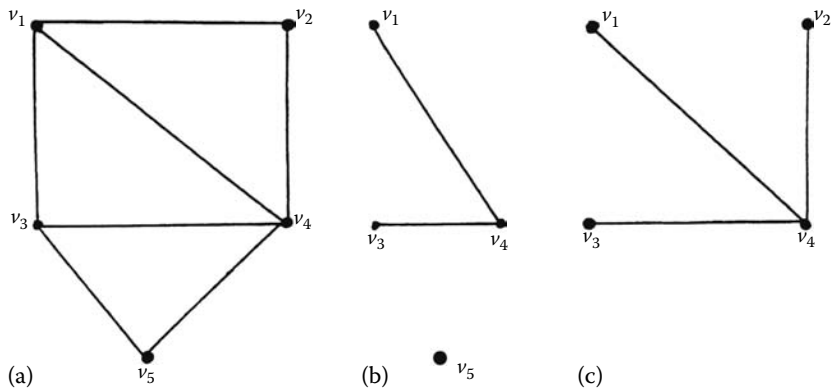


FIGURE 7.2 (a) Graph G; (b) subgraph of G; (c) an edge-induced subgraph of G.

* In electrical network theory literature, the term "loop" is also used to refer to a circuit.

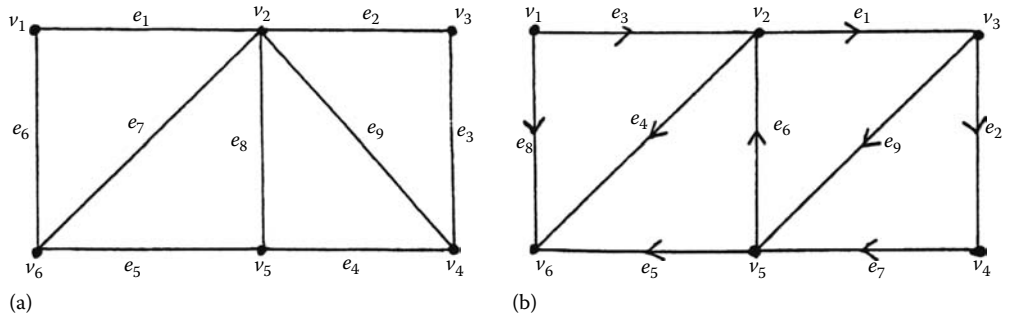


FIGURE 7.3 (a) An undirected graph; (b) a directed graph.

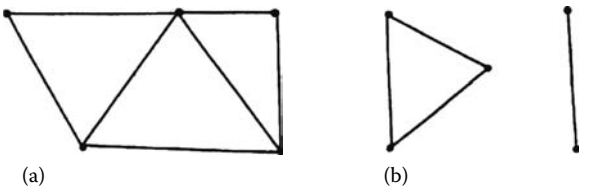


FIGURE 7.4 (a) A connected graph; (b) a disconnected graph.

A tree is a graph that is connected and has no circuits. Consider a connected graph G. A subgraph of G is a spanning tree* of G if the subgraph is a tree and contains all the vertices of G. A tree and a spanning tree of the graph of Figure 7.5a are shown in Figure 7.5b and c, respectively.

The edges of a spanning tree T are called the branches of T. Given a spanning tree of a connected graph G, the cospanning tree* relative to T is the subgraph of G induced by the edges that are not present in T. For example, the cospanning tree relative to the spanning tree T of Figure 7.5c consists of the edges e_3, e_6, and e_7. The edges of a cospanning tree are called chords.

A subgraph of a graph G is a k-tree of G if the subgraph has exactly k components and has no circuits. For example, a 2-tree of the graph of Figure 7.5a is shown in Figure 7.5d. If a graph has k components, then a forest of G is a spanning subgraph that has k components and no circuits. Thus, each component of the forest is a spanning tree of a component of G. A graph G and a forest of G are shown in Figure 7.6.

Consider a directed graph G. A spanning tree T of G is called a directed spanning tree with root v_i if T is a spanning tree of G, and $d_{\text{in}}(v_i) = 0$ and $d_{\text{in}}(v_j) = 1$ for all $v_j \neq v_i$. A directed graph G and a directed spanning tree with root v_1 are shown in Figure 7.7.

It can easily be verified that, in a tree, exactly one path connects any two vertices.

THEOREM 7.3

A tree on n vertices has $n - 1$ edges.

Proof: Proof is by induction on the number of vertices of the tree. Clearly, the result is true if a tree has one or two vertices. Assume that the result is true for trees on $n \geq 2$ or fewer vertices. Consider now a

* In electrical network theory literature, the terms "tree" and "cotree" are usually used to mean spanning tree and cospanning tree, respectively.

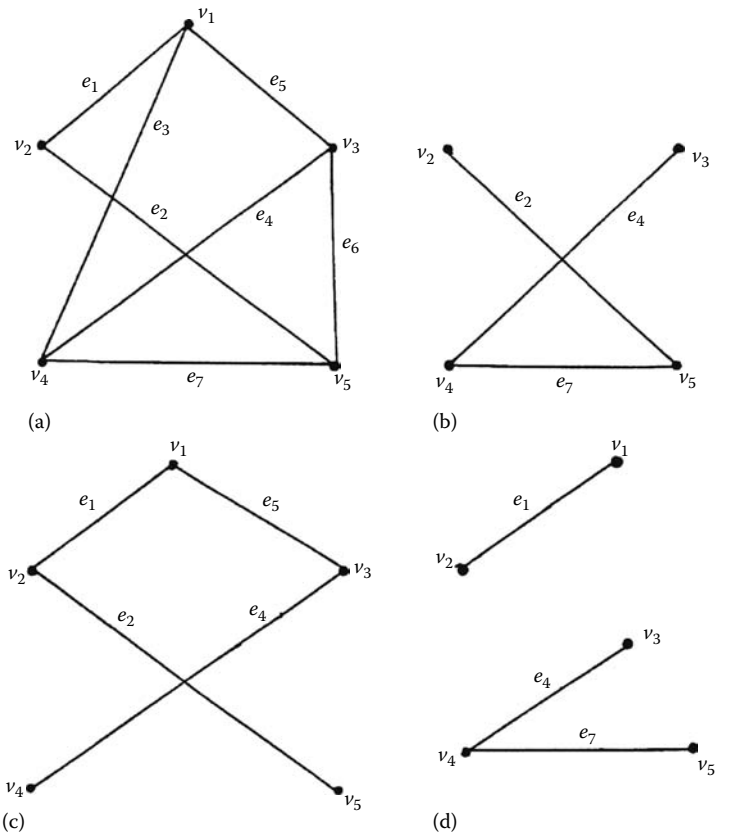


FIGURE 7.5 (a) Graph G; (b) a tree of graph G; (c) a spanning tree of G; (d) a 2-tree of G.

tree T on $n+1$ vertices. Pick an edge $e=(v_i, v_j)$ in T. Removing e from T would disconnect it into exactly two components T_1 and T_2. Both T_1 and T_2 are trees. Let n_1 and m_1 be the number of vertices and the number of edges in T_1, respectively. Similarly n_2 and m_2 are defined. Then, by the induction hypothesis

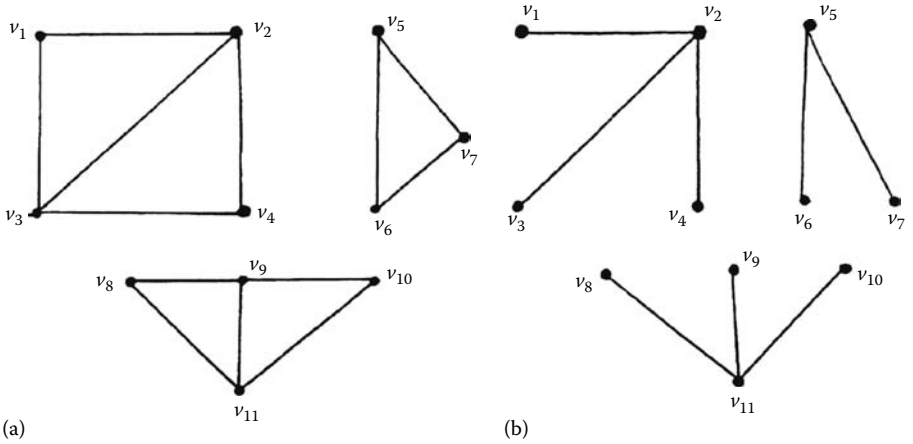


FIGURE 7.6 (a) Graph G; (b) a forest of G.

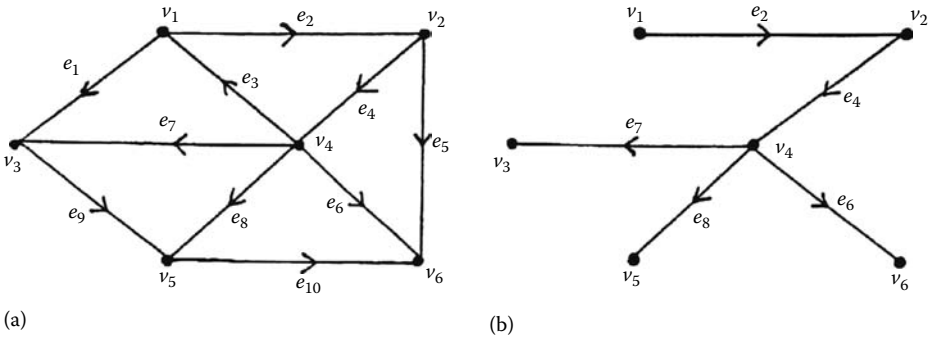


FIGURE 7.7 (a) Directed graph G; (b) a directed spanning tree of G with root v_1.

$$m_1 = n_1 - 1$$

and

$$m_2 = n_2 - 1$$

Thus, the number m of edges in T is given by

$$\begin{aligned} m &= m_1 + m_2 + 1 \\ &= (n_1 - 1) + (n_2 - 1) + 1 \\ &= n_1 + n_2 - 1 \\ &= n - 1 \end{aligned}$$

This completes the proof of the theorem.

If a connected graph G has n vertices, m edges, and k components, then the rank ρ and nullity μ of G are defined as follows:

$$\rho(G) = n - k \tag{7.1}$$

$$\mu(G) = m - n + k \tag{7.2}$$

Clearly, if G is connected, then any spanning tree of G has $\rho = n - 1$ branches and $\mu = m - n + 1$ chords.

We conclude this section with the following theorems. Proof of these theorems may be found in Ref. [2].

THEOREM 7.4

A tree on $n \geq 2$ vertices has at least two pendant vertices.

THEOREM 7.5

A subgraph of an n-vertex connected graph G is a spanning tree of G if and only if the subgraph has no circuits and has $n - 1$ edges.

THEOREM 7.6

If a subgraph G' of a connected graph G has no circuits then there exists a spanning tree of G that contains G'.

7.3 Cuts, Circuits, and Orthogonality

We introduce here the notions of a cut and a cutset and develop certain results which bring out the dual nature of circuits and cutsets.

Consider a connected graph $G = (V, E)$ with n vertices and m edges. Let V_1 and V_2 be two mutually disjoint nonempty subsets of V such that $V = V_1 \approx V_2$. Thus, $V_2 = \overline{V}_1$, the complement of V_1 in V. V_1 and V_2 are also said to form a partition of V. Then the set of all those edges which have one end vertex in V_1 and the other in V_2 is called a cut of G and is denoted by $\langle V_1, V_2 \rangle$. As an example, a graph G and a cut $\langle V_1, V_2 \rangle$ of G are shown in Figure 7.8.

The graph G' which results after removing the edges in a cut will have at least two components and so will not be connected. G' may have more than two components. A cutset S of a connected graph G is a minimal set of edges of G such that removal of S disconnects G into exactly two components. Thus, a cutset is also a cut. Note that the minimality property of a cutset implies that no proper subset of a cutset is a cutset.

Consider a spanning tree T of a connected graph G. Let b be a branch of T. Removal of the branch b disconnects T into exactly two components, T_1 and T_2. Let V_1 and V_2 denote the vertex sets of T_1 and T_2, respectively. Note that V_1 and V_2 together contain all the vertices of G. We can verify that the cut $\langle V_1, V_2 \rangle$ is a cutset of G and is called the fundamental cutset of G with respect to branch b of T. Thus, for a given connected graph G and a spanning tree T of G, we can construct $n - 1$ fundamental cutsets, one for each branch of T. As an example, for the graph shown in Figure 7.8, the fundamental cutsets with respect to the spanning tree $T = [e_1, e_2, e_6, e_8]$ are

Branch e_1: (e_1, e_3, e_4)
Branch e_2: (e_2, e_3, e_4, e_5)
Branch e_6: (e_6, e_4, e_5, e_7)
Branch e_8: (e_8, e_7)

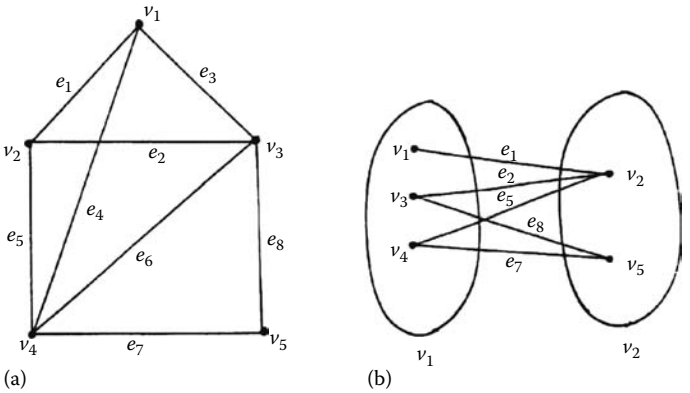


FIGURE 7.8 (a) Graph G; (b) cut $\langle V_1, V_2 \rangle$ of G.

Note that the fundamental cutset with respect to branch b contains b. Furthermore, the branch b is not present in any other fundamental cutset with respect to T.

Next, we identify a special class of circuits of a connected graph G. Again, let T be a spanning tree of G. Because exactly one path exists between any two vertices of T, adding a chord c to T produces a unique circuit. This circuit is called the fundamental circuit of G with respect to chord c of T. Note again that the fundamental circuit with respect to chord c contains c, and the chord c is not present in any other fundamental circuit with respect to T. As an example, the set of fundamental circuits with respect to the spanning tree $T = (e_1, e_2, e_6, e_8)$ of the graph shown in Figure 7.8 is

Chord e_3: (e_3,e_1,e_2)
Chord e_4: (e_4,e_1,e_2,e_6)
Chord e_5: (e_5,e_2,e_6)
Chord e_7: (e_7,e_8,e_6)

We now present a result that is the basis of what is known as the orthogonality relationship.

THEOREM 7.7

A circuit and a cutset of a connected graph have an even number of common edges.

Proof: Consider a circuit C and a cutset $S = \langle V_1, V_2 \rangle$ of G. The result is true if C and S have no common edges. Suppose that C and S possess some common edges. Let us traverse the circuit C starting from a vertex, e.g., v_1 and V_1. Because the traversing should end at v_1, it is necessary that every time we encounter an edge of S leading us from V_1 to V_2 an edge of S must lead us from V_2 back to V_1. This is possible only if S and C have an even number of common edges.

The above result is the foundation of the theory of duality in graphs. Several applications of this simple result are explored in different parts of this chapter.

A comprehensive treatment of the duality theory and its relationship to planarity may be found in Ref. [2]. The following theorem establishes a close relationship between fundamental circuits and fundamental cutsets.

THEOREM 7.8

1. *The fundamental circuit with respect to a chord of a spanning tree T of a connected graph consists of exactly those branches of T whose fundamental cutsets contain the chord.*
2. *The fundamental cutset with respect to a branch of a spanning tree T of a connected graph consists of exactly those chords of T whose fundamental circuits contain the branch.*

Proof: Let C be the fundamental circuit of a connected graph G with respect to a chord c of a spanning tree T of G. Let C contain, in addition to the chord c, the branches $b_1, b_2, \ldots, b_k$ of T. Let S_i be the fundamental cutset with respect to branch b_i.

We first show that each S_i, $1 \leq i \leq k$ contains c. Note that b_i is the only branch common to S_i and C, and c is the only chord in C. Because by Theorem 7.7, S_i and C must have an even number of common edges, it is necessary that S_i contains c.

Next, we show that no other fundamental cutset of T contains c. Suppose the fundamental cutset S_{k+1} with respect to some branch b_{k+1} of T contains c. Then c will be the only edge common to S_{k+1} and C, contradicting Theorem 7.7. Thus the chord c is present only in those cutsets defined by the branches $b_1, b_2, \ldots, b_k$.

The proof for item 2 of the theorem is similar to that of item 1.

7.4 Incidence, Circuit, and Cut Matrices of a Graph

The incidence, circuit, and cut matrices are coefficient matrices of Kirchhoff's equations which describe an electrical network. We develop several properties of these matrices that have proved useful in the study of electrical networks. Our discussions are mainly in the context of directed graphs. The results become valid in the case of undirected graphs if addition and multiplication are in $GF(2)$, the field of integers modulo 2. (Note that $1 + 1 = 0$ in this field.)

7.4.1 Incidence Matrix

Consider a connected directed graph G with n vertices and m edges and have no self-loop. The all-vertex incidence matrix $A_c = [a_{ij}]$ of G has n rows, one for each vertex, and m columns, one for each edge. The element a_{ij} of A_c is defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if the } j\text{th edge is incident out of the } i\text{th vertex} \\ -1, & \text{if the } j\text{th edge is incident into the } i\text{th vertex} \\ 0, & \text{if the } j\text{th edge is not incident on the } i\text{th vertex} \end{cases}$$

A row of A_c will be referred to as an incidence vector. As an example, for the directed graph shown in Figure 7.9, the matrix A_c is

$$A_c = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_5 & e_6 & e_3 & e_4 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \end{array}$$

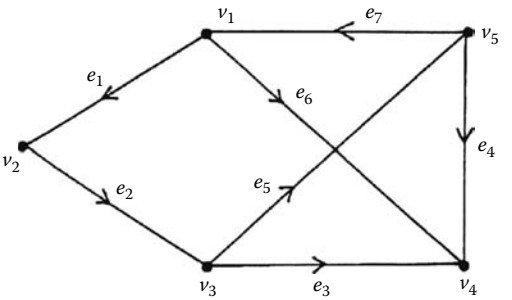


FIGURE 7.9 Directed graph.

From the definition of A_c it should be clear that each column of this matrix has exactly two nonzero entries, one $+1$ and one -1, and therefore, we can obtain any row of A_c from the remaining rows. Thus,

$$\text{rank}(A_c) \leq n - 1 \tag{7.3}$$

An $(n-1)$ rowed submatrix of A_c is referred to as an incidence matrix of G. The vertex which corresponds to the row of A_c that is not in A is called the reference vertex of A.

THEOREM 7.9

The determinant of an incidence matrix of a tree is ± 1.

Proof: Proof is by induction on the number m of edges in the tree. We can easily verify that the result is true for any tree with $m \leq 2$ edges. Assume that the result is true for all trees having $m \geq 2$ or fewer edges. Consider a tree T with $m+1$ edges. Let A be the incidence matrix of T with reference vertex v_r. Because, by Theorem 7.4, T has at least two pendant vertices, we can find a pendant vertex $v_i \neq v_r$. Let (v_i, v_j) be the only edge incident on v_i. Then, the remaining edges form a tree T_1. Let A_1 be the incidence matrix of T_1 with vertex v_r as reference. Now let us rearrange the rows and columns of A so that the first $n-2$ rows correspond to the vertices in T_1 (except v_r) and the first $n-1$ columns correspond to the edges of T_1. Then, we have

$$A = \begin{bmatrix} A_1 & A_3 \\ 0 & \pm 1 \end{bmatrix}$$

So

$$\det A = \pm(\det A_1) \tag{7.4}$$

A_1 is the incidence matrix of T_1 and T_1 has m edges, it follows from the induction hypothesis that det $A_1 = \pm 1$. Hence the theorem.

Because a connected graph has at least one spanning tree, it follows from the above theorem that any incidence matrix A of a connected graph has a nonsingular submatrix of order $n-1$. Therefore,

$$\text{rank}(A_c) \geq n - 1 \tag{7.5}$$

Combining Equations 7.3 and 7.5 yields the following theorem.

THEOREM 7.10

The rank of any incidence matrix of a connected directed graph G is equal to $n-1$, the rank of G.

7.4.2 Cut Matrix

Consider a cut $\langle V_a, \overline{V}_a \rangle$ in a connected directed graph G with n vertices and m edges. Recall that $\langle V_a, \overline{V}_a \rangle$ consists of all those edges connecting vertices in V_a to those in $\overline{V}_a$. This cut may be assigned on orientation from V_a to $\overline{V}_a$ or from $\overline{V}_a$ to V_a. Suppose the orientation of $\langle V_a, \overline{V}_a \rangle$ is from V_a to $\overline{V}_a$. Then the orientation of an edge (v_i, v_j) is said to agree with the cut orientation if $v_i \in V_a$, and $v_j \in \overline{V}_a$.

The cut matrix $Q_c = [q_{ij}]$ of G has m columns, one for each edge, and has one row for each cut. The element q_{ij} is defined as follows:

$$q_{ij} = \begin{cases} 1, & \text{if the } j\text{th edge is in the } i\text{th cut and its orientation agrees with the cut orientation} \\ \pm 1, & \text{if the } j\text{th edge is in the } i\text{th cut and its orientation does not agree with the cut orientation} \\ 0, & \text{if the } j\text{th edge is not in the } i\text{th cut} \end{cases}$$

Each row of Q_c is called a cut vector.

The edges incident on a vertex form a cut. Thus, it follows that the matrix A_c is a submatrix of Q_c. Next we identify another important submatrix of Q_c.

Recall that each branch of a spanning tree T of a connected graph G defines a fundamental cutset. The submatrix of Q_c corresponding to the $n-1$ fundamental cutsets defined by T is called the fundamental cutset matrix Q_f of G with respect to T.

Let $b_1, b_2, \ldots, b_{n-1}$ denote the branches of T. Let us assume that the orientation of a fundamental cutset is chosen so as to agree with that of the defining branch. Suppose we arrange the rows and the columns of Q_f so that the ith column corresponds to branch b_i, and the ith row corresponds to the fundamental cutset defined by b_i. Then, the matrix Q_f can be displayed in a convenient form as follows:

$$Q_f = [U | Q_{fc}] \tag{7.6}$$

where U is the unit matrix of order $n-1$ and its columns correspond to the branches of T.

As an example, the fundamental cutset matrix of the graph in Figure 7.9 with respect to the spanning tree

$$T = (e_1, e_2, e_5, e_6) \text{ is}$$

$$Q_f = \begin{array}{c} \\ e_1 \\ e_2 \\ e_5 \\ e_6 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_5 & e_6 & e_3 & e_4 & e_7 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

It is clear from Equation 7.6 that the rank of Q_f is $n-1$. Hence,

$$\text{rank}(Q_c) \geq n - 1 \tag{7.7}$$

7.4.3 Circuit Matrix

Consider a circuit C in a connected directed graph G with n vertices and m edges. This circuit can be traversed in one of two directions, clockwise or counterclockwise. The direction we choose for traversing C is called the orientation of C. If an edge $e = (v_i, v_j)$ directed from v_i to v_j is in C, and if v_i appears before v_j as we traverse C in the direction specified by its orientation, then we say that the orientation of e agrees with the orientation of C.

The circuit matrix $B_c = [b_{ij}]$ of G has m columns, one for each edge, and has one row for each circuit in G. The element b_{ij} is defined as follows:

$$b_{ij} = \begin{cases} 1, & \text{if the } j\text{th edge is in the i th circuit and its orientation agrees with the circuit orientation} \\ \pm 1, & \text{if the } j\text{th edge is in the } i\text{th circuit and its orientation does not agree with the orientation} \\ 0, & \text{if the } j\text{th edge is not in the } i\text{th circuit} \end{cases}$$

Each row of B_c is called a circuit vector.

The submatrix of B_c corresponding to the fundamental circuits defined by the chords of a spanning tree T is called the fundamental circuit matrix B_f of G with respect to the spanning tree T.

Let $c_1, c_2, \ldots, c_{m-n+1}$ denote the chords of T. Suppose we arrange the columns and the rows of B_f so that the ith row corresponds to the fundamental circuit defined by the chord c_i, and the ith column corresponds to the chord c_i.

If, in addition, we choose the orientation of a fundamental circuit to agree with the orientation of the defining chord, we can write B_f as

$$B_f = [U|B_{ft}] \tag{7.8}$$

where U is the unit matrix of order $m-n+1$, and its columns correspond to the chords of T.

As an example, the fundamental circuit matrix of the graph shown in Figure 7.9 with respect to the tree $T=(e_1, e_2, e_5, e_6)$ is given below:

$$B_f = \begin{array}{c} \\ e_3 \\ e_4 \\ e_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_3 & e_4 & e_7 & e_1 & e_2 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

It is clear from Equation 7.8 that the rank of B_f is $m-n+1$. Hence,

$$\text{rank}(B_c) \geq m-n+1 \tag{7.9}$$

7.5 Orthogonality Relation and Ranks of Circuit and Cut Matrices

THEOREM 7.11

If a cut and a circuit in a directed graph have 2k edges in common, then k of these edges have the same relative orientation in the cut and in the circuit, and the remaining k edges have one orientation in the cut and the opposite orientation in the circuit.

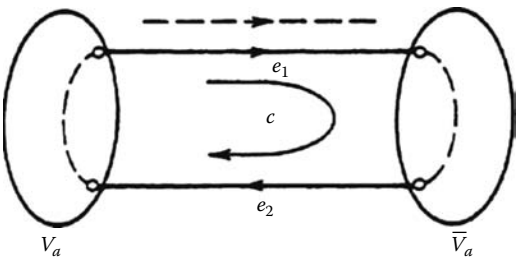


FIGURE 7.10 Relative orientations of an edge in a cut and a circuit.

Proof: Consider a cut $\langle V_a, \overline{V}_a \rangle$ and a circuit C in a directed graph. Suppose we traverse C starting from a vertex in V_a. Then, for every edge e_1 that leads from V_a to $\overline{V}_a$, an edge e_2 leads from $\overline{V}_a$ to V_a. Suppose the orientation of e_1 agrees with the orientation of the cut and that of the circuit. Then we can easily verify that e_2 has one orientation in the cut and the opposite orientation in the circuit (Figure 7.10). On the other hand, we can also verify that if e_1 has one orientation in the cut and the opposite orientation in the circuit, then e_2 will have the same relative orientation in the circuit and in the cut. This proves the theorem.

Next we prove the orthogonality relation.

THEOREM 7.12

If the columns of the circuit matrix B_c and the columns of the cut matrix Q_c are arranged in the same edge order, then

$$B_c Q_c^t = 0 \tag{7.10}$$

Proof: Each entry of the matrix $B_c Q_c^t$ is the inner product of a circuit vector and a cut vector. Suppose a circuit and a cut have $2k$ edges in common. The inner product of the corresponding vectors is zero, because by Theorem 7.11, this product is the sum of k 1's and $k-1$'s.

The orthogonality relation is a profound result with interesting applications in electrical network theory. Consider a connected graph G with m edges and n vertices. Let Q_f be the fundamental cutset matrix and B_f be the fundamental circuit matrix of G with respect to a spanning tree T. If we write Q_f and B_f as in Equations 7.6 and 7.8, then using the orthogonality relation we get

$$B_f Q_f^t = 0$$

that is,

$$[B_{ft} U] \begin{bmatrix} U \\ Q_{fc}^t \end{bmatrix} = 0$$

that is,

$$B_{ft} = -Q_{fc}^t \tag{7.11}$$

Using Equation 7.11, each circuit vector can now be expressed as a linear combination of the fundamental circuit vectors. Consider a circuit vector $\beta = \beta_1, \beta_2, \ldots, \beta_\rho \,|\beta_{\rho+1}, \ldots, \beta_m$ of G where $\rho = n-1$, is the rank of G. Then, again by the orthogonality relation we have

$$\beta Q_f^t = [\beta_1, \beta_2, \ldots, \beta_\rho | \beta_{\rho+1} \cdots \beta_m] \begin{bmatrix} U \\ Q_{fc}^t \end{bmatrix} = 0 \tag{7.12}$$

Therefore,

$$\begin{aligned} [\beta_1, \beta_2, \ldots, \beta_\rho] &= -[\beta_{\rho+1}, \beta_{\rho+2}, \ldots, \beta_m] Q_{fc}^t \\ &= [\beta_{\rho+1}, \beta_{\rho+2}, \ldots, \beta_m] B_{ft} \end{aligned}$$

So,

$$\begin{aligned} [\beta_1, \beta_2, \ldots, \beta_m] &= [\beta_{\rho+1}, \beta_{\rho+2} \cdots \beta_m] \, [\, B_{ft} \quad U \,] \\ &= [\beta_{\rho+1}, \beta_{\rho+2} \cdots \beta_m] B_f \end{aligned} \tag{7.13}$$

Thus, any circuit vector can be expressed as a linear combination of the fundamental circuit vectors. So

$$\text{rank}(B_c) \leq \text{rank}(B_f) = m - n + 1$$

Combining the above with Equation 7.9 we obtain

$$\text{rank}(B_c) = m - n + 1 \tag{7.14}$$

Starting from a cut vector and using the orthogonality relation, we can prove in an exactly similar manner that

$$\text{rank}(Q_c) \leq \text{rank}(Q_f) = n - 1$$

Combining the above with Equation 7.7, we get

$$\text{rank}(Q_c) = n - 1$$

Summarizing, we have the following theorem.

THEOREM 7.13

For a connected graph G with m edges and n vertices

$$\text{rank}(B_c) = m - n + 1$$
$$\text{rank}(Q_c) = n - 1$$

We wish to note from Equation 7.13 that the vector corresponding to a circuit C can be expressed as an appropriate linear combination of the fundamental circuit vectors corresponding to the chords present in C. Similarly, the vector corresponding to a cut can be expressed as an appropriate linear combination of the fundamental cut vectors corresponding to the branches present in the cut. Because modulo 2 addition of two vectors corresponds to the ring sum of the corresponding subgraphs, we have the following results for undirected graphs.

THEOREM 7.14

Let G be a connected undirected graph.

1. *Every circuit can be expressed as a ring sum of the fundamental circuits with respect to a spanning tree.*
2. *Every cut can be expressed as a ring sum of the fundamental cutsets with respect to a spanning tree.*

We can easily verify the following consequences of the orthogonality relation:

1. A linear relationship exists among the columns of the cut matrix (also of the incidence matrix), which correspond to the edges of a circuit.
2. A linear relationship exists among the columns of the circuit matrix, which correspond to the edges of a cut.

The following theorem characterizes the submatrices of A_c, Q_c, and B_c which correspond to spanning trees and cospanning trees. Proof follows from the above results and may be found in Ref. [2].

THEOREM 7.15

Let G be a connected graph G with n vertices, and m edges.

1. *A square submatrix of order $n-1$ of Q_c (also of A_c) is nonsingular iff the edges corresponding to the columns of this submatrix form a spanning tree of G.*
2. *A square submatrix of order $m-n+1$ of B_c is nonsingular iff the edges corresponding to the columns of this submatrix form a cospanning tree of G.*

7.6 Spanning Tree Enumeration

Here, we first establish a formula for counting the number of spanning trees of an undirected graph. We then state a generalization of this result for the case of a directed graph. These formulas have played key roles in the development of topological formulas for electrical network functions. A detailed development of topological formulas for network functions may be found in Swamy and Thulasiraman [1].

The formula for counting the number of spanning trees of a graph is based on Theorem 7.9 and a result in matrix theory, known as the Binet–Cauchy theorem.

A major of a matrix is a determinant of a maximum order. Consider a matrix P of order $p \times q$ and a matrix Q of order $q \times p$, with $p \leq q$. The majors of P and Q are of order p. If a major of P consists of columns $i_1, i_2, \ldots, i_p$, the corresponding major of Q is formed by rows $i_1, i_2, \ldots, i_p$ of Q. For example, if

$$P = \begin{bmatrix} 1 & -2 & -2 & 4 \\ 2 & 3 & -1 & 2 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} -5 & 0 \\ 2 & 1 \\ -2 & 2 \\ 3 & 1 \end{bmatrix}$$

then for the major

$$\begin{vmatrix} -2 & 4 \\ 3 & 2 \end{vmatrix}$$

of P

$$\begin{vmatrix} 2 & 1 \\ 3 & 1 \end{vmatrix}$$

is the corresponding major of Q.

The Binet–Cauchy theorem is stated next. Proof of this theorem may be found in Hohn [3].

THEOREM 7.16

If P is a $p \times q$ matrix and Q is a $q \times p$ matrix, with $p \leq q$, then

$$\det(PQ) = \Sigma \text{ (product of the corresponding majors of } P \text{ and } Q).$$

THEOREM 7.17

Let G be a connected undirected graph and A an incidence matrix of a directed graph obtained by assigning orientations to the edges of G. Then

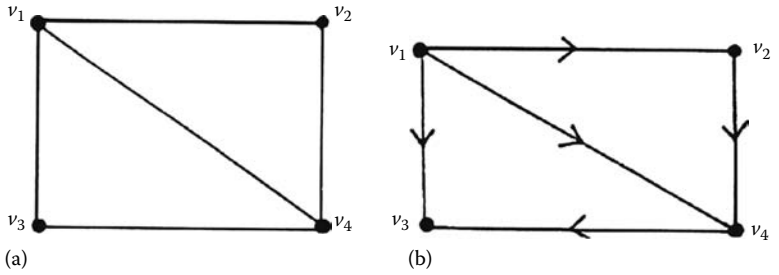


FIGURE 7.11 (a) An undirected graph G; (b) directed graph obtained after assigning arbitrary orientations to the edges of G.

$$\tau(G) = \det(AA^t) \tag{7.15}$$

where $\tau(G)$ is the number of spanning trees of G.

Proof: By the Binet–Cauchy theorem, we have

$$\det(AA^t) = \Sigma \text{ (product of the corresponding majors of } A \text{ and } A^t) \tag{7.16}$$

Recall from Theorem 7.15 that a major of A is nonzero iff the edges corresponding to the columns of the major form a spanning tree of G. Also, the corresponding majors of A and A^t have the same value equal to 0, 1, or -1 (Theorem 7.9). Thus, each nonzero term in the sum on the right-hand side of Equation 7.16 has the value 1, and it corresponds to a spanning tree and vice versa—hence the theorem.

For example, consider the undirected graph G shown in Figure 7.11a. Assigning arbitrary orientations to the edges of G, we obtain the directed graph in Figure 7.11b. If A is the incidence matrix of this directed graph with vertex v_4 as reference vertex then it can be verified that

$$AA^t = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix}$$

and $\det(AA^t) = 8$. Thus, G has eight spanning trees.

An interesting and useful interpretation of the matrix AA^t now follows. Let $v_1, v_2, \ldots, v_n$ be the vertices of an undirected graph G. The degree matrix $K = [k_{ij}]$ of G is an $n \times n$ matrix defined as follows.

$$k_{ij} = \begin{cases} -p, & \text{if } i \neq j \text{ and } p \text{ parallel edges connect } v_i \text{ and } v_j \\ d(v_i), & \text{if } i = j \end{cases}$$

We may easily verify that $K = A_c \cdot A_c^t$, and that it is independent of the choice of orientations for the edges of G. Also, if v_i is the reference vertex, AA^t is obtained by removing row i and column i of K. In other words, $\det(AA^t)$ is the (i, i) cofactor of K. It then follows from Theorem 7.17 that all of the cofactors of K are equal to the number of spanning trees of G. Thus, Theorem 7.17 may be stated in the following form originally presented by Kirchhoff [4].

THEOREM 7.18

All the cofactors of the degree matrix of an undirected graph G have the same value equal to the number of spanning trees of G.

Consider a connected undirected graph G. Let A be the incidence matrix of G with reference vertex v_n. Let $\tau_{i,n}$ denote the number of spanning 2-trees of G such that the vertices v_i and v_n are in different components of these spanning 2-trees. Also, let $\tau_{ij,n}$ denote the number of spanning 2-trees such that vertices v_i and v_j are in the same component, and vertex v_n is in a different component of these spanning 2-trees. If Δ_{ij} denotes the (i, j) cofactor of (AA^t), then we have the following result, proof of which may be found in Ref. [2].

THEOREM 7.19

For a connected graph G,

$$\tau_{i,n} = \Delta_{ii} \tag{7.17}$$

$$\tau_{ij,n} = \Delta_{ij} \tag{7.18}$$

Consider next a directed graph $G = (V, E)$ without self-loops and with $V = (v_1, v_2, \ldots, v_n)$. The in-degree matrix $K = [k_{ij}]$ of G is an $(n \times n)$ matrix defined as follows:

$$k_{ij} = \begin{cases} -p & \text{if } i \neq j \text{ and } p \text{ parallel edges are directed from } v_i \text{ to } v_j \\ d_{\text{in}}(v_i), & \text{if } i = j \end{cases}$$

The following result is due to Tutte [5]. Proof of this result may also be found in Ref. [2].

THEOREM 7.20

Let K be the in-degree matrix of a directed graph G without self-loops. Let the ith row of K correspond to vertex v_i. Then, the number τ_d of directed spanning trees of G having v_r as root is given by

$$\tau_d = \Delta_{rr} \tag{7.19}$$

where Δ_{rr} is the (r, r) cofactor of K.

Note the similarity between Theorem 7.18 and Theorem 7.20.

To illustrate Theorem 7.20, consider the directed graph G shown in Figure 7.12. The in-degree matrix K and G is

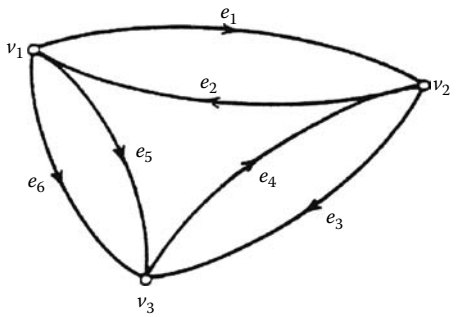


FIGURE 7.12 A directed graph G.

$$K = \begin{bmatrix} 1 & -1 & -2 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

Then

$$\Delta_{11} = \begin{bmatrix} 2 & -1 \\ -1 & 3 \end{bmatrix} = 5$$

The five directed spanning trees of G with vertex v_1 as root are (e_1, e_5), (e_1, e_6), (e_1, e_3), (e_4, e_5), and (e_4, e_6).

7.7 Graphs and Electrical Networks

An electrical network is an interconnection of electrical network elements such as resistances, capacitances, inductances, voltage and current sources, etc. Each network element is associated with two variables, the voltage variable $v(t)$ and the current variable $i(t)$. We also assign reference directions to the network elements (Figure 7.13) so that $i(t)$ is positive whenever the current is in the direction of the arrow, and $v(t)$ is positive whenever the voltage drop in the network element is in the direction of the arrow. Replacing each element and its associated reference direction by a directed edge results in the directed graph representing the network. For example, a simple electrical network and the corresponding directed graph are shown in Figure 7.14.

The physical relationship between the current and voltage variables of a network element is specified by Ohm's law. For voltage and current sources, the voltage and current variables are required to have specified values. The linear dependence among the voltage variables in the network and the linear dependence among the current variables are governed by Kirchhoff's voltage and current laws.

Kirchhoff's Voltage Law (KVL): The algebraic sum of the voltages around any circuit is equal to zero.

Kirchhoff's Current Law (KCL): The algebraic sum of the currents flowing out of a node is equal to zero.

As an example, the KVL equation for the circuit 1, 3, 5 and the KCL equation for the vertex b in the graph of Figure 7.14 are

$$\text{Circuit: } 1, 3, 5 \quad v_1 + v_3 + v_5 = 0$$

$$\text{Vertex } b\text{: } -i_1 + i_2 + i_3 = 0$$

It can easily be seen that KVL and KCL equations for an electrical network N can be conveniently written as

$$A_c I_e = 0 \tag{7.20}$$

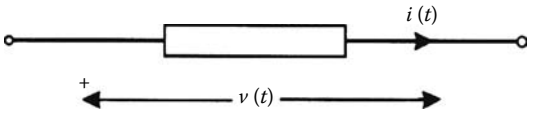


FIGURE 7.13 A network element with reference convention.

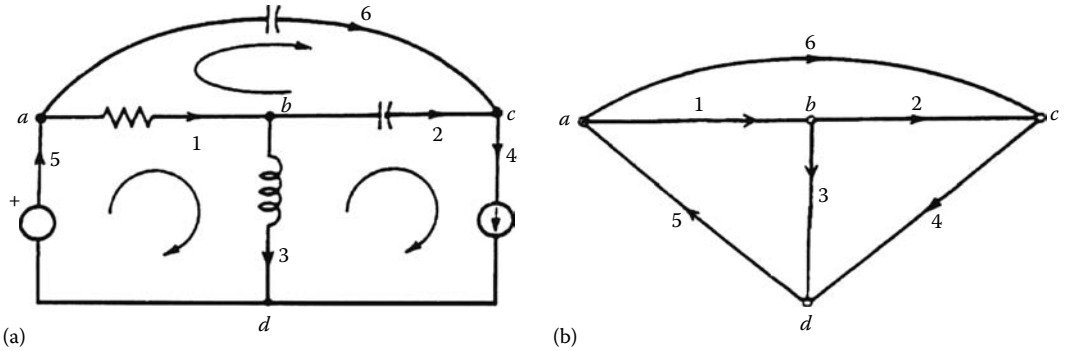


FIGURE 7.14 (a) An electrical network N; (b) directed graph representation of N.

and

$$B_c V_e = 0 \tag{7.21}$$

where

A_c and B_c are, respectively, the incidence and circuit matrices of the directed graph representing N
I_e and V_e are, respectively, the column vectors of element currents and voltages in N

Because each row in the cut matrix Q_c can be expressed as a linear combination of the rows of the matrix, in Equation 7.20 we can replace A_c by Q_c. Thus, we have

$$\text{KCL:} \quad Q_c I_e = 0 \tag{7.22}$$

$$\text{KVL:} \quad B_e V_e = 0 \tag{7.23}$$

From Equation 7.22, we can see that KCL can also be stated as: The algebraic sum of the currents in any cut of N is equal to zero.

If a network N has n vertices, m elements, and its graph is connected then there are only $(n-1)$ linearly independent cuts and only $(m-n+1)$ linearly independent circuits (Theorem 7.13). Thus, in writing KVL and KCL equations we need to use only, B_f, a fundamental circuit matrix and Q_f, a fundamental cutset matrix, respectively. Thus, we have

$$\text{KCL:} \quad Q_f I_e = 0 \tag{7.24}$$

$$\text{KVL:} \quad B_f V_e = 0 \tag{7.25}$$

We note that the KCL and the KVL equations depend only on the way the network elements are interconnected and not on the nature of the network elements. Thus, several results in electrical network theory are essentially graph theoretic in nature. Some of these results and their usefulness in electrical network analysis are presented in the remainder of this chapter. In the following a network N and its directed graph are both denoted by N.

THEOREM 7.21

Consider an electrical network N. Let T be a spanning tree of N, and let B_f and Q_f denote the fundamental circuit and the fundamental cutset matrices of N with respect to T. If I_e and V_e are the column vectors of element currents and voltages and I_c and V_t are, respectively, the column vector of currents associated with the chords of T and the column vector of voltages associated with the branches of T, then

$$\text{Loop transformation:} \quad I_e = B_f^t I_c \tag{7.26}$$

$$\text{Cutset transformation:} \quad V_e = Q_f^t V_t \tag{7.27}$$

Proof: From Kirchhoff's laws, we have

$$Q_f I_e = 0 \tag{7.28}$$

and

$$B_f V_e = 0 \tag{7.29}$$

Let us partition I_e and V_e as

$$I_e = \begin{bmatrix} I_c \\ I_t \end{bmatrix}$$

and

$$V_e = \begin{bmatrix} V_c \\ V_t \end{bmatrix}$$

where the vectors that correspond to the chords and branches of T are distinguished by the subscripts c and t, respectively. Then Equations 7.28 and 7.29 can be written as

$$[\, Q_{fc} \quad U \,] \begin{bmatrix} I_c \\ I_t \end{bmatrix} = 0 \tag{7.30}$$

and

$$[\, U \quad B_{ft} \,] \begin{bmatrix} V_c \\ V_t \end{bmatrix} = 0 \tag{7.31}$$

Recall Equation 7.11 that

$$B_{ft} = -Q_{fc}^t$$

Then, we get from Equation 7.30

$$\begin{aligned} I_t &= -Q_{fc} I_c \\ &= B_{ft}^t I_c \end{aligned}$$

Thus,

$$I_e = \begin{bmatrix} U \\ B_{ft}^t \end{bmatrix} I_c = B_f^t I_c$$

This establishes the loop transformation.

Starting from Equation 7.31, we can show in a similar manner that

$$V_e = Q_f^t V_t$$

thereby establishing the cutset transformation.

In the special case in which the incidence matrix A is used in place of the fundamental cutset matrix, the cutset transformation (Equation 7.27) is called the node transformation. The loop, cutset, and node transformations have been extensively employed to develop different methods of network analysis. The loop method of analysis develops a system of network equations which involve only the chord currents as

variables. The cutset (node) method of analysis develops a system of equations involving only the branch (node) voltages as variables. Thus, the loop and cutset (node) methods result in systems of equations involving $m - n + 1$ and $n - 1$ variables, respectively. In the mixed-variable method of analysis, which is essentially a combination of both the loop and cutset methods, some of the independent variables are currents and the others are voltages. The minimum number of variables required in the mixed-variable method of analysis is determined by what is known as the principal partition of a graph introduced by Kishi and Kajitani in a classic paper [6]. Ohtsuki et al. [7] discuss several issues relating to the mixed-variable method of analysis. A detailed discussion of the principal partition of a graph and the different methods of network analysis including the state-variable method may be found in Ref. [1].

7.8 Tellegen's Theorem and Network Sensitivity Computation

Here, we first present a simple and elegant theorem due to Tellegen [8]. The proof of this theorem is essentially graph theoretic in nature and is based on the loop and the cutset transformations, Equations 7.26 and 7.27, and the orthogonality relation (Theorem 7.12). Using this theorem, we develop the concept of the adjoint of a network and its application in network sensitivity computations.

THEOREM 7.22

Consider two electrical networks, N and $\hat{N}$, such that the graphs associated with them are identical. Let V_e and ψ_e denote the element voltage vectors of N and $\hat{N}$, respectively, and let I_e and Λ_e be the corresponding element current vectors. Then,

$$V_e^t \Lambda_e = 0$$
$$I_e^t \psi_e = 0$$

Proof: If B_f and Q_f are the fundamental circuit and cutset matrices of N (and hence also of $\hat{N}$), then from the loop and cutset transformations, we obtain

$$V_e = Q_f^t V_t$$

and

$$\Lambda_e = B_f^t \Lambda_c$$

So

$$\begin{aligned} V_e^t \Lambda_e &= V_t^t (Q_f B_f^t) \Lambda_c \\ &= 0, \text{ by Theorem 7.12} \end{aligned}$$

Proof of the second part follows in a similar manner.

The adjoint network was introduced by Director and Rohrer [9], and our discussion is based on their work. A more detailed discussion may be found in Ref. [1].

Consider a lumped, linear time-invariant network N. We assume, without loss of generality, that N is a 2-port network. Let $\hat{N}$ be a 2-port network that is topologically equivalent to N. In other words, the graph of $\hat{N}$ is identical to that of N. The corresponding elements of N and $\hat{N}$ are denoted by the same symbol.

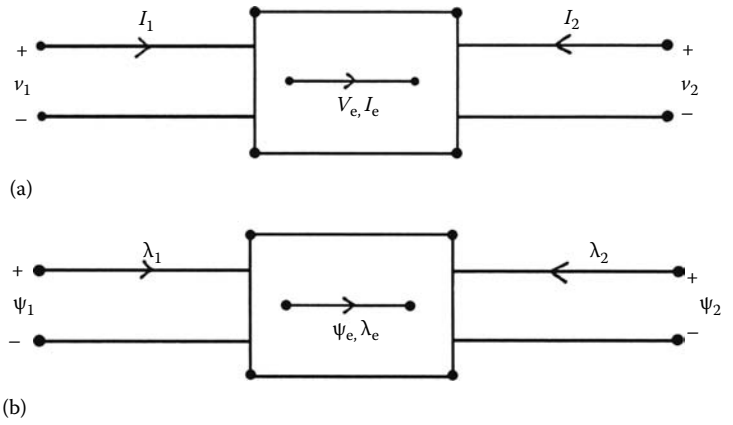


FIGURE 7.15 (a) A 2-port network N; (b) adjoint network $\hat{N}$ of N.

Our goal now is to define the elements of $\hat{N}$ so that $\hat{N}$ in conjunction with N can be used in computing the sensitivities of network functions of N.

Let V_e and I_e denote, respectively, the voltage and the current associated with the element e in N, and ψ_e and λ_e denote, respectively, the voltage and the current associated with the corresponding element e in $\hat{N}$. Also, V_i and I_i, $i = 1, 2$, denote the voltage and current variables associated with the ports of N, and ψ_i and λ_i, $i = 1, 2$, denote the corresponding variables for the ports of $\hat{N}$ (Figure 7.15).

Applying Tellegen's theorem to N and $\hat{N}$, we get

$$V_1\lambda_1 + V_2\lambda_2 = \sum_e V_e\lambda_e \tag{7.32}$$

and

$$I_1\psi_1 + I_2\psi_2 = \sum_e I_e\psi \tag{7.33}$$

Suppose we now perturb the values of elements of N and apply Tellegen's theorem to $\hat{N}$ and the perturbed network N:

$$(V_1 + \Delta V_1)\lambda_1 + (V_2 + \Delta V_2)\lambda_2 = \sum_e (V_e + \Delta V_e)\lambda_e \tag{7.34}$$

and

$$(I_1 + \Delta I_1)\psi_1 + (I_2 + \Delta I_2)\psi_2 = \sum_e (I_e + \Delta I_e)\psi_e \tag{7.35}$$

where ΔV and ΔI represent the changes in the voltage and current which results as a consequence of the perturbation of the element values in N. Subtracting Equation 7.32 from Equation 7.34 and subtracting Equation 7.33 from Equation 7.35,

$$\Delta V_1\lambda_1 + \Delta V_2\lambda_2 = \sum_e \Delta V_e\lambda_e \tag{7.36}$$

and

$$\Delta I_1\psi_1 + \Delta I_2\psi_2 = \sum_{\mathrm{e}} \Delta I_{\mathrm{e}}\psi_{\mathrm{e}} \tag{7.37}$$

Subtracting Equation 7.37 from Equation 7.36 yields

$$(\Delta V_1\lambda_1 - \Delta I_1\psi_1) + (\Delta V_2\lambda_2 - \Delta I_2\psi_2) = \sum_{\mathrm{e}} (\Delta V_{\mathrm{e}}\lambda_{\mathrm{e}} - \Delta I_{\mathrm{e}}\psi_{\mathrm{e}}) \tag{7.38}$$

We wish to define the corresponding element of $\hat{N}$ for every element in N so that each term in the summation on the right-hand side of Equation 7.38 reduces to a function of the voltage and current variables and the change in value of the corresponding network element. We illustrate this for resistance elements. Consider a resistance element R in N. For this element, we have

$$V_{\mathrm{R}} = RI_{\mathrm{R}} \tag{7.39}$$

Suppose we change R to ΔR, then

$$(V_{\mathrm{R}} + \Delta V_{\mathrm{R}}) = (R + \Delta R)(I_{\mathrm{R}} + \Delta I_{\mathrm{R}}) \tag{7.40}$$

Neglecting second-order terms, Equation 7.40 simplifies to

$$V_{\mathrm{R}} + \Delta V_{\mathrm{R}} = RI_{\mathrm{R}} + R\Delta I_{\mathrm{R}} + I_{\mathrm{R}}\Delta R \tag{7.41}$$

Subtracting Equation 7.39 from Equation 7.41,

$$\Delta V_{\mathrm{R}} = R\Delta I_{\mathrm{R}} + I_{\mathrm{R}}\Delta R \tag{7.42}$$

Now using Equation 7.42 the terms in Equation 7.38 corresponding to the resistance elements of N can be written as

$$\sum_{\mathrm{R}} [R\lambda_{\mathrm{R}} - \psi_{\mathrm{R}}]\Delta I_{\mathrm{R}} + I_{\mathrm{R}}\lambda_{\mathrm{R}}\Delta R \tag{7.43}$$

If we now choose

$$\psi_{\mathrm{R}} = R\lambda_{\mathrm{R}} \tag{7.44}$$

then Equation 7.43 reduces to

$$\sum_{R} I_{\mathrm{R}}\lambda_{\mathrm{R}}\Delta R \tag{7.45}$$

which involves only the network variables in N (before perturbation) and $\hat{N}$ and the changes in resistance values. Equation 7.44 is the relation for a resistance. Therefore, the element in $\hat{N}$ corresponding to a resistance element of value R in N is also a resistance of value R.

Proceeding in a similar manner we can determine the element of $\hat{N}$ corresponding to other types of network elements (inductance, capacitance, controlled sources, etc.) The network $\hat{N}$ so obtained is called the adjoint of N. A table defining adjoint elements corresponding to different types of network elements may be found in Ref. [1].

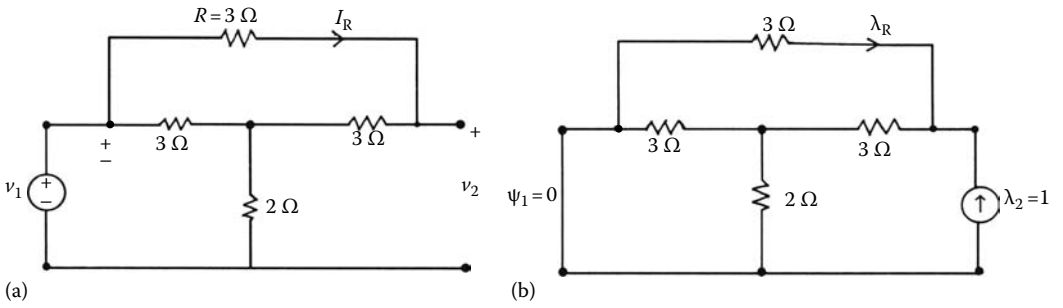


FIGURE 7.16 (a) A 2-port network N; (b) adjoint network $\hat{N}$.

We now illustrate the application of the adjoint network in the computation of the sensitivity of a network function. Note that the sensitivity of a network function F with respect to a parameter x is a measure of the effect on F of an incremental change in x. Computing this sensitivity essentially involves determining $\partial F/\partial x$.

For the sake of simplicity, consider the resistance network shown in Figure 7.16a. Let us assume that resistance R is perturbed from its nominal value of 3 Ω. Assume that no changes occur in the values of the other resistance elements. We wish to compute $\partial F/\partial R$ where F is the open-circuit voltage ratio, that is,

$$F = \left.\frac{V_2}{V_1}\right|_{I_{2=0}}$$

In other words, to compute F, we connect a voltage source of value $V_1 = 1$ across port 1 of N and open-circuit port 2 of N (so that $I_2 = 0$). So, $\Delta V_1 = 0$ and $\Delta I_2 = 0$ and Equation 7.38 reduces to

$$-\Delta I_1 \psi_1 + \Delta V_2 \lambda_2 = I_R \lambda_R \Delta R \tag{7.46}$$

Now we need to determine ΔV_2 as a function of ΔR. This could be achieved if we set $\psi_1 = 0$ and $\lambda_2 = 1$ for the adjoint network $\hat{N}$. Connect a current source of value $\lambda_2 = 1$ across port 2 and short circuit port 1 of $\hat{N}$. The resulting adjoint network is shown in Figure 7.16b. With port variables of $\hat{N}$ defined as above, Equation 7.46 reduces to

$$\Delta V_2 = I_R \lambda_R \Delta_R$$

Thus,

$$\partial F/\partial R = \partial V_2/\partial R = I_R \lambda_R$$

where I_R and λ_R are the currents in the networks N and $\hat{N}$ shown in Figure 7.16. Thus, in general, computing the sensitivity of a network function essentially reduces to the analysis of N and $\hat{N}$ under appropriate excitations at their ports. Note that we do not need to express the network function explicitly in terms of network elements, nor do we need to calculate partial derivatives.

For the example under consideration, we calculate $I_R = 1/12$ A and $\lambda_R = -7/12$ A with the result that $\partial F/\partial R = -7/144$. A further discussion of the adjoint network and related results may be found in Section 7.3.

7.9 Arc Coloring Theorem and the No-Gain Property

We now derive a profound result in graph theory, the arc coloring theorem for directed graphs, and discuss its application in establishing the no-gain property of resistance networks. In the special case of undirected graphs the arc coloring theorem reduces to the "painting" theorem. Both of these theorems [10] are based on the notion of painting a graph.

Given an undirected graph with edge set E, a painting of the graph is a partitioning of E into three subsets, R, G, and B, such that $|G| = 1$. We may consider the edges in the set R as being "painted red," edge in G as being "painted green," and the edges in B as being "painted blue."

THEOREM 7.23

For any painting of a graph, there exists a circuit C consisting of the green edge and no blue edges, or a cutset C^* consisting of the green edge and no red edges.

Proof: Consider a painting of the edge set E of a graph G. Assuming that there does not exist a required circuit, we shall establish the existence of a required cutset.

Let $E' = R \cup G$ and T' denote a spanning forest of the subgraph induced by E' containing the green edge. (Note that the subgraph induced by E' may not be connected). Then, construct a spanning tree T of G such that $T' \subseteq T$.

Now consider any red edge y that is not in T', and hence not in T. Because the fundamental circuit of y with respect to T is the same as the fundamental circuit of y with respect to T', this circuit consists of no blue edges. Furthermore, this circuit will not contain the green edge, for otherwise a circuit consisting of the green edge and no blue edges would exist contrary to our assumption. Thus, the fundamental circuit of a red edge with respect to T does not contain the green edge. Then, it follows from Theorem 7.8 that the fundamental cutset of the green edge with respect to T contains no red edges. Thus, this cutset satisfies the requirements of the theorem.

A painting of a directed graph with edge set E is a partitioning of E into three sets R, G, and B, and the distinguishing of one element of the set G. Again, we may regard the edges of the graph as being colored red, green, or blue with exactly one edge of G being colored dark green. Note that the dark green edge is also to be treated as a green edge.

Next, we state and prove Minty's arc coloring theorem.

THEOREM 7.24

For any painting of a directed graph exactly one of the following is true.

1. *A circuit exists containing the dark green edge, but no blue edges, in which all the green edges are similarly oriented.*
2. *A cutset exists containing the dark green edge, but no red edges, in which all the green edges are similarly oriented.*

Proof: Proof is by induction on the number of green edges. If only one green edge exists, then the result will follow from Theorem 7.23. Assume then that the result is true when the number of green edges is

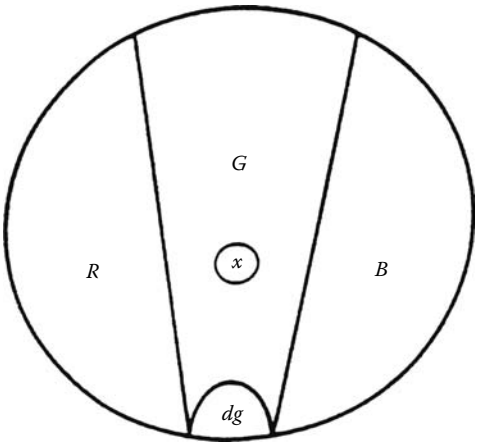


FIGURE 7.17 Painting of a directed graph.

$m \geq 1$. Consider a painting in which $m+1$ edges are colored green. Pick a green edge x other than the dark green edge (Figure 7.17). Color the edge x red. In the resulting painting we find m green edges. If a cutset of type 2 is now found, then the theorem is proved. On the other hand if we color the edge x blue and in the resulting painting a circuit of type 1 exists, then the theorem is proved.

Suppose neither occurs. Then, using the induction hypothesis we have following:

1. Cutset of type 2 exists when x is colored blue.
2. Circuit of type 1 exists when x is colored red.

Now let the corresponding rows of the circuit and cutset matrices be

	dg	R		B	G	x
Cutset	$+1$	$00\cdots0$	0	$1-1\cdots01$	$111\cdots0$	?
Circuit	$+1$	$-11\cdots0-1$	0	$0\cdots00$	$011\cdots0$	?

Here, we have assumed, without loss of generality, that $+1$ appears in the dark green position of both rows.

By the orthogonality relation (Theorem 7.12) the inner product of these two row vectors is zero. No contribution is made to this inner product from the red edges or from the blue edges. The contribution from the green edges is a non-negative integer p. The dark green edge contributes 1 and the edge x contributes an unknown integer q which is 0, 1, or -1. Thus, we have $1+p+q=0$. This equation is satisfied only for $p=0$ and $q=-1$. Therefore, in one of the rows, the question mark is $+1$ and in the other it is -1. The row in which the question mark is 1 corresponds to the required circuit or cutset. Thus, either statement 1 or 2 of the theorem occurs. Both cannot occur simultaneously because the inner product of the corresponding circuit and cutset vectors will then be nonzero.

THEOREM 7.25

Each edge of a directed graph belongs to either a directed circuit or to a directed cutset, but no edge belongs to both. (Note: A cutset is a directed cutset if all its edges are similarly oriented.)

Proof: Proof will follow if we apply the arc coloring theorem to a painting in which all the edges are colored green and the given edge is colored dark green.

We next present an application of the arc coloring theorem in the study of electrical networks. We prove what is known as the no-gain property of resistance networks. Our proof is the result of the work of Wolaver [11] and is purely graph theoretic in nature.

THEOREM 7.26

In a network of sources and (linear/nonlinear) positive resistances, the magnitude of the current through any resistance with nonzero voltage is not greater than the sum of the magnitudes of the currents through the sources.

Proof: Let us eliminate all the elements with zero voltage by considering them to be short-circuits and then assign element reference directions so that all element voltages are positive.

Consider a resistance with nonzero voltage. Thus, no directed circuit can contain this resistance, for if such a directed circuit were present, the sum of all the voltages in the circuit would be nonzero, contrary to Kirchhoff's voltage law. It then follows from Theorem 7.25 that a directed cutset contains the resistance under consideration.

Pick a directed cutset that contains the considered resistance. Let the current through this resistance be i_0. Let R be the set of all other resistances in this cutset and let S be the set of all sources. Then, applying Kirchhoff's current law to the cutset, we obtain

$$i_0 + \sum_{k \in R} i_k + \sum_{s \in S} \pm i_s = 0 \tag{7.47}$$

Because all the resistances and voltages are positive, every resistance current is positive. Therefore, we can write the above equation as

$$|i_0| + \sum_{k \in R} |i_k| + \sum_{s \in S} \pm i_s = 0 \tag{7.48}$$

and so

$$|i_0| \leq \sum_{s \in S} \mp i_s \leq \sum_{s \in S} |i_s| \tag{7.49}$$

Thus follows the theorem.

The following result is the dual of the above theorem. Proof of this theorem follows in an exactly dual manner, if we replace current with voltage, voltage with current, and circuit with cutset in the proof of the above theorem.

THEOREM 7.27

In a network of sources and (linear/nonlinear) positive resistances, the magnitude of the voltage across any resistance is not greater than the sum of the voltages across all the sources.

Chua and Green [12] used the arc coloring theorem to establish several properties of nonlinear networks and nonlinear multiport resistive networks.

References

1. M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms,* New York: Wiley-Interscience, 1981.
2. K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms,* New York: Wiley-Interscience, 1992.
3. F. E. Hohn, *Elementary Matrix Algebra,* New York: Macmillan, 1958.
4. G. Kirchhoff, Uber die Auflosung der Gleichungen auf welche mon bei der untersuchung der linearen Verteilung galvanischer strome gefuhrt wind, *Annalen Physik und Chemie*, 72, 497–508, 1847.
5. W. T. Tutte, The dissection of equilateral triangles into equilateral triangles, *Proceedings of the Cambridge Philosophical Society*, 44, 203–217, 1948.
6. G. Kishi and Y. Kajitani, Maximally distant trees and principal partition of a linear graph, *IEEE Transactions on Circuit Theory*, CT-15, 247–276, 1968.
7. T. Ohtsuki, Y. Ishizaki, and H. Watanabe, Topological degrees of freedom and mixed analysis of electrical networks, *IEEE Transactions on Circuit Theory*, CT-17, 491–499, 1970.
8. B. D. H. Tellegen, A general network theorem with applications, *Philips Research Reports*, 7, 259–269, 1952.
9. S. W. Director and R. A. Rohrer, Automated network design—The frequency domain case, *IEEE Transactions on Circuit Theory*, CT-16, 330–337, 1969.
10. G. J. Minty, On the axiomatic foundations of the theories of directed linear graphs, electrical networks and network programming, *Journal of Mathematics and Mechanics*, 15, 485–520, 1966.
11. D. H. Wolaver, Proof in graph of the ''no-gain'' property of resistor networks, *IEEE Transactions on Circuit Theory*, CT-17, 436–437, 1970.
12. L. O. Chua and D. N. Green, Graph-theoretic properties of dynamic nonlinear networks, *IEEE Transactions on Circuit Theory*, CAS-23, 292–312, 1976.

8

Signal Flow Graphs

8.1 Introduction **8**-1
8.2 Adjacency Matrix of a Directed Graph **8**-1
8.3 Coates' Gain Formula **8**-4
8.4 Mason's Gain Formula **8**-8
References **8**-11

Krishnaiyan Thulasiraman
University of Oklahoma

8.1 Introduction

Signal flow graph theory is concerned with the development of a graph theoretic approach to solving a system of linear algebraic equations. Two closely related methods proposed by Coates [1] and Mason [2,3] have appeared in the literature and have served as elegant aids in gaining insight into the structure and nature of solutions of systems of equations. In this chapter, we develop these two methods. Our development follows these methods closely [4].

An extensive discussion of signal flow theory may be found in Ref. [5]. Applications of signal flow theory in the analysis and synthesis electrical networks may be found in Sections 8.3 and 8.4. Coates' and Mason's methods may be viewed as generalizations of a basic theorem in graph theory due to Harary [6], which provides a formula for finding the determinant of the adjacency matrix of a directed graph. Thus, our discussion begins with the development of this theorem. For graph theoretic terminology, the reader may refer to Chapter 7.

8.2 Adjacency Matrix of a Directed Graph

Consider a directed graph $G = (V, E)$ with no parallel edges. Let $V = \{v_1, \ldots, v_n\}$. The adjacency matrix $M = [m_{ij}]$ of G is an $n \times n$ matrix defined as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

The graph in Figure 8.1 has the following adjacency matrix:

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

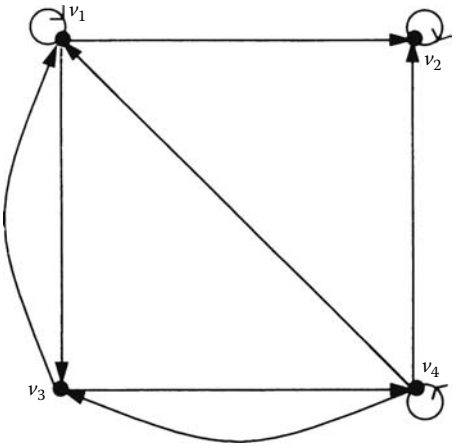


FIGURE 8.1 Graph *G*.

In the following, we shall develop a topological formula for det *M*. Toward this end, we introduce some basic terminology. A 1-factor of a directed graph *G* is a spanning subgraph of *G* in which the in-degree and the out-degree of every vertex are both equal to 1. It is easy to see that a 1-factor is a collection of vertex-disjoint directed circuits. Because a self-loop at a vertex contributes 1 to the in-degree and 1 to the out-degree of the vertex, a 1-factor may have some self-loops. As an example, the three 1-factors of the graph of Figure 8.1 are illustrated in Figure 8.2.

A permutation $(j_1, j_2, \ldots, j_n)$ of integers $1, 2, \ldots, n$ is even (odd) if an even (odd) number of interchanges are required to rearrange it as $(1, 2, \ldots, n)$. The notation

$$\begin{pmatrix} 1, 2, \ldots, n \\ j_1, j_2, \ldots, j_n \end{pmatrix}$$

is also used to represent the permutation $(j_1, j_2, \ldots, j_n)$. As an example, the permutation (4, 3, 1, 2) is odd because it can be rearranged as (1, 2, 3, 4) using the following sequence of interchanges:

1. Interchange 2 and 4
2. Interchange 1 and 2
3. Interchange 2 and 3

For a permutation $(j) = (j_1, j_2, \ldots, j_n)$, $\varepsilon_{j_1, j2, \ldots, jn}$, is defined as equal to 1, if (j) is an even permutation; otherwise, $\varepsilon_{j_1, j2, \ldots, jn}$, is equal to -1.

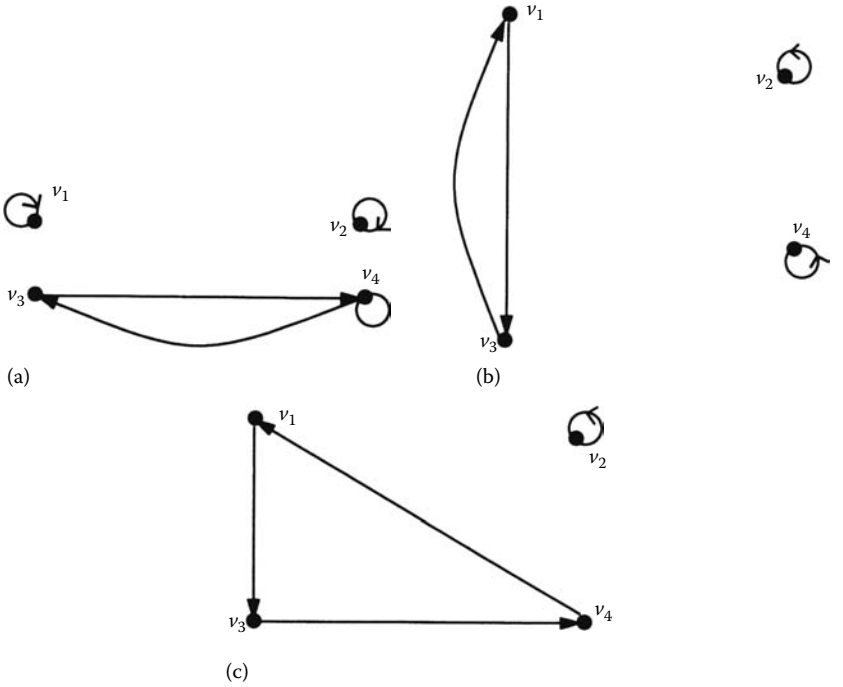


FIGURE 8.2 Three 1-factors of the graph of Figure 8.1.

Given an $n \times n$ square matrix $X = [x_{ij}]$, we note that det X is given by

$$\det X = \sum_{(j)} \varkappa_{j_1, j_2, j_3, \ldots, j_n} \, x_{1j_1}, x_{2j_2}, \ldots, x_{n_{jn}}$$

where the summation $\Sigma_{(j)}$ is over all permutations of 1, 2, …, n [7].

The following theorem is due to Harary [6].

THEOREM 8.1

Let H_i, $i = 1, 2, \ldots, p$ be the 1-factors of an n-vertex directed graph G. Let L_i denote the number of directed circuits in H_i, and let M denote the adjacency matrix of G. Then,

$$\det M = (-1)^n \sum_{i=1}^{p} (-1)^{L_i}$$

Proof. From the definition of a determinant, we have

$$\det M = \sum_{(j)} \varepsilon_{j_1, j_2, \ldots, j_n} m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n} \tag{8.1}$$

Proof will follow if we establish the following:

1. Each nonzero term $m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n}$ corresponds to a 1-factor of G, and conversely, each 1-factor of G corresponds to a nonzero term $m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n}$.
2. $\varepsilon_{j1,\, j2, \ldots,\, jn} = (-1)^{n+L}$ if the 1-factor corresponding to a nonzero $m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n}$ has L directed circuits.

A nonzero term $m_{1j_1} \cdot m_{1j_2} \cdots m_{nj_n}$ corresponds to the set of edges (v_1, v_{j1}), $(v_2, v_{j2}), \ldots, (v_n, v_{jn})$. Each vertex appears exactly twice in this set, once as an initial vertex and once as a terminal vertex of a pair of edges. Therefore, in the subgraph induced by these edges, for each vertex its in-degree and its out-degree are both equal to 1, and this subgraph is a 1-factor of G. In other words, each nonzero term in the sum in Equation 8.1 corresponds to a 1-factor of G. The fact that each 1-factor of G corresponds to a nonzero term $m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n}$ is obvious.

Regarding $\varepsilon_{j_1,\, j2, \ldots,\, jn}$, consider a directed circuit C in the 1-factor corresponding to $m_{1j_1} \cdot m_{2j_2} \cdots m_{nj_n}$. Without loss of generality, assume that C consists of the w edges

$$(v_1, v_2), (v_2, v_3), \ldots, (v_w, v_1)$$

It is easy to see that the corresponding permutation (2, 3, …, w, 1) can be rearranged as (1, 2, …, w) using $w - 1$ interchanges. If the 1-factor has L directed circuits with lengths $w_1, \ldots, w_L$, the permutation $(j_1, \ldots, j_n)$ can be rearranged as (1, 2, …, n) using

$$(w_1 - 1) + (w_2 - 1) + \cdots + (w_L - 1) = n - L$$

interchanges. So,

$$\varepsilon_{j_1, j_2, j_n} = (-1)^{n+L}$$

As an example, for the 1-factors (Figure 8.2) of the graph of Figure 8.1, the corresponding L_i are $L_1 = 3$, $L_2 = 3$, and $L_3 = 2$. So, the determinant of the adjacency matrix of the graph of Figure 8.1 is

$$(-1)^4\left[(-1)^3 + (-1)^3 + (-1)^2\right] = -1$$

Consider next a weighted directed graph G in which each edge (v_i, v_j) as associated with a weight w_{ij}. Then we may define the adjacency matrix $M = [m_{ij}]$ of G as follows:

$$m_{ij} = \begin{cases} w_{ij} & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Given a subgraph H of G, let us define weight $w(H)$ of H as the product of the weights of all edges in H. If H has no edges, then we define $w(H) = 1$. The following result is an easy generalization of Theorem 8.1.

THEOREM 8.2

The determinant of the adjacency matrix of an n-vertex directed graph G is given by

$$\det M = (-1)^n \sum_H (-1)^{L_H} w(H),$$

where H is a 1-factor, w(H) is the weight of H, and L_H is the number of directed circuits in H.

8.3 Coates' Gain Formula

Consider a linear system described by the equation

$$AX = Bx_{n+1} \tag{8.2}$$

where

A is a nonsingular $n \times n$ matrix
X is a column vector of unknown variables $x_1, x_2, \ldots, x_n$
B is a column vector of elements $b_1, b_2, \ldots, b_n$
x_{n+1} is the input variable

It is well known that

$$\frac{x_k}{x_{n+1}} = \frac{\sum_{i=1}^{n} b_i \Delta_{ik}}{\det A} \tag{8.3}$$

where Δ_{ik} is the (i, k) cofactor of A.

To develop Coates' topological formulas for the numerator and the denominator of Equation 8.3, let us first augment the matrix A by adding $-B$ to the right of A and adding a row of zeroes at the bottom of the resulting matrix. Let this matrix be denoted by A'. The Coates flow graph* $G_c(A')$, or simply the Coates graph, associated with matrix A' is a weighted directed graph whose adjacency matrix is the transpose of the matrix A'. Thus, $G_c(A')$ has $n+1$ vertices $x_1, x_2, \ldots, x_{n+1}$, and if $a_{ji} \ldots \neq 0$, $G_c(A')$ has an edge directed from x_i to x_j with weight a_{ji}. Clearly, the Coates graph $G_c(A)$ associated with matrix A can be obtained from $G_c(A')$ by removing the vertex x_{n+1}.

* In network and systems theory literature, the Coates graph is referred to as a flow graph.

As an example, for the following system of equations

$$\begin{bmatrix} 3 & -2 & 1 \\ -1 & 2 & 0 \\ 3 & -2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -2 \end{bmatrix} x_4 \tag{8.4}$$

the matrix A' is

$$A' = \begin{bmatrix} 3 & -2 & 1 & -3 \\ -1 & 2 & 0 & -1 \\ 3 & -2 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The Coates' graphs $G_c(A')$ and $G_c(A)$ are depicted in Figure 8.3.

Because a matrix and its transpose have the same determinant value, and because A is the transpose of the adjacency matrix of $G_c(A)$, we obtain the following result from Theorem 8.2.

THEOREM 8.3

If a matrix A is nonsingular, then

$$\det A = (-1)^n \sum_H (-1)^{L_H} w(H) \tag{8.5}$$

where H is a 1-factor of $G_c(A)$, w(H) is the weight of H, and L_H is the number of directed circuits in H.

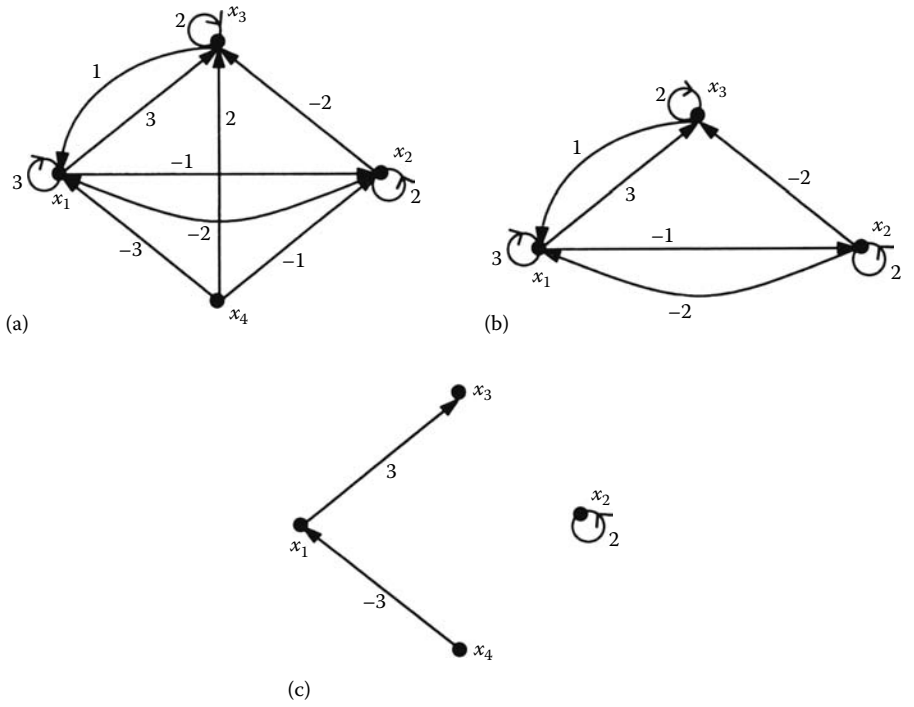


FIGURE 8.3 (a) The Coates graph $G_c(A')$; (b) the graph $G_c(A)$; (c) a 1-factorial connection $H_{4,3}$ of the graph $G_c(A')$.

To derive a similar expression for the sum in the numerator of Equation 8.3, we first define the concept of a 1-factorial connection. A 1-factorial connection H_{ij} from x_i to x_j in $G_c(A)$ is a spanning subgraph of G that contains a directed path P from x_i to x_j and a set of vertex-disjoint directed circuits, which include all the vertices of $G_c(A)$ other than those that lie on P. Similarly, a 1-factorial connection of $G_c(A')$ can be defined. As an example, a 1-factorial connection from x_4 to x_3 of the graph $G_c(A')$ of Figure 8.3a is also in Figure 8.3c.

THEOREM 8.4

Let $G_c(A')$ be the Coates graph associated with an $n \times n$ matrix A. Then,

1. $\Delta_{ii} = (-1)^{n-1} \sum_{H} (-1)^{L_H} w(H)$
2. $\Delta_{ij} = (-1)^{n-1} \sum_{H_{ij}} (-1)^{L'_H} w(H_{ij}) \quad i \neq j$

where H is a 1-factor in the graph obtained by removing vertex x_i from $G_c(A)$, H_{ij} is a 1-factorial connection in $G_c(A)$ from vertex x_i to vertex x_j, and L_H and L'_H are the numbers of directed circuits in H and H_{ij}, respectively.

Proof

1. Note that Δ_{ii} is the determinant of the matrix obtained from A by removing its row i and column i. Also, the Coates graph of the resulting matrix can be obtained from $G_c(A)$ by removing vertex x_i. Proof follows from these observations and Theorem 8.3.
2. Let A_α denote the matrix obtained from A by replacing its jth column by a column of zeroes, except for the element in row i, which is 1. Then it is easy to see that

$$\Delta_{ij} = \det A_\alpha$$

Now, the Coates graph $G_c(A_\alpha)$ can be obtained from $G_c(A)$ by removing all edges incident out of vertex x_j and adding an edge directed from x_j to x_i with weight 1. Then from Theorem 8.3, we get

$$\begin{aligned} \Delta_{ij} &= \det A_\alpha \\ &= (-1)^n \sum_{H_\alpha} (-1)^{L_\alpha} w(H_\alpha) \end{aligned} \tag{8.6}$$

where
H_α is a 1-factor of $G_c(A_\alpha)$
L_α is the number of directed circuits in H_α

Consider now a 1-factor H_α in $G_c(A_\alpha)$. Let C be the directed circuit of H_α containing x_i. Because in $G_c(A_\alpha)$, (x_j, x_i) is the only edge incident out of x_j, it follows that x_j also lies in C. If we remove the edge (x_j, x_i) from H_α we get a 1-factorial connection, H_{ij}. Furthermore, $L'_H = L_\alpha - 1$ and $w(H_{ij}) = w(H_\alpha)$ because (x_j, x_i) has weight equal to 1. Thus, each H_α corresponds to a 1-factorial connection H_{ij} of $G_c(A_\alpha)$ with $w(H_\alpha) = w(H_{ij})$ and $L'_H = L_\alpha - 1$. The converse of this is also easy to see. Thus, in Equation 8.6 we can replace H_α by H_{ij} and L_α by $(L'_H + 1)$. Then we obtain

$$\Delta_{ij} = (-1)^{n-1} \sum_{H_{ij}} (-1)^{L'_H} w(H_{ij})$$

Having shown that each Δ_{ij} can be expressed in terms of the weights of the 1-factorial connections H_{ij} in $G_c(A)$, we now show that $\Sigma b_i\Delta_{ik}$ can be expressed in terms of the weights of the 1-factorial connections $H_{n+1,k}$ in $G_c(A')$.

First, note that adding the edge (x_{n+1}, x_i) to H_{ik} results in a 1-factorial connection $H_{n+1,k}$, with $w(H_{n+1,k}) = -b_i\, w(H_{ik})$. Also, $H_{n+1,k}$ has the same number of directed circuits as H_{ik}. Conversely, from each $H_{n+1,k}$ that contains the edge (x_{n+1}, x_i) we can construct a 1-factorial connection H_{ik} satisfying $w(H_{n+1,k}) = -b_i\, w(H_{ik})$. Also, $H_{n+1,k}$ and the corresponding H_{ik} will have the same number of directed circuits. Thus, a one-to-one correspondence exists between the set of all 1-factorial connections $H_{n+1,k}$ in $G_c(A')$ and the set of all 1-factorial connections in $G_c(A)$ of the form H_{ik} such that each $H_{n+1,k}$ and the corresponding H_{ik} have the same number of directed circuits and satisfy the relation $w(H_{n+1,k}) = -b_i\, w(H_{ik})$. Combining this result with Theorem 8.4, we get

$$\sum_{i=1}^{n} b_i\Delta_{ik} = (-1)^n \sum_{H_{n+1,k}} (-1)^{L'_H}\, w(H_{n+1,k}) \tag{8.7}$$

where the summation is over all 1-factorial connections, $H_{n+1,k}$ in $G_c(A')$, and L'_H is the number of directed circuits in $H_{n+1,k}$. From Equations 8.5 and 8.7 we get the following theorem.

THEOREM 8.5

If the coefficient matrix A is nonsingular, then the solution of Equation 8.2 is given by

$$\frac{x_k}{x_{n+1}} = \frac{\sum_{H_{n+l,k}} (-1)^{L'_H}\, w(H_{n+1,k})}{\sum_H (-1)^{L_H}\, w(H)} \tag{8.8}$$

for $k = 1, 2, \ldots, n$, where $H_{n+1,k}$ is a 1-factorial connection of $G_c(A')$ from vertex x_{n+1} to vertex x_k, H is a 1-factor of $G_c(A)$, and L'_H and L_H are the numbers of directed circuits in $H_{n+1,k}$ and H, respectively.

Equation 8.8 is the called Coates' gain formula. We now illustrate Coates' method by solving the system, Equation 8.4, for x_2/x_4. First, we determine the 1-factors of the Coates' graph $G_c(A)$ shown in Figure 8.3b. These 1-factors, along with their weights, are listed below. The vertices enclosed within parentheses represent a directed circuit.

1-Factor, H	Weight, $w(H)$	L_H
$(x_1)\ (x_2)\ (x_3)$	12	3
$(x_2)\ (x_1, x_3)$	6	2
$(x_3)\ (x_1, x_2)$	4	2
(x_1, x_2, x_3)	2	1

From the above we get the denominator in Equation 8.8 as

$$\sum_H (-1)^{L_H}\, w(H) = (-1)^3 \cdot 12 + (-1)^2 \cdot 6 + (-1)^2 \cdot 4 + (-1)^1 \cdot 2 = -4$$

To compute the numerator in Equation 8.8 we need to determine the 1-factorial connections $H_{4,2}$ in the Coates graph $G_c(A')$, shown in Figure 8.3a. They are listed below along with their weights. The vertices in a directed path from x_4 to x_2 are given within parentheses.

1-Factorial Connection, $H_{4,2}$	$w(H_{4,2})$	L'_H
$(x_4, x_1, x_2)\ (x_3)$	6	1
$(x_4, x_2)\ (x_1)\ (x_3)$	−6	2
$(x_4, x_2)\ (x_1, x_3)$	−3	1
(x_4, x_3, x_1, x_2)	−2	0

From the above we get the numerator in Equation 8.8 as

$$\sum_{H_{4,2}} (-1)^{L'_H} w(H_{4,2}) = (-1)^1 \cdot 6 + (-1)^2(-6) + (-1)^1(-3) + (-1)^0(-2) = -11$$

Thus, we get

$$\frac{x_2}{x_4} = \frac{11}{4}$$

8.4 Mason's Gain Formula

Consider again the system of equations

$$AX = Bx_{n+1}$$

We can rewrite the above as

$$x_j = (a_{jj} + 1)x_j + \sum_{\substack{k=1 \\ k \neq j}}^{n} a_{jk}x_k - b_j x_{n+1}, \quad j = 1, 2, \ldots, n, \quad x_{n+1} = x_{n+1} \tag{8.9}$$

Letting X' denote the column vector of the variables $x_1, x_2, \ldots, x_{n+1}$, and U_{n+1} denote the unit matrix of order n, we can write Equation 8.9 in matrix form as follows:

$$(A' + U_{n+1})X' = X' \tag{8.10}$$

where A' is the matrix defined earlier in Section 8.3.

The Coates graph $G_c(A' + U_{n+1})$ is called the Mason's signal flow graph or simply the Mason graph* associated with A' and it is denoted by $G_m(A')$. The Mason graph $G_m(A)$ is defined in a similar manner. The Mason graphs $G_m(A')$ and $G_m(A)$ associated with the system (Equation 8.4) are illustrated in Figure 8.4. Mason's graph elegantly represents the flow of variables in a system. If we associate each vertex with a variable and if an edge is directed from x_i to x_j, then we may consider the variable x_i as contributing $(a_{ji}x_i)$ to the variable x_j. Thus, x_j is equal to the sum of the products of the weights of the edges incident into vertex x_j and the variables corresponding to the vertices from which these edges emanate.

Note that, to obtain the Coates graph $G_c(A)$ from the Mason graph $G_m(A)$, we simply subtract one from the weight of each self-loop. Equivalently, we may add at each vertex of the Mason graph a self-loop of weight –1. Let S denote the set of all such loops of weight –1 added to construct the Coates graph G_c from the Mason graph $G_m(A)$.

* In network and systems theory literature, Mason graphs are usually referred to as signal flow graphs.

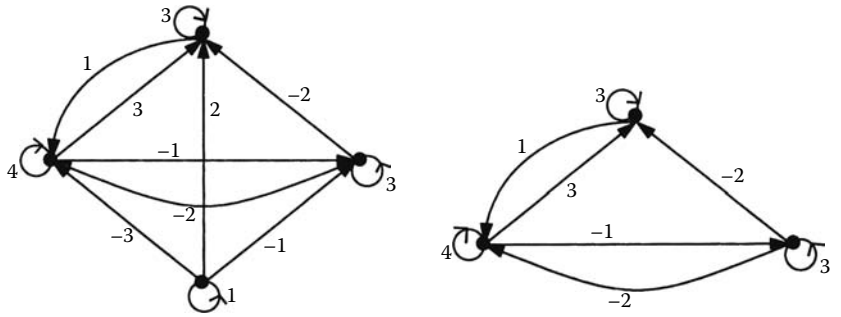


FIGURE 8.4 (a) The Mason graph $G_m(A')$; (b) the Mason graph $G_m(A)$.

Consider now the Coates graph G_c constructed as above and a 1-factor H in G_c having j self-loops from the set S. If H has a total of $L_Q + j$ directed circuits, then removing the j self-loops from H will result in a subgraph Q of $G_m(A)$, which is a collection of L_Q vertex disjoint directed circuits. Also,

$$w(H) = (-1)^j w(Q)$$

Then, from Theorem 8.3 we get

$$\begin{aligned} \det A &= (-1)^n \sum_H (-1)^{L_Q + j} w(H) \\ &= (-1)^n \sum_Q (-1)^{L_Q} w(Q) \\ &= (-1)^n \left[1 + \sum_Q (-1)^{L_Q} w(Q) \right] \end{aligned} \tag{8.11}$$

We can rewrite the above as

$$\det A = (-1)^n \left[1 - \sum_j Q_{j1} + \sum_j Q_{j2} - \sum_j Q_{j3} \cdots \right] \tag{8.12}$$

where each term in $\Sigma_j Q_{ji}$ is the weight of a collection of i vertex-disjoint directed circuits in $G_m(A)$.

Suppose we refer to $(-1)^n \det A$ as the determinant of the graph $G_m(A)$. Then, starting from $H_{n+1,k}$ and reasoning exactly as above we can express the numerator of Equation 8.3 as

$$\sum_{i=1}^{n} b_i \Delta_{ik} = (-1)^n \sum_j w\left(P^j_{n+1,k}\right) \Delta_j \tag{8.13}$$

where

$P^j_{n+1,k}$ is a directed path from x_{n+1} to x_k of $G_m(A')$

Δ_j is the determinant of the subgraph of $G_m(A')$, which is vertex disjoint from the path $P^j_{n+1,k}$

From Equations 8.12 and 8.13, we get the following theorem.

THEOREM 8.6

If the coefficient matrix A in Equation 8.2 *is nonsingular, then*

$$\frac{x_k}{x_{n+1}} = \frac{\sum_j w\left(P^j_{n+l,k}\right)\Delta_j}{\Delta}, \quad k = 1, 2, \ldots, n \tag{8.14}$$

where $P^j_{n+1,k}$ *is the jth directed path from* x_{n+1} to x_k *of* $G_m(A')$, Δ_j *is the determinant of the subgraph of* $G_m(A')$, *which is vertex disjoint from the jth directed path* $P^j_{n+1,k}$, *and* Δ *is the determinant of the graph of* $G_m(A)$.

Equation 8.14 is known as Mason's gain formula. In network and systems theory, $P^j_{n+1,k}$ is referred to as a forward path from vertex x_{n+1} to vertex x_k. The directed circuits of $G_m(A')$ are called the feedback loops.

We now illustrate Mason's method by solving the system (Equation 8.4) for x_2/x_4. To compute the denominator in Equation 8.14, we determine the different collections of vertex-disjoint directed circuits of the Mason graph $G_m(A)$ shown in Figure 8.4b. They are listed below along with their weights.

Collection of Vertex-Disjoint Directed Circuits of $G_m(A)$	Weight	No. of Directed Circuits
(x_1)	4	1
(x_2)	3	1
(x_3)	3	1
(x_1, x_2)	2	1
(x_1, x_3)	3	1
(x_1, x_2, x_3)	2	1
$(x_1)\ (x_2)$	12	2
$(x_1)\ (x_3)$	12	2
$(x_2)\ (x_3)$	9	2
$(x_2)\ (x_1, x_3)$	9	2
$(x_3)\ (x_1, x_2)$	6	2
$(x_1)\ (x_2)\ (x_3)$	36	3

From the above, we obtain the denominator in Equation 8.14

$$\begin{aligned}\Delta = 1 &+ (-1)^1[4 + 3 + 3 + 2 + 3 + 2] \\ &+ (-1)^2[12 + 12 + 9 + 9 + 6] + (-1)^3\ 36 = -4\end{aligned}$$

To compute the numerator in Equation 8.14 we need the forward paths in $G_m(A')$ from x_4 to x_2. They are listed below with their weights.

j	$P^j_{4,2}$	Weight
1	(x_4, x_2)	–1
2	(x_4, x_1, x_2)	3
3	(x_4, x_3, x_1, x_2)	–2

The directed circuits which are vertex disjoint from $P^1_{4,2}$ are (x_1), (x_3), (x_1, x_3). Thus

$$\Delta_1 = 1 - (4 + 3 + 3) + 12 = 1 - 10 + 12 = 3$$

(x_3) is the only directed circuit which is vertex disjoint from $P^2_{4,2}$. So,

$$\Delta_2 = 1 - 3 = -2$$

No directed circuit is vertex disjoint from $P^3_{4,2}$ so $\Delta_3 = 1$. Thus, the numerator in Equation 8.14 is

$$P^1_{4,2}\Delta_1 + P^1_{4,2}\Delta_2 + P^1_{4,3}\Delta_3 = -3 - 6 - 2 = -11$$

and

$$\frac{x_2}{x_4} = \frac{11}{4}$$

References

1. C. L. Coates, Flow graph solutions of linear algebraic equations, *IRE Transactions on Circuit Theory*, CT-6, 170–187, 1959.
2. S. J. Mason, Feedback theory: Some properties of signal flow graphs, *Proceedings of the IRE*, 41, 1144–1156, 1953.
3. S. J. Mason, Feedback theory: Further properties of signal flow graphs, *Proceedings of the IRE*, 44, 920–926, 1956.
4. K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, New York: Wiley Interscience, 1992.
5. W. K. Chen, *Applied Graph Theory*, Amsterdam: North Holland, 1971.
6. F. Harary, The determinant of the adjacency matrix of a graph, *SIAM Review*, 4, 202–210, 1962.
7. F. E. Hohn, *Elementary Matrix Algebra*, New York: Macmillan, 1958.

9

Theory of Two-Dimensional Hurwitz Polynomials

9.1 Introduction **9**-1
9.2 Preliminaries and Notations **9**-2
Infinite Distant Points • Analog Biplane • Isolated and Continuum of Zeroes
9.3 Value of a Two-Variable Polynomial at Infinity **9**-3
Nonessential Singularities of the Second Kind
9.4 Various Analog Hurwitz Polynomials **9**-4
Self-Paraconjugate Polynomial • Broad-Sense Hurwitz Polynomial • Narrow-Sense Hurwitz Polynomial • Scattering Hurwitz Polynomial • Hurwitz Polynomial in the Strict Sense • Very Strict Hurwitz Polynomial • Self-Paraconjugate Hurwitz Polynomial • Reactance Hurwitz Polynomial • Immittance Hurwitz Polynomial • Summary
9.5 Testsets for Analog Hurwitz Polynomials **9**-7
Continuity Property of the Zeroes of 2-V Polynomials
9.6 Two-Variable Very Strict Hurwitz Polynomials **9**-10
9.7 Application of Two-Dimensional Hurwitz Polynomials for Two-Variable Passive Networks and Stability **9**-11
Application to Two-Dimensional Analog System Stability
9.8 Conclusion **9**-13
Acknowledgments **9**-13
References **9**-13

Hari C. Reddy
California State University and National Chiao-Tung University

9.1 Introduction

The advances in two-dimensional (2-D) signal and image processing activities have stimulated active research in 2-D circuits and systems area. Two-variable (2-V) or 2-D Hurwitz polynomial study finds application in areas such as generation and testing of (2-V) reactance functions, bounded/positive real functions, and matrices; testing the stability of 2-D digital filters; and the generation of stable 2-D digital transfer functions. Stability analysis is an important aspect of the design of dynamic systems. This analysis is often carried out by examining for the absence of zeroes of the denominator polynomial of a system transfer function in some specified regions of the complex plane. 1-D systems are studied through the characterization whether or not the denominator polynomial is Hurwitz. By expanding this idea, we can define and study 2-D (also called bivariate, 2-V) Hurwitz polynomials. In view of the diverse needs of

several different applications a number of 2-D Hurwitz polynomials have been defined and their test procedures established. In this chapter, a detailed presentation of various 2-D Hurwitz polynomials and their relationships to one another is given. We also study their relevant applications.

To highlight the relationships among the various Hurwitz polynomials, the definitions of all the Hurwitz polynomials are presented. This is done in terms of the absence of or the nature of their zerosets in specified regions such as the open or closed right half of the (S_1, S_2)-biplane. The goal is to make a tutorial exposition on 2-D Hurwitz polynomials.

Section 9.2 gives some preliminaries and notations. We next present the definitions of a number of (S_1, S_2)-domain Hurwitz polynomials. Based on the continuity property of the zeroes of 2-V polynomials, testsets for the various Hurwitz polynomials are given in Section 9.5. Following that, a 2-D, very strict Hurwitz polynomial is discussed in detail because this is the counterpart of strict Hurwitz in the 1-D case. Some of the applications of the various Hurwitz polynomials are described in Section 9.7.

9.2 Preliminaries and Notations

9.2.1 Infinite Distant Points

The following discussion is crucial to the understanding of certain classes of stable 2-D polynomials. The points at infinite distances in the (S_1, S_2)-biplane play an important role in the definition of certain 2-D Hurwitz polynomials. Some of the confusion that resulted in the application of these Hurwitz polynomials can be attributed to the neglect or omission of these infinite distant points. This chapter considers the extended (S_1, S_2)-biplane, which includes the infinite distance points. For the sake of clarity, we also explicitly indicate whether the infinite distant points are included in or excluded from the regions considered. The behavior of 2-V polynomials at infinite distant points is well described in the literature [2,3]. Seemingly many infinite distant points in the 1-D plane, such as the S_1-plane or the S_2-plane, may be assumed to merge to a single point. Thus, infinity is treated as a single point, and any shift from this infinite distant point, however small, leads to a finite distant point.

9.2.2 Analog Biplane

$\mathrm{Re}(s) =$ Real part of the variable s

For $i = 1, 2$

$S_{i\,+} = \{s_i | \mathrm{Re}(s_i) > 0,\ |s_i| < \infty\}$, open right half of the S_i-plane

$S_{io} = \{s_i | \mathrm{Re}(s_i) = 0,\ |s_i| \leq \infty\}$, imaginary axis of the S_i-plane

$S_{i\oplus} = \{s_i | \mathrm{Re}(s_i) \geq 0,\ |s_i| \leq \infty\}$, closed right half of the S_i-plane

$S^2_{+0} = \{(s_1, s_2) \mid \mathrm{Re}(s_1) > 0,\ \mathrm{Re}(s_2) = 0,\ |s_1| < \infty,\ |s_2| \leq \infty\}$, open right half of the S_1-plane and the imaginary axis of the S_2-plane

$S^2_{0+} = \{(s_1, s_2) \mid \mathrm{Re}(s_1) = 0,\ \mathrm{Re}(s_2) > 0,\ |s_1| \leq \infty,\ |s_2| < \infty\}$, open right half of the S_2-plane and the imaginary axis of the S_1-plane

$S^2_{++} = \{(s_1, s_2) \mid \mathrm{Re}(s_1) > 0, \mathrm{Re}(s_2) > 0,\ |s_1| < \infty,\ |s_2| < \infty\}$, open right half of the (S_1, S_2)-biplane

$S^2_{00} = \{(s_1, s_2) \mid \mathrm{Re}(s_1) = 0,\ \mathrm{Re}(s_2) = 0,\ |s_1| \leq \infty,\ |s_2| \leq \infty\}$, distinguished boundary of the (S_1, S_2)-biplane

$S^2_{\oplus\oplus} = \{(s_1, s_2) \mid \mathrm{Re}(s_1) \geq 0, \mathrm{Re}(s_2) \geq 0,\ |s_1| \leq \infty,\ |s_2| \leq \infty\}$, closed right half of the (S_1, S_2)-biplane

PRF: Positive real function
RF: Reactance function
TPRF: 2-Variable positive real function
TRF: 2-Variable reactance function
TBRF: 2-Variable bounded real function

TLBRF: 2-Variable lossless bounded real function
$B_*(s_1, s_2) =$ Paraconjugate of $B(s_1, s_2) = [B(-s_1{}^*, -s_2{}^*)]^*$, where s^* represents complex conjugate of s
$B_e(s_1, s_2) =$ Para-even part of $B(s_1, s_2) = [B(s_1, s_2) + B_*(s_1, s_2)]/2$
$B_0(s_1, s_2) =$ Para-odd part of $B(s_1, s_2) = [B(s_1, s_2) - B_*(s_1, s_2)]/2$

Definition A: A rational function $F(s)$ with real coefficients such that $\mathrm{Re}[F(s)] > 0$ for $\mathrm{Re}(s) > 0$ is called a positive real function (PRF).

Definition B: A PRF $F(s)$ is said to be a strict PRF if $\mathrm{Re}[F(s)] > 0$ for $\mathrm{Re}(s) \geq 0$.

Definition C: A PRF $F(s)$ is said to be a minimum reactive and susceptive if it has neither poles nor zeroes on the imaginary axis of the S-plane.

Definition D: A PRF $F(s)$ is called a reactance function (RF) if $\mathrm{Re}[F(s)] = 0$ for $\mathrm{Re}(s) = 0$.

Definition E: A 2-V rational function $F(s_1, s_2)$ with real coefficients such that $\mathrm{Re}[F(s_1, s_2)] > 0$ for S^2_{++} is called a TPRF. A TPRF $F(s_1, s_2) = -F(-s_1, -s_2)$ is called a TRF.

Definition F: A 2-V rational function $H(s_1, s_2)$ with real coefficients such that $|H(s_1, s_2)| < 1$ for S^2_{++} is called a TBRF. A TBRF $H(s_1, s_2)$ satisfying the condition $|H(s_1, s_2)| = 1$ or $0/0$ for S^2_{00} is called TLBRF.

9.2.3 Isolated and Continuum of Zeroes

Some types of Hurwitz polynomials are distinguished on the basis of whether they have isolated zeroes or a continuum of zeroes on S^2_{00}. As a point on S^2_{00} is characterized by $s_1 = jw_1$ and $s_2 = jw_2$, where w_1 and w_2 are real quantities, the region S^2_{00} can be graphically represented by the (W_1, W_2)-plane. The isolated zeroes on S^2_{00} are, thus, points on this plane and the continuum of zeroes is represented by continuous curves. Isolated and continuum of zeroes for the 2-D case are illustrated on the (W_1, W_2)-plane in Figure 9.1, where zeroes of some simple polynomials are shown.

9.3 Value of a Two-Variable Polynomial at Infinity

Because later in the chapter it is necessary to determine the value of a 2-V polynomial at infinite distance points, the following explanation is in order. In 1-D complex plane S the infinite distant points can be represented by a single point, and the value of any function at this point is found by applying some transformation which transforms the point at infinity to some finite point s', and the value of the transformed function at s' is determined. Often $s = 1/u$ is the transformation used and infinity is mapped onto the origin. Using this transformation, the value of $B(s)$ at infinity can be defined as $B(\infty) = B_T(0)$, where $B_T(u) = B(1/u)$.

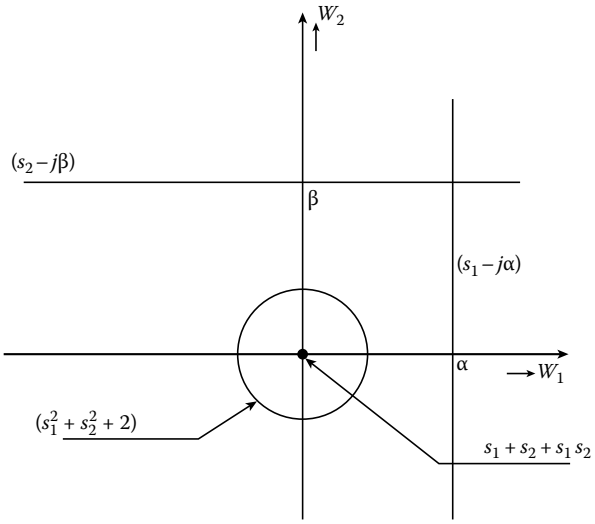


FIGURE 9.1 Zero distribution of some simple polynomials in (W_1, W_2) plane (S_{00}^2). (Polynomial $(s_1 + s_2 + s_1 s_2)$ has an isolated zero, whereas polynomials $(s_1 - j\alpha)$, $(s_2 - j\beta)$, and $(S_1^2 + S_2^2 + 2)$ have a continuum of zeroes.)

In the 2-D biplane (S_1, S_2) consisting of two complex planes, S_1 and S_2, an infinite distant point can have infinite coordinates in either one or both of these planes, and thus an infinite number of infinite distant points exists. They can be classified into three categories [3]:

$$1.\ s_1 = \infty \text{ and } s_2 = \text{finite} \tag{9.1a}$$

$$2.\ s_1 = \text{finite and } s_2 = \infty \tag{9.1b}$$

$$3.\ s_1 = \infty \text{ and } s_2 = \infty \tag{9.1c}$$

Applying the transformation method to each variable, the value of the function at each of the above points is defined as

$$B(\infty, s_2') = B_1(0, s_2') \quad \text{where } B_1(u, s_2) = B(1/u, s_2), \left|s_2'\right| < \infty \tag{9.2a}$$

$$B(s_1', \infty) = B_2(s_1', 0) \quad \text{where } B_2(s_1, v) = B(s_1, 1/v), \left|s_1'\right| < \infty \tag{9.2b}$$

$$B(\infty, \infty) = B_3(0, 0) \quad \text{where } B_3(u, v) = B(1/u, 1/v) \tag{9.2c}$$

9.3.1 Nonessential Singularities of the Second Kind

It is well known [3] that 2-V polynomials may have nonessential singularities of the second kind (NSSK) at some infinite distant points where the value of the polynomial is indeterminate. For the sake of notational convenience and to indicate the type of indeterminacy involved we write $B(s_{10}, s_{20}) = 0/0$ to say that $B(s_1, s_2)$ has an NSSK at (s_{10}, s_{20}). Of course, for a polynomial $B(s_1, s_2)$, this can occur only if s_{10}, s_{20}, or both, have infinite value.

9.4 Various Analog Hurwitz Polynomials

Following up the definition of 1-V Hurwitz polynomials, 2-D Hurwitz polynomials have been traditionally defined on the basis of the regions in the S_2-biplane where the 2-V polynomials have no zeroes or have only some constrained types of zeroes. At least eight types of Hurwitz polynomials have been

defined. Below we present their definitions so as to bring out their zero-free regions and the constraints (if any) on their zeroes in some regions.

9.4.1 Self-Paraconjugate Polynomial

A 2-V polynomial $B(s_1, s_2)$ is said to be an (analog) self-paraconjugate polynomial if $B_*(s_1, s_2) = cB(s_1, s_2)$, where c is a unimodular complex constant, i.e., $|c| = 1$. When $c = +1$, $B(s_1, s_2)$ is said to be a para-even polynomial, and when $c = -1$, a para-odd polynomial. It is easy to verify that the para-even and para-odd parts of a polynomial are self-paraconjugate, i.e., $B_{e^*}(s_1, s_2) = B_e(s_1, s_2)$ and $B_{0^*}(s_1, s_2) = -B_0(s_1, s_2)$. In the case of real polynomials para-even and para-odd polynomials are simply called even and odd polynomials.

9.4.2 Broad-Sense Hurwitz Polynomial

This represents the weakest class of Hurwitz polynomials in the sense that their zero-free region in the S_2-biplane is the smallest of all the 2-D Hurwitz polynomials. $B(s_1, s_2)$ is a broad-sense Hurwitz polynomial (BHP) if $B(s_1, s_2) \neq 0$ in S^2_{++}. For example,

$$B_1(s_1, s_2) = (s_2^2 + 4)^i (s_1 + s_2)^j (s_1 s_2 + s_1 + s_2)^k (s_1 + s_2 + 1)^l (s_1 s_2 + s_1 + s_2 + 1)^m$$

where i, j, k, l, m are nonnegative integers. In the literature this Hurwitz polynomial has been known by different names: Hurwitz polynomial in the extended sense, widest sense Hurwitz, or simply Hurwitz [4–7,10].

9.4.3 Narrow-Sense Hurwitz Polynomial

A subclass of BHP is narrow-sense Hurwitz polynomial (NHP), which was introduced by Ansell [1] in the study of two-variable reactance functions (TRFs). They may be characterized by the following equivalent definitions:

1. $B(s_1, s_2)$ is an NHP if $B(s_1, s_2) \neq 0$ in $S^2_{\oplus\oplus} - S^2_{00} = S^2_{++} + S^2_{+0} + S^2_{0+}$ where the minus sign is used to denote set theoretic subtraction and the plus sign, set theoretic union.
2. $B(s_1, s_2)$ is an NHP if $B(s_1, s_2)$ is a BHP [i.e., $B(s_1, s_2) \neq 0$ in S^2_{++}] and $B(s_1, s_2)$ has no 1-V factor having zeroes on the imaginary axis of the S_1- or S_2-plane. For example,

$$B_1(s_1, s_2) = (s_1 + s_2)^i (s_1 s_2 + s_1 + s_2)^j (s_1 + s_2 + 1)^m (s_1 s_2 + s_1 + s_2 + 1)^n$$

   where i, j, m, n are nonnegative constants, is an NHP, whereas $(S_1^2 + 1)(s_1 + s_2 + 1)$ is not an NHP because of the factor $(S_1^2 + 1)$.

We see that an NHP may have zeroes on S^2_{00}. The irreducible 2-V factors that give rise to zeroes on S^2_{00} with no zeroes on S^2_{++} can be shown to belong to one of two types: (1) those that become zero only at isolated points on S^2_{00}; (2) those that become zero on a continuum of points on S^2_{00}. The fact that no irreducible factor can have isolated zeroes as well as continuum of zeroes without having zeroes in S^2_{++} can be established based on the continuity property of the zeroes of 2-V polynomials. For example, the factor $(s_1 s_2 + s_1 + s_2)$ with a zero at (0,0) on S^2_{00} corresponds to the first type, and the factor $s_1 + s_2$ with a continuum of zeroes given by $s_1 = -s_2$ on S^2_{00} corresponds to the second type.

9.4.4 Scattering Hurwitz Polynomial

Based on the above observation, the next class is identified as a subclass of NHP which can have only discrete (isolated) zeroes on S^2_{00}. Scattering Hurwitz polynomial (SHP) or principal Hurwitz polynomial corresponds to this class [7,13,18]. The following equivalent definitions characterize an SHP:

1. A 2-V polynomial $B(s_1, s_2)$ is an SHP if $B(s_1, s_2)$ is an NHP and if $B(s_1, s_2)$ has no (1-D) continuum of zeroes on S^2_{00}.
2. A 2-V polynomial $B(s_1, s_2)$ is an SHP if $B(s_1, s_2)$ is a BHP and if $B(s_1, s_2)$ has no (1-D) continuum of zeroes on S^2_{00}.
3. A 2-V polynomial $B(s_1, s_2)$ is an SHP if $B(s_1, s_2)$ is a BHP and if $B(s_1, s_2)$ has no 1- or 2-V self-paraconjugate factors.
4. A 2-V polynomial $B(s_1, s_2)$ is an SHP if $B(s_1, s_2)$ is a BHP, i.e., $B(s_1, s_2) \neq 0$ in S^2_{++}, and if $B(s_1, s_2)$ and $B_*(s_1, s_2)$ are relatively prime. For example,

$$B(s_1, s_2) = (s_1 s_2 + s_1 + s_2)^i (s_1 + s_2 + 4)^j (3s_1 s_2 + s_1 + s_2 + 1)^k$$

   where *i, j, k* are nonnegative integers.

Fettweis [14] identified and popularized this class and suggested the names "principal Hurwitz polynomials" and "scattering Hurwitz polynomials." He studied the properties of SHP in depth, and his pioneering efforts established SHP as a very important class of Hurwitz polynomials in stability and passivity studies.

9.4.5 Hurwitz Polynomial in the Strict Sense

Increasing the zero-free regions of a Hurwitz polynomial further by the addition of the whole finite imaginary axes of the S_1- and S_2-planes, we get the definition of a Hurwitz polynomial in the strict sense: $B(s_1, s_2)$ is a Hurwitz polynomial in the strict sense (HPSS) if $B(s_1, s_2) \neq 0$ in $\{S^2_{\oplus\oplus}$–infinite distant points$\}$. For example,

$$B(s_1, s_2) = (s_1 + s_2 + 7)^m (4s_1 s_2 + s_1 + s_2 + 1)^n$$

where integers m and n are nonnegative.

This class appears to have been first defined and used by Saito [4] and Youla [5,17]. This strict Hurwitz polynomial definition has been used by a number of authors such as Huang [8], in the derivation of stability tests for 2-D digital filters, Strintzis [15], in the extension of digital filter stability tests to continuous domain functions, and Goodman [16,23], in the study of double bilinear transformation. In order to get a 2-D Hurwitz polynomial that is a counterpart of 1-D strict Hurwitz we need to include the infinite distant points in the 2-D plane. (Note that, in certain cases, 2-D scattering Hurwitz appears to be the counterpart of the 1-D strict Hurwitz polynomial.)

9.4.6 Very Strict Hurwitz Polynomial

By including the infinite distant points we get the most strict Hurwitz polynomial, the very strict Hurwitz polynomial (VSHP), named by Reddy et al. [2,24]. Similar conclusions were reached by Delsarte et al. [9]. The following equivalent definitions characterize this type of Hurwitz polynomial:

1. $B(s_1, s_2)$ is a VSHP if $B(s_1, s_2) \neq 0$ or if $B(s_1, s_2) \neq 0/0$ in $S^2_{\oplus\oplus}$
2. $B(s_1, s_2)$ is a VSHP if $1/B(s_1, s_2)$ has no (first or second kind) singularities in $S^2_{\oplus\oplus}$. For example,

$$B(s_1, s_2) = (5s_1 s_2 + s_1 + s_2 + 8)$$

   We discuss 2-D VSHP in detail in a later section of this chapter.

9.4.7 Self-Paraconjugate Hurwitz Polynomial

$B(s_1, s_2)$ is said to be a self-paraconjugate Hurwitz polynomial (SPHP) if $B(s_1, s_2)$ is a self-paraconjugate polynomial and if $B(s_1, s_2) \neq 0$ in S^2_{++}. In other words, $B(s_1, s_2)$ is a BHP. For example,

$$B(s_1, s_2) = (s_1^2 + 1)^j (s_1 s_2 + 1)^k$$

where j, k are nonnegative integers.

9.4.8 Reactance Hurwitz Polynomial

A reactance Hurwitz polynomial (RHP) is defined as the para-even or para-odd part of a SHP. $B(s_1, s_2)$ is said to be a RHP if $B(s_1, s_2)$ is an SPHP and if each zero locus of $B(s_1, s_2)$ on S^2_{00} is of multiplicity unity; i.e., $B(s_1, s_2)$ has no repeated factors. For example,

$$B(s_1, s_2) = (s_1^2 + 1)(s_1 s_2 + 1)^k$$

is an RHP whereas

$$B(s_1, s_2) = (s_1^2 + 1)^2 (3 s_1 s_2 + 1)^2$$

is not an RHP.

9.4.9 Immittance Hurwitz Polynomial

An immittance Hurwitz polynomial (IHP) is defined as the product of an RHP and an SHP [10]. An alternate definition is given in terms of the zero-free regions.

$B(s_1, s_2)$ is said to be an IHP if $B(s_1, s_2)$ is a BHP and if each continuum zero locus of $B(s_1, s_2)$ on S^2_{00} is of multiplicity unity; i.e., $B(s_1, s_2)$ has no repeated self-paraconjugate factors. For example, $B(s_1, s_2) = (s_1^2 + 1)(s_1 s_2 + 1)(2 s_1 s_2 + 7 s_1 + s_2)(s_1 + s_2 + 1)$.

9.4.10 Summary

The definitions of BHP, NHP, SHP, HPSS, and VSHP are such that in the above sequence each Hurwitz polynomial satisfies the conditions required for each of the preceding polynomials, and hence form a subset of each. In other words,

$$\{\text{VSHP}\} \subset \{\text{HPSS}\} \subset \{\text{SHP}\} \subset \{\text{NHP}\} \subset \{\text{BHP}\}$$
$$\{\text{IHP}\} \subset \{\text{BHP}\}$$
$$\{\text{SPHP}\} \subset \{\text{BHP}\}$$
$$\{\text{RHP}\} = \{\text{IHP}\} \cap \{\text{SPHP}\}$$

9.5 Testsets for Analog Hurwitz Polynomials

9.5.1 Continuity Property of the Zeroes of 2-V Polynomials

Let $B(s_1, s_2)$ be a 2-V polynomial of degree m in s_1 and n in s_2. Let x_1 be a point in the S_1-plane. Then, $B(x_1, s_2)$ is a 1-V polynomial in s_2 and has n zeroes (possible some at infinity) in the S_2-plane. When x_1 is such that the coefficients of r consecutive terms s_2^n, $s_2^{n-1}, \ldots, s_2^{n-r+1}$ become zero, $B(x_1, s_2)$ becomes a

polynomial of degree $n-r$ in s_2 and has only $(n-r)$ zeroes in the finite part of the S_2-plane. If this polynomial is considered an $n-r$ degree polynomial in s_2, then it has n poles at $s_2=\infty$. Further, we also know that for a polynomial (in general, for a rational function), the number of poles and zeroes are equal when those that are at infinity are also counted. Hence, in the present use, where r leading coefficients become zero, the r missing zeroes are accounted to be at infinity.

As discussed earlier, the infinite distant points of the S_2-plane are assumed to converge to a single point at infinity. We further note that at an NSSK the function possesses zeroes as well as poles. Consider a zero s_2' of $B(x_1, s_2)$. When x_1 is moved on a continuous line in the S_1-plane, S_2' either moves continuously or remains stationary. Further, when $K(K \leq n)$, zeroes are present at a point, among which K' zeroes are stationary, $(K-K')$ incoming and $(K-K')$ outgoing loci exist at that point. We may associate an incoming locus with an outgoing locus arbitrarily on a one-to-one basis. With these remarks in mind, the continuity property of the zeroes of 2-V polynomials can be stated in the form of a theorem.

THEOREM 9.1

Let $B(x_1, s_2)$ be not identically zero. [*If $B(x_1, s_2) \equiv 0, (s_1 - x_1)^k$, $K \leq M$, is a factor of $B(s_1, s_2)$. Divide this factor out and consider the resulting polynomial*]. *Then, the locus of a zero $\hat{s}_2$ of $B(x_1, s_2)$ generated by the movement of x_1 on a continuous line in the S_1-plane is a continuous line or a fixed point in the S_2 plane. In the latter case, $(s_2 - \hat{s}_2)$ is a factor of $B(s_1, s_2)$.*

Proof: The proof of Theorem 9.1 can be given based on the properties of algebraic functions discussed by Bliss [27]. Also note that the locus may pass through the point at infinity; this is treated just like any other point in the S_2-plane. A similar property also holds for the zeroes s_1 of $B(s_1, x_2)$.

As a first step to identifying smaller testsets, we show that, because of the continuity property of the zeroes of 2-V polynomials, tests at some isolated points can be omitted. We do this for the tests in the region s_{0+}^2 in the following lemma [24].

LEMMA 9.1

A 2-V polynomial $B(s_1, s_2) \neq 0$ or $0/0$ in the region S_{0+}^2 if $B(s_1, s_2) \neq 0$ or $0/0$ in $\{S_{10}$–some isolated points$\} \times S_{2+}$ (i.e., the Cartesian product of the whole imaginary axis of the S_1-plane, except some isolated points, and the open right half of the S_2-plane), and if $B(\hat{s}_1, s_2) \neq 0$, where $\hat{s}_1$ is any such isolated point in S_{10} [*i.e., no 1-V factor of the form $(s_1 - \hat{s}_1)$ is present in $B(s_1, s_2)$*].

COROLLARY 9.1

A 2-V polynomial $B(s_1, s_2) \neq 0$ or $0/0$ in the region S_{0+}^2 if $B(jw_1, s_2) \neq 0$ for $-\infty < w_1 < \infty$ and $Re(s_2) > 0$, $|s_2| < \infty$.

Note that to test for the absence of the zeroes of $B(s_1, s_2)$ in $s_{0\oplus}^2$, we must test all the boundary points and we cannot leave any isolated point untested, as in the case of s_{0+}^2.

Next, we discuss test procedure for the various Hurwitz polynomials.

THEOREM 9.2

A 2-V polynomial $B(s_1, s_2)$ is a BHP iff $B(jw_1, s_2) \neq 0$ in S_{2+} for all real finite w_1, except possibly some isolated w_1 and if the polynomial $B(s_1, b)$ has no zeroes in S_{1+} for some $b \in S_{2+}$.

THEOREM 9.3

A 2-V polynomial $B(s_1, s_2)$ is an NHP iff for all real finite w_1, $B(jw_1, s_2) \neq 0$ in S_{2+}; the polynomial in s_1, $B(s_1, b) \neq 0$ in S_{1+} for some $b \in S_{2+}$, and $B(s_1, s_2)$ has no factor of the type $(s_2 - j\alpha)$, where α is a real constant.

THEOREM 9.4

A 2-V polynomial $B(s_1, s_2)$ is an SHP iff for all real finite w_1, $B(jw_1, s_2)$ has no zeroes in S_{2+}; the polynomial $B(s_1, b)$ has no zeroes in S_{1+} for some $b \in S_{2+}$, and $B(s_1, s_2)$ and $B_*(s_1, s_2)$ are relatively prime.

THEOREM 9.5

A 2-V polynomial $B(s_1, s_2)$ is a HPSS iff for all finite w_1, $B(jw_1, s_2)$ has no zeroes in $\{s_2: Re(s_2) \geq 0, |s_2| < \infty\}$ and the polynomial, $B(s_1, b)$ has no zeroes in Re $s_1 \geq 0$, for some $b \in S_{2+}$. It is easy to verify that only infinite distant points on S_{00}^2 are omitted from S_{00}^2 in testing for the zero locations of $B(s_1, s_2)$. Hence, $B(s_1, s_2)$ is a HPSS.

THEOREM 9.6

The necessary and sufficient conditions for a 2-V polynomial $B(s_1, s_2)$ to be a VSHP are [19]

1. *$B(s_1, s_2)$ is a HPSS*

2*a*. *$B(\infty, s_2) \neq 0/0$ for $Re(s_2) = 0$ and $|s_2| < \infty$*

2*b*. *$B(s_1, \infty) \neq 0/0$ for $Re(s_1) = 0$ and $|s_1| < \infty$*

2*c*. *$B(\infty, \infty) \neq 0/0$*

The infinite point testing method shown in Equations 9.2a *through* c *could be followed for the testing of the three conditions under number* 2 *above.* Let

$$B(s_1, s_2) = \sum_{i=0}^{M} \sum_{j=0}^{N} b_{ij} s_1^i s_2^j$$

Then, conditions 2a to 2c are equivalent to:

$$A_M(s_2) = \sum_{j=0}^{N} b_{Mj} s_2^j \neq 0 \quad \text{for Re}(s_2) = 0$$

$$B_N(s_1) = \sum_{i=0}^{M} b_{iN} s_1^i \neq 0 \quad \text{for Re}(s_1) = 0$$

$$b_{MN} \neq 0$$

THEOREM 9.7

Let

$$\begin{aligned} B(s_1, s_2) &= \sum_{j=0}^{N} a_j(s_1) s_2^j \\ &= \sum_{i=0}^{M} b_i(s_2) s_1^i \\ &= \sum_{i=0}^{M} \sum_{j=0}^{N} c_{ij}\ s_1^i s_2^j \end{aligned}$$

Then, $B(s_1, s_2)$ is a VSHP iff $B(jw_1, jw_2) \neq 0, -\infty < w_i < \infty\ i = 1, 2$, and $a_N(s_1) \neq 0$ in $S_{1\oplus}$, and $b_M(s_2) \neq 0$ in $S_{2\oplus}$, and $c_{MN} \neq 0$.

Testsets for SPHP, RHP, and IHP can be formulated easily based on their definitions and the testset of a BHP.

9.6 Two-Variable Very Strict Hurwitz Polynomials

A brief, additional discussion of VSHPs is given in this section because this class of 2-V Hurwitz polynomials is a counterpart to the 1-V SHP, at least from the domain description of a closed right-half plane. We now state some of the properties of 2-V VSHPs. Let a two-variable transfer function $T(s_1, s_2)$ be expressed as

$$T(s_1, s_2) = P(s_1, s_2)/Q(s_1, s_2) \tag{9.3}$$

where

$$P(s_1, s_2) = \sum_i \sum_j p_{ij} s_1^i s_2^j \quad (i = 0, 1, \ldots, k; j = 0, 1, \ldots, l)$$

$$Q(s_1, s_2) = \sum_i \sum_j q_{ij} s_1^i s_2^j \quad (i = 0, 1, \ldots, m; j = 0, 1, \ldots, n)$$

By applying the transformation method of Equations 9.2a through c it can be shown that unless $m \geq k$ and $n \geq l$, polar singularities exist at a set of infinite distant points in the closed right half of the $\{S_1, S_2\}$-biplane. Therefore, assume the $m \geq k$ and $n \geq 1$. Then, the following theorem regarding the singularity in the closed right-half biplane can be stated.

THEOREM 9.8

$T(s_1, s_2)$ does not possess any singularity on the closed right half of $\{S_1, S_2\}$-biplane defined by $S^2_{\oplus\oplus}$ iff $Q(s_1, s_2)$ is a VSHP.

Proof: The proof of the theorem is straightforward if the infinite distant points are also taken into account.

Some other useful properties of VSHPs are [2]

- $B(s_1, s_2) = [B_1(s_1, s_2).\ B_2(s_1, s_2)]$ is a VSHP iff $B_1(s_1, s_2)$ and $B_2(s_1, s_2)$ are VSHPs.
- If $B(s_1, s_2)$ is a VSHP, then $\partial/\partial s_i\ [B(s_1, s_2)]$, $i = 1, 2$ are also VSHPs. This property is not true for other 2-D Hurwitz polynomials. Let

$$B(s_1, s_2) = A_M(s_2)s_1^M + A_{M-1}(s_2)s_1^{M-1} + \cdots + A_1(s_2)s_1 + A_0(s_2) \tag{9.4}$$

$$= C_N(s_1)s_2^N + C_{N-1}(s_1)s_2^{N-1} + \cdots + C_1(s_1)s_2 + C_0(s_1) \tag{9.5}$$

- Let $B(s_1, s_2)$ be expressed as in Equations 9.4 and 9.5. Then, $A_i(s_2)$, $i = 0, 1, \ldots, M$ and $C_j(s_1)$, $j = 0, 1, \ldots, N$ are 1-V strict Hurwitz polynomials. This property readily follows from the preceding partial derivative property.
- Let $B(s_1, s_2) = \Sigma_i\ \Sigma_j\ b_{ij}\ s_1^i s_2^j$ be a real 2-D VSHP. Then $b_{MN}\ b_{ij} > 0$ for all i and j ($i = 0, 1, \ldots, M$; $j = 0, 1, \ldots, N$).
- Let $B(s_1, s_2)$ be expressed as in Equations 9.4 and 9.5. Then, $A_i(s_2)/A_{i-1}\ (s_2)$ for $i = 1, \ldots, M$ and $C_j(s_1)/C_{j-1}(s_1)$ for $j = 1, \ldots, N$ are minimum reactive, susceptive, strict PRFs.

The preceding property gives the following the necessary and sufficient condition for $B(s_1, s_2)$, which has a first degree in s_1 and any degree in s_2.

The necessary and sufficient condition that allows a 2-V polynomial $B(s_1, s_2) = B_1(s_2)s_1 + B_0(s_2)$ to be a VSHP is that the 1-V function $F(s_2) = B_1(s_2)/B_0(s_2)$ be a minimum reactive, susceptive, strict PRF.

Finally, we give a transformation theorem that transforms a 1-D strict Hurwitz polynomial into a 2-D VSHP. This is called a reactance transformation.

THEOREM 9.9

Let $D(s)$ be any strict Hurwitz polynomial of order n. Generate a 2-D polynomial in the following way:

$$B(s_1, s_2) = [N(s_1, s_2)\,]^n \cdot \{D(s)\}|s = M(s_1, s_2)/N(s_1, s_2)$$

where M and N are, respectively, the even and odd 2-D polynomials. The necessary and sufficient condition for $B(s_1, s_2)$ to be a VSHP is that $M(s_1, s_2) + N(s_1, s_2)$ be a VSHP [2].

The odd TPRF $Z(s_1, s_2) = M(s_1, s_2)/N(s_1, s_2)$ does not possess NSSK on the distinguished boundary S_{00}^2 and is called a proper or strict 2-D reactance function [2].

9.7 Application of Two-Dimensional Hurwitz Polynomials for Two-Variable Passive Networks and Stability

This section enumerates some properties of 2-V passive network functions, with particular reference to the Hurwitz nature of the polynomials [24]. (The following assumes Re $F(1, 1) > 0$.)

Let $F(s_1, s_2) = N(s_1, s_2)/D(s_1, s_2)$ be the driving-point immittance of a passive network. Then, $N(s_1, s_2)$ and $D(s_1, s_2)$ are BHPs. Let the common factors of $N(s_1, s_2)$ and $D(s_1, s_2)$ be canceled out and the resulting polynomials be called $N_1(s_1, s_2)$ and $D_1(s_1, s_2)$. Then, $N_1(s_1, s_2)$ and $D_1(s_1, s_2)$ are immittance Hurwitz polynomials.

Let $F(s_1, s_2) = A(s_1, s_2)/B(s_1, s_2)$ be a relatively prime 2-V odd rational function. Then, $F(s_1, s_2)$ is a 2-V reactance function if $A(s_1, s_2) + B(s_1, s_2)$ is a VSHP. $F(s_1, s_2)$ is A 2-V reactance function iff $A(s_1, s_2) + B(s_1, s_2)$ is a scattering Hurwitz polynomial. The self-paraconjugate polynomials $A(s_1, s_2)$ and $B(s_1, s_2)$ satisfy the reactance Hurwitz properties.

A relatively prime 2-V odd function $F(s_1, s_2) = A(s_1, s_2)/B(s_1, s_2)$ having no second-kind singularities is a reactance function iff $A(s_1, s_2) + B(s_1, s_2)$ is a VSHP. Such functions are called proper or strict reactance functions [2], and are useful as transformation functions to generate a (structurally stable) 2-D network from a stable 1-D network. This is one of the main applications of VSHP.

Let us now consider a relatively prime function $F(s_1, s_2) = N(s_1, s_2)/D(s_1, s_2)$. $F(s_1, s_2)$ is a TPRF iff $N(s_1, s_2) + D(s_1, s_2)$ is a scattering Hurwitz polynomial. Further, if no second-kind singularities exist for $F(s_1, s_2)$ on S^2_{00}, $N(s_1, s_2) + D(s_1, s_2)$ will be a VSHP. From the previous discussion, we can conclude that the Hurwitz nature determines important necessary conditions (and in some cases necessary and sufficient conditions) of 2V positive lossless functions. Hurwitz polynomials can be used to generate 2-V positive and lossless functions as in 1-V case through partial derivative operations [6].

The following property relates to sum separability and Hurwitz nature [20]. Let $F(s_1, s_2) = N(s_1, s_2)/D(s_1, s_2)$ be a 2-V positive function. Assume $D(s_1, s_2)$ is an immittance Hurwitz polynomial having self-paraconjugate factors. In other words, $D(s_1, s_2)$ is written as $D(s_1, s_2) = D_1(s_1, s_2)D_2(s_1, s_2)$, where $D_1(s_1, s_2)$ is a reactance Hurwitz and $D_1(s_1, s_2)$ is a scattering Hurwitz. Then, $F(s_1, s_2)$ is sum separable as

$$F(s_1, s_2) = \frac{N_1(s_1, s_2)}{D_1(s_1, s_2)} + \frac{N_2(s_1, s_2)}{D_2(s_1, s_2)}$$

where N_1/D_1 is a reactance function and N_2/D_2 is a positive function.

Now, we turn our attention to some applications concerning transfer functions. Let $T(s_1, s_2) = A(s_1, s_2)/B(s_1, s_2)$ be the transfer function of a singly terminated or doubly terminated 2-V lossless network. Then, $B(s_1, s_2)$ is a scattering Hurwitz polynomial. References [21,22] provide a detailed discussion of networks with transfer functions having scattering and VSHP denominators. It is not necessary that the denominator of all RLC 2-V network transfer functions be scattering Hurwitz. In the most general case it could be a broad-sense Hurwitz.

Another interesting observation is that in the 1-V case the voltage transfer function cannot have a pole at origin and infinity. Extending this to the 2-V situation, we find that the 2-V voltage transfer function, $T(s_1, s_2)$ cannot have first-kind (polar) singularities at $s_i = 0$ or ∞ $(i = 1, 2)$, but $T(s_1, s_2)$ can be 0/0 at $s_i = 0$ or ∞ $(i = 1, 2)$.

Let $H(s_1, s_2) = P(s_1, s_2)/Q(s_1, s_2)$ be a 2-V bounded real or lossless bounded real function. Then, $Q(s_1, s_2)$ is a scattering Hurwitz polynomial. If $H(s_1, s_2)$ has no NSSK on S^2_{00} then $Q(s_1, s_2)$ must be a VSHP.

9.7.1 Application to Two-Dimensional Analog System Stability

We consider the following important theorem [12].

THEOREM 9.10

The 2-D analog transfer function $T(s_1, s_2) = A(s_1, s_2)/B(s_1, s_2)$ is bounded-input–bounded-output (BIBO) stable only if $B(s_1, s_2)$ is a scattering Hurwitz polynomial. The sufficient *condition for stability is that $B(s_1, s_2)$ be a VSHP (assume that $T(s_1, s_2)$ has no polar singularities at infinite distant points).*

We conclude this section with the following unresolved problem of BIBO stability of 2-D continuous time systems [12]:

Conjecture: *The 2-D analog transfer function $T(s_1, s_2)$ described in Theorem* 9.10 *is BIBO stable with no NSSK on S^2_{00} iff $B(s_1, s_2)$ is a VSHP.*

The sufficiency part of this statement is proved. The necessity has yet to be established.

9.8 Conclusion

This chapter provides a comprehensive, yet compact treatment of the theory of 2-D (analog) Hurwitz polynomials. With the help of double bilinear transformation $s_1 = (1 - z_i)/(1 + z_i)$, $i = 1, 2$, most of the theory could easily be translated to the 2-D discrete case, and thus to the stability theory and design of 2-D digital filters [2,24]. As in the 1-D case the 2-D Hurwitz polynomials play a critical role in the study of 2-D circuits, systems, and filters. In this chapter, a detailed classification and testing (see theorems) of various 2-D Hurwitz polynomials is presented. Discussion of the properties of 2-D very strict Hurwitz polynomials is also given. The various testing procedures (algorithms) are not discussed. The test procedures can be found in Refs. [11,12,25,26]. The chapter concludes by discussing how various Hurwitz polynomials arise in passive 2-V circuit theory and 2-D analog stability.

Acknowledgments

As always, Dr. P. K. Rajan of the Tennessee Technological University and Dr. E. I. Jury of the University of California at Berkeley and the University of Miami provided constant encouragement. Dr. Rajan made significant contributions to the theory of stable 2-D polynomials discussed in this chapter. The material in this chapter is based mainly on Refs. [2,24] (including Figure 9.1).

References

1. H. G. Ansell, On certain two-variable generalization of circuit theory with applications to networks and transmission lines of lumped reactances, *IEEE Trans. Circuit Theory*, CT-11, 214–223, June 1964.
2. H. C. Reddy et al., Generation of two-dimensional digital transfer functions without nonessential singularities of the second kind, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*., Washington, D.C., pp. 13–19, Apr. 1979. Also see P. K. Rajan et al., *IEEE Trans. Acoust., Speech, Signal Process*., pp. 216–223, Apr. 1980.
3. V. S. Valdimirov, *Method of Theory of Functions of Many Complex Variables*, Cambridge, MA: MIT Press, 1966, pp. 36–38.
4. M. Saito, Synthesis of transmission line networks by multivariable techniques, in *Proc. Symp. Generalized Networks*, PIB, New York, 1966, pp. 353–393.
5. D. C. Youla, Synthesis of networks containing lumped and distributed elements, in *Proc. Symp. Generalized Networks*, PIB, New York, 1966, pp. 289–343.
6. V. Ramachandran, Some similarities and dissimilarities between single variable and two-variable reactance functions, *IEEE Circuits Syst. Newsl*., pp. 11–14, 1976.
7. A. Fettweis, On the scattering matrix and the scattering transfer matrix of multidimensional lossless two-ports, *Arch. Elk. Ubertragung*, 36, 374–381, Sept. 1982.
8. T. S. Huang, Stability of two-dimensional recursive digital filters, *IEEE Trans. Audio Electroacoust*., AU–20, 158–163, June 1972.
9. Ph. Delsarte, Y. Genin, and Y. Kamp, Two–variable stability criteria, *Proc. IEEE Trans. Int. Symp. Circuits Syst*., Tokyo, Japan, pp. 495–498, Jul. 1979.
10. A. Fettweis, On Hurwitz polynomials in several variables, in *Proc.* 1983 *IEEE Int. Symp. Circuits and Syst*., Newport Beach, CA, 1983, pp. 382–385.
11. N. K. Bose, *Applied Multidimensional Systems Theory*, New York: Van Nostrand Reinhold, 1982.
12. E. I. Jury, Stability of multidimensional systems and related problems, in *Multidimensional Systems—Techniques and Applications*, S.G. Tzafestas, Ed., New York: Marcel Dekker, 1986.
13. A. Fettweis and S. Basu, On discrete scattering Hurwitz polynomials, *Int. J. Circuit Theory Appl*., 13, 47–59, Jan. 1985.

14. A. Fettweis, Some properties of scattering Hurwitz polynomials, *Arch. Elk. Ubertragung*, 38, 171–176, 1984.
15. M. G. Strintzis, Tests of stability of multidimensional filters, *IEEE Trans. Circuits Syst.*, CAS-24, 432–437, Aug. 1977.
16. D. Goodman, Some difficulties with double bilinear transformation in 2-D filter design, *Proc. IEEE*, 66, 905–914, June 1977.
17. D. C. Youla, The analysis and synthesis of lumped passive *n*-dimensional networks—Part I—Analysis, Polytechnic Inst. New York, Brooklyn, Report MIR-1437-84, Jul. 1984.
18. A. Fettweis and S. Basu, New results on multidimensional Hurwitz polynomials, in *Proc. Int. Symp. Circuits Syst.*, Kyoto, Japan, June 1985, pp. 1359–1362.
19. H. C. Reddy and P. K. Rajan, A simpler test-set for very strict Hurwitz polynomials, *Proc. IEEE*, 890–891, June 1986.
20. H. C. Reddy et al., Separability of mutivariable network driving point functions, *IEEE Trans. Circuits Syst.*, CAS-29, 833–840, Dec. 1982.
21. H. C. Reddy et al., Realization of resistively terminated two-variable lossless ladder networks, *IEEE Trans. Circuits Syst.*, CAS–29, 827–832, Dec. 1982.
22. H. C. Reddy et al., Design of two-dimensional digital filters using analog reference filters without second kind singularities, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, Georgia, Apr. 1981, pp. 692–695.
23. D. Goodman, Some stability properties of linear shift invariant digital filters, *IEEE Trans. Circuits Syst.*, 26, 201–208, Apr. 1971.
24. H. C. Reddy and P. K. Rajan, A comprehensive study of two-variable Hurwitz polynomials, *IEEE Trans. Educ.*, 32, 198–209, Aug. 1989.
25. H. C. Reddy and P. K. Rajan, A test procedure for the Hurwitz nature of two-dimensional analog polynomials using complex lossless function theory, *Proc. IEEE Int. Symp. Circuits Syst.*, Philadelphia, Pennsylvania, 702–705, May 1987.
26. P. K. Rajan and H. C. Reddy, Hermite matrix test for very strict Hurwitz polynomials, *Proc. Midwest Symp. Circuits Syst.*, Lincoln, Nebraska, 670–673, Aug. 1986.
27. G. A. Bliss, *Algebraic Functions*, New York: American Mathematical Society, 1933.

10

Application of Symmetry: Two-Dimensional Polynomials, Fourier Transforms, and Filter Design

10.1 Introduction **10**-2

10.2 Basic Symmetry Definitions **10**-2
Nature of T-Operations • Nature of ψ-Operations • Various T–ψ Symmetries

10.3 Two-Dimensional Fourier Transform Pairs with Symmetry **10**-7
Continuous-Time Continuous-Frequency Case • Continuous-Time Discrete-Frequency Case (Fourier Series) • Discrete-Time Continuous-Frequency Case • Discrete-Time Discrete-Frequency Case (Discrete Fourier Transform)

10.4 Symmetry and 2-D Fast Fourier Transform **10**-10
Symmetry-Based 1-D FFT • 2-D Symmetries • Symmetrical Decomposition for Data without Symmetry

10.5 Symmetry in 2-D Magnitude Response **10**-15
Continuous-Time Frequency Response • Discrete-Time Frequency Response • Symmetry in Magnitude Response

10.6 Polynomial Operations and Definitions **10**-17

10.7 Spectral Forms for Magnitude-Squared Function **10**-19

10.8 Determination of 2-D Polynomials Possessing Various Symmetries **10**-21
Quadrantal Symmetry for Analog Polynomials • Diagonal Symmetry for Analog Polynomials • Fourfold (90°) Rotational Symmetry for Analog Polynomials • Octagonal Symmetry for Analog Polynomials • Summary of Analog and Digital Polynomial Factors Possessing Symmetry

10.9 Symmetry and Stability **10**-27
Stability of Analog Filters with Symmetric Magnitude Response • Stability of Digital Filters with Symmetric Magnitude Response

Hari C. Reddy
California State University and National Chiao-Tung University

I-Hung Khoo
California State University

P. K. Rajan
Tennessee Tech University

10.10 Filter Design Procedure **10**-30
10.11 Filter Design Examples **10**-31
Bandpass Filter • Low-Pass Filter • High-Pass Filter • Bandstop Filter
Appendix: MATLAB Programs for Filter Design. **10**-36
References **10**-39

10.1 Introduction

Two-dimensional (2-D) digital filters find applications in many fields such as image processing and seismic signal processing. Design of 2-D digital filters is more complicated than that of 1-D digital filters because of the increase in the number of coefficients with the increase in the dimension and also the difficulty of testing their stability. Fortunately, 2-D frequency responses possess many types of symmetries and the presence of these symmetries can be used to reduce the complexity of the design as well as the implementation of these filters. Symmetry in the frequency response of a filter induces certain constraints on the coefficients of the filter, which in turn reduces the filter design complexity [1–28]. Therefore, a study of the symmetries of the filter frequency responses and the resulting constraints on the filter coefficients is undertaken in this chapter. As there is a close relationship between digital and analog filter functions, symmetry properties are discussed in this chapter for both analog and digital domain functions. In addition to filter design, symmetry can also be applied in the computation of Fourier transform of 2-D signals. To facilitate this, the symmetry properties in the Fourier transform pairs will be presented. It will also be shown that the presence of symmetry will reduce the complexity in the implementation of the fast Fourier transform (FFT) of 2-D signals [29].

The following is the layout of the sections. First, the symmetries are defined. Then, the Fourier transform pairs with symmetry and the use of symmetry in the implementation of FFT are presented. Next, symmetry in the magnitude functions of filters is discussed. This is followed by the symmetry constraints on polynomials and a procedure to design 2-D filters employing the constraints. Finally, several examples are given to illustrate the application of the symmetry-based filter design procedure.

10.2 Basic Symmetry Definitions

Let us consider a real rational function $f(x_1, x_2)$ in two independent variables x_1 and x_2, where the function f assumes a unique value for each pair of x_1 and x_2. The existence of symmetry in $f(x_1, x_2)$ implies the value of the function at (x_1, x_2) is related to the value at (x_{1T}, x_{2T}), where (x_{1T}, x_{2T}) is obtained by some operation on (x_1, x_2) [30–32]. Expanding on this concept, a function f is said to possess a symmetry if a pair of operations, T and ψ, performed simultaneously on the vector of variables, (x_1, x_2), and the function value leaves the shape of the function f undisturbed. The $T - \psi$ symmetry of a function is defined as follows:

Definition 10.1: A function $f(\underline{x})$ is said to possess a $T - \psi$ symmetry over a domain D if

$$\psi[f(T[\underline{x}])] = f(\underline{x}) \quad \text{for all } x \in D \tag{10.1}$$

where

ψ is an operation on the value of $f(\underline{x})$

T is an operation on $\underline{x}$ that maps D onto itself on a one-to-one basis

Depending on the $T-\psi$ operations, different symmetries are obtained such as $x_1 = x_2$ diagonal reflection antisymmetry, fourfold rotational conjugate symmetry, and so on. The nature of T and ψ operations is discussed next.

10.2.1 Nature of *T*-Operations

In the study of symmetry, the basic T-operation can be represented by the transformation:

$$T[\underline{x}] = \underline{A} \cdot \underline{x} + \underline{b} \tag{10.2}$$

where

$\underline{A}$ is a nonsingular (2×2) real square matrix
$\underline{b}$ is a (2×1) real vector

Some of the common T-transformations are displacement transformation, rotational transformation, and reflection transformation. In Table 10.1, we list only the basic reflection and rotational transformations where $\underline{b} = \underline{0}$ and the $\underline{A}$ matrices are formed using 1 or -1 as elements on the diagonal or off the diagonal. Of the seven operations listed, only the first five are needed as the product of them will give the remaining two. Thus we will only focus on operations (i)–(v) listed in the table.

It is to be noted that the T-operations can be compounded. For example, T_1T_2 refers to the compound operation consisting of operation T_2 followed by T_1. The compounding of transformations always obey the associative law, i.e., $T_1(T_2T_3) = (T_1T_2)T_3$.

The T-operations can also be classified by the number of cycles. A T-operation is said to be k-cyclic if k repeated T-operations on $\underline{x}$ yield the same original $\underline{x}$. That is: $T^k[\underline{x}] = \underline{x}$ or stated in another way $T^k = \underline{I}$ (where $\underline{I}$ is the identity matrix). For example, $T_1^2 = \underline{I}$ and $T_5^4 = \underline{I}$. So, operations (i)–(iv) listed in Table 10.1 are 2-cyclic, while (v) is 4-cyclic.

TABLE 10.1 Basic T-Operations

	$\underline{A}$	$\underline{b}$	Name of Operation	Symbol
(i)	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\underline{0}$	Reflection about x_1-axis	T_1
(ii)	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\underline{0}$	Reflection about x_2-axis	T_2
(iii)	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\underline{0}$	Reflection about $x_1 = x_2$ diagonal	T_3
(iv)	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$	$\underline{0}$	Reflection about $x_1 = -x_2$ diagonal	T_4
(v)	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\underline{0}$	90° clockwise rotation about origin	T_5
(vi)	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\underline{0}$	90° anticlockwise rotation about origin	T_6
(vii)	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$	$\underline{0}$	180° rotation about origin	T_7

10.2.2 Nature of ψ-Operations

The basic ψ-operation can be represented by

$$\psi[f(\underline{x})] = |f(\underline{x})|e^{j(\delta \angle f(\underline{x}) + \underline{\beta}^t \underline{x} + \phi)} \tag{10.3}$$

where

$\delta = \pm 1$
$\angle f(\underline{x})$ denotes the argument of $f(\underline{x})$
$\underline{\beta}$ is a (2×1) real constant vector
ϕ is a real constant

In Equation 10.3, it can be seen that ψ does not alter the magnitude of $f(\underline{x})$ as the parameters δ, $\underline{\beta}$, and φ affect only the argument of $f(\underline{x})$. Most commonly occurring symmetries have $\underline{\beta} = 0$. The term $\underline{\beta}'$ has been included in the ψ operation to account for delay type symmetries that may be present in some functions. In Table 10.2, we list four common ψ-operations that are used in various symmetry descriptions.

10.2.3 Various *T*–ψ Symmetries

The T and ψ operations in Equations 10.2 and 10.3 can be combined to form a composite symmetry operation that is described by the symmetry parameters ($\underline{A}$, $\underline{b}$, δ, $\underline{\beta}$, φ). This composite operation can be used to generate various symmetries. For example, using the T-operations (i)–(v) in Table 10.1 and assuming the identity ψ_I, we have the following standard symmetries. (Note that the corresponding antisymmetry, conjugate-symmetry, and conjugate-antisymmetry can be obtained by applying the ψ_A, ψ_C, and ψ_{CA} operations, respectively, shown in Table 10.2.)

1. Reflection about x_1-axis symmetry: $f(\underline{x}) = f(T_1[\underline{x}])$.
2. Reflection about x_2-axis symmetry: $f(\underline{x}) = f(T_2[\underline{x}])$.
3. Reflection about $x_1 = x_2$ diagonal symmetry: $f(\underline{x}) = f(T_3[\underline{x}])$.
4. Reflection about $x_1 = -x_2$ diagonal symmetry: $f(\underline{x}) = f(T_4[\underline{x}])$.
5. 90° clockwise rotational symmetry: $f(\underline{x}) = f(T_5[\underline{x}])$. This also implies $f(T_5[\underline{x}]) = f(T_5^2[\underline{x}])$ and $f(T_5^2[\underline{x}]) = f(T_5^3[\underline{x}])$, through repeated substitution of $\underline{x} = T_5[\underline{x}]$.
6. Centrosymmetry (180° rotation): $f(\underline{x}) = f(-\underline{x})$. This can also be expressed as $f(\underline{x}) = f(T_5^2[\underline{x}])$ or $f(\underline{x}) = f(T_1T_2[\underline{x}])$ or $f(\underline{x}) = f(T_3T_4[\underline{x}])$.

These symmetries are shown in Figure 10.1a through f. It is to be noted that the values of the function at corresponding points in the shaded regions are the same since we assume $\psi = \psi_I$.

The various symmetries can also be classified according to their cycles. Reflection symmetries about x_1-axis, x_2-axis, $x_1 = x_2$ diagonal, $x_1 = -x_2$ diagonal, and centrosymmetry are all 2-cyclic symmetries. On the other hand, 90° clockwise rotational symmetry is 4-cyclic, and we usually call it fourfold (90°) rotational symmetry.

TABLE 10.2 Basic ψ-Operations

δ	$\underline{\beta}$	φ	ψ-Operations	Symmetry Name	Symbol
1	$\underline{0}$	0	$\psi[f(\underline{x})] = f(\underline{x})$	Identity symmetry	ψ_I
1	$\underline{0}$	π	$\psi[f(\underline{x})] = -f(\underline{x})$	Antisymmetry	ψ_A
−1	$\underline{0}$	0	$\psi[f(\underline{x})] = [f(\underline{x})]^*$	Conjugate symmetry	ψ_C
−1	$\underline{0}$	π	$\psi[f(\underline{x})] = -[f(\underline{x})]^*$	Conjugate antisymmetry	ψ_{CA}

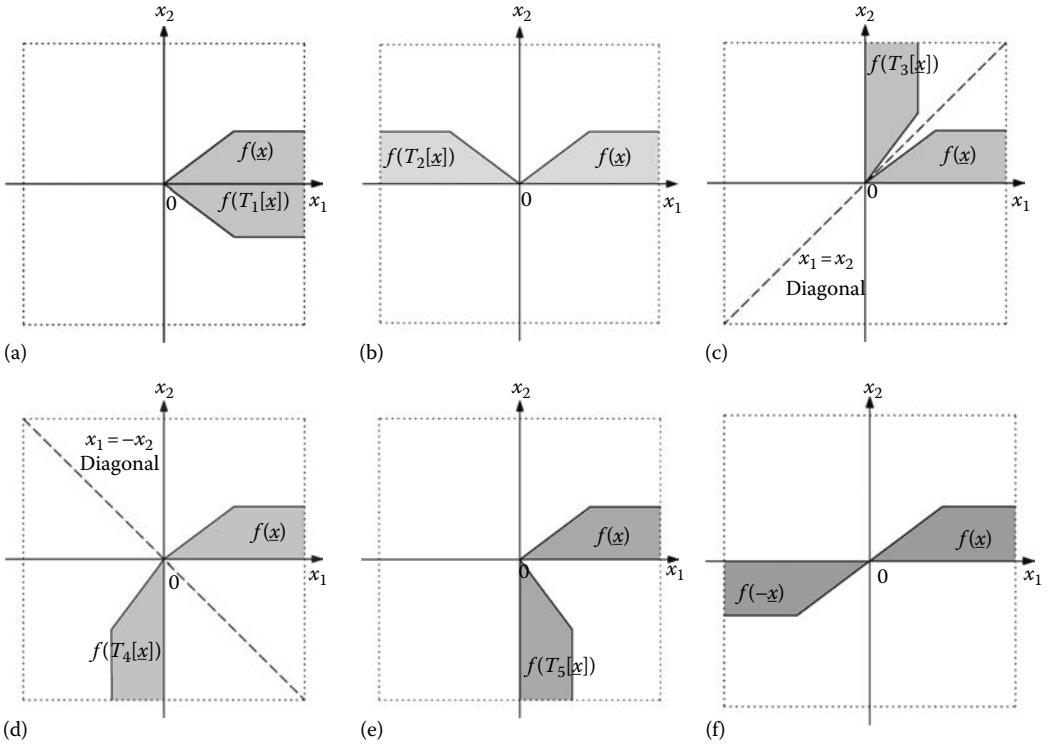


FIGURE 10.1 (a) x_1-Axis reflection symmetry, (b) x_2-axis reflection symmetry, (c) $x_1 = x_2$ diagonal reflection symmetry, (d) $x_1 = -x_2$ diagonal reflection symmetry, (e) 90° clockwise rotation symmetry, and (f) centrosymmetry (180° rotation).

In addition to these basic symmetries, more complex symmetries can be generated using a combination of different T-operations. For example, using a combination of T_1 and T_2 operations, we can obtain quadrantal symmetry. These compound symmetries are listed in Table 10.3. (Once again, for illustration, we show only the symmetries resulting from the ψ_I operation.)

We can make some important observations on the compound symmetries in Table 10.3:

1. Quadrantal symmetry is a combination of x_1-axis reflection, x_2-axis reflection, and centrosymmetries. The presence of any two of the symmetries implies the existence of the third, i.e., $T_1 \cdot T_2 \Rightarrow -\underline{I}$, $-\underline{I} \cdot T_1 \Rightarrow T_2$, and $-\underline{I} \cdot T_2 \Rightarrow T_1$. So only two of the three symmetries are needed to ensure quadrantal symmetry. In terms of cycles, quadrantal symmetry is a double 2-cyclic symmetry as $(T_1 \cdot T_2) \cdot (T_1 \cdot T_2) = \underline{I}$.

TABLE 10.3 Definitions of Compound T–ψ_I Symmetries

Type of Symmetry	Conditions
Quadrantal symmetry	$f(\underline{x}) = f(T_1[\underline{x}]) = f(T_1T_2[\underline{x}]) = f(T_2[\underline{x}])$
Diagonal symmetry	$f(\underline{x}) = f(T_3[\underline{x}]) = f(T_3T_4[\underline{x}]) = f(T_4[\underline{x}])$
Fourfold (90°) rotational symmetry	$f(\underline{x}) = f(T_5[\underline{x}]) = f(T_5^2[\underline{x}]) = f(T_5^3[\underline{x}])$
Octagonal symmetry	$f(\underline{x}) = f(T_1[\underline{x}]) = f(T_2[\underline{x}])$ $= f(T_3[\underline{x}]) = f(T_4[\underline{x}])$ $= f(T_5[\underline{x}]) = f(T_5^2[\underline{x}]) = f(T_5^3[\underline{x}])$

2. Diagonal symmetry is a combination of $x_1 = x_2$ diagonal, $x_1 = -x_2$ diagonal, and centrosymmetries. The presence of any two of the three symmetries implies the existence of the third and is enough to ensure diagonal symmetry. Diagonal symmetry is a double 2-cyclic symmetry.
3. Fourfold (90°) rotational symmetry is a 4-cyclic symmetry. It is formed by four repeated applications of T_5.
4. Octagonal symmetry is a combination of quadrantal, diagonal, and fourfold rotational symmetries. The presence of any two of the three symmetries implies the existence of the third, and is sufficient to guarantee octagonal symmetry. Octagonal symmetry can be classified as a triple 2-cyclic symmetry, or a combination of 4-cyclic and 2-cyclic symmetries.

To aid in understanding, Figure 10.2a through d shows the graphical interpretation of the quadrantal, diagonal, rotational, and octagonal symmetries. Once again, the values of the function in the shaded regions are related to one another in the manner specified by the ψ operation (i.e., identity, negative, conjugate, or negative conjugate). From the figures, one can also see that there are four regions of

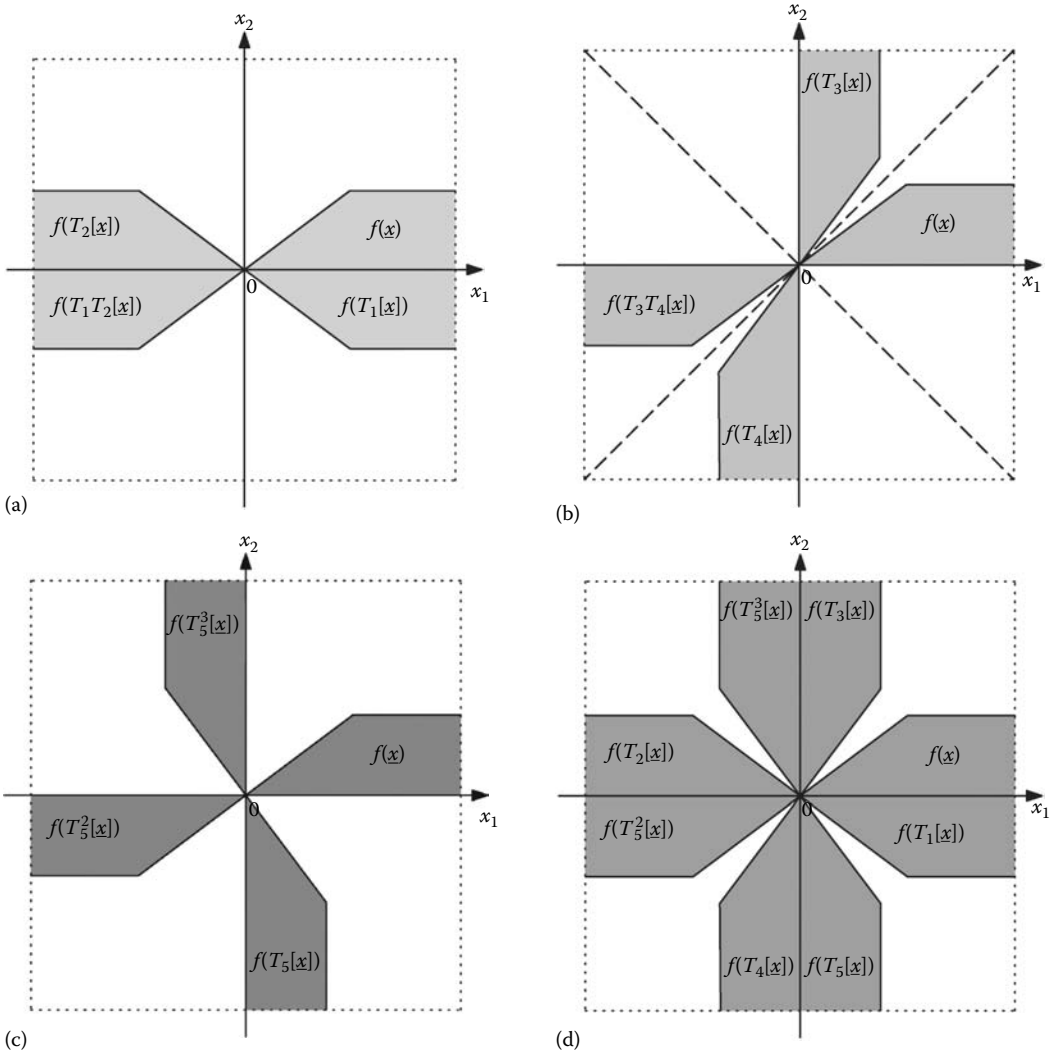


FIGURE 10.2 (a) Quadrantal symmetry, (b) diagonal symmetry, (c) fourfold rotational symmetry, and (d) octagonal symmetry.

symmetry in the X-plane for quadrantal, diagonal, and fourfold rotational symmetries, and eight regions for octagonal symmetry.

10.3 Two-Dimensional Fourier Transform Pairs with Symmetry

Two-dimensional Fourier transform is a very important tool in the analysis and design of 2-D linear systems. Since the transform uniquely relates the impulse or unit sample response of a linear system with its frequency response, symmetry present in one response may be expected to induce some form of symmetry in the other response. The existence of such symmetries can then be utilized to simplify the analysis and design of these systems. In this section, we present, for both continuous and discrete-time signals, the type of symmetry induced in one function (Fourier transform or inverse Fourier transform) as a result of a particular symmetry in the other function.

10.3.1 Continuous-Time Continuous-Frequency Case

Let $h(\underline{l}) = h(l_1, l_2)$ be a 2-D signal, and $H(\underline{\omega}) = H(\omega_1, \omega_2)$ be the 2-D Fourier transform of $h(\underline{l})$. If $h(\underline{l})$ is the impulse response of a stable continuous-time system, then its Fourier transform $H(\underline{\omega})$ always exists and the Fourier transform pair are given as follows:

$$H(\underline{\omega}) = \int_{\underline{l} \in L} h(\underline{l}) \cdot e^{-j\underline{\omega}^t \underline{l}} \cdot d\underline{l} \tag{10.4}$$

and

$$h(\underline{l}) = \frac{1}{(2\pi)^2} \int_{\underline{\omega} \in W} H(\underline{\omega}) \cdot e^{j\underline{l}^t \underline{\omega}} \cdot d\underline{\omega} \tag{10.5}$$

where

$d\underline{l} \triangleq dl_1 \cdot dl_2$

$d\underline{\omega} \triangleq d\omega_1 \cdot d\omega_2$

The following theorem describes the nature of the symmetry that will be present in $h(\underline{l})$, given a symmetry in $H(\underline{\omega})$, and vice versa [29].

THEOREM 10.1

Let $h(\underline{l})$ and $H(\underline{\omega})$ form a 2-D Fourier transform pair. Then $H(\underline{\omega})$ possesses a $T - \psi$ symmetry with parameters $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)$, $|\underline{A}| = \pm 1$, if and only if $h(\underline{l})$ possesses a $T - \psi$ symmetry with parameters $(\delta \cdot (\underline{A}^{-1})^t, \delta\underline{\beta}, \delta, -\delta\underline{b}, \underline{b}^t\,\underline{\beta} + \phi)$ where $\underline{A}$, $\underline{b}$, δ, $\underline{\beta}$, ϕ are as defined in Section 10.2.

Observations

We can make the following observations based on Theorem 10.1:

1. If $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_\omega$ and $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_l$ are respectively the $\underline{\omega}$-domain and the $\underline{l}$-domain symmetry parameters, the corresponding $\underline{l}$ and $\underline{\omega}$ domain parameters are obtained by the following relations:

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{\omega}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, \delta\underline{\beta}, \delta, -\delta\underline{b}, \underline{b}^t\underline{\beta} + \phi)_{\underline{l}} \tag{10.6}$$

and

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{l}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, \ -\delta\underline{\beta}, \ \delta, \delta\underline{b}, \ \underline{b}^t\underline{\beta} + \phi)_{\underline{\omega}} \tag{10.7}$$

This illustrates the duality present in $\underline{\omega}$ and $\underline{l}$ domain symmetries.

2. Nature of symmetry transformation, such as rotation, reflection, etc., as identified by the $\underline{A}$-matrix remains the same in both the $\underline{\omega}$ and $\underline{l}$ domains.
3. Identical symmetries resu1t in both $\underline{\omega}$ and $\underline{l}$ domains if $\delta = 1$, $\underline{b} = 0$, and $\underline{\beta} = 0$.

10.3.2 Continuous-Time Discrete-Frequency Case (Fourier Series)

Let $h(\underline{l}) = h(l_1, l_2)$ be a 2-D continuous-time periodic signal with periods L_1 and L_2, respectively in l_1 and l_2 directions. Let $H[\underline{k}] = H[k_1, k_2]$ be the 2-D Fourier series coefficients of $h(l_1, l_2)$. The relations between $h(\underline{l})$ and $H[\underline{k}]$ are given by

$$H[\underline{k}] = \frac{1}{L_1 \cdot L_2} \cdot \int_{l_1=0}^{L_1} \int_{l_2=0}^{L_2} h(\underline{l}) \cdot e^{-j \cdot \underline{k}^t \cdot \underline{V} \cdot \underline{l}} \cdot d\underline{l} \tag{10.8}$$

and

$$h(\underline{l}) = \sum_{\underline{k}} H[\underline{k}] \cdot e^{j \cdot \underline{k}^t \cdot \underline{V} \cdot \underline{l}} \tag{10.9}$$

where

$$\underline{V} = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix} = \begin{bmatrix} \frac{2 \cdot \pi}{L_1} & 0 \\ 0 & \frac{2 \cdot \pi}{L_2} \end{bmatrix}$$

It may be verified that $h(\underline{l})$ is a doubly periodic function indicating the existence of displacement symmetries. In the following, we will consider the general $T - \psi$ symmetry relations.

THEOREM 10.2

Let $H[\underline{k}]$ be the Fourier series representation of a periodic function $h(\underline{l})$. Then $h(\underline{l})$ possesses a $T - \psi$ symmetry with parameters $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)$, $|\underline{A}| = \pm 1$, if and only if $H[\underline{k}]$ possesses a $T - \psi$ symmetry with parameters $(\delta \cdot (\underline{A}^{-1})^t, -\delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, \delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t \underline{\beta} + \phi)$ where $\underline{A}$, $\underline{b}$, δ, $\underline{\beta}$, ϕ are as defined in Section 10.2. It should be noted that $\underline{b}$ and $-\delta \cdot \underline{V}^{-1} \cdot \underline{\beta}$ are integer vectors.

Since the observations made on the continuous domain case are also applicable here, the relations between $\underline{k}$ and $\underline{l}$ domain parameters are given by

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{l}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, -\delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, \delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t \underline{\beta} + \phi)_{\underline{k}} \tag{10.10}$$

and

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{k}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, \delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, -\delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t \underline{\beta} + \phi)_{\underline{l}} \tag{10.11}$$

10.3.3 Discrete-Time Continuous-Frequency Case

Let $h(\underline{n}) = h(n_1, n_2)$, where n_1, n_2 are integers, be a 2-D discrete-domain signal and $H(\underline{\theta}) = H(e^{-j\theta_1}, e^{-j\theta_2})$ be the 2-D Fourier transform of $h(\underline{n})$. The Fourier transform pair connecting $h(\underline{n})$ and $H(\underline{\theta})$ are given by

$$H(\underline{\theta}) = \sum_{\underline{n} \in N^2} h(\underline{n}) \cdot e^{-j\underline{\theta}^t \underline{n}} \tag{10.12}$$

and

$$h(\underline{n}) = \frac{1}{(2\pi)^2} \int_{\underline{\theta} \in \theta_p} H(\underline{\theta}) \cdot e^{j\underline{n}^t \underline{\theta}} \cdot d\underline{\theta} \tag{10.13}$$

where

$\theta_p = \{\underline{\theta} | -\pi \le \theta_1 \le \pi, \ -\pi \le \theta_2 \le \pi\}$

$d\underline{\theta} = d\theta_1 \cdot d\theta_2$

It may easily be verified from Equation 10.12 that $H(\underline{\theta})$ is a periodic function of $\underline{\theta}$ with a period of 2π in the θ_1 and θ_2 directions. Because of the periodic nature of $H(\underline{\theta})$, displacement symmetries in θ_1 and θ_2 directions are always present. This is due to the discrete nature of $h(\underline{n})$. We will consider the effects of the remaining symmetries that will be present in $h(\underline{n})$, given a symmetry in $H(\underline{\theta})$ and vice versa, in the following theorem.

THEOREM 10.3

Let $h(\underline{n})$ and $H(\underline{\theta})$ be a 2-D Fourier transform pair. Then $H(\underline{\theta})$ possesses a $T - \psi$ symmetry with parameters $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)$, $|\underline{A}| = \pm 1$, if and only if $h(\underline{n})$ possesses a $T - \psi$ symmetry with parameters $(\delta \cdot (\underline{A}^{-1})^t, \delta\underline{\beta}, \delta, -\delta\underline{b}, \underline{b}^t \underline{\beta} + \phi)$ where $\underline{A}$, $\underline{b}$, δ, $\underline{\beta}$, ϕ are as defined in Section 10.2.

Because of the similarity between Theorem 10.3 and Theorem 10.1, the observations made in the continuous domain case are also applicable here. Consequently, the relations between $\underline{\theta}$ and $\underline{n}$ symmetry parameters are given by

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{\theta}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, \delta\underline{\beta}, \delta, -\delta\underline{b}, \underline{b}^t\underline{\beta} + \phi)_{\underline{n}} \tag{10.14}$$

and

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{n}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, -\delta\underline{\beta}, \delta, \delta\underline{b}, \underline{b}^t\underline{\beta} + \phi)_{\underline{\theta}} \tag{10.15}$$

In the application of this theorem, it should be noted that $h(\underline{n})$ is a function of discrete-domain variable $\underline{n}$ and $h(\underline{n}) = \underline{0}$ if $\underline{n}$ is not an integer vector.

10.3.4 Discrete-Time Discrete-Frequency Case (Discrete Fourier Transform)

Let $h[\underline{n}] = h[n_1, n_2]$ be a 2-D discrete-domain signal defined over $0 \le n_1 \le (N_1 - 1)$ and $0 \le n_2 \le (N_2 - 1)$. Then its (N_1, N_2) length 2-D discrete Fourier transform (DFT) is given by

$$H[\underline{k}] = H[k_1, k_2] = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} h[\underline{n}] \cdot e^{-j \cdot \underline{k}^t \cdot \underline{V} \cdot \underline{n}}, \quad 0 \le k_1 \le (N_1 - 1), \quad 0 \le k_2 \le (N_2 - 1) \tag{10.16}$$

where

$$\underline{V} = \begin{bmatrix} \frac{2\cdot\pi}{N_1} & 0 \\ 0 & \frac{2\cdot\pi}{N_2} \end{bmatrix}$$

The inverse DFT is given by

$$h[\underline{n}] = h[n_1, n_2] = \frac{1}{N_1 \cdot N_2} \cdot \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} H[\underline{k}] \cdot e^{j\cdot \underline{k}^t \cdot \underline{V} \cdot \underline{n}} \tag{10.17}$$

It may be noted that even though $h[n_1, n_2]$ and $H[k_1, k_2]$ are defined in the rectangle $[0, N_1 - 1] \times [0, N_2 - 1]$, they satisfy the doubly periodic relations:

$$h[n_1 + r_1 \cdot N_1, n_2 + r_2 \cdot N_2] = h[n_1, n_2] \quad \text{for any integer } r_1 \text{ and } r_2$$

and similarly

$$H[k_1 + r_1 \cdot N_1, k_2 + r_2 \cdot N_2] = H[k_1, k_2] \quad \text{for any integer } r_1 \text{ and } r_2$$

The symmetry relations between h and H are given in terms of $T - \psi$ parameters in the following theorem.

THEOREM 10.4

Let $H[\underline{k}]$ be the DFT of $h[\underline{n}]$. Then $h[\underline{n}]$ possesses a $T - \psi$ symmetry with parameters $(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)$ if and only if $H[\underline{k}]$ possesses a $T - \psi$ symmetry with parameters $(\delta \cdot (\underline{A}^{-1})^t, -\delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, \delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t\ \underline{\beta} + \phi)$ where $\underline{A}$, $\underline{b}$, δ, $\underline{\beta}$, ϕ are as defined in Section 10.2.

As in the previous cases, the symmetry parameters of $\underline{n}$ and $\underline{k}$ domains can be written as

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{n}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, -\delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, \delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t \underline{\beta} + \phi)_{\underline{k}} \tag{10.18}$$

and

$$(\underline{A}, \underline{b}, \delta, \underline{\beta}, \phi)_{\underline{k}} \Rightarrow (\delta \cdot (\underline{A}^{-1})^t, \delta \cdot \underline{V}^{-1} \cdot \underline{\beta}, \delta, -\delta \cdot \underline{V} \cdot \underline{b}, \underline{b}^t \underline{\beta} + \phi)_{\underline{n}} \tag{10.19}$$

In the application of this theorem it should be noted that $\underline{n}$ and $\underline{k}$ are integer vectors and $h[\underline{n}]$ and $H[\underline{k}]$ are periodic with periods N_1 and N_2 in the two dimensions.

10.4 Symmetry and 2-D Fast Fourier Transform

FFT is a frequently used operation on 2-D signals. For large size signals, it is also a very time-consuming operation. So, any technique that reduces this complexity is desirable. As noted earlier, presence of symmetry in a signal yields some constraints on the spectrum. Therefore, use of symmetry constraints can be expected to reduce the complexity of evaluation of 2-D Fourier transforms. In this section, we will see how symmetry constraints can be used to speed up the computation of the FFT. First we will consider 1-D signals and then we will discuss symmetry application to 2-D FFTs.

10.4.1 Symmetry-Based 1-D FFT

Let $x[n]$, $n = 0, 1, \ldots, (N-1)$ be an N-length 1-D complex signal and $X[k]$, $k = 0, 1, \ldots, (N-1)$, be the N-length DFT of $x[n]$. $X[k]$ is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{k \cdot n}, \quad k = 0, 1, \ldots, (N-1) \tag{10.20}$$

where $W_N = e^{-j\frac{2\pi}{N}}$.

The DFT requires approximately N^2 complex multiplication and N^2 complex additions. When N is a power of 2, this can be reduced to $\frac{N}{2} \log_2 N$ complex multiplications and N^2 additions using what is called the Cooley–Tukey FFT.

1-D signals may possess reflection symmetries, translation symmetries, and identity symmetries. Reflection symmetries themselves can be with respect to the point at $N/2$ or $(N-1)/2$. To illustrate the application of symmetries and the resultant reduction in the complexity, we use reflection symmetry about the point $(N-1)/2$. Application of other symmetries yields similar reductions in complexity.

If $x[n]$ possesses reflection symmetry about $n = (N-1)/2$, then $x[N-1-n] = x[n]$ for $n = 0, 1, \ldots, (N-1)$. In terms of $T - \psi$ parameters, this corresponds to $A = [-1]$, $b = [N-1]$, $\delta = 1$, $\beta = 0$, and $\phi = 0$. The corresponding symmetry parameters in DFT domain is given by

$$A_k = [-1], \quad b_k = [0], \quad \delta_k = 1, \quad \beta_k = N-1, \quad \text{and} \quad \phi_k = 0,$$

which yields the symmetry relation after employing the periodic property of $X[k]$,

$$X[k] = W_N^{-k} \cdot X[N-k], \quad k = 0, 1, \ldots, (N-1) \tag{10.21}$$

Therefore, only half of $X[k]$ and $x[n]$ are independent and once they are known, the other half can be determined using the symmetry relation (Equation 10.21).

Assuming N to be even, we next show how $X[k]$ can be determined with less number of operations than the FFT. Using even and odd sample decomposition we can write $X[k]$ as

$$\begin{aligned} X[k] &= \sum_{\substack{n=0 \\ n \text{ even}}}^{N-1} x[n] \cdot W_N^{k \cdot n} + \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x[n] \cdot W_N^{k \cdot n} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x[2r] \cdot W_N^{k \cdot 2r} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] \cdot W_N^{k \cdot (2r+1)} \\ &= S_0[k] + W_N^k \cdot S_1[k] \end{aligned} \tag{10.22}$$

where S_0 and S_1 are $N/2$ point DFTs. Now using the symmetry properties $x[N-1-n] = x[n]$, we can write $S_1[k]$ in terms of $S_0[k]$ as

$$S_1[k] = W_N^{-2k} \cdot S_0\left[\frac{N}{2} - k\right] \tag{10.23}$$

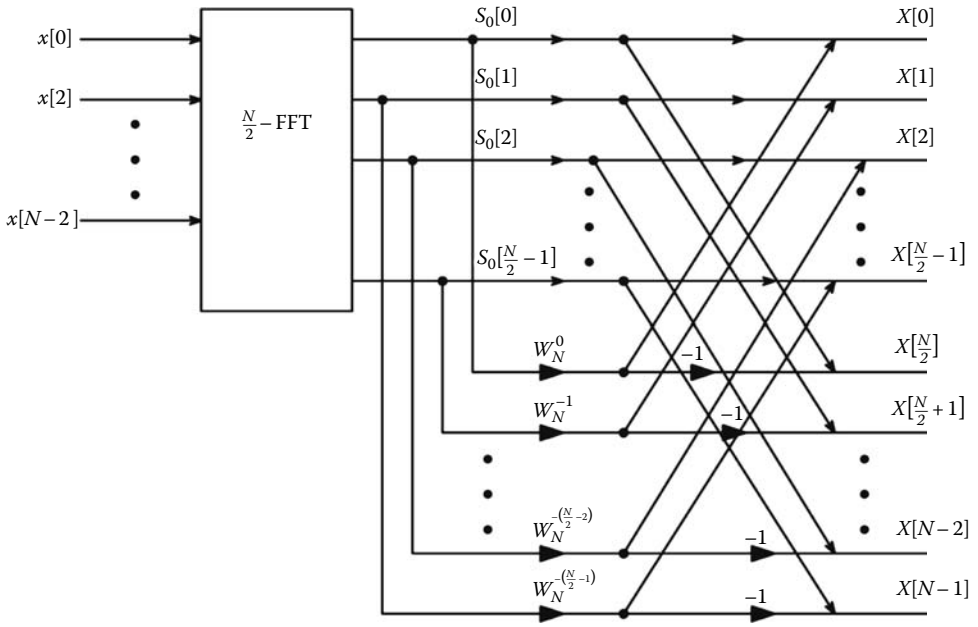


FIGURE 10.3 Symmetry-based FFT diagram.

So, for $k = 0, 1, 2, \ldots, \frac{N}{2} - 1$,

$$\begin{aligned} X[k] &= S_0[k] + W_N^k \cdot W_N^{-2k} \cdot S_0\left[\frac{N}{2} - k\right] \\ &= S_0[k] + W_N^{-k} \cdot S_0\left[\frac{N}{2} - k\right] \\ X\left[\frac{N}{2} + k\right] &= S_0[k] - W_N^{-k} \cdot S_0\left[\frac{N}{2} - k\right] \end{aligned} \tag{10.24}$$

Thus, $X[k]$ for $k = 0, 1, 2, \ldots, (N-1)$ can be determined using $N/2$ even indexed samples. This is illustrated in the flow diagram shown in Figure 10.3.

It is noted that we need only one $\frac{N}{2}$-point FFT instead of the normal two. As a result, the total number of complex multiplications needed is reduced to $\frac{N}{4}\log_2\left(\frac{N}{2}\right) + \frac{N}{2}$ instead of $\frac{N}{2}\log_2 N$ in the nonsymmetrical case. Thus use of symmetry results in approximately 50% reduction in the computational complexity.

10.4.2 2-D Symmetries

As in the 1-D case, symmetry can be utilized to reduce the computational complexity of 2-D DFTs. In the following, we illustrate the symmetry application for centrosymmetry and quadrantal symmetry cases.

As in the 1-D case, the reflection and rotation symmetries can be defined with respect to $\left(\frac{N}{2}, \frac{N}{2}\right)$ point or $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ point. Because of periodicity of h, the symmetries with respect to $\left(\frac{N}{2}, \frac{N}{2}\right)$ will also correspond to symmetries with respect to the origin (0,0) for the periodically extended signal. On the other hand, with the choice of $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ point for reflection or rotation, the various symmetries will occur for $h[\underline{n}]$ with respect to the center of the given data. In the following, $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ is chosen in the definition of various symmetries for $h[\underline{n}]$.

10.4.2.1 Centrosymmetry about $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$

An $N \times N$ array 2-D signal $x[m, n]$ is said to possess centrosymmetry about $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ if

$$x[N-1-m, N-1-n] = x[m, n] \quad \text{for } 0 \leq m, n \leq N-1$$

The corresponding symmetry relation for its 2-D DFT $X[k, l]$ is given by

$$X[N-k, N-l] = W_N^{k+l} \cdot X[k, l] \tag{10.25}$$

Now employing the even- and odd-indexed samples decomposition, $X[k, l]$ can be written as

$$X[k, l] = S_{00}[k, l] + W_N^k \cdot S_{10}[k, l] + W_N^l \cdot S_{01}[k, l] + W_N^{k+l} \cdot S_{11}[k, l] \tag{10.26}$$

where

$$S_{00}[k, l] = \sum_{\substack{p=0 \\ p \text{ even}}}^{\frac{N}{2}-1} \sum_{\substack{q=0 \\ q \text{ even}}}^{\frac{N}{2}-1} x[2p, 2q] \cdot W_{\frac{N}{2}}^{k \cdot p + l \cdot q}$$

$$S_{10}[k, l] = \sum_{p=0}^{\frac{N}{2}-1} \sum_{q=0}^{\frac{N}{2}-1} x[2p+1, 2q] \cdot W_{\frac{N}{2}}^{k \cdot p + l \cdot q}$$

$$S_{01}[k, l] = \sum_{p=0}^{\frac{N}{2}-1} \sum_{q=0}^{\frac{N}{2}-1} x[2p, 2q+1] \cdot W_{\frac{N}{2}}^{k \cdot p + l \cdot q}$$

$$S_{11}[k, l] = \sum_{p=0}^{\frac{N}{2}-1} \sum_{q=0}^{\frac{N}{2}-1} x[2p+1, 2q+1] \cdot W_{\frac{N}{2}}^{k \cdot p + l \cdot q}$$

Now applying the symmetry relation we can write

$$S_{01}[k, l] = W_N^{-2(k+l)} \cdot S_{10}\left[\frac{N}{2} - k, \frac{N}{2} - l\right]$$

$$\text{and} \quad S_{11}[k, l] = W_N^{-2(k+l)} \cdot S_{00}\left[\frac{N}{2} - k, \frac{N}{2} - l\right]$$

Then,

$$X[k, l] = S_{00}[k, l] + W_N^k \cdot S_{10}[k, l] + W_N^{-2k-l} \cdot S_{10}\left[\frac{N}{2} - k, \frac{N}{2} - l\right] + W_N^{-(k+l)} \cdot S_{00}\left[\frac{N}{2} - k, \frac{N}{2} - l\right] \tag{10.27}$$

From the above expression it is seen that when the centrosymmetry relation is employed, we need to compute only two $\left(\frac{N}{2} \times \frac{N}{2}\right)$ size 2-D FFTs instead of four $\left(\frac{N}{2} \times \frac{N}{2}\right)$ size 2-D FFTs. Thus there is a 50% reduction in the computational complexity.

10.4.2.2 Quadrantal Symmetry

A 2-D array $x[m, n]$ is said to possess quadrantal symmetry if the following condition is satisfied:

$$x[m,n] = x[N-1-m,n] = x[m,N-1-n] = x[N-1-m,N-1-n] \tag{10.28}$$

It may be noted that quadrantal symmetry is defined here with respect to the center of the array, $\left(\frac{N-1}{2},\frac{N-1}{2}\right)$. In the expression for $X[k, l]$ in Equation 10.26, $S_{10}[k, l]$, $S_{01}[k, l]$, and $S_{11}[k, l]$ can be expressed in terms of $S_{00}[k, l]$ as

$$\begin{aligned} S_{10}[k,l] &= \sum_{p=0}^{\frac{N}{2}-1}\sum_{q=0}^{\frac{N}{2}-1} x[2p+1,2q]\cdot W_{\frac{N}{2}}^{p\cdot k+q\cdot l} \\ &= W_N^{-2k}\cdot S_{00}\left[\frac{N}{2}-k,l\right] \\ S_{01}[k,l] &= W_N^{-2l}\cdot S_{00}\left[k,\frac{N}{2}-l\right] \\ \text{and}\quad S_{11}[k,l] &= W_N^{-2(k+l)}\cdot S_{00}\left[\frac{N}{2}-k,\frac{N}{2}-l\right]. \end{aligned}$$

Then $X[k, l]$ can be written in terms of S_{00} solely as

$$X[k,l] = S_{00}[k,l] + W_N^{-k}\cdot S_{00}\left[\frac{N}{2}-k,l\right] + W_N^{-l}\cdot S_{00}\left[k,\frac{N}{2}-l\right] + W_N^{-(k+l)}\cdot S_{00}\left[\frac{N}{2}-k,\frac{N}{2}-l\right] \tag{10.29}$$

It is then noted that evaluation of $X[k, l]$ requires the evaluation of four $\frac{N}{2}\times\frac{N}{2}$ size 2-D FFTs when $x[m, n]$ does not possess any symmetry whereas only one $\frac{N}{2}\times\frac{N}{2}$ size 2-D FFT is required when quadrantal symmetry is present in $x[m, n]$. In other words, utilization of the quadrantal symmetry in $x[m, n]$ reduces the computational complexity by approximately 75%. It may also be noted that quadrantal symmetry in $x[m,n]$ as defined in Equation 10.28 results in a form of quadrantal symmetry in $X[k, l]$ as

$$X[k,l] = W_N^{-k}\cdot X[N-k,l] = W_N^{-l}\cdot X[k,N-l] = W_N^{-(k+l)}\cdot X[N-k,N-l] \tag{10.30}$$

10.4.3 Symmetrical Decomposition for Data without Symmetry

In many situations, signals do not possess any symmetries. In those cases, symmetry results cannot be applied if we process the signals as they are. One way of dealing with this situation is to decompose the signal into signals possessing symmetries and antisymmetries [20,33]. For example, $x[m, n]$ can be decomposed as

$$x[m,n] = x_{00}[m,n] + x_{10}[m,n] + x_{01}[m,n] + x_{11}[m,n] \tag{10.31}$$

where

x_{00} possesses quadrantal symmetry
x_{10} possesses quadrantal antisymmetry of type 1 (antisymmetry w.r.t. m and symmetry w.r.t. n)
x_{01} possesses quadrantal antisymmetry of type 2 (symmetry w.r.t. m and antisymmetry w.r.t. n)
x_{11} possesses quadrantal antisymmetry of type 3 (antisymmetry w.r.t. both m and n)

Then each component can be processed using appropriate symmetry properties. While this method may not reduce the overall complexity, it will facilitate parallel processing of major computations.

10.5 Symmetry in 2-D Magnitude Response

In the design of 2-D filters, the design specifications are usually given in terms of the magnitude spectrum which possesses certain symmetries, while the phase characteristic is either not known or is not important. In such cases, it is desirable to know the types of transfer functions that can support the specified symmetry in the magnitude response. Recall that the ψ operation only affects the phase of the frequency response and not the magnitude. Hence, when dealing with magnitude symmetry, we assume $\psi = \psi_I$ and call our $T-\psi$ symmetry as simply T-symmetry. Now, in order for the magnitude response of a transfer function to possess a particularly symmetry, both the numerator and denominator polynomials have to possess the symmetry individually. In other words, when studying the symmetry constraints on the transfer function, one need only focus on the polynomial symmetry constraints on the numerator and denominator. In this section, we discuss the frequency response of the continuous-time and discrete-time 2-D polynomials and present the symmetry that might be present in the magnitude responses.

10.5.1 Continuous-Time Frequency Response

Let $P(s_1, s_2)$ be a 2-D s-domain polynomial with complex coefficients. Its frequency response is given by $P(j\omega_1, j\omega_2)$ where ω_1 and ω_2 are the radian frequencies. This is equivalent to evaluating the polynomial on the imaginary axes of the (s_1, s_2) biplane as shown in Figure 10.4. The magnitude-squared function of the frequency response is then given by

$$\begin{aligned} F(\omega_1, \omega_2) &= P(j\omega_1, j\omega_2) \cdot P^{\star}(-j\omega_1, -j\omega_2) \\ &= P(s_1, s_2) \cdot P^{\star}(-s_1, -s_2)\big|_{s_i = j\omega_i}, \quad i = 1, 2 \end{aligned} \tag{10.32}$$

where $P^{\star}$ is obtained by complex conjugating the coefficients of P.

10.5.2 Discrete-Time Frequency Response

For a 2-D z-domain polynomial $Q(z_1, z_2)$ with complex coefficients, its frequency response is given by $Q(e^{j\theta_1}, e^{j\theta_2})$. This is the same as evaluating the polynomial on the boundary of the unit circle in the (z_1, z_2) biplane as shown in Figure 10.5. Note that $\theta_i = \omega_i T$, where ω_1 and ω_2 are the same analog radian frequencies as above, and T is the sampling period. The magnitude-squared function of the frequency response is given by

$$\begin{aligned} F(\theta_1, \theta_2) &= Q\left(e^{j\theta_1}, e^{j\theta_2}\right) \cdot Q^{\star}\left(e^{-j\theta_1}, e^{-j\theta_2}\right) \\ &= Q(z_1, z_2) \cdot Q^{\star}\left(z_1^{-1}, z_2^{-1}\right)\Big|_{z_i = e^{j\theta_i}}, \quad i = 1, 2 \end{aligned} \tag{10.33}$$

where $Q^{\star}$ is obtained by complex conjugating the coefficients of Q.

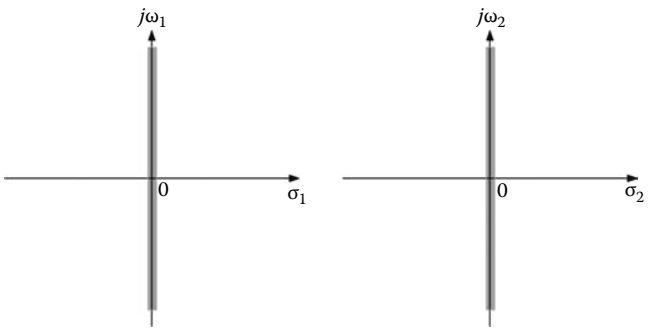


FIGURE 10.4 (s_1, s_2) biplane. (The frequency response is evaluated on the imaginary axes, $s_i = j\omega_i$.)

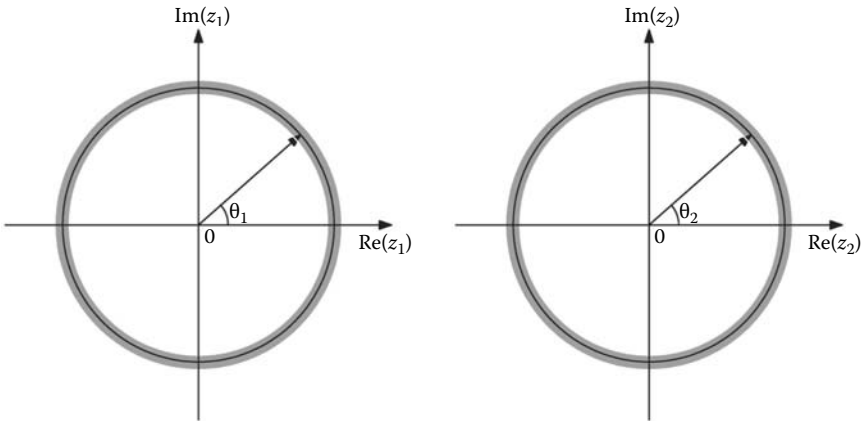


FIGURE 10.5 (z_1, z_2) biplane. (The frequency response is evaluated on the unit circles, $z_i = e^{j\theta_i}$.)

10.5.3 Symmetry in Magnitude Response

It can be seen in Equations 10.32 and 10.33 that if P and Q are real coefficient polynomials, $F(\omega_1, \omega_2) = F(-\omega_1, -\omega_2)$ and $F(\theta_1, \theta_2) = F(-\theta_1, -\theta_2)$. Therefore, a real polynomial, analog or digital, always possesses centrosymmetry. Because of this, only the compound symmetries in Table 10.3 need to be considered. We summarize in Table 10.4 these symmetry conditions for the magnitude-squared functions in both continuous and discrete domains. These symmetries are the same as those shown in Figure 10.2a–d, with the appropriate change in function and variable names.

TABLE 10.4 Symmetries Conditions for Continuous and Discrete Domain Magnitude-Squared Functions

Type of Symmetry	Condition on $F(\omega_1, \omega_2)$, $\forall$ (ω_1, ω_2)	Condition on $F(\theta_1, \theta_2)$, $\forall$ (θ_1, θ_2)
Quadrantal	$F(\omega_1, \omega_2) = F(-\omega_1, \omega_2)$ $= F(\omega_1, -\omega_2)$ $= F(-\omega_1, -\omega_2)$	$F(\theta_1, \theta_2) = F(-\theta_1, \theta_2)$ $= F(\theta_1, -\theta_2)$ $= F(-\theta_1, -\theta_2)$
Diagonal	$F(\omega_1, \omega_2) = F(\omega_2, \omega_1)$ $= F(-\omega_1, -\omega_2)$ $= F(-\omega_2, -\omega_1)$	$F(\theta_1, \theta_2) = F(\theta_2, \theta_1)$ $= F(-\theta_1, -\theta_2)$ $= F(-\theta_2, -\theta_1)$
Fourfold (90°) Rotational	$F(\omega_1, \omega_2) = F(-\omega_2, \omega_1)$ $= F(-\omega_1, -\omega_2)$ $= F(\omega_2, -\omega_1)$	$F(\theta_1, \theta_2) = F(-\theta_2, \theta_1)$ $= F(-\theta_1, -\theta_2)$ $= F(\theta_2, -\theta_1)$
Octagonal	$F(\omega_1, \omega_2) = F(\omega_2, \omega_1)$ $= F(-\omega_2, \omega_1)$ $= F(-\omega_1, \omega_2)$ $= F(-\omega_1, -\omega_2)$ $= F(-\omega_2, -\omega_1)$ $= F(\omega_2, -\omega_1)$ $= F(\omega_1, -\omega_2)$	$F(\theta_1, \theta_2) = F(\theta_2, \theta_1)$ $= F(-\theta_2, \theta_1)$ $= F(-\theta_1, \theta_2)$ $= F(-\theta_1, -\theta_2)$ $= F(-\theta_2, -\theta_1)$ $= F(\theta_2, -\theta_1)$ $= F(\theta_1, -\theta_2)$

10.6 Polynomial Operations and Definitions

In this section, we present the polynomial operations and definitions that will be used later for deriving the polynomial factors with symmetry.

Definition 10.2: Paraconjugate operation

The paraconjugate of an analog polynomial $P(s_1, s_2)$ is defined as

$$P_*(s_1, s_2) = P^*(-s_1, -s_2) \tag{10.34}$$

The subscript "*" indicates the paraconjugate of a polynomial, whereas the superscript "*" indicates the complex conjugation of coefficients alone.

Then in terms of the paraconjugate of a polynomial, the magnitude-squared function of $P(s_1, s_2)$ can be expressed as

$$F(\omega_1, \omega_2) = P(s_1, s_2) \cdot P_*(s_1, s_2)\big|_{s_i = j\omega_i}, \quad i = 1, 2 \tag{10.35}$$

The paraconjugate operation can also be performed w.r.t. a single variable, s_1 or s_2:

$$P_{*_{s_1}}(s_1, s_2) = P^*(-s_1, s_2) \tag{10.36}$$

$$P_{*_{s_2}}(s_1, s_2) = P^*(s_1, -s_2) \tag{10.37}$$

The paraconjugate of a discrete-time polynomial $Q(z_1, z_2)$ is called the inverse polynomial, and is defined as

$$Q_*(z_1, z_2) = Q^*(z_1^{-1}, z_2^{-1}) \tag{10.38}$$

It may be noted that $Q(z_1, z_2)$ is considered to be a pseudo polynomial which may have negative powers also.

The inverse operation for discrete-time polynomials can also be performed w.r.t. a single variable, z_1 or z_2.

$$Q_{*_{z_1}}(z_1, z_2) = Q^*(z_1^{-1}, z_2) \tag{10.39}$$

$$Q_{*_{z_2}}(z_1, z_2) = Q^*(z_1, z_2^{-1}) \tag{10.40}$$

Definition 10.3: Self-paraconjugate polynomial

$P(s_1, s_2)$ is said to be self-paraconjugate or para-even if

$$P_*(s_1, s_2) = P(s_1, s_2) \tag{10.41}$$

Henceforth, $P_e(s_1, s_2)$ will denote a para-even polynomial.

A polynomial can also be para-even w.r.t. a single variable s_i:

$$P_{*_{s_i}}(s_1, s_2) = P(s_1, s_2), \quad i = 1, 2 \tag{10.42}$$

An example of a polynomial that is para-even w.r.t. s_1 is $P_{e,s_1}(s_1, s_2) = s_1^2 + s_2 + 1$.

Definition 10.4: Anti-self-paraconjugate polynomial

$P(s_1, s_2)$ is said to be anti-self-paraconjugate or para-odd if

$$P*(s_1, s_2) = -P(s_1, s_2) \tag{10.43}$$

Henceforth, $P_o(s_1, s_2)$ will denote a para-odd polynomial.

A polynomial can also be para-odd w.r.t. a single variable s_i:

$$P_{*_{s_i}}(s_1, s_2) = -P(s_1, s_2), \quad i = 1, 2 \tag{10.44}$$

Para-even and para-odd in (s_1, s_2) correspond to self-inverse and anti-self-inverse in (z_1, z_2) respectively.

A discrete-time polynomial $Q(z_1, z_2)$ is self-inverse if

$$Q*(z_1, z_2) = Q(z_1, z_2) \tag{10.45}$$

$Q(z_1, z_2)$ is anti-self-inverse if

$$Q*(z_1, z_2) = -Q(z_1, z_2) \tag{10.46}$$

The discrete-time polynomial can also be self-inverse or anti-self-inverse w.r.t. a single variable, z_1 or z_2.

Property 1: Unique decomposition of polynomials

Any analog polynomial can be expressed as the sum of a para-even and a para-odd polynomial. That is,

$$P(s_1, s_2) = P_e(s_1, s_2) + P_o(s_1, s_2) \tag{10.47}$$

From the above, $P_e(s_1, s_2)$ and $P_o(s_1, s_2)$ are given by

$$P_e(s_1, s_2) = \frac{P(s_1, s_2) + P*(s_1, s_2)}{2} \tag{10.48}$$

$$P_o(s_1, s_2) = \frac{P(s_1, s_2) - P*(s_1, s_2)}{2} \tag{10.49}$$

The same holds when we deal with the operation w.r.t. a single variable s_i:

$$P(s_1,s_2) = P_{e,s_i}(s_1,s_2) + P_{o,s_i}(s_1, s_2) \tag{10.50}$$

where

$$P_{e,s_i}(s_1,s_2)=\frac{P(s_1,s_2)+P_{\star_{s_i}}(s_1,s_2)}{2} \tag{10.51}$$

$$P_{o,s_i}(s_1,s_2)=\frac{P(s_1,s_2)-P_{\star_{s_i}}(s_1,s_2)}{2} \tag{10.52}$$

In Equations 10.51 and 10.52 above, if $i=1$, then we have a polynomial restricted in s_1 and general in s_2. In such a case, we can further decompose into para-even and para-odd w.r.t. s_2 variable. In other words, $P(s_1, s_2)$ can be expressed as

$$P(s_1,s_2)=P_{e,s_1;e,s_2}(s_1,s_2)+P_{o,s_1;o,s_2}(s_1,s_2)+P_{o,s_1;e,s_2}(s_1,s_2)+P_{e,s_1;o,s_2}(s_1,s_2) \tag{10.53}$$

More specifically,

$$P(s_1,s_2)=P_1\left(s_1^2,s_2^2\right)+s_1s_2P_2\left(s_1^2,s_2^2\right)+s_1P_3\left(s_1^2,s_2^2\right)+s_2P_4\left(s_1^2,s_2^2\right) \tag{10.54}$$

The above (s_1, s_2) decomposition is the most general w.r.t. paraconjugate operation. It compares as follows to the standard decomposition in discrete-time (z_1, z_2):

$$Q(z_1,z_2)=Q_1(x_1,x_2)+y_1y_2Q_2(x_1,x_2)+y_1Q_3(x_1,x_2)+y_2Q_4(x_1,x_2) \tag{10.55}$$

where

$$\begin{aligned}x_i&=z_i+z_i^{-1}\\ y_i&=z_i-z_i^{-1},\quad i=1,2\end{aligned}$$

10.7 Spectral Forms for Magnitude-Squared Function

In this section, we present the spectral forms of the magnitude-squared function for the various symmetries.

Recall that the magnitude-squared function, $F(\omega_1, \omega_2)$, of a polynomial $P(s_1, s_2)$ is defined as

$$F(\omega_1,\omega_2)=P(s_1,s_2)\cdot P^*(-s_1,-s_2)|_{s_i=j\omega_i},\quad i=1,\,2 \tag{10.56}$$

Analogous to Equation 10.54, we can express the general magnitude-squared function as

$$F(\omega_1,\omega_2)=F_1\left(\omega_1^2,\omega_2^2\right)+\omega_1\omega_2F_2\left(\omega_1^2,\omega_2^2\right)+\omega_1F_3\left(\omega_1^2,\omega_2^2\right)+\omega_2F_4\left(\omega_1^2,\omega_2^2\right) \tag{10.57}$$

A polynomial with real coefficients always possesses centrosymmetry in its magnitude response. So,

$$F(\omega_1,\omega_2)=F(-\omega_1,-\omega_2)$$

Using Equation 10.57, this can be written as

$$\begin{aligned}&F_1\left(\omega_1^2,\omega_2^2\right)+\omega_1\omega_2F_2\left(\omega_1^2,\omega_2^2\right)+\omega_1F_3\left(\omega_1^2,\omega_2^2\right)+\omega_2F_4\left(\omega_1^2,\omega_2^2\right)\\ &\quad=F_1\left(\omega_1^2,\omega_2^2\right)+\omega_1\omega_2F_2\left(\omega_1^2,\omega_2^2\right)-\omega_1F_3\left(\omega_1^2,\omega_2^2\right)-\omega_2F_4\left(\omega_1^2,\omega_2^2\right)\\ \Leftrightarrow\quad&\omega_1F_3\left(\omega_1^2,\omega_2^2\right)+\omega_2F_4\left(\omega_1^2,\omega_2^2\right)=-\omega_1F_3\left(\omega_1^2,\omega_2^2\right)-\omega_2F_4\left(\omega_1^2,\omega_2^2\right)\end{aligned} \tag{10.58}$$

The above is only possible if $F_3(\omega_1^2, \omega_2^2)=0$ and $F_4(\omega_1^2, \omega_2^2)=0$.

Thus,

$$F(\omega_1,\omega_2) = F_1\left(\omega_1^2,\omega_2^2\right) + \omega_1\omega_2 F_2\left(\omega_1^2,\omega_2^2\right) \tag{10.59}$$

Using analytic continuation and Equation 10.56, this can be written as

$$\begin{aligned} P(s_1,s_2)\cdot P(-s_1,-s_2) &= F_1\left(s_1^2,s_2^2\right) + s_1 s_2 F_2\left(s_1^2,s_2^2\right) \\ &= \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j} + s_1 s_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} b_{ij}s_1^{2i}s_2^{2j} \end{aligned} \tag{10.60}$$

The above is the spectral form of the magnitude-squared function possessing centrosymmetry.

Now, for a real-coefficient $P(s_1, s_2)$ to possess quadrantal symmetry in its magnitude-squared response, we need

$$F(\omega_1,\omega_2) = F(\omega_1,-\omega_2)$$

Using Equation 10.59, this can be written as

$$F_1\left(\omega_1^2,\omega_2^2\right) + \omega_1\omega_2 F_2\left(\omega_1^2,\omega_2^2\right) = F_1\left(\omega_1^2,\omega_2^2\right) - \omega_1\omega_2 F_2\left(\omega_1^2,\omega_2^2\right) \tag{10.61}$$

The above is only possible if $F_2(\omega_1^2, \omega_2^2) = 0$. Thus,

$$F(\omega_1,\omega_2) = F_1\left(\omega_1^2,\omega_2^2\right), \tag{10.62}$$

TABLE 10.5 Spectral Forms of Magnitude-Squared Function for Various Symmetries

Analog	Digital[a]
Centrosymmetry	
$P(s_1,s_2)\cdot P(-s_1,-s_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j} + s_1 s_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} b_{ij}s_1^{2i}s_2^{2j}$	$Q(z_1,z_2)\cdot Q(z_1^{-1},z_2^{-1}) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} c_{ij}x_1^i x_2^j + y_1 y_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} d_{ij}x_1^i x_2^j$
Quadrantal symmetry	
$P(s_1,s_2)\cdot P(-s_1,-s_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j}$	$Q(z_1,z_2)\cdot Q(z_1^{-1},z_2^{-1}) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} c_{ij}x_1^i x_2^j$
Diagonal symmetry	
$P(s_1,s_2)\cdot P(-s_1,-s_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j} + s_1 s_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} b_{ij}s_1^{2i}s_2^{2j}$	$Q(z_1,z_2)\cdot Q(z_1^{-1},z_2^{-1}) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} c_{ij}x_1^i x_2^j + y_1 y_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} d_{ij}x_1^i x_2^j$
where $a_{ij} = a_{ji}$ and $b_{ij} = b_{ji}$	where $c_{ij} = c_{ji}$ and $d_{ij} = d_{ji}$
Fourfold (90°) rotational symmetry	
$P(s_1,s_2)\cdot P(-s_1,-s_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j} + s_1 s_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} b_{ij}s_1^{2i}s_2^{2j}$	$Q(z_1,z_2)\cdot Q(z_1^{-1},z_2^{-1}) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} c_{ij}x_1^i x_2^j + y_1 y_2 \sum_{i=0}^{m_2}\sum_{j=0}^{n_2} d_{ij}x_1^i x_2^j$
where $a_{ij} = a_{ji}$ and $b_{ij} = -b_{ji}$	where $c_{ij} = c_{ji}$ and $d_{ij} = -d_{ji}$
Octagonal symmetry	
$P(s_1,s_2)\cdot P(-s_1,-s_2) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} a_{ij}s_1^{2i}s_2^{2j}$	$Q(z_1,z_2)\cdot Q(z_1^{-1},z_2^{-1}) = \sum_{i=0}^{m_1}\sum_{j=0}^{n_1} c_{ij}x_1^i x_2^j$
where $a_{ij} = a_{ji}$	where $c_{ij} = c_{ji}$

[a] Note that $x_i = z_i + z_i^{-1}$ and $y_i = z_i - z_i^{-1}$, $i = 1, 2$.

and so the spectral form for the magnitude-squared function that possesses quadrantal symmetry is

$$\begin{aligned} P(s_1, s_2) \cdot P(-s_1, -s_2) &= F_1\left(s_1^2, s_2^2\right) \\ &= \sum_{i=0}^{m_1} \sum_{j=0}^{n_1} a_{ij} s_1^{2i} s_2^{2j} \end{aligned} \tag{10.63}$$

We can obtain the spectral forms for the other symmetries in a similar manner. They are listed in Table 10.5, together with the spectral forms for the z-domain functions.

10.8 Determination of 2-D Polynomials Possessing Various Symmetries

In this section, we present the polynomial factors that possess symmetry in their magnitude responses. We begin by showing the conditions that the s-domain polynomials must satisfy in order to possess the required symmetry. Then, the s-domain polynomial factors that satisfy those conditions are given. (The proof is provided for quadrantal symmetry). Finally, at the end of the section, the z-domain polynomial factors possessing symmetry are listed, alongside the analog polynomial factors.

10.8.1 Quadrantal Symmetry for Analog Polynomials

THEOREM 10.5

A polynomial $P(s_1, s_2)$ possesses quadrantal symmetry in its magnitude response if its factors either alone or jointly satisfy one of the following conditions:

$$1.\ P(s_1, s_2) = P(s_1, -s_2) \tag{10.64}$$

$$2.\ P(s_1, s_2) = -P(s_1, -s_2) \tag{10.65}$$

$$3.\ P(s_1, s_2) = P(-s_1, s_2) \tag{10.66}$$

$$4.\ P(s_1, s_2) = -P(-s_1, s_2) \tag{10.67}$$

$$5.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(s_1, -s_2) \tag{10.68}$$

$$6.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(-s_1, s_2) \tag{10.69}$$

Proof of Theorem 10.5: From Table 10.4, a magnitude-squared response possesses quadrantal symmetry if

$$F(\omega_1, \omega_2) = F(\omega_1, -\omega_2) \tag{10.70}$$

Applying Equation 10.32, this can be written as

$$P(j\omega_1, j\omega_2) \cdot P(-j\omega_1, -j\omega_2) = P(j\omega_1, -j\omega_2) \cdot P(-j\omega_1, j\omega_2)$$

Using analytic continuation, this becomes

$$P(s_1, s_2) \cdot P(-s_1, -s_2) = P(s_1, -s_2) \cdot P(-s_1, s_2) \tag{10.71}$$

If we assume $P(s_1, s_2)$ to be irreducible, the unique factorization property of 2-variable polynomials [34] states that $P(s_1, s_2)$ should satisfy one of the following two conditions:

$$\text{(i) } P(s_1, s_2) = k_1 \cdot P(s_1, -s_2) \text{ where } k_1 \text{ is a real constant} \tag{10.72}$$

$$\text{(ii) } P(s_1, s_2) = k_2 \cdot P(-s_1, s_2) \text{ where } k_2 \text{ is a real constant} \tag{10.73}$$

We first consider case (i) above. Substituting $s_2 = -s_2$ into Equation 10.72, we obtain

$$P(s_1, -s_2) = k_1 \cdot P(s_1, s_2)$$

Substituting this back into Equation 10.72, we get

$$P(s_1, s_2) = k_1^2 \cdot P(s_1, s_2) \tag{10.74}$$

So, $k_1^2 = 1$, i.e., $k_1 = +1$ or -1. Thus, Equation 10.72 becomes

$$P(s_1, s_2) = \pm 1 \cdot P(s_1, -s_2) \tag{10.75}$$

With that, we have the first two conditions of Theorem 10.5, Equations 10.64 and 10.65.

We now work on case (ii). Substituting $s_1 = -s_1$ into Equation 10.73, we have

$$P(-s_1, s_2) = k_2 \cdot P(s_1, s_2) \tag{10.76}$$

Substituting Equation 10.76 back into Equation 10.73, we get $k_2^2 = 1$, i.e., $k_2 = +1$ or -1. This gives us the next two conditions for Theorem 10.5, Equations 10.66 and 10.67.

So far, we have assumed $P(s_1, s_2)$ to be irreducible. If $P(s_1, s_2)$ is reducible, then it can be expressed as a product of irreducible factors:

$$P(s_1, s_2) = k \cdot \prod_{i=1}^{N} P_i(s_1, s_2) \tag{10.77}$$

Let $P_i(s_1, s_2)$ be one such irreducible factor in $P(s_1, s_2)$. Invoking the unique factorization property on Equation 10.71, we can see that $P_i(s_1, s_2)$ must be present in either $P(s_1, -s_2)$ or $P(-s_1, s_2)$ and thus must satisfy one of the following four conditions:

(a) $P_i(s_1, s_2) = k_1 \cdot P_i(s_1, -s_2)$
(b) $P_i(s_1, s_2) = k_2 \cdot P_j(s_1, -s_2)$ where $i \neq j$
(c) $P_i(s_1, s_2) = k_3 \cdot P_i(-s_1, s_2)$
(d) $P_i(s_1, s_2) = k_4 \cdot P_j(-s_1, s_2)$ where $i \neq j$

Cases (a) and (c) are identical to the ones for irreducible polynomials, so the previous results apply. We need only investigate cases (b) and (d) here.

For case (b),

$$P_i(s_1, s_2) = k_2 \cdot P_j(s_1, -s_2) \tag{10.78}$$

We solve for $P_j(s_1, s_2)$ by multiplying both sides by $1/k_2$ and substituting $s_2 = -s_2$:

$$P_j(s_1, s_2) = \frac{1}{k_2} \cdot P_i(s_1, -s_2) \tag{10.79}$$

$P_i(s_1, s_2) \cdot P_j(s_1, s_2)$ is a factor in $P(s_1, s_2)$. Using Equation 10.79, this can be expressed as

$$P_i(s_1, s_2) \cdot P_j(s_1, s_2) = P_i(s_1, s_2) \cdot \frac{1}{k_2} \cdot P_i(s_1, -s_2) \tag{10.80}$$

So, $P_i(s_1, s_2) \cdot P_i(s_1, -s_2)$ is a possible factor in $P(s_1, s_2)$. This proves Equation 10.68 in Theorem 10.5.

Now, we work on case (d),

$$P_i(s_1, s_2) = k_4 \cdot P_j(-s_1, s_2) \tag{10.81}$$

Solving for $P_j(s_1, s_2)$:

$$P_j(s_1, s_2) = \frac{1}{k_4} \cdot P_i(-s_1, s_2) \tag{10.82}$$

Now using Equation 10.82, the product of $P_i(s_1, s_2) \cdot P_j(s_1, s_2)$ can be expressed as

$$P_i(s_1, s_2) \cdot P_j(s_1, s_2) = P_i(s_1, s_2) \cdot \frac{1}{k_4} \cdot P_i(-s_1, s_2) \tag{10.83}$$

Therefore $P_i(s_1, s_2) \cdot P_i(-s_1, s_2)$ is a factor in $P(s_1, s_2)$ and this corresponds to Equation 10.69 in Theorem 10.5. This concludes the proof for Theorem 10.5.

Theorem 10.5 specifies the conditions each factor of a polynomial should satisfy to ensure quadrantal symmetry in the magnitude response of the overall polynomial. However, these conditions (except 5 and 6) do not show the way of generating the polynomial factor that possesses the symmetry. In the following, we will derive the forms of the polynomial factors that satisfy conditions 1 to 4 of Theorem 10.5.

Corollary of Condition 1

Using Equation 10.54, we can write Equation 10.64 as the following:

$$\begin{aligned} &P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \\ &\quad = P_1\left(s_1^2, s_2^2\right) - s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) - s_2 P_4\left(s_1^2, s_2^2\right) \end{aligned}$$

$$\Leftrightarrow \quad s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) = -s_1 s_2 P_2\left(s_1^2, s_2^2\right) - s_2 P_4\left(s_1^2, s_2^2\right)$$

The above is only possible if $P_2(s_1^2, s_2^2) = 0$ and $P_4(s_1^2, s_2^2) = 0$.

Therefore,

$$P(s_1, s_2) = P_1\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right)$$

This can be expressed as

$$P_5(s_1, s_2^2) \tag{10.84}$$

Corollary of Condition 2
Using Equation 10.54, we can write Equation 10.65 as the following:

$$\begin{aligned} &P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \\ &\quad = -P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) - s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \end{aligned}$$

$$\Leftrightarrow \quad P_1\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) = -P_1\left(s_1^2, s_2^2\right) - s_1 P_3\left(s_1^2, s_2^2\right)$$

The above is only possible if $P_1(s_1^2, s_2^2) = 0$ and $P_3(s_1^2, s_2^2) = 0$. So,

$$P(s_1, s_2) = s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right)$$

This can be expressed as

$$s_2 \cdot P_5(s_1, s_2^2) \tag{10.85}$$

Corollary of Condition 3
Using Equation 10.54, we can write Equation 10.66 as the following:

$$\begin{aligned} &P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \\ &\quad = P_1\left(s_1^2, s_2^2\right) - s_1 s_2 P_2\left(s_1^2, s_2^2\right) - s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \end{aligned}$$

$$\Leftrightarrow \quad s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) = -s_1 s_2 P_2\left(s_1^2, s_2^2\right) - s_1 P_3\left(s_1^2, s_2^2\right)$$

The above is only possible if $P_2(s_1^2, s_2^2) = 0$ and $P_3(s_1^2, s_2^2) = 0$.
Therefore,

$$P(s_1, s_2) = P_1\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right)$$

This can be expressed as

$$P_6(s_1^2, s_2) \tag{10.86}$$

Corollary of Condition 4
Using Equation 10.54, we can write Equation 10.67 as the following:

$$\begin{aligned} &P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) \\ &\quad = -P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right) - s_2 P_4\left(s_1^2, s_2^2\right) \end{aligned}$$

$$\Leftrightarrow \quad P_1\left(s_1^2, s_2^2\right) + s_2 P_4\left(s_1^2, s_2^2\right) = -P_1\left(s_1^2, s_2^2\right) - s_2 P_4\left(s_1^2, s_2^2\right)$$

The above is only possible if $P_1(s_1^2, s_2^2) = 0$ and $P_4(s_1^2, s_2^2) = 0$. So,

$$P(s_1, s_2) = s_1 s_2 P_2\left(s_1^2, s_2^2\right) + s_1 P_3\left(s_1^2, s_2^2\right)$$

This can be expressed as

$$s_1 \cdot P_6(s_1^2, s_2) \tag{10.87}$$

Reviewing all the above, we can see that the general polynomial form, $P_5\ (s_1,\ s_2^2) \cdot P_6\ (s_1^2,\ s_2)$, satisfies Equations 10.84 through 10.87.

Now, we can state the polynomials factors that satisfy conditions 1 through 6 of Theorem 10.5 in the following corollary.

Corollary of Theorem 10.5
The following polynomial factors possess quadrantal symmetry in their magnitude responses:

$$1.\ P(s_1, s_2^2) \tag{10.88}$$

$$2.\ P(s_1^2, s_2) \tag{10.89}$$

$$3.\ P(s_1, s_2) \cdot P(s_1, -s_2) \tag{10.90}$$

$$4.\ P(s_1, s_2) \cdot P(-s_1, s_2) \tag{10.91}$$

It is obvious that the product of any of the above polynomial factors satisfy the symmetry as well.

Examples

We now provide examples of (2,2) degree polynomials satisfying Equations 10.88 and 10.90.

1. $P(s_1, s_2^2) = a_0 + a_1 s_1 + a_2 s_1^2 + a_3 s_2^2 + a_4 s_1 s_2^2 + a_5 s_1^2 s_2^2.$
2. $P(s_1, s_2) \cdot P(s_1, -s_2)$

Let $P(s_1,\ s_2) = b_0 + b_1\ s_1 + b_2\ s_2 + b_3\ s_1\ s_2$
Then,

$$\begin{aligned} P(s_1, s_2) \cdot P(s_1, -s_2) &= (b_0 + b_1 s_1 + b_2 s_2 + b_3 s_1 s_2) \cdot (b_0 + b_1 s_1 - b_2 s_2 - b_3 s_1 s_2) \\ &= b_0^2 + 2b_0 b_1 s_1 + b_1^2 s_1^2 - b_2^2 s_2^2 - 2b_2 b_3 s_1 s_2^2 - b_3^2 s_1^2 s_2^2 \end{aligned}$$

Using the same derivation procedure outlined above for quadrantal symmetry, we can obtain the conditions and polynomial factors for the other symmetries.

10.8.2 Diagonal Symmetry for Analog Polynomials

THEOREM 10.6

A polynomial $P(s_1,\ s_2)$ possesses diagonal symmetry in its magnitude response if its factors either alone or jointly satisfy one of the following conditions:

$$1.\ P(s_1, s_2) = P(s_2, s_1) \tag{10.92}$$

$$2.\ P(s_1, s_2) = -P(s_2, s_1) \tag{10.93}$$

$$3.\ P(s_1, s_2) = P(-s_2, -s_1) \tag{10.94}$$

$$4.\ P(s_1, s_2) = -P(-s_2, -s_1) \tag{10.95}$$

$$5.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(s_2, s_1) \tag{10.96}$$

$$6.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(-s_2, -s_1) \tag{10.97}$$

Corollary of Theorem 10.6
The following polynomial factors possess diagonal symmetry in their magnitude responses:

$$1.\ P_1(s_1, s_2) \tag{10.98}$$

$$2.\ P_2\left(s_1^2, s_2^2\right) + s_1 s_2 P_3\left(s_1^2, s_2^2\right) + s_1 P_4\left(s_1^2, s_2^2\right) - s_2 P_4(s_2^2, s_1^2) \tag{10.99}$$

$$3.\ P(s_1, s_2) \cdot P(s_2, s_1) \tag{10.100}$$

$$4.\ P(s_1, s_2) \cdot P(-s_2, -s_1) \tag{10.101}$$

where

$$P_1(s_1, s_2) = P_1(s_2, s_1)$$

$$P_k\left(s_1^2, s_2^2\right) = P_k(s_2^2, s_1^2) \quad \text{for } k = 2, 3$$

10.8.3 Fourfold (90°) Rotational Symmetry for Analog Polynomials

---

THEOREM 10.7

A polynomial $P(s_1, s_2)$ possesses fourfold rotational symmetry in its magnitude response if its factors either alone or jointly satisfy one of the following conditions:

$$1.\ P(s_1, s_2) = P(-s_2, s_1) \tag{10.102}$$

$$2.\ P(s_1, s_2) = -P(-s_2, s_1) \tag{10.103}$$

$$3.\ P(s_1, s_2) = P(s_2, -s_1) \tag{10.104}$$

$$4.\ P(s_1, s_2) = -P(s_2, -s_1) \tag{10.105}$$

$$5.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(-s_2, s_1) \tag{10.106}$$

$$6.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(s_2, -s_1) \tag{10.107}$$

$$7.\ P(s_1, s_2) = P_1(s_1, s_2) \cdot P_1(-s_2, s_1) \cdot P_1(-s_1, -s_2) \cdot P_1(s_2, -s_1) \tag{10.108}$$

Corollary of Theorem 10.7
The following polynomial factors possess fourfold rotational symmetry in their magnitude responses:

$$1.\ P_1\left(s_1^2, s_2^2\right) + s_1 s_2 \cdot (s_1^2 - s_2^2) \cdot P_2\left(s_1^2, s_2^2\right) \tag{10.109}$$

$$2.\ (s_1^2 - s_2^2) P_1\left(s_1^2, s_2^2\right) + s_1 s_2 P_2\left(s_1^2, s_2^2\right) \tag{10.110}$$

$$3.\ P(s_1, s_2) \cdot P(-s_2, s_1) \tag{10.111}$$

$$4.\ P(s_1, s_2) \cdot P(s_2, -s_1) \tag{10.112}$$

$$5.\ P(s_1, s_2) \cdot P(-s_2, s_1) \cdot P(-s_1, -s_2) \cdot P(s_2, -s_1) \tag{10.113}$$

where $P_k(s_1^2, s_2^2) = P_k(s_2^2, s_1^2)$ for $k = 1, 2$.

10.8.4 Octagonal Symmetry for Analog Polynomials

THEOREM 10.8

The following polynomial factors possess octagonal symmetry in their magnitude responses:

$$1.\ (s_1^2 - s_2^2)^{\alpha} \cdot P_1\left(s_1^2, s_2^2\right), \quad \text{where } \alpha = 0 \text{ or } 1 \tag{10.114}$$

$$2.\ P(s_1^2, s_2) \cdot P(s_2^2, s_1) \tag{10.115}$$

$$3.\ P(s_1^2, s_2) \cdot P(s_2^2, -s_1) \tag{10.116}$$

$$4.\ P(s_1, s_2^2) \cdot P(-s_2, s_1^2) \tag{10.117}$$

$$5.\ P_2(s_1, s_2) \cdot P_2(-s_1, s_2) \tag{10.118}$$

$$6.\ P_2(s_1, s_2) \cdot P_2(s_1, -s_2) \tag{10.119}$$

where

$$P_1\left(s_1^2, s_2^2\right) = P_1(s_2^2, s_1^2)$$

$$P_2(s_1, s_2) = P_2(s_2, s_1)$$

10.8.5 Summary of Analog and Digital Polynomial Factors Possessing Symmetry

In Table 10.6, we state the digital polynomial factors that possess the various symmetries, alongside the analog polynomial factors presented earlier. The digital polynomial factors can be derived by assuming the general z-domain polynomial form in Equation 10.55 and following the same derivation steps as the analog case.

10.9 Symmetry and Stability

A sufficient condition for the stability of 2-D analog filters [35–37] is that their transfer functions do not have any poles in the region of the (s_1, s_2) biplane defined by $\mathrm{Re}(s_1) \geq 0$ and $\mathrm{Re}(s_2) \geq 0$, including infinite distant points. Similarly, a sufficient condition for the stability of 2-D digital filters is that their transfer functions do not have any poles in the region of the (z_1, z_2) biplane defined by $|z_1| \geq 1$ and $|z_2| \geq 1$. We will next consider the analog domain and the digital domain stability conditions under the presence of symmetry.

10.9.1 Stability of Analog Filters with Symmetric Magnitude Response

The transfer function of an analog 2-D filter can be represented as a rational function in s_1 and s_2 as

$$T(s_1, s_2) = \frac{P(s_1, s_2)}{Q(s_1, s_2)} \tag{10.120}$$

It can be shown that $T(j\omega_1, j\omega_2)$ possesses various magnitude symmetries if $P(j\omega_1, j\omega_2)$ and $Q(j\omega_1, j\omega_2)$ possess the same symmetries individually. Then for $T(s_1, s_2)$ to possess a symmetry and be stable, $Q(s_1, s_2)$ should satisfy the conditions for the symmetry as well as the conditions for stability. The type of $Q(s_1, s_2)$ that satisfies these two conditions for quadrantal symmetry is presented in the following theorem.

TABLE 10.6 Analog and Digital Polynomial Factors Possessing Symmetry

Analog s-Domain	Digital z-Domain[a]
Quadrantal symmetry	
1. $P(s_1, s_2^2)$	1. $Q(z_1, x_2)$
2. $P(s_1^2, s_2)$	2. $Q(x_1, z_2)$
3. $P(s_1, s_2)\cdot P(s_1, -s_2)$	3. $Q(z_1, z_2)\cdot Q(z_1, z_2^{-1})$
4. $P(s_1, s_2)\cdot P(-s_1, s_2)$	4. $Q(z_1, z_2)\cdot Q(z_1^{-1}, z_2)$
Diagonal symmetry	
1. $P_1(s_1, s_2)$	1. $Q_1(z_1, z_2)$
2. $P_2(s_1^2, s_2^2) + s_1 s_2 P_3(s_1^2, s_2^2) + s_1 P_4(s_1^2, s_2^2) - s_2 P_4(s_2^2, s_1^2)$	2. $Q_2(x_1, x_2) + y_1 y_2 Q_3(x_1, x_2) + y_1 Q_4(x_1, x_2) - y_2 Q_4(x_2, x_1)$
3. $P(s_1, s_2)\cdot P(s_2, s_1)$	3. $Q(z_1, z_2)\cdot Q(z_2, z_1)$
4. $P(s_1, s_2)\cdot P(-s_2, -s_1)$	4. $Q(z_1, z_2)\cdot Q(z_2^{-1}, z_1^{-1})$
where $P_1(s_1, s_2) = P_1(s_2, s_1)$ and $P_k(s_1^2, s_2^2) = P_k(s_2^2, s_1^2)$ for $k = 2, 3$.	where $Q_1(z_1, z_2) = Q_1(z_2, z_1)$ and $Q_k(x_1, x_2) = Q_k(x_2, x_1)$ for $k = 2, 3$.
Fourfold (90°) rotational symmetry	
1. $P_1(s_1^2, s_2^2) + s_1 s_2\cdot(s_1^2 - s_2^2)\cdot P_2(s_1^2, s_2^2)$	1. $Q_1(x_1, x_2) + y_1 y_2\cdot(x_1 - x_2)\cdot Q_2(x_1, x_2)$
2. $(s_1^2 - s_2^2) P_1(s_1^2, s_2^2) + s_1 s_2 P_2(s_1^2, s_2^2)$	2. $(x_1 - x_2) Q_1(x_1, x_2) + y_1 y_2 Q_2(x_1, x_2)$
3. $P(s_1, s_2)\cdot P(-s_2, s_1)$	3. $Q(z_1, z_2)\cdot Q(z_2^{-1}, z_1)$
4. $P(s_1, s_2)\cdot P(s_2, -s_1)$	4. $Q(z_1, z_2)\cdot Q(z_2, z_1^{-1})$
5. $P(s_1, s_2)\cdot P(-s_2, s_1)\cdot P(-s_1, -s_2)\cdot P(s_2, -s_1)$	5. $Q(z_1, z_2)\cdot Q(z_2^{-1}, z_1)\cdot Q(z_1^{-1}, z_2^{-1})\cdot Q(z_2, z_1^{-1})$
where $P_k(s_1^2, s_2^2) = P_k(s_2^2, s_1^2)$ for $k = 1, 2$	where $Q_k(x_1, x_2) = Q_k(x_2, x_1)$ for $k = 1, 2$
Octagonal symmetry	
1. $(s_1^2 - s_2^2)^{\alpha}\cdot P_1(s_1^2, s_2^2)$, where $\alpha = 0$ or 1	1. $(x_1 - x_2)^{\alpha}\cdot Q_1(x_1, x_2)$, where $\alpha = 0$ or 1
2. $P(s_1^2, s_2)\cdot P(s_2^2, s_1)$	2. $Q(x_1, z_2)\cdot Q(x_2, z_1)$
3. $P(s_1^2, s_2)\cdot P(s_2^2, -s_1)$	3. $Q(x_1, z_2)\cdot Q(x_2, z_1^{-1})$
4. $P(s_1, s_2^2)\cdot P(-s_2, s_1^2)$	4. $Q(z_1, x_2)\cdot Q(z_2^{-1}, x_1)$
5. $P_2(s_1, s_2)\cdot P_2(-s_1, s_2)$	5. $Q_2(z_1, z_2)\cdot Q_2(z_1^{-1}, z_2)$
6. $P_2(s_1, s_2)\cdot P_2(s_1, -s_2)$	6. $Q_2(z_1, z_2)\cdot Q_2(z_1, z_2^{-1})$
where $P_1(s_1^2, s_2^2) = P_1(s_2^2, s_1^2)$ and $P_2(s_1, s_2) = P_2(s_2, s_1)$	where $Q_1(x_1, x_2) = Q_1(x_2, x_1)$ and $Q_2(z_1, z_2) = Q_2(z_2, z_1)$

[a] Note that $x_i = z_i + z_i^{-1}$ and $y_i = z_i - z_i^{-1}$ for $i = 1, 2$.

10.9.1.1 Quadrantal Symmetry

THEOREM 10.9

For $T(s_1, s_2)$ to be stable and to possess quadrantal magnitude symmetry, $P(s_1, s_2)$ should satisfy the quadrantal symmetry conditions listed in Table 10.6 and the denominator polynomial should be product separable as $Q(s_1, s_2) = Q_1(s_1)\cdot Q_2(s_2)$ where Q_1 and Q_2 are stable 1-D polynomials.

Proof of Theorem 10.9: As mentioned above for $T(s_1, s_2)$ to possess quadrantal magnitude symmetry in its frequency response, $Q(s_1, s_2)$ should be expressible as

$$Q(s_1, s_2) = Q_A(s_1, s_2^2)\cdot Q_B(s_1^2, s_2) \tag{10.121}$$

Now it can be easily seen that for any s_1, $Q_A(s_1, s_2^2)$ will have zeros in the right half of s_2 plane unless Q_A is a polynomial of s_1 alone. Similarly it can be argued that Q_B should be a polynomial of s_2 alone. In other words, for $Q(s_1, s_2)$ to possess stability and quadrantal magnitude symmetry, it should be expressible as a product of two 1-D polynomials as

$$Q(s_1, s_2) = Q_1(s_1) \cdot Q_2(s_2) \tag{10.122}$$

such that Q_1 and Q_2 are stable analog polynomials.Q.E.D.

10.9.1.2 Octagonal Symmetry

As quadrantal symmetry is an integral part of octagonal symmetry, for $T(s_1, s_2)$ to possess octagonal symmetry it should first satisfy the conditions for quadrantal symmetry, i.e., the denominator polynomial should be separable. Then it can be shown that as octagonal symmetry also implies diagonal symmetry, $Q(s_1, s_2)$ should be expressible as

$$Q(s_1, s_2) = Q_1(s_1) \cdot Q_1(s_2) \tag{10.123}$$

10.9.1.3 Rotational Symmetry

In a similar manner examining the various factors that can be present in the polynomials possessing rotational symmetry, it can be shown that the denominator polynomial of a stable 2-D analog filter possessing rotational magnitude symmetry should be expressible as

$$Q(s_1, s_2) = Q_1(s_1) \cdot Q_1(s_2) \tag{10.124}$$

10.9.1.4 Diagonal Symmetry

It can be shown that the denominator of a stable 2-D analog filter possessing diagonal magnitude symmetry should satisfy the stability conditions and satisfy the symmetry condition:

$$Q(s_1, s_2) = Q(s_2, s_1) \tag{10.125}$$

10.9.2 Stability of Digital Filters with Symmetric Magnitude Response

Now applying the stability conditions on the polynomial factors that possess the various magnitude symmetries, the following conditions on the denominator polynomials of 2-D digital filters are obtained.

10.9.2.1 Quadrantal Symmetry

The denominator polynomial of a stable 2-D digital filter possessing quadrantal symmetry in its magnitude response can be expressed as

$$Q(z_1, z_2) = Q_1(z_1) \cdot Q_2(z_2) \tag{10.126}$$

where Q_1 and Q_2 are 1-D stable polynomials.

10.9.2.2 Octagonal Symmetry

The denominator polynomial of a stable 2-D digital filter possessing octagonal symmetry in its magnitude response can be expressed as

$$Q(z_1, z_2) = Q_1(z_1) \cdot Q_1(z_2) \tag{10.127}$$

10.9.2.3 Rotational Symmetry

The denominator polynomial of a stable 2-D digital filter possessing rotational symmetry in its magnitude response can be expressed as

$$Q(z_1, z_2) = Q_1(z_1) \cdot Q_1(z_2) \tag{10.128}$$

10.9.2.4 Diagonal Symmetry

The denominator polynomial of a stable 2-D digital filter possessing diagonal symmetry in its magnitude response should satisfy the stability condition and satisfy the symmetry condition:

$$Q(z_1, z_2) = Q(z_2, z_1) \tag{10.129}$$

10.10 Filter Design Procedure

We now present the design steps for 2-D z-domain separable denominator IIR filters with symmetry in the magnitude response. Computer optimization is used in this procedure to obtain the filter coefficients such that the filter specifications are satisfied.

Steps:

1. Identify the type of symmetry in the magnitude response specifications for the filter.
2. Assume a transfer function of the form

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1) \cdot D(z_2)} \tag{10.130}$$

   Select the numerator from the list of polynomials in Table 10.6, such that it satisfies the required symmetry identified in Step 1. The denominator is chosen to be separable so that its stability can be easily assured. This denominator possesses all four types of symmetries.
3. Select a suitable order for the filter such that the specifications can be met.
4. Choose a region in the (θ_1, θ_2) frequency plane to specify the desired magnitude response. (Note that $\theta_i = \omega_i \cdot T$). For quadrantal, diagonal, and rotational symmetries, the region need only be a 90° sector in the frequency plane. For octagonal symmetry, it need only be a 45° sector. Specify the frequency sample points in this region using concentric circles or rectangular grids.
5. Form an objective function to be minimized. This will be based on the difference between the magnitude response of the transfer function and the desired magnitude response, at the selected frequency points. One such objective function is

$$J = \sum_k \sum_l w_{kl}[F(\theta_{1k}, \theta_{2l}) - F_d(\theta_{1k}, \theta_{2l})]^2 \tag{10.131}$$

where

F is the magnitude-squared response

θ_{1k}, θ_{2l} are the sample frequency points where the desired response is specified

w_{kl} are the weights on errors

In this function, the variables to be optimized are the coefficients of the transfer function. The weights w_{kl} are chosen so as to provide the emphasis on the accuracy needed at different frequency points.

6. Use any minimization algorithm, such as those provided in MATLAB, to minimize the objective function J and obtain the optimal values for the filter coefficients. Verify that the filter specifications can be met with this set of filter coefficients. If not, make adjustment to the minimization algorithm and repeat the process.
7. Check the stability of the filter by finding the poles. Any unstable pole can be stabilized by replacing it with its inverse pole.

10.11 Filter Design Examples

10.11.1 Bandpass Filter

Using the procedure discussed in Section 10.10, we now design a bandpass filter with the ideal filter specification shown in Figure 10.6.

It can be seen that the filter possesses diagonal symmetry. So we select the numerator to be $N(z_1, z_2) = N(z_2, z_1)$, which is case (1) in the list of polynomials for diagonal symmetry in Table 10.6. We choose the order of the filter to be (4,4) and w_{kl} equal to 1 for all k and l. Because of the diagonal symmetry constraints, the number of variables to optimize is reduced from 33 to 19. The following are the forms for the numerator and denominator:

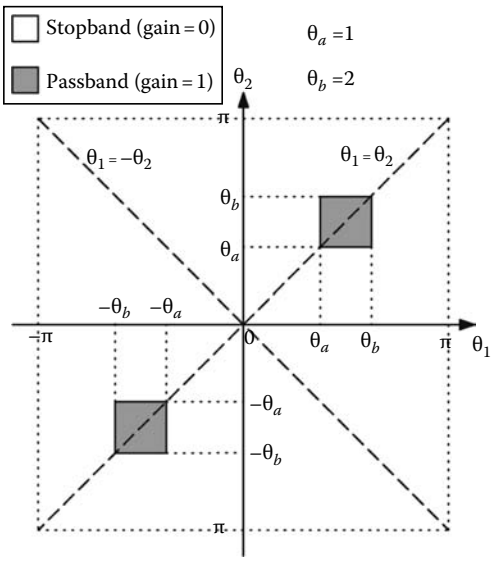


FIGURE 10.6 Specification for a bandpass filter with diagonal symmetry.

$$N(z_1, z_2) = \begin{array}{c} \\ z_2^0 \\ z_2^1 \\ z_2^2 \\ z_2^3 \\ z_2^4 \end{array} \begin{array}{c} \begin{array}{ccccc} z_1^0 & z_1^1 & z_1^2 & z_1^3 & z_1^4 \end{array} \\ \begin{bmatrix} a_{00} & a_{10} & a_{20} & a_{30} & a_{40} \\ \boxed{a_{10}} & a_{11} & a_{21} & a_{31} & a_{41} \\ \boxed{a_{20}} & \boxed{a_{21}} & a_{22} & a_{32} & a_{42} \\ \boxed{a_{30}} & \boxed{a_{31}} & \boxed{a_{32}} & a_{33} & a_{43} \\ \boxed{a_{40}} & \boxed{a_{41}} & \boxed{a_{42}} & \boxed{a_{43}} & a_{44} \end{bmatrix} \end{array} \tag{10.132}$$

and $D(z_i) = b_0 + b_1 z_i + b_2 z_i^2 + b_3 z_i^3 + z_i^4$, $i = 1, 2$

Also, because of symmetry, we need only specify the desired response in a reduced region (90° sector) in the frequency plane. In this case, we use the sector from 45° to 135°, and specify the sample points using rectangular grids.

We use the ''lsqnonlin'' routine in MATLAB Optimization Toolbox to minimize the objective function. The MATLAB M-files used to optimize the filter are listed in the Appendix at the end of the chapter. Optimize.m is the main program. Z2diir1.m calculates the objective and gradient functions. Diasym.m, Diasym1.m, and Diasym2.m are supporting M-files that impose the diagonal symmetry constraints on the objective and gradient functions. The coefficients to be optimized are arranged in vector form: xzo = [a_{00} a_{10} a_{20} a_{30} a_{40} a_{11} a_{21} a_{31} a_{41} a_{22} a_{32} a_{42} a_{33} a_{43} a_{44} b_0 b_1 b_2 b_3].

The transfer function coefficients of the optimized filter are listed below, together with the contour and 3-D magnitude plots (Figure 10.7). We verified that the filter is stable and meets the specification.

$a_{00} = 0.05691572886174$ $\quad a_{10} = -0.02461862190217$
$a_{20} = 0.00986568924226$ $\quad a_{30} = -0.01034804473379$
$a_{40} = 0.05820287661798$ $\quad a_{11} = -0.02118017629239$
$a_{21} = -0.00404160157329$ $\quad a_{31} = 0.03303916563349$
$a_{41} = -0.01033357674522$ $\quad a_{22} = 0.00255022502754$
$a_{32} = -0.00398097317269$ $\quad a_{42} = 0.00985123577827$
$a_{33} = -0.02118784417373$ $\quad a_{43} = -0.02462035270181$
$a_{44} = 0.05691802180139$
$b_0 = 0.45583704915947$ $\quad b_1 = -0.09763672454601$
$b_2 = 0.87533799314168$ $\quad b_3 = -0.03923759607499$

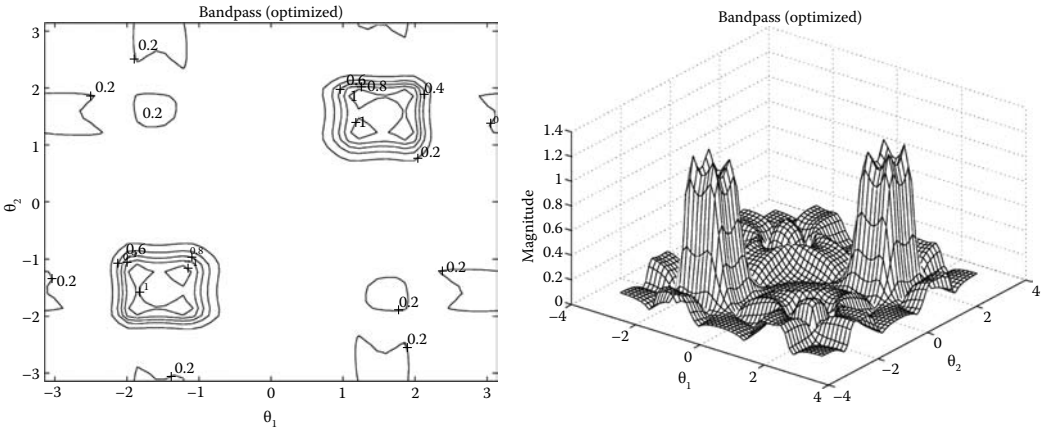


FIGURE 10.7 Contour and 3-D magnitude plots of the optimized bandpass filter.

Other types of filters can be designed using the same procedure. In the following, we provide examples of low-pass, high-pass, and bandstop filters with diagonal symmetry. We assume the same forms for the numerator and denominator as in Equation 10.132, and the same (4,4) filter order. Only minor modifications to optimize.m are needed to accommodate the different filter specifications. For each example, the initial filter specification, together with the final optimized filter coefficients and magnitude plots are shown (Figures 10.8 through 10.13).

10.11.2 Low-Pass Filter

Optimized low-pass filter coefficients:

$$
\begin{aligned}
a_{00} &= 0.00529914307068 & a_{10} &= -0.01633617926751\\
a_{20} &= -0.02625235827634 & a_{30} &= 0.01872969469730\\
a_{40} &= 0.01576064607841 & a_{11} &= 0.04957401022387\\
a_{21} &= 0.00624440464280 & a_{31} &= 0.02162875055170\\
a_{41} &= -0.05297244650352 & a_{22} &= -0.03401010515043\\
a_{32} &= 0.01453009659277 & a_{42} &= 0.03055298078630\\
a_{33} &= -0.02221031405884 & a_{43} &= -0.02655353678757\\
a_{44} &= 0.03073948639145 & &\\
b_0 &= 0.35738777321908 & b_1 &= -1.34613176291069\\
b_2 &= 2.36229363863415 & b_3 &= -2.3649826259138
\end{aligned}
$$

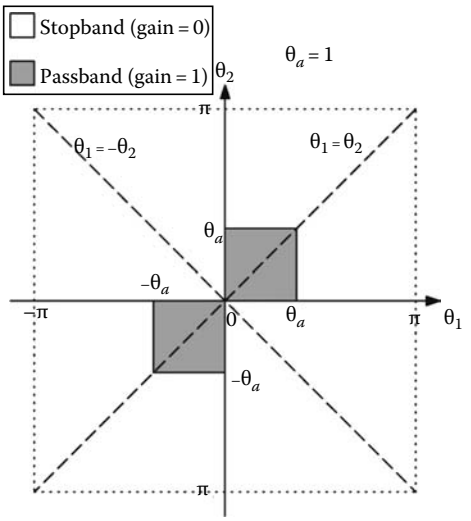


FIGURE 10.8 Specification for a low-pass filter with diagonal symmetry.

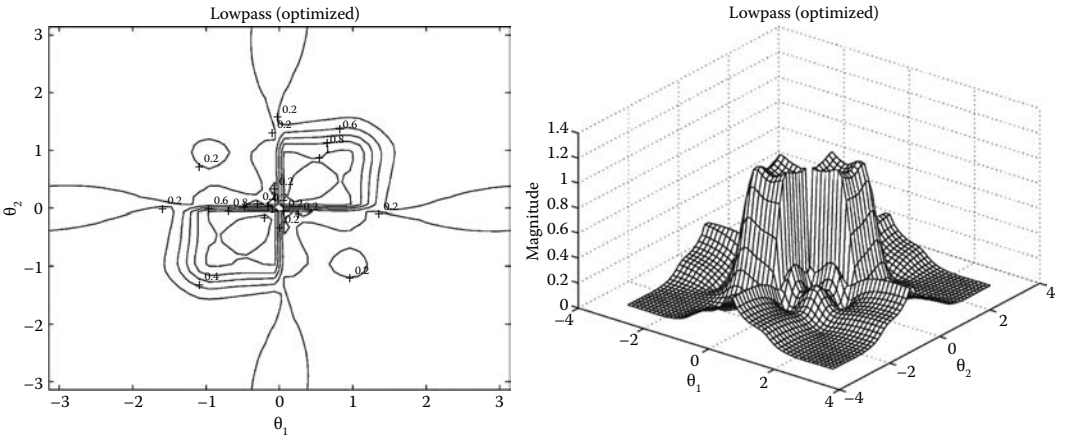


FIGURE 10.9 Contour and 3-D magnitude plots of the optimized low-pass filter.

10.11.3 High-Pass Filter

Optimized high-pass filter coefficients:

$$
\begin{aligned}
a_{00} &= 0.11673568605736 & a_{10} &= -0.23177721297900\\
a_{20} &= 0.43419109379123 & a_{30} &= -0.29863295283199\\
a_{40} &= 0.00449321269615 & a_{11} &= 0.46365543368518\\
a_{21} &= -1.00357273137798 & a_{31} &= 0.64451471450031\\
a_{41} &= 0.00269015989674 & a_{22} &= 1.92318260896734\\
a_{32} &= -1.18903964365735 & a_{42} &= 0.04108825448793\\
a_{33} &= 0.80208033691209 & a_{43} &= -0.06731123126511\\
a_{44} &= 0.04023458665886 & &\\
b_0 &= 0.0138207726422 & b_1 &= -0.29223860287727\\
b_2 &= 0.72903725441097 & b_3 &= -1.30068296054482
\end{aligned}
$$

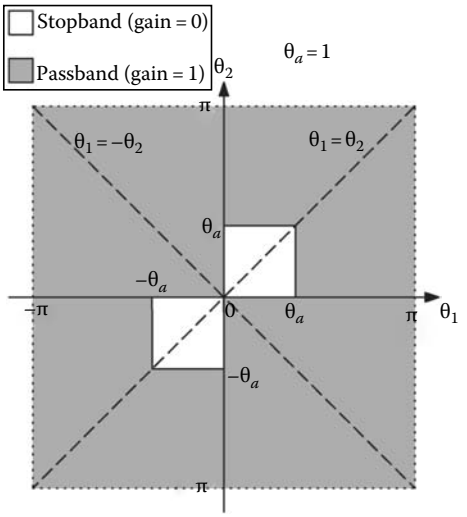


FIGURE 10.10 Specification for a high-pass filter with diagonal symmetry.

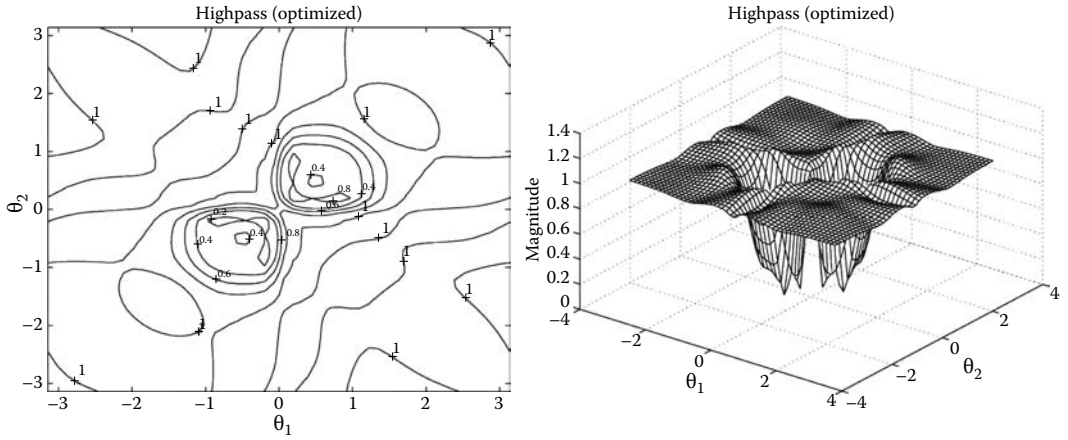


FIGURE 10.11 Contour and 3-D magnitude plots of the optimized high-pass filter.

10.11.4 Bandstop Filter

Optimized bandstop filter coefficients:

$$
\begin{aligned}
a_{00} &= 0.14253026125402 & a_{10} &= -0.08096362308810\\
a_{20} &= 0.46623126507478 & a_{30} &= -0.08095523783769\\
a_{40} &= 0.14066200552071 & a_{11} &= 0.06747075188723\\
a_{21} &= -0.24257225896378 & a_{31} &= 0.01771229512751\\
a_{41} &= -0.07763999364081 & a_{22} &= 1.34634826088519\\
a_{32} &= -0.24794622466503 & a_{42} &= 0.46598192581512\\
a_{33} &= 0.06876898730115 & a_{43} &= -0.08375671190833\\
a_{44} &= 0.14179014155689 & &\\
b_0 &= 0.13819412133529 & b_1 &= -0.11427757850242\\
b_2 &= 0.79195400054775 & b_3 &= -0.302423248254
\end{aligned}
$$

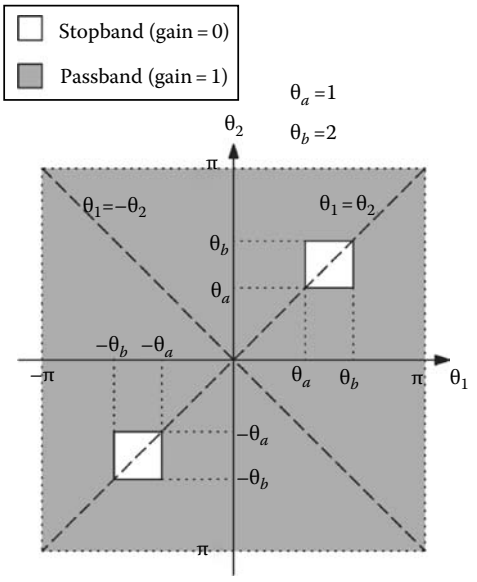


FIGURE 10.12 Specification for a bandstop filter with diagonal symmetry.

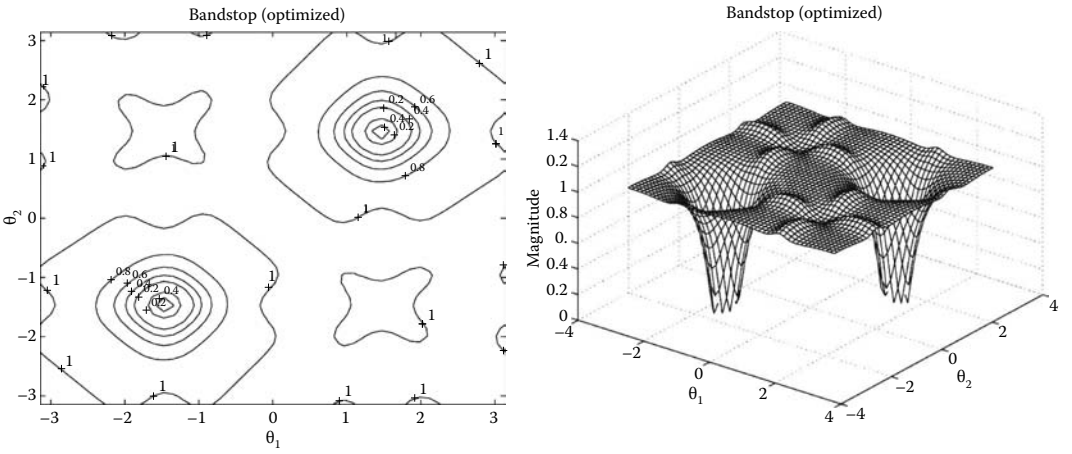


FIGURE 10.13 Contour and 3-D magnitude plots of the optimized bandstop filter.

Appendix: MATLAB Programs for Filter Design

Program 1: optimize.m

```
% Program to optimize for the 2-D bandpass filter in Figure 10.6

clear all;
close all;
order1 = 4;
order2 = 4;
pdiv = 15;
sdiv = 15;
pts = 20;

% use a known IIR transfer function for the initial values
[numz10,denz10] = cheby1(order1/2,0.5,[1/pi 2/pi]);
[numz20,denz20] = cheby1(order2/2,0.5,[1/pi 2/pi]);

% convert this transfer function to a form for optimization
xz0 = [];
for n = 1:order2+1
            xz0 = [xz0,numz10*numz20(n)];
end
xz0 = [fliplr(xz0),fliplr(denz10(2:order1+1)),fliplr(denz20(2:order2
+1))];
xz0 = diasym2(xz0,order1); %xz0 contains the initial values

% specify the desired response in the frequency plane
```

```
div = pdiv+sdiv;
t2 = [linspace(0,1,pdiv+1) linspace(1.1,pi,sdiv)];
t2 = t2(2:div+1);
theta1 = []; %theta1 and theta2 are the sample frequency points
theta2 = [];
for n = 1:length(t2)
   pt = round(t2(n)*pts);
   theta1 = [theta1 linspace(-t2(n),t2(n),pt)];
   theta2 = [theta2 ones(1,pt)*t2(n)];
end
fd = []; %fd contains the desired response
for n = 1:length(theta1)
   if theta1(n)>=1 & theta2(n)<=2
      fd = [fd 1];
   else
      fd = [fd 0];
   end
end

% optimize for the transfer function
optionz = optimset('Jacobian','on','Display','iter');
xzo = lsqnonlin('z2diir1',xz0,[],[],optionz,order1,order2,theta1,the
ta2,fd); %xzo contains the optimized filter coefficients

%plot the optimized magnitude response
points = 50;
theta1 = linspace(-pi,pi,points);
theta2 = linspace(-pi,pi,points);
[X,Y] = meshgrid(theta1,theta2);
X1 = reshape(X,1,points^2);
Y1 = reshape(Y,1,points^2);
dummy = zeros(1,points^2);
zresp = sqrt(z2diir1(xzo,order1,order2,X1,Y1,dummy));
zresp1 = reshape(zresp,points,points);
figure;
c = contour(X,Y,zresp1);
clabel(c);
xlabel('\it\theta_1')
ylabel('\it\theta_2')
title('Band-pass (optimized)');
figure
mesh(X,Y,zresp1);
view(37.5,37.5);
title('Band-pass (optimized)');
xlabel('\theta_1')
ylabel('\theta_2')
zlabel('Magnitude')
```

Program 2: z2diir1.m

```
function [F,J] = z2diir1(x,order1, order2,theta1,theta2,fd)
x = diasym(x,order1);
order = (order1+1)*(order2+1);
xnum = x(1:order);
xden1 = [x(order+1:order+order1) 1];
xden2 = [x(order+1+order1:length(x)) 1];
n = 0:order1;
m = 0:order2;
Ltheta = ones(1,length(theta1));
theta1a = (ones(order1+1,1)*exp(-i*theta1)).^( n'*Ltheta );
theta1b = repmat(theta1a,order2+1,1);
theta2a = (ones(order2+1,1)*exp(-i*theta2)).^(m'*Ltheta);
theta2b = repmat(theta2a.',order1+1,1);
theta2b = (reshape(theta2b,length(theta2),order)).';
A = xnum.'*ones(1,length(theta1));
thetablock = theta1b.*theta2b;
fnum = sum(A.*thetablock);
fden1 = sum( (xden1.'*Ltheta).*theta1a);
fden2 = sum( (xden2.'*Ltheta).*theta2a);
fden = fden1.*fden2;
f = fnum./fden;
fstar = conj(f);
F = f.*fstar-fd;
if nargout> 1
            A1 = (ones(order,1)*(fstar./fden)).*thetablock;
            B1 = (ones(order1+1,1)*(-f.*fstar./fden1)).*theta1a;
            B2 = B1(1:order1,:);
            C1 = (ones(order2+1,1)*(-f.*fstar./fden2)).*theta2a;
            C2 = C1(1:order2,:);
            R = [A1;B2;C2];
            result = R+conj(R);
            J = diasym1(result,order1).';
end
```

Program 3: diasym.m

```
function x1 = diasym(x,order1)
A = [];
ord = order1+1;
ind = 1;
for n = 1:ord
            k = ord-n+1;
            A(:,n) = [zeros(1,ord-k), x(ind:(ind+k-1))].';
            ind = ind+k;
end
A1 = tril(A,-1);
A2 = A1.';
```

```
A3 = A+A2;
x1 = reshape(A3,1,ord*ord);
x2 = x(ind:ind+order1-1);
x1 = [x1 x2 x2];
```

Program 4 – diasym1.m

```
function result = diasym1(A,order1)
ord = order1+1;
ord1 = ord^2;
a = 1:ord1;
b = reshape(a,ord,ord);
c1 = triu(b,1)';
d1 = c1(:);
e2 = find(d1)';
e1 = d1(e2)';
f1 = [e1 ord1+ord:ord1+2*order1];
f2 = [e2 ord1+1:ord1+order1];
A(f2,:) = A(f1,:)+A(f2,:);
A(f1,:) = [];
result = A;
```

Program 5: diasym2.m

```
function result = diasym2(A,order1)
ord = order1+1;
ord1 = ord^2;
a = 1:ord1;
b = reshape(a,ord,ord);
c1 = triu(b,1)';
d1 = c1(:);
e2 = find(d1)';
e1 = d1(e2)';
f1 = [e1 ord1+ord:ord1+2*order1];
A(f1) = [];
result = A;
```

References

1. A. Fettweis, Symmetry requirements for multidimensional digital filters, *International Journal of Circuit Theory and Application*, 5, 343–353, 1977.
2. P. K. Rajan and M. N. S. Swamy, Quadrantal symmetry associated with two-dimensional digital filter transfer functions, *IEEE Transactions on Circuits and Systems*, CAS-25, 340–343, June 1978.
3. M. Narasimha and A. Peterson, On using symmetry of FIR filters for digital interpolation, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-26, 267–268, 1978.
4. P. K. Rajan and M. N. S. Swamy, Symmetry conditions on two-dimensional half-plane digital filter transfer functions, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27, 506–511, October 1979.
5. D. M. Goodman, *Symmetry Conditions for Two-Dimensional FIR Filters*, Lawrence Livermore Lab Rep. UCID-311, 1979.

6. S. A. H. Aly, J. Lodge, and M. M. Fahmy, The design of two-dimensional digital filters with symmetrical or anti-symmetrical specifications, *Proceedings of the 1980 European Conference on Circuit Theory and Design*, Warsaw, Poland, pp. 145–150, 1980.
7. D. M. Goodman, Quadrantal symmetry calculations for nonsymmetric half-plane filters, *Proceedings of the 14th Asilomar Conference*, Pacific Grove, CA, 14, 1980.
8. S. A. H. Aly and M. M. Fahmy, Symmetry exploitation in the design and implementation of two-dimensional rectangularly sampled filters, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29, 973–982, 1981.
9. P. K. Rajan, H. C. Reddy, and M. N. S. Swamy, Further results on 4-fold rotational symmetry in 2-D functions, *Proceedings of the 1982 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paris, France, May 1982.
10. P. K. Rajan, H. C. Reddy, and M. N. S. Swamy, Fourfold rotational symmetry in two-dimensional functions, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30, 488–499, June 1982.
11. J. H. Lodge and M. M. Fahmy, K-cyclic symmetries in multidimensional sampled signals, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31, 847–860, 1983.
12. D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
13. B. P. George and A. N. Venetsanopoulos, Design of two-dimensional recursive digital filters on the basis of quadrantal and octagonal symmetry, *Circuits Systems and Signal Processing*, 3, 59–78, 1984.
14. J. Pitas and A. N. Venetsanopoulos, The use of symmetries in the design of multidimensional digital filters, *IEEE Transactions on Circuits and Systems*, CAS-33, 863–873, 1986.
15. M. N. S. Swamy and P. K. Rajan, Symmetry in 2-D filters and its application, in *Multidimensional Systems: Techniques and Applications*, Chapter 9, S. G. Tzafestas (Ed.), Marcel Dekker, New York, 1986.
16. A. Fettweis, Multidimensional digital filters with closed loss behavior designed by complex network theory approach, *IEEE Transactions on Circuits and Systems*, CAS-34, 338–344, 1987.
17. V. Rajaravivarma, P. K. Rajan, and H. C. Reddy, Study of phase symmetry in 3-D filters, *Proceedings of Southeast Conference 1989*, Columbia, South Carolina, April 1989.
18. V. Rajaravivarma, P. K. Rajan, and H. C. Reddy, Symmetry study on 2-D complex analog and digital filter functions, *Multidimensional Systems and Signal Processing*, 2, 161–187, 1991.
19. V. Rajaravivarma, P. K. Rajan, and H. C. Reddy, Planar symmetries in 3-D filter responses and their application in 3-D filter design, *IEEE Transactions on Circuits and Systems II*, 39(6), 356–368, June 1992.
20. V. Rajaravivarma, P. K. Rajan, and H. C. Reddy, Design of multidimensional FIR digital filters using symmetrical decomposition technique, *IEEE Transactions on Signal Processing*, 42(1), 164–174, January 1994.
21. H. C. Reddy, P. K. Rajan, G. S. Moschytz, and A. R. Stubberud, Study of various symmetries in the frequency response of two-dimensional delta operator formulated discrete-time systems, *Proceedings of the 1996 IEEE-ISCAS*, 2, 344–347, May 1996.
22. H. C. Reddy, I. H. Khoo, G. S. Moschytz, and A. R. Stubberud, Theory and test procedure for symmetries in the frequency response of complex two-dimensional delta operator formulated discrete-time systems, *Proceedings of the 1997 IEEE-ISCAS*, 4, 2373–2376, June 1997.
23. H. C. Reddy, I. H. Khoo, and P. K. Rajan, Symmetry in the frequency response of two-dimensional delta operator formulated discrete-time systems, *Proceedings of the 1997 ECCTD*, 3, 1118–1123, August 1997.
24. H. C. Reddy, I. H. Khoo, P. K. Rajan, and A.R. Stubberud, Symmetry in the frequency response of two-dimensional (γ_1, γ_2) complex plane discrete-time systems, *Proceedings of the 1998 IEEE-ISCAS*, 5, 66–69, May 1998.

25. I. H. Khoo, H. C. Reddy, and P. K. Rajan, Delta operator based 2-D filter design using symmetry constraints, *Proceedings of the 2001 IEEE-ISCAS*, 2, 781–784, May 2001.
26. I. H. Khoo, H. C. Reddy, P. K. Rajan, Delta operator based 2-D filters: symmetry, stability, and design, *Proceedings of the 2003 IEEE-ISCAS*, 3, 25–28, May 2003.
27. H. C. Reddy, I. H. Khoo, and P. K. Rajan, 2-D symmetry: Theory and filter design applications, *IEEE Circuits and Systems Magazine*, 3(3), 4–33, Third Quarter 2003.
28. I. H. Khoo, H. C. Reddy, and P. K. Rajan, Symmetry study for delta-operator-based 2-D digital filters, *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(9), 2036–2047, September 2006.
29. P. K. Rajan, H. C. Reddy, and M. N. S. Swamy, Symmetry relations in two-dimensional Fourier transforms, *Proceedings of the 1984 IEEE-ISCAS*, 3, 1022–1025, 1984.
30. H. Weyl, *Symmetry*, Princeton University Press, Princeton, NJ, 1952.
31. A. V. Shubinov and V. M. Koptsik, *Symmetry in Science and Art*, Plenum Press, New York, 1974.
32. J. Rosen, *Symmetry Discovered*, Cambridge University Press, Cambridge, England, 1975.
33. P. K. Rajan and H. C. Reddy, Symmetrical decomposition and transformation, *Mathematics and Computers in Simulation*, 27(5 & 6), 585–598, October 1985.
34. G. A. Bliss, *Algebraic Functions*, Dover, New York, 1966.
35. P. K. Rajan and H. C. Reddy, A test procedure for 2-D discrete scattering Hurwitz polynomials, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-37, 118–120, January 1989.
36. H. C. Reddy and P. K. Rajan, A comprehensive study of two-variable Hurwitz polynomials, *IEEE-Transactions on Education*, 198–209, August 1989.
37. H. C. Reddy, Theory of two-dimensional Hurwitz polynomials, *The Circuits and Filters Handbook*, W. K. Chen (Ed.), CRC Press, Boca Raton, FL, 2002.

II

Circuit Elements, Devices, and Their Models

John Choma, Jr.
University of Southern California

11 Passive Circuit Elements *Stanisław Nowak, Tomasz W. Postupolski, Gordon E. Carlson, and Bogdan M. Wilamowski* **11**-1
Resistor • References • Capacitor • References • Inductor • Defining Terms • References • Further Information • Appendices • Transformer • Defining Terms • References • Further Information • Semiconductor Diode • References

12 RF Passive IC Components *Tomas H. Lee, Maria del Mar Hershenson, Sunderarajan S. Mohan, Hirad Samavati, and C. Patrick Yue* **12**-1
Introduction • Fractal Capacitors • Spiral Inductors • On-Chip Transformers • References

13 Circuit Elements, Modeling, and Equation Formulation *Josef A. Nossek* **13**-1
Lumped Circuit Approximation • Circuit Elements and Connecting Multiport • Characterizations of Circuit Elements • Connecting Multiport • Tableau Formulation • References

14 Controlled Circuit Elements *Edwin W. Greeneich and James F. Delansky* **14**-1
Controlled Sources • References • Signal Converters • Defining Terms • References

15 Bipolar Junction Transistor Amplifiers *David J. Comer and Donald T. Comer* **15**-1
Introduction • Amplifier Types and Applications • Differences between Discrete and IC Amplifier Design • Building Blocks for IC Amplifiers • Compound Stage Building Blocks for IC Amplifiers • Operational Amplifiers • Wideband Amplifiers • IC Power Output Stages • References

16 Operational Amplifiers *David G. Nairn and Sergio B. Franco* **16**-1
Ideal Operational Amplifier • References • Nonideal Operational Amplifier • References • Frequency- and Time-Domain Considerations • Acknowledgment • References

17 High-Frequency Amplifiers *Chris Toumazou and Alison Payne* **17**-1
Introduction • Current Feedback Op-Amp • RF Low-Noise Amplifiers • Optical Low-Noise Preamplifiers • Fundamentals of RF Power Amplifier Design • Applications of High-Q Resonators in IF-Sampling Receiver Architectures • Log-Domain Processing • References

11

Passive Circuit Elements

11.1 Resistor **11**-1
Linear Resistor • Nonlinear • Dependence on Material Properties • Dependence on Geometry of Material • Nonideal Linear Resistor
References **11**-23
11.2 Capacitor **11**-24
Introduction • Linear Capacitor • Nonlinear Capacitor • Dependence on Material Properties • Dependence on Geometry of Material • Nonideal Linear Capacitor
References **11**-40
11.3 Inductor **11**-40
Basics • Air Inductor • Cored Inductor • Nonlinearity in Inductors • Magnetic Core Materials • Magnetic Materials Measurements: Basic Methodological Principles • Physical Irregularities • International Standards on Inductors
Defining Terms **11**-63
References **11**-64
Further Information **11**-65
Appendices **11**-65
11.4 Transformer **11**-66
Introduction • Ideal Transformer • Nonideal Transformer
Defining Terms **11**-72
References **11**-72
Further Information **11**-72
11.5 Semiconductor Diode **11**-73
Nonlinear Static *I–V* Characteristics • Diode Capacitances • Depletion Capacitance • Diode as a Switch • Temperature Properties • Piecewise Linear Model
References **11**-84

Stanisław Nowak
University of Mining and Metallurgy

Tomasz W. Postupolski
Institute of Electronic Materials Technology

Gordon E. Carlson
University of Missouri–Rolla

Bogdan M. Wilamowski
Auburn University

11.1 Resistor

11.1.1 Linear Resistor

11.1.1.1 Introduction

An ideal resistor is an electronic component, the fundamental feature of which is resistance R according to Ohm's law expressed by the equation

$$V = RI \tag{11.1}$$

where
V represents voltage in volts
I is the current in amperes
R is the resistance in ohms

The main parameters of a resistor are nominal resistance value, nominal power dissipation, and limited voltage value. According to their construction and technology, we can divide resistors into five groups: wirewound resistors, foil resistors, thin film resistors, thick film resistors, and bulk resistors. Each group has some advantages and disadvantages; until now it has been impossible to manufacture all of the needed resistors within one technology. It is more interesting to divide resistors with respect to their application into two groups as follows:

1. Fixed resistors, including low-power resistors of 0.05–2 W, high-power resistors of 2–100 W, high-voltage resistors, high-ohmic resistors, chip resistors, resistive networks
2. Variable resistors (potentiometers), including rotary control potentiometers, slide control potentiometers, preset potentiometers, and special potentiometers

11.1.1.2 Fixed Resistor

An ideal fixed resistor is an electronic component, the resistance value of which is constant with time and different environmental conditions. In practice, we can observe some changes of resistance in time and under high temperature, high humidity, frequency, and electrical load conditions, and so on. Those changes of a resistance, called the instability of resistor, are the basis for classification of resistors according to the requirements of the International Electrical Commission (IEC) and the International Organization for Standardization (ISO 9000–9004) in order to build in a reliability system.

Figure 11.1 presents different kinds of fixed resistors. Each resistor is marked mainly by resistance value *R* and production tolerance $\delta_{p(\pm)}$. Nominal resistance is rated according to the E6, E12, E24, E48, and E96 series. It is very important for the user to know not only the production deviation δ_p but also dynamic tolerance Δ.

Dynamic tolerance Δ and resistor class: The author's proposal for calculation of the dynamic tolerance Δ is given by Equations 11.2 and 11.3:

$$\Delta_+ = \delta_{p(+)} + \overline{\delta} + c * s \tag{11.2}$$

$$\Delta_- = \delta_{p(-)} - \overline{\delta} - c * s \tag{11.3}$$

where

$$\overline{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i \tag{11.4}$$

$$\delta_i = \frac{R_i(t) - R_i(0)}{R_i(0)} \times 100\% \tag{11.5}$$

where
n is the quantity of samples in test
t is the test time
R_i is the resistance of *i*, the resistor

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\delta_i - \overline{\delta})^2} \tag{11.6}$$

$c = 1.28$ for probability level 90% cases inside the range $\overline{\delta} \pm 1.28^* s$, $c = 1.64$ for probability level 95% cases inside the range $\overline{\delta} \pm 1.64^* s$.

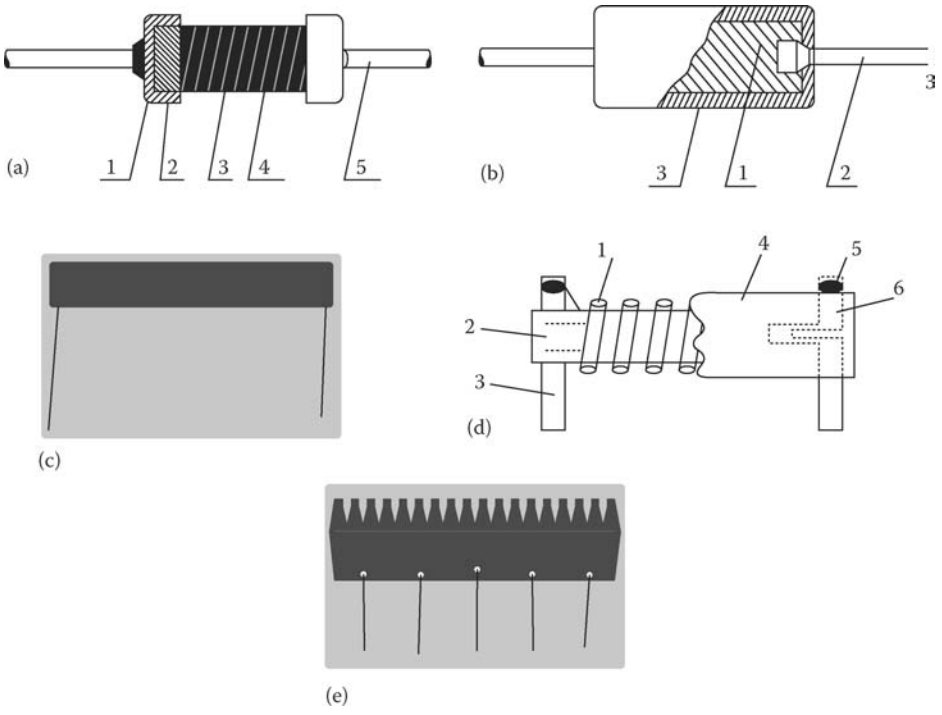


FIGURE 11.1 Typical fixed resistors: (a) film resistor: 1—metal cap, 2—electrode, 3—resistive layer, 4—groove, cut along screw line, 5—termination; (b) bulk composition resistor: 1—resistive composition, 2—termination, 3—pressed encapsulation; (c) high-voltage 100 MΩ thick film resistor; (d) wirewound power resistor, 1—resistive wire, 2—ceramic tube (substrate), 3—termination, 4—cement overcoating, 5—welded point, 6—cut for mechanical fixing of termination; and (e) thick film resistor with radiator.

Figure 11.2 illustrates dynamic tolerance Δ as an example of a thick film resistor endurance test prolonged up to 10,000 h. Resistors of 20 kΩ were manufactured by the Telpod factory (Poland) from Birox 1441 Du Pont paste. In Equations 11.2 and 11.3, for the long-life stability test, the coefficient c is chosen for normal distribution of changes δ_i for $t_j = 1000$ h because many experiments performed by author improved it with a high level of confidence. The hot humidity test causes distribution of changes asymmetrical, however.

The results of tests obtained for 400 resistors are shown in Figure 11.3a and b. For asymmetrical distribution, the following values for c parameters are suggested: $c_1 = -1$ for $\delta_i < \overline{\delta}$ and $c_2 = +3$ for $\delta_i > \overline{\delta}$. Dynamic tolerance Δ is recommended by IEC Publ. 115, 1982 because quality classes are connected directly, with instability δ_{max}, which is presented in Table 11.1. In practice, $\delta_{max} = \delta + cs$ according to Equation 11.2.

In accordance with IEC Publ. 115-5, 1982, the following classes of resistors are ranked according to precision group (see Table 11.2).

Temperature coefficient or resistance: The influence of temperature on resistance might be observed:

1. When the resistor is exposed to high temperature for a long time; this results in irreversible changes in resistance.
2. When the resistor is exposed to thermal condition of short duration (0.5 h); this results in reversible changes in resistance that are measured as temperature coefficient of resistance (TCR).

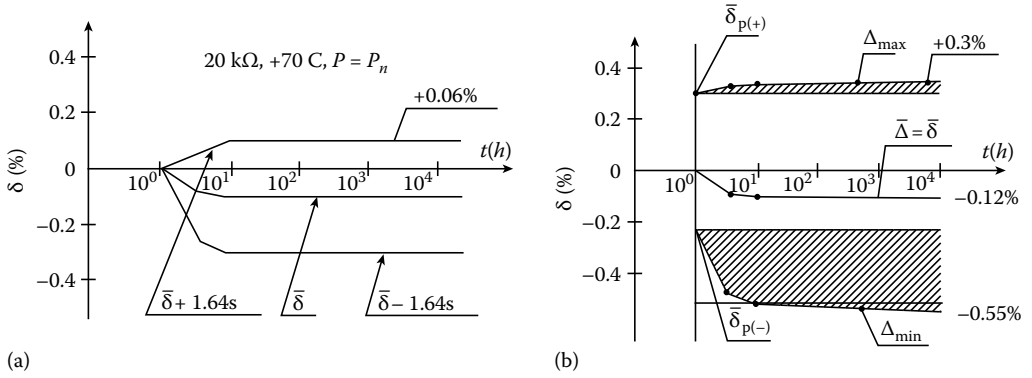


FIGURE 11.2 Resistor tolerance in exploitation. (a) The changes of resistance δ in time t $\overline{t\delta} = \varphi_1(t)$, $\overline{\delta} + 1.64^{*}s = \varphi_2(t)\overline{\delta} - 1.64^{*}s = \varphi_3(t)$ and (b) dynamic tolerance Δ as a function of time. Test time 10,000 h, $\delta_p = \pm 0.25\%$.

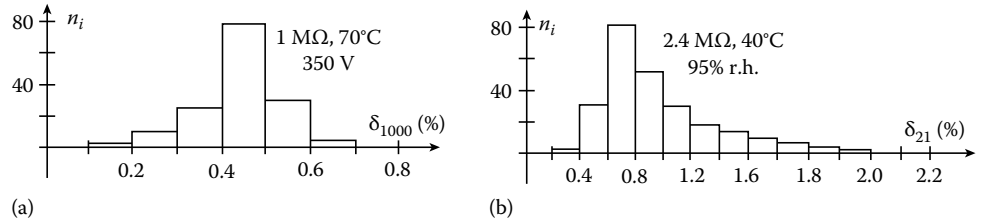


FIGURE 11.3 Distribution of δ (histogram). (a) Results after 1000 h endurance test: $\chi^2 = 5.01$, $\chi^2_{0.05,5} = 11.7$, $n = 150$ pcs, $-\overline{\delta} = 0.45\%$, $s = 0.24\%$; (b) result after 21-day hot humidity test: $\chi^2 = 100$, $\chi^2_{0.05,8} = 15.5$, $n = 246$ pcs, $\overline{\delta} = 0.79\%$, $s = 0.47\%$, χ^2—chi square distribution.

TABLE 11.1 Instability after Tests for Different Quality Resistor Classes

Classes	δ_{max} after Tests: Endurance Test Hot Humidity Test Climate Cycles Test	Recommended δ_p	Remarks
15	±(15% + 0.5 Ω)	±20%, ±10%, ±5%	
10	±(10% + 0.5 Ω)	±2%, ±1%	Common use resistor
5	±(5% + 0.1 Ω)		
3	±(3% + 0.1 Ω)		
2	±(2% + 0.1 Ω)	±5%, ±2%, ±1%	
1	±(1% + 0.05 Ω)	±0.5%, ±0.25%	Stable resistor
0.5	±(0.5% + 0.05 Ω)	±0.1%	
0.25	±(0.25% + 0.05 Ω)		

TABLE 11.2 Different Resistor Classes Ranked according to Precision Group

Classes	δ_{max} after Tests	Recommended δ_p
0.5	±(0.5% + 0.05 Ω)	±1%, ±0.5%, ±0.25%
0.25	±(0.25% + 0.05 Ω)	±0.1%, ±0.05%
0.1	±(0.1% + 0.01 Ω)	±0.025%
0.05	±(0.05% + 0.01 Ω)	±0.01%

TCR may be calculated from Equation 11.7 and in practice from Equation 11.8

$$\text{TCR} = \frac{1}{R}\frac{\partial R}{\partial T} \tag{11.7}$$

$$\text{TCR} = \frac{1}{R_0}\frac{R_T - R_0}{T - T_0} \tag{11.8}$$

where

R_0 is the resistance measured at room temperature T_0
R_T is the resistance measured at temperature T
T_0 is the room temperature

In Figure 11.4, some curves $R(T)$ are presented versus temperature for four types of resistors. It can be seen that in the tested range of temperature, TCR is positive and constant for curves 1 and 2 but negative and constant for curve 4. A different result is obtained for curve 3; at temperatures lower than T_m, TCR is negative, at higher temperatures than T_m, TCR is positive, and at T_m, TCR = 0. When $T_m = 40°C$, that type of resistor is the most interesting for users, because in the operating temperature range of 20°C–60°C, TCR is very small, almost zero.

As recommended by IEC Publ. 115, 1982, TCR limit values for different quality classes of resistors are shown in Table 11.3. Data in positions 1–5 refer to common use resistors, in positions 6–10, data refer to stable resistors, and in positions 11–15 to precision resistors.

Resistive network: In electronic circuits, resistors are often used as elements of dividers. In that case, it is more convenient to apply resistive networks (see Figure 11.5).

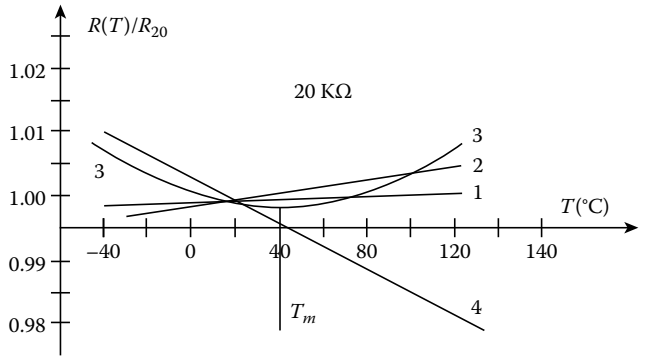


FIGURE 11.4 Dependence of resistance on ambient temperature. 1—wirewound resistor, 2—thin film resistor, 3—thick film ruthenium based resistor, 4—pyrolitic carbon resistor.

TABLE 11.3 TCR Limit Values for Different Quality Classes of Resistors

Number	TCR (ppm/K)	Number	TCR (ppm/K)	Number	TCR (ppm/K)
1	±2500	6	±250	11	±25
2	−800 to +2500	7	±100	12	±20
3	±1000	8	±50	13	±15
4	−400 to −1000	9	±25	14	±10
5	±500	10			

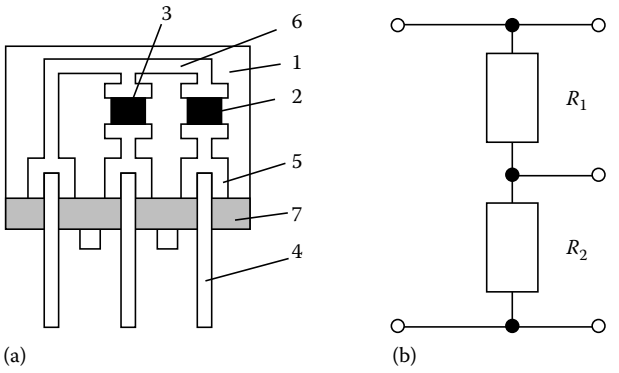


FIGURE 11.5 Thick film resistive network. (a) Topography of divider: 1—substrate, 2—resistor R_1, 3—resistor R_2, 4—wire termination, 5—soldering point, 6—conductive path, 7—insulating saddle and (b) electrical circuit.

Because the resistive network is deposited on a substrate in one technological cycle, both the TCR and time instability δ are almost the same for different resistors in the network. It appears to be more important for a user of a circuit to know differences Δ TCR and Δδ instead of absolute values of TCR or δ. It is estimated that Δ TCR can exhibit values of 1–3 ppm/K for TCR $\leq$ 50 ppm/K and Δδ can get value of 0.02% for $\delta = 0.5\%$.

Chip resistor: The development of electronic circuit mounting technology is going toward reliability and flexibility improvement and this results in a new assembly technique, known as surface mounting technology (SMT). Because SMT components have no terminations, their dimensions can be reduced. The smallest chip resistor is 2.5 mm long. Figure 11.6 depicts a 0.25 W chip resistor. Requirement

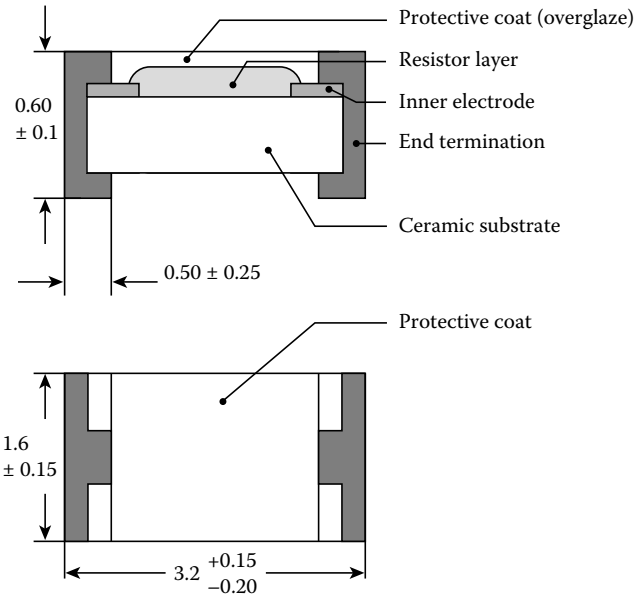


FIGURE 11.6 Chip resistor, nominal power 0.25 W (Philips Components Catalogue, 1989). Dimensions are in millimeters.

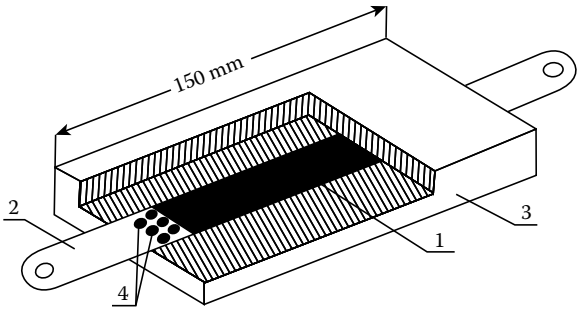


FIGURE 11.7 High-frequency resistor, nominal power 60 W. 1—resistive band, 2—band termination, 3—hot pressed overcoat, 4—holes for decreasing contact resistance.

parameters for chip resistors are the same as for fixed resistors with terminations. Thick film technology is often used in manufacturing chip resistors.

High-ohmic, high-voltage resistor: Usually resistors have resistances below 10 MΩ but sometimes resistors up to 10 GΩ are needed (e.g., in pH measurements, radiation particle detection, and so on). Only thick film technology enables production resistors of such high resistance values. Because the range of sheet resistance of the thick film layer changes from 1 Ω/square to 100 MΩ/square, we can easily get a resistance range from 1 to 10 GΩ. Laser trimming and shaping of the layer allows easily to get from 100 to 1000 squares of resistive layer.

A very high value of sheet resistance decreases the thermal and long-life stability of resistors, so it is advisable to design stable resistance with inks of 1 MΩ/square and to obtain the required resistance value by multiplying the number of squares.

High-ohmic resistors can be used as high-voltage resistors if their resistive paths are long enough.

The required voltage strength is a maximum of 2 kV/cm of resistive path. These types of resistors are used up to 10 kV in TV focusing systems.

High-power resistor: Very often, in electrical systems as well as in some electronic circuits (e.g., power suppliers, power amplifiers, R-TV transmitters, and radar equipment), resistors with dissipation power above 5 W are necessary. For dc and low-frequency applications up to 100 W, power resistors are realized by a cement layer but high parasitic inductance makes them useless for higher frequency performances. Film and band resistors have very good high-frequency characteristics and they are suggested for high-frequency applications. Resistive bands are made of boron–carbon or other compositions. Nominal resistance ranges are as follows: 50, 75, 300, and 600 Ω. In Figure 11.7, a band resistor of low inductance is shown.

11.1.1.3 Variable Resistor

The variable resistor, very often called a potentiometer, is an electronic component in which by movement of mechanical wiper the resistance is regulated. A variable resistor can regulate voltage when it is joined to the circuit as a fourth-pole element (Figure 11.8a). It can regulate current when it is joined to the circuit as a twin-pole element in series with load R_L (Figure 11.8b). Requirements for variable resistors are similar as for fixed resistors but several additional problems must be considered: mechanical endurance, rotational noise, contact resistance variation (CRV), the type of regulation curve (taper), and parallel curves in tandem potentiometers (stereo potentiometers).

Variable potentiometers can be divided into three groups: rotary control potentiometers, slide control potentiometers, and preset potentiometers.

In Figure 11.9, photos of different types of potentiometers are shown. With respect to their application we can divide potentiometers into several categories: standard type (common use) potentiometers with a

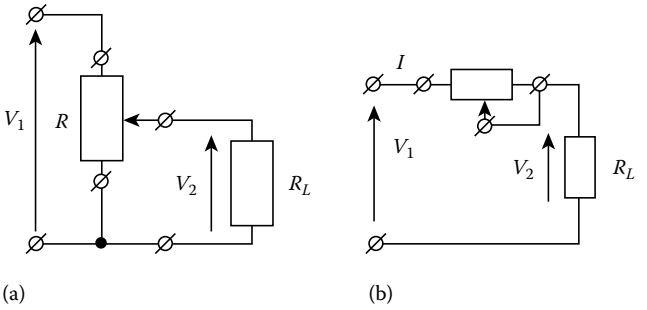


FIGURE 11.8 Potentiometer in an electric circuit: (a) as four-pole, voltage divider and (b) as twin-pole, rheostat.

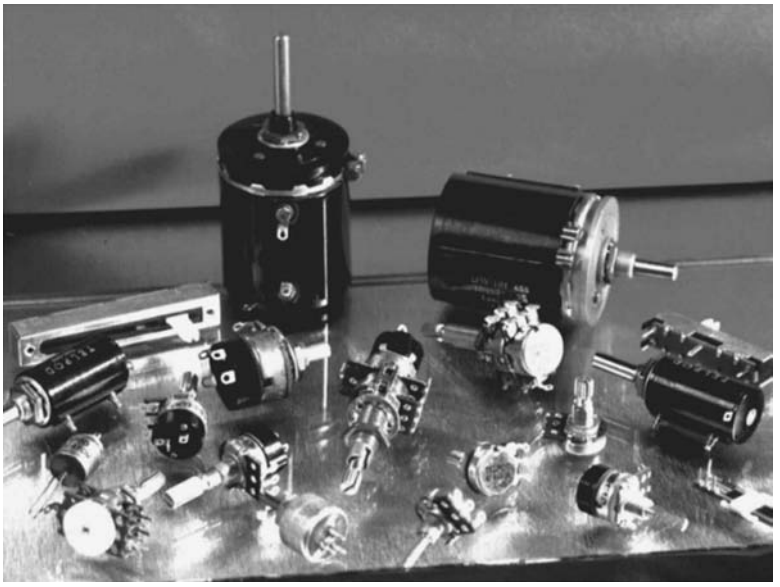


FIGURE 11.9 Photos of different potentiometers.

carbon polymer resistive layer, high-stability potentiometers with a cermet layer, precision potentiometers formed as wirewound, or thin film ones.

To increase sensitivity of regulation, a lead screw actuated potentiometer is used, in which the screw moves the nut connected with the wiper. Slow displacement of the wiper causes fluent resistance regulation.

Specially built potentiometers (helipots) are used for precise regulation (see Figure 11.10). In that case, the wiper is moving along a screw line. This means that for 10 rotations of shaft, the total angle is 3600° and the way of wiper is 10 times longer than in a simple rotary potentiometer. In helipots, precision of adjustment depends on the diameter of potentiometer. This type

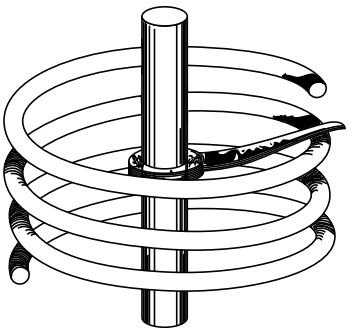


FIGURE 11.10 Helipot—a principle of work. (From Bourns, Inc., *The Potentiometer Handbook*, McGraw-Hill, New York, 1975. With permission.)

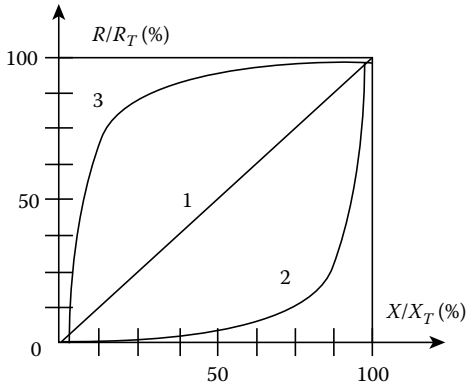


FIGURE 11.11 The main tapers of potentiometers. 1—linear, 2—exponential, 3—antiexponential. X_T—total wiper direction, R_T—total resistance. At the beginning and at the end of wiper movements there are very low resistive paths.

of a potentiometer is manufactured by Beckman, Bourns, and others.

Mechanical endurance: During the test of mechanical endurance, the wiper has to perform many total cycles. For present potentiometers, the number of cycles is 100–500; for control potentiometers it is 10^5–10^6 cycles.

***Regulation curves* (*taper*)**: The most popular regulation curve is a straight line; but as it is well known, our ear has a logarithmic characteristic. Therefore, for volume regulation in radios, audio amplifiers, and television sets, potentiometers with an exponential curve must be used. Figure 11.11 shows typical curves (tapers) of potentiometers.

In practice, a nonlinear curve is exactly realized by a few linear segments of resistive layer. Because each segment has another resistivity, it is necessary to use several segments to obtain a better exponential or antiexponential approximation. The minimum number of segments to perform an exponential curve is two, but then some steps in regulation are unavoidable. In production, we can achieve potentiometers performing sinus, cosinus, and other curves.

Curve parallelism of stereo tandem potentiometers: A stereo potentiometer with an exponential curve has to fulfill the additional requirement for parallel curves; both potentiometers are controlled by one roller (e.g., at attenuation of 40 dB the difference between both curves must be smaller than 2 dB).

11.1.2 Nonlinear

11.1.2.1 Varistor

A voltage-dependent resistor (VDR), called a varistor, is a resistor whose characteristic V versus I is not a straight line and a small change of voltage causes a significant change of current according to the equation

$$V = CI^{\beta} \tag{11.9}$$

where

β is the nonlinearity coefficient = 0.03–0.4 (it depends on the material and manufacturing technology)

C is the varistor-dependent coefficient

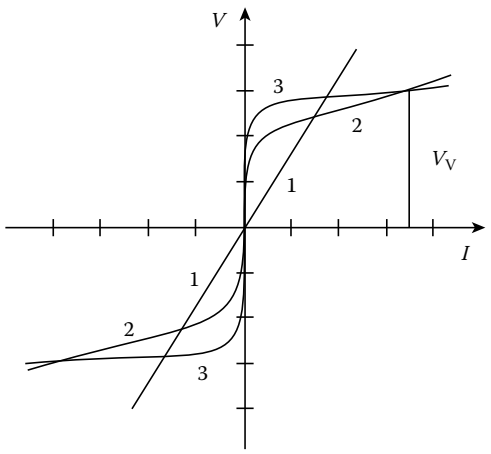


FIGURE 11.12 Voltage versus current characteristics. 1—linear resistor, 2—SiC varistor, 3—ZnO varistor, V_v—varistor voltage.

The main parameters of varistor are nonlinearity coefficient β and varistor voltage V_v measured at constant current, for example, 1 mA. Comparisons of the characteristics of a linear resistor and two types of varistors are shown in Figure 11.12.

A varistor can be used for dc voltage stabilization and especially for electronic circuit protection against

overvoltage pulses caused by industrial distortions and atmospheric discharges. Coefficient β is calculated from Equation 11.10:

$$\beta = \frac{\lg(V_2)/(V_1)}{\lg(I_2)/(I_1)} \tag{11.10}$$

when $I_2/I_1 = 10$, the denominator is equal to 1 and

$$\beta = \lg V_2 - \lg V_1 \tag{11.11}$$

To explain the above relation, Figure 11.13 is helpful, where both V and I are in logarithmic scale. The slope of the straight line segment of this curve equals β. For SiC varistors (curve 2 in Figure 11.12), $\beta = 0.12–0.4$; for ZnO varistors however $\beta = 0.03–0.1$.

Varistor voltage is in the range of 4 V up to 2 kV; it depends on varistor thickness (length). To get a higher operating voltage, disk varistors should be connected in pile. Maximum pulse current is in the range of 0.2 A up to 2 kA; it depends on the diameter of the varistor body. For pulse work, the following additional parameters are important: the capacity of the varistor (it is in the range of 100 pF up to 1 μF) and absorption energy (it is in the range 1–2200 J) [10].

11.1.2.2 Thermistor

A temperature-dependent resistor, called a thermistor, is a resistor with significant TCR, which can be positive (PTC) or negative (NTC). Some groups of thermistors are characterized by a very rapid change of resistance in temperature. Those thermistors are called critical temperature resistors (CTR). They can be positive CTR_P or negative CTR_N. Figure 11.14 presents typical characteristics R versus T for different types of thermistors.

NTC and PTC thermistors are used for stabilization of the working point in temperature for different electric circuits and as well as for temperature measurement. CTR thermistors are applied as protective elements against overheating in electronic circuits. CTR_Ps are used in degaussing circuits in color TV tubes. In some catalogs, CTR_Ps are called PTCs [7]. The electrical, climatic, and mechanical requirements of thermistors are almost the same as for fixed resistors but some additional parameters were introduced as well, such as thermal time constant (in seconds), heat capacity (in J/K), dissipation factor (in mW/K), switch temperature or critical temperature (in °C) (for CTRs only). The first three

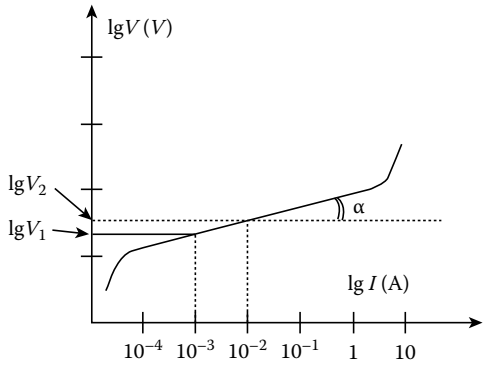


FIGURE 11.13 log(*V*)/log(*I*) characteristic of a varistor for nonlinearity coefficient β description. $\beta = \text{tg}\alpha = \log V_2 - \log V_1$; $I_2/I = 10$.

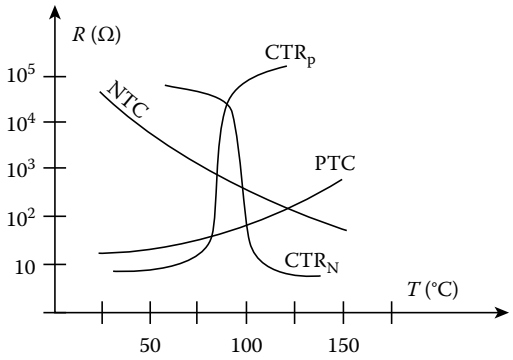


FIGURE 11.14 Characteristic $R = \varphi(T)$ of different thermistors. NTC—negative temperature coefficient thermistor; PTC—positive temperature coefficient thermistor; CTR—thermistor with critical temperature of resistance; index *N* curve falling with temperature; index *P*-curve rising with temperature.

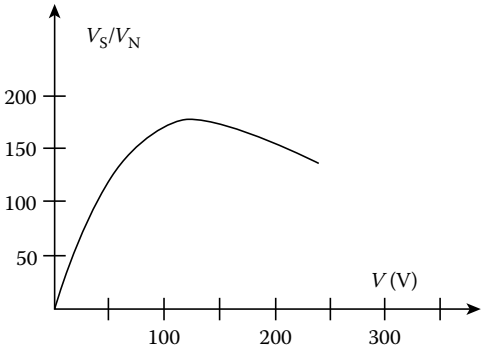


FIGURE 11.15 Signal-to-noise ratio versus polarization voltage V_{DC} of PbS photoresistor, Mullard type, 615 V. V_S—signal voltage, V_N–noise voltage, photo source temperature 473 K, $f = 800$ Hz, $\Delta f = 50$ Hz. (From Nowak, S., Wenta, A., and Kuzma, E., *Handbuch der Elektronik*, Vol. 2 and Ambroziak, A. et al., Vol. 7, Franzis-Verlag, Munich, 1979. With permission.)

are related to the thermistor dimensions, the remaining to the row material and technological process.

11.1.2.3 Photoresistor

A photoresistor is a film resistor whose resistance is sensitive to the light, that is, it depends on light intensity and light wavelength λ. The latter, in turn, depends on the kind of material used as follows.

For Cd S: $\lambda = 0.4$–0.7 μm, (visible light), for Pb S: $\lambda = 0.9$–3.5 μm (infrared), for Ge Si-doped Zn: $\lambda = 4$–15 μm, for Ge doped Sb: $\lambda = 30$–100 μm. This means that for a different photoresistor there exists an optimal wavelength at which maximum sensitivity (maximum change of resistance between lightness and darkness) occurs. During the design of a circuit with a photoresistor it is necessary to know at which polarization voltage the smallest noise exists. Figure 11.15 presents V_S/V_N versus polarization voltage for a PbS photoresistor made by Mullard.

11.1.2.4 Magnetoresistor

Some thin film multilayer ferromagnetic structures cause changes in magnetic field H. This phenomenon is called the magnetoresistive effect. An electronic component in which the magnetoresistive effect occurs is called a magnetoresistor and is usually used as a sensor. Special preparation of a ferromagnetic multilayer allows achievement of magnetosensitivity up to 100 MHz. The change of resistance is 1%–5% at the change of magnetic field H of about 10 Oe. Very often, two magnetoresistors are joined in a Wheatstone bridge and then sensitivity of the sensor is doubled.

11.1.3 Dependence on Material Properties

Materials selected for resistors have played a fundamental role in resistor production. Resistive elements are composed for a metal alloy, carbon, metal oxide, and mixtures of insulating and conducting particles such as polymer and black carbon, glass, and bismuth ruthenate, as well as glass and metal oxide. Semiconductors are also good materials for resistors, especially for nonlinear resistors such as varistors, thermistors, and photoresistors.

11.1.3.1 Influence of Resistive Material on TCR

Nonmagnetic metals: According to Grüneisenn's principle [2,3] temperature influences resistivity as follows:

$$\rho_T = \rho_\Theta \frac{T - 0.15\Theta}{0.85\Theta} \quad \text{for } T \geq 0.15\Theta \tag{11.12}$$

and

$$\mathrm{TCR}_{20} = \frac{1}{293 - 0.15\Theta} \tag{11.13}$$

$$\Theta = \frac{h\nu_{\max}}{k} \tag{11.14}$$

TABLE 11.4 Debye Temperature and Resistivity of Nonmagnetic Metals

Metal	ρ_{20} at $T = 293$ K (10^{-8} Ω m)	Θ (K)	0.15Θ (K)	ρ at Θ (10^{-8} Ω m)
Ag	1.62	214	32	1.16
Cu	1.68	320	48	1.94
Au	2.22	160	24	1.17
Al	2.73	374	56	3.79
Zn	6.12	180	27	3.65
Pt	10.6	220	33	7.91
Pb	20.8	84.5	12.7	5.5
W	5.39	346	52	6.76

Source: Neudeck, G.W., *The PN Junction Diode*, Vol. II, Modular Series on Solid-State Devices, Reading, MA: Addison-Wesley, 1983. With permission.

where

$h = 6.625$ Js = Planck's constant
$k = 1.38$ J/K = Boltzmann's constant
Θ = the Debye temperature (for several nonmagnetic metals is given in Table 10.4)
ν_{max} = the maximal elastic frequency of the atom in a lattice of metal
ρ_Θ = the resistivity of the metal at the Debye temperature

From Table 11.4, 0.15Θ is in the range 10–60 K, which means that

$$\text{TCR}_{20} = \frac{1}{283} \text{ to } \frac{1}{233} \tag{11.15}$$

that is, TCR = +3500 up to +4300 ppm/K.

For nonmagnetic metal, TCR_{20} is constant, and temperate dependency of resistivity can be written as follows:

$$\rho_T = A\rho_\Theta T \tag{11.16}$$

where A = constant.

It appears that for nonmagnetic metal in the range of temperature $T \geq 0.15\Theta$, resistivity is proportional to the ambient temperature. On this basis, the resistive platinum thermometer is built.

Magnetic metal: For magnetic metal, for example, Fe, Ni, Co, the relation $\rho_T = f(T)$ is nonlinear and given by following equation:

$$\rho_T = CT^{1.7} \quad \text{for } T \leq T_c \tag{11.17}$$

where T_c is the Curie temperature (see Table 11.5).

TCRs measured for iron and nickel are about +4500 ppm/K. Pure metal, both magnetic and nonmagnetic, is not useful for resistor design due to its large TCR and low resistivity.

TABLE 11.5 Curie Temperature for Different Metals

Metal	T_c [K]
Fe	1043
Ni	635
Co	1400

Metal alloy: Increasing of resistivity with temperature can be explained by atomic vibrations in the crystalline lattice. Those atoms cause obstructions to free electrons and the higher temperature causes an increase of resistance (obstruction).

Such a resistivity is the first component ρ_s of an alloy resistivity ρ. According to Mathiessen's rule [2], we can add the second component of resistivity ρ_i that presents obstruction of free electrons by atoms of impurities in the metal lattice.

$$\begin{aligned} \rho &= \rho_s + \rho_i \\ \rho_i &= \text{const}, \quad \rho_i \neq \varphi(T) \end{aligned} \tag{11.18}$$

where ρ_s is given by Equations 11.16 and 11.17.

When $\rho_i \gg \rho_s$, TCR is very small. In this case $\rho \gg \rho_s$ as well. This means that a resistor made of a specially prepared metal alloy can have very small TCR and large resistivity ρ. That result is very remarkable for a metal alloy resistor design.

Example:

Constantin 60% Cu + 40% Ni: $TCR_{20} = 1$–5 ppm/K, $\rho_{20} = 0.49 \times 10^{-6}$ Ωm in contrast with pure copper: TCR = 4000 ppm/K and $\rho_{20} = 0.0168 \times 10^{-6}$ Ωm; Canthal 70% Fe + 23% Cr + 4.5% Al + 1% Co + 1.5% other metals: $TCR_{20} = 50$ ppm/K, $\rho_{20} = 1.4 \times 10^{-6}$ Ωm.

This example demonstrates that, for alloys, TCR is from 100 up to 1000 times smaller and ρ_{20} is from 20 up to 100 times greater in comparison to pure metals (see also Table 11.4). This information is useful for wirewound and foil resistor design.

11.1.3.2 Thin Film Resistor

Over 90% of thin film resistors are made of metal alloy. A thin layer of metal alloy is deposited on ceramic or glass substrate. Sheet resistance $R(\Omega/\text{square})$ of a metal alloy thin film resistor varies from 10 Ω/square up to 200 Ω/square; the nominal resistance range is from 10 Ω to 10 MΩ; TCR ≈ 15–150 ppm/K; classes 5, 3, 2, 1; precision resistors are available.

For a high nominal resistance value, the resistance layer is specially shaped by laser trimming to get enough squares. Some features of thin film layer are a bit different from bulk metal alloy.

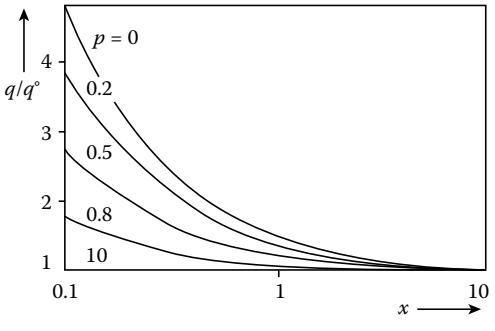


FIGURE 11.16 Influence of layer thickness on resistivity of metal thin film. ρ_0—resistivity of bulk metal; p—quantity of reflected electrons; $1 - p$—quantity of absorbed electrons. (From Maissel, L.I. and Glang, R. eds., *Handbook of Thin Film Technology*, McGraw-Hill, New York, 1970. With permission.)

***Fuchs's and Sondheimer's effect* [2]**: When the thickness of the resistive layer is smaller than a free path of the free electron, the resistivity increases.

Figure 11.16 shows the influence of normalized thickness κ on resistivity ρ:

$$\kappa = \frac{t}{\lambda} \tag{11.19}$$

where

t is the thickness of a layer

λ is the free path of the electron, which equals 20–30 nm [2]

κ is expressed by Equation 11.19

Grain effect: Because free electrons are reflected and absorbed by edges of grains, the resistivity increases when the quantity of grains becomes greater. Annealing processes of the layer limit that effect.

Figure 11.17 illustrates the result of the annealing process [2]. Small grains of the resistive layer are also the reason for the large absorption of humidity and gases that causes instability of the resistor.

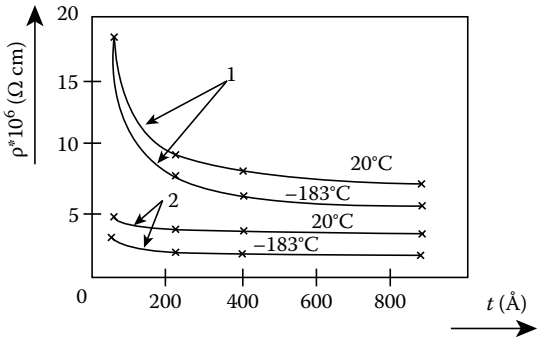


FIGURE 11.17 Influence of annealing process on resistance of thin film. 1—before annealing; 2—after annealing. (From Maissel, L.I. and Glang, R. eds., *Handbook of Thin Film Technology*, McGraw-Hill, New York, 1970. With permission.)

11.1.3.3 Pyrolytic Carbon Resistor

Carbon deposited in vacuum on a ceramic substrate performs very inexpensively and produces a quite good quality resistive layer. Pyrolytic carbon has graphite structure sp^3; three electrons create bonds and the fourth is a free electron. Surface resistance is 10 Ω/square up to 2 kΩ/square; the nominal resistance value is in range 10 Ω to 10 MΩ; encountered classes are 15, 10, 5, 3 (see Table 11.1); TCR ≡ −200 to −1000 ppm/K. About 50% of the produced resistors are carbon pyrolytic ones.

11.1.3.4 Thick Film Resistor

A thick film resistive layer is prepared from a composition called paste or ink that is a mixture of conducting particles and glass. Conducting particles have a metallic conduction mechanism. By mixing different quantities of both particles, we can obtain a high-range surface resistance of 1 Ω/square to 10 MΩ/square. This means that the nominal resistance range might be 0.5 Ω up to 1 GΩ; classes are 5, 3, 2, 1, 0.5; TCR = ±100 and ±50 ppm/K; in a resistive network, ΔTCR = 3 and 1 ppm/K.

The theory of conduction mechanisms is not well known. Several models of conduction mechanisms are used, but one of them proposed by Pike and Seager is used [9]. Each conductive particle is surrounded by glass. There particles form the chains of metal–insulator–metal (MIM). Electrons are passed through the insulator by tunneling. In the layer of very high resistivity, electrons travel according to the hopping effect [6]. Investigations show that for the very low resistivity layer, chains without glass [6] are formed. The formula for the total resistance of such a chain is as follows:

$$R(T) = R_{\mathrm{M_IM}} + R_{\mathrm{m}} + R_{\mathrm{c}} \tag{11.20}$$

where

$$R_{\mathrm{M_IM}} = \frac{1}{2} R_{\mathrm{bo}} \frac{\sin aT}{aT} \left(1 + \exp \frac{\Delta E}{T}\right) \tag{11.21}$$

$$R_{\mathrm{m}} = R_{\mathrm{mo}}(1 + bT) \tag{11.22}$$

$$R_{\mathrm{c}} = \kappa_3 p_c (T_s - T)^{-\frac{1}{3}} \tag{11.23}$$

where

$R_{\mathrm{M_IM}}$ is the resistance connected by tunneling [4]
R_{bo} is the resistance in $T = 0$
a is the distance between particles
ΔE is the activation energy
k is the Boltzmann constant
R_{m} is the resistance of the conduction particle (the resistivity value is much higher than in bulk material because its diameter is very small (0.5 μm) and grain effect occurs)
R_{mo} is the resistance in $T = 0$ K (obtained by line extrapolation)

b is the TCR of the conducting particle
R_c is the resistance of contact between two particles without glass)
T_s is the temperature of glass melting
k_3 is the constant coefficient

For high resistance, the most important parameter is R_{M_iM}. In that range of resistance, TCR is negative and the surrounding glass layers give large voltage coefficients of resistance, which is shown in Figure 11.30. For low resistance, VCR is negligible, and TCR is positive, which means that R_m and R_c are important while an influence of R_{M_iM} may be neglected.

Equations 11.20 through 11.23 also explain the physical meaning of T_m for curve 3 in Figure 11.4. For the very high resistivity layer, both tunneling and hopping conduction occur. Parameters for this sort of resistor are not particularly good, but only in that technology can we get high-ohmic resistor in the range of 10 MΩ up to 1 GΩ.

11.1.3.5 Polymer Resistor

About 70% of all potentiometers are manufactured as polymer resistor. The resistive layer consists of conductive particle (black carbon, graphite, as well as metal powder) and thermosetting or thermoplastic resin. Also, a polymer layer is used for printing resistors on PC boards or for manufacturing chips in SMT. The layer is deposited by screen printing, painting, or by some other methods. Classes are 20, 15, 10, 5; TCR = −1000 or ±400 ppm/K; surface resistance 50–1 MΩ/square; the nominal resistance range is from 100 Ω to 10 MΩ. Thermoplastic polymer resistors have generated great interest [7] recently. At the softening temperature of a polymer, a strong increase of resistance is observed. TCR ∼ 100%/K; and alter cooling resistance retains a previous value. This phenomenon is applied to the multifuse resistor, whose characteristics are shown in Figure 11.18a and b.

A multifuse is needed for protection of the electronic power circuit against fire. Polymer layers are also used in keyboards as very hard and uncorrodible contact material. For example, that layer contains carbon and copper powder.

11.1.3.6 Comparison of Parameters of Different Resistors—Suggestion for Application

Figures 11.20 and 11.21 present average instability $\overline{\delta}$ of resistors made in four technologies: metal thin film, thick film, pyrolytic carbon, and polymer carbon. An analysis of these figures, as well as Figure 11.19, gives us a clue as to how to choose a resistor for a specific application. For example, we would like to find a resistor of a nominal resistance value of 1 MΩ to operate under high humidity conditions. A thick film resistor would be the best. Taking into consideration the price, a pyrolytic carbon resistor, class 10 or 5, should be chosen. In an operational amplifier application, a thin film resistor or thick film network would be the best choice, although for a small-signal preamplifer considering the small noise level, a thin film resistor would be preferred. Considering our decision, we should pay attention to the results of the following tests: TCR = φ(R), endurance test φ(R) for 1000 h, humidity test, 21 days: δ = φ(R).

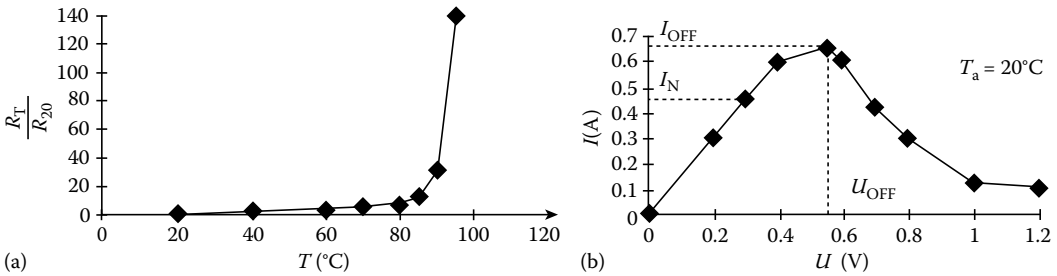


FIGURE 11.18 Characteristics of multifuse resistor. (a) Normalized resistance vs. ambient temperature and (b) current vs. voltage V_{off}—voltage above which current decreases.

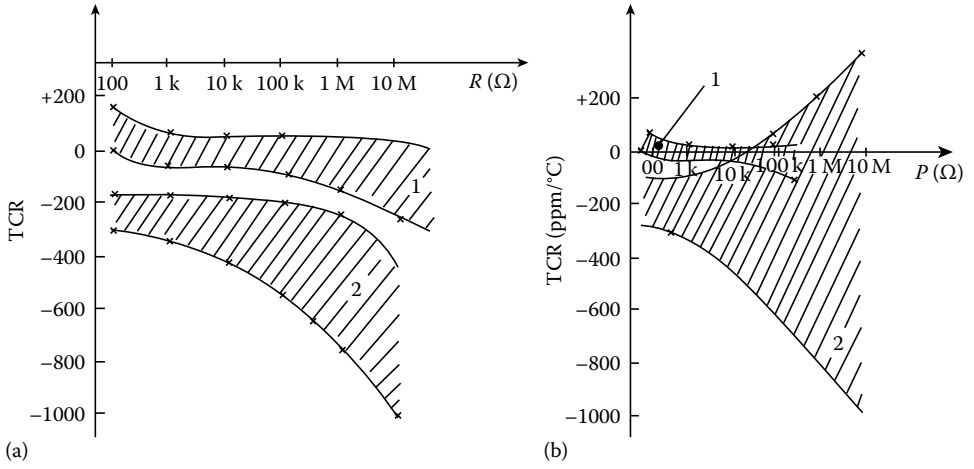


FIGURE 11.19 TCR dependence on nominal resistance value. (a) 1—thick film ruthenium-based resistors, 2—pyrolytic carbon resistors and (b) 1—thin film metal alloy resistors, 2—polymer carbon resistors. The envelope of the TCR range is calculated statistically at ±1.64 * s.

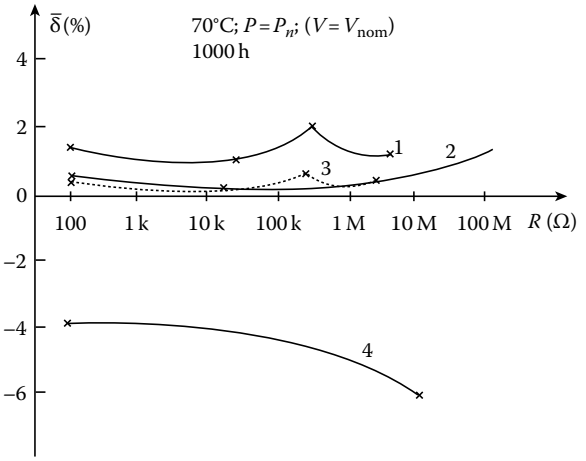


FIGURE 11.20 The average change $\bar{\delta}$ after 1000 h endurance test in relation to a nominal resistance value for different types of resistors. 1—pyrolytic carbon, 2—thick film ruthenium-based resistor, 3—thin film metal alloy, 4—polymer carbon. For the range 100 Ω–240 kΩ $P = P_n$, above 240 kΩ $P < P_n$ and $V = V_{max}$.

11.1.3.7 Influence of Ceramic Substrate on Parameters of Resistor

All resistive films are deposited on the ceramic substrate. Only polymer film can be put on the fenolic paper or on the epoxy resin substrate. Thick film, pyrolytic carbon, and metal alloy film are also deposited on the ceramic. Some of thin films may be deposited on the glass. It is observed that substrate affects the resistive layer in two ways as follows:

1. When ion current related to alkali metals in substrate flows. Its destructive effect is shown in Figure 11.22, where some pyrolytic carbon layers manufactured by the same technology were deposited on the three types of ceramic substrates. The densities of the ion current were 56, 7, and 0.5 μA/mm^2. The results of long-life stability changes for resistors deposited on various substrates differ greatly from each other. This effect is observed only at dc voltages and at high temperatures. The smaller thickness of a layer and the destructive effect of ion current are more remarkable.

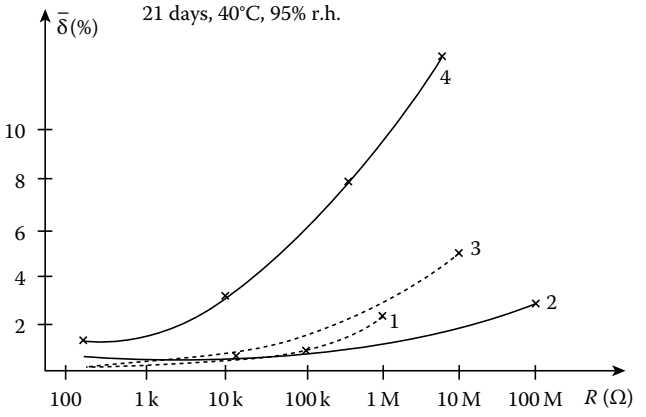


FIGURE 11.21 The average change δ after 21-day humidity test dependence on the nominal resistance value of different types resistors. 1—thin film metal alloy, 2—thick film ruthenium-based resistor, 3—pyrolytic carbon, 4—polymer carbon.

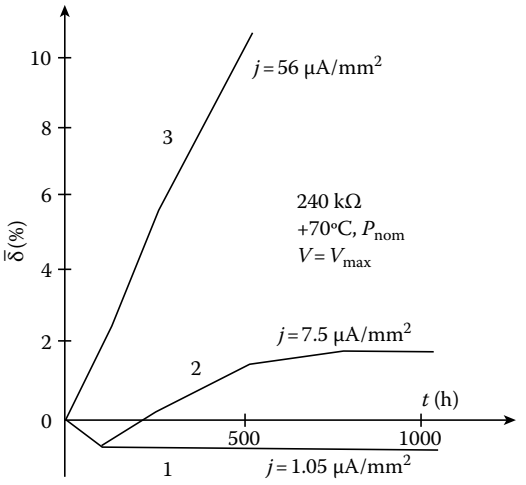


FIGURE 11.22 Influence of ion current density j on resistor stability $\bar{\delta}$ versus time during endurance test. $R = 240$ kΩ (critical nominal resistance value); j is measured according to GOST 10449-63, that is, 400°C and 400 V; distance of cut layer 0.5 mm.

During long-life tests at high temperatures, this phenomenon can also be observed for high-ohmic wirewound resistors [1].

2. When the thermal expansion coefficient of a layer and substrate are much different, thermal dilatability becomes a large problem and results in plus or minus changes of TCR [1]. Encapsulation of resistors with lacquer, cement, and transfer molding cover causes similar problems.

11.1.4 Dependence on Geometry of Material

The geometry of a resistive element affects either its high-frequency characteristic or its maximum temperature.

11.1.4.1 Influence of Resistive Element Shape on the Frequency Range

A wirebound resistor has many advantages but its inductance is very high. In Figure 11.23, different ways of resistive wire winding are presented in order to decrease parasitic inductance. A special wirewound resistor can work up to 200 kHz. For higher frequencies, a thin film resistor must be used. Though a film resistor can work at high frequencies up to 1 GHz, some limitations occur in this area (see Figure 11.24). For resistors of a low resistance value, that limitation is inductance of resistive elements and terminations. For resistors of a resistance above 10 kΩ, distributed capacitance is the main problem.

Figure 11.25 presents the part of thin film layer deposited on a cylindrical ceramic substrate. To get many more squares, cutting along the screw line is performed. Each step of the screw wind (Figure 11.27a) gives some elementary resistance R_r and elementary parasitic capacitance C_r. The maximum operating frequently f_m is given by Equation 11.24 as

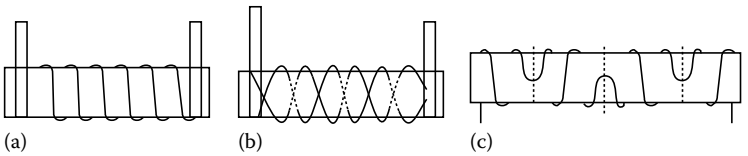


FIGURE 11.23 Different means of wire winding in wirewound resistors: (a) flat; (b) cross; and (c) bifilar.

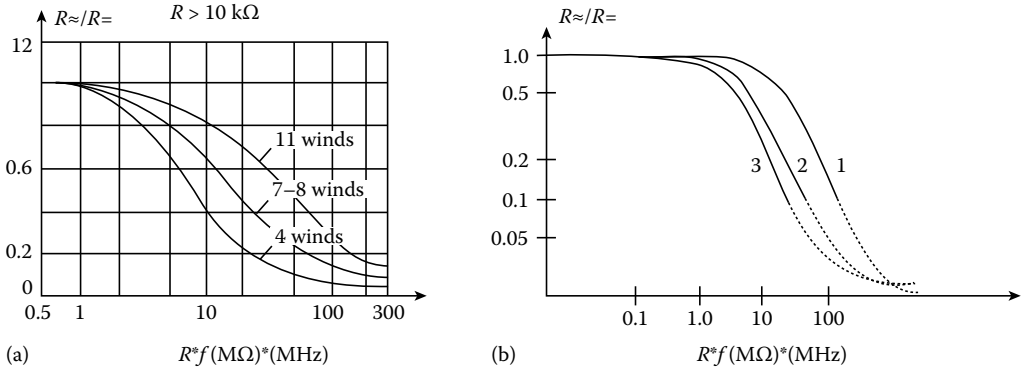


FIGURE 11.24 Dependence of $R_{\sim}/R_{=}$ on the product of nominal resistance value and frequency. (a) Pyrolytic carbon resistors, 0.25 W, cut according to screw line; different curves present resistors with different winds of cutting and (b) 1—thin film, 2—pyrolytic carbon, 3—thick film ruthenium-based resistor. (From Nowak, S., Wenta, A., and Kuzma, E., *Handbuch der Elektronik*, Vol. 2 and Ambroziak, A. et al., Vol. 7, Franzis-Verlag, Munich, 1979. With permission.)

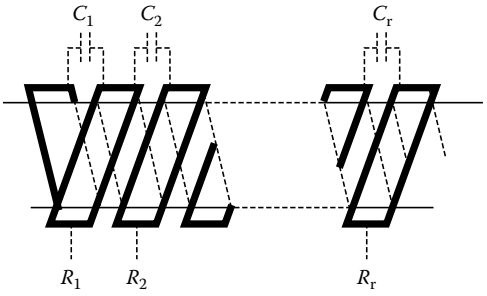


FIGURE 11.25 Part of a film resistor cut according to the screw line; R_r—elementary resistance, C_r—elementary capacitance.

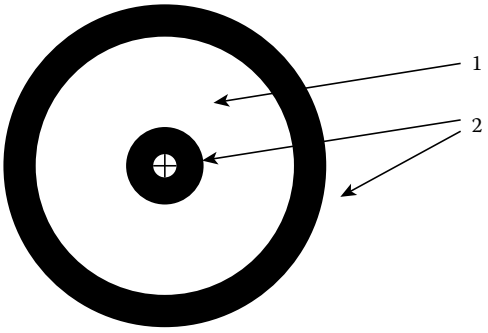


FIGURE 11.26 Coaxial resistor: 1—resistive layer, 2—electrode layer.

$$2\pi f_m = \frac{1}{R_r C_r} \tag{11.24}$$

where C_r is bigger when the thickness of the layer is greater and the groove becomes narrower, but also when ε_r of the insulating cover is high; R_r is smaller when Ω/square is lower.

This information suggests how to choose a resistor for high frequency. It is seen that a thin film with a low Ω/square is recommended. To reduce the termination capacitance for coaxial lines or cables, coaxial resistors are used (see Figure 11.26), which can work up to 20 GHz. For high-frequency applications,

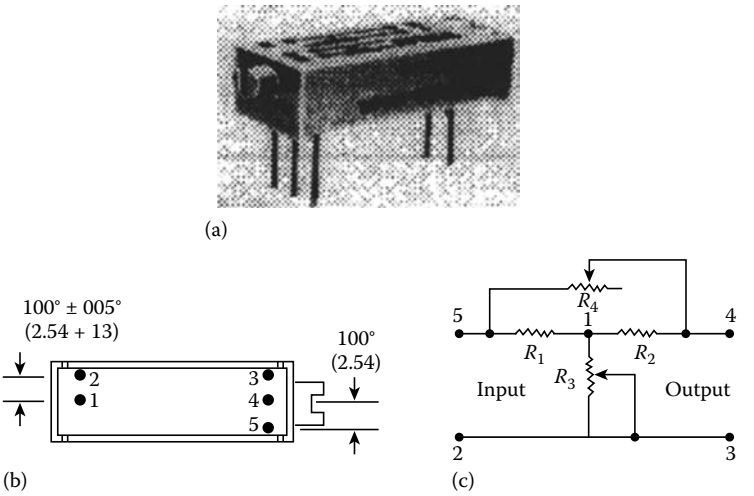


FIGURE 11.27 High-frequency attenuator: (a) overview; (b) pin localization; and (c) electrical circuit. (From *Philips Components Catalogue*, Eindhoven, 1989. With permission.)

special shapes of resistors have been designed (Figure 11.7). Strip resistors are connected to a microwave strip line. A special potentiometer of 75 Ω input and 75 Ω output works up to 1 GHz. This potentiometer has fluent regulation of the attenuation obtained by specific shapes of three resistive elements inside the potentiometer, which, in turn, are regulated by one appropriately shaped slider (a similar attenuator for 300 MHz is shown in Figure 11.27).

11.1.4.2 Sensitivity Improvement of Precision Potentiometer by Complication of Shape

In Figure 11.10, how a helipot works is explained. The wiper makes a rotation of 3600° and the precision is about 0.01%. The range of resistance is 100 Ω to 100 kΩ. The producers are Beckman, Bourns, and others.

11.1.4.3 Influence of Resistor Shape on Its Application

The typical requirement for resistor construction is to establish a uniform temperature on the entire surface of the resistive element. This problem does not appear in the designing of a fail-safe resistor. The fail-safe resistor is shown in Figure 11.28.

At the center of the resistive layer a narrow resistive path is placed [6]. Under normal working conditions, the resistance of the circuit is stable. When a defect occurs in the circuit, the current doubles or triples in path and breaks the fail-safe resistor. This is important in protection against fire and

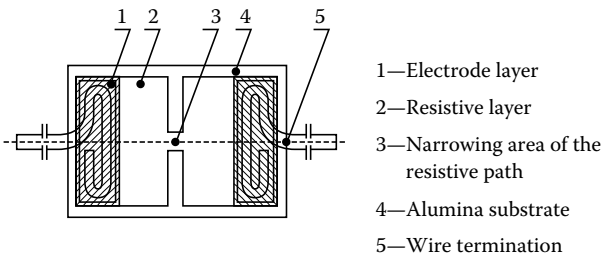


FIGURE 11.28 Fail-safe resistor.

avalanche devastation of the electronic set. The resistors utilizing the piezoresistive effect for pressure detection are also specially shaped.

11.1.5 Nonideal Linear Resistor

The main features of the nonideal resistor is its instability in time, which is described in the introduction to the section on resistors and instability related to temperature, presented in Sections 11.1.1.1 and 11.1.3.1. Catastrophic failure rate is estimated: for thin film resistors on the level of 10^{-9}/h, for thick film resistors on the level of 10^{-10}/h. Noise is also an important feature of the nonideal linear resistor.

11.1.5.1 Noise

Total noise is the sum of a number of factors, summarized as follows:

Thermal noise called Johnson noise or white noise: The value of thermal noise can be calculated from Equation 11.25:

$$V_{t\text{rms}} = \sqrt{4kRT\Delta f} \tag{11.25}$$

where

$V_{t\text{rms}}$ is the root-mean-square value of the noise voltage (V)
R is the resistance value (Ω)
k is the Boltzmann constant (1.38×10^{-23} J/K)
T is the temperature (K)
Δf is the frequency bandwidth (Hz) over which the noise energy is measured

The spectral density of thermal noise is constant in the total frequency bandwidth (white noise). Total noise is the sum of $V_{t\text{rms}}$ and current noise $V_{i\text{rms}}$:

$$V_{\text{rms}} = V_{t\text{rms}} + V_{i\text{rms}} \tag{11.26}$$

Current noise or structural noise: When dc voltage is applied to the resistor, dc current causes ac voltage fluctuation. That ac voltage fluctuation depends on the structure of the resistive element and an applied voltage. Density of that noise can be described by Equation 11.27:

$$\frac{V_{i\text{rms}}}{\Delta f} = cf^{-\gamma}V_{=}^{\alpha} \tag{11.27}$$

where

c, γ, and α are the constants
$V_{=}$ is the applied dc voltage
Δf is the frequency bandwidth

$\gamma = 0.98$–1.2; very often $\gamma = 1$ and this kind of noise is called "$1/f$ noise"; $\alpha = 1$–2 and depends on the structure of the resistive element.

$V_{i\text{rms}}$ is measured in μ V/V or in dB, where 0 dB = 1 μ V/V. For foil and wirewound resistors, $V_{i\text{rms}} = 0$ and this means that only thermal noise occurs there. Figure 11.29 illustrates simplified characteristics of current noise for different types of film resistors.

11.1.5.2 Voltage Coefficient of Resistance

A linear resistor can exhibit some deviations from Ohm's law (Equation 11.1). This nonlinearity, called the voltage coefficient of resistance (VCR), is measured in %/V and is calculated from Equation 11.28:

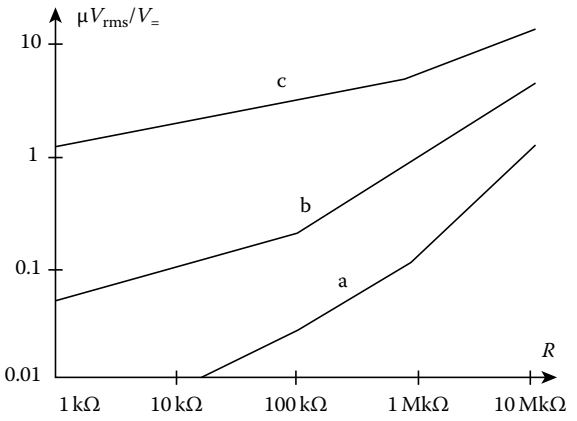


FIGURE 11.29 Simplified current noise characteristics versus nominal resistance value for different types of resistors: (a) thin film; (b) pyrolytic carbon; and (c) ruthenium-based thick film.

$$\text{VCR} = \frac{R_1 - R_2}{R_2(V_1 - V_2)} \times 100 \ [\%/V] \tag{11.28}$$

where

R_1 is the resistance at the rated voltage V_1

R_2 is the resistance at 10% of the rated voltage V_2

Metal alloy film resistors and pyrolytic resistors show a negligible small voltage coefficient but polymer resistors and thick film resistors have a remarkable one. In Figure 11.30, normalized R/R_{10} versus voltage stress for the ruthenium-based thick film resistors are presented, where R_{10} is resistance measured at stress 10 V/mm. Results are collected for low stress voltage in Figure 11.30a and for high stress voltage in Figure 11.30b.

It can be observed that for a resistor made of a low-resistivity ink up to 100 KΩ/square, VCR is small, but for inks of high resistance/square (e.g., 500 MΩ/square), VCR is large [6]. VCR depends on the ink producer, as well (see R310 in Figure 11.30b).

11.1.5.3 Rotational Noise and Contact Resistance Variation

When a potentiometer is supplied by dc voltage and its wiper is moving from the beginning to the end of resistive layer, some ac voltage appears on the output. It is noise, which is measured in $\text{mV/V}_{=}$. The IEC standard requires that for a quality potentiometer this noise has to be smaller than 2 $\text{mV/V}_{=}$. The CRV is important when the resistor works in series with the load or in a very sensitive instrument. This parameter is measured in percentage of total resistance (1% is a typical value). CRV and rotational noise decrease if a multipoint wiper is used in potentiometers, for example, a wiper made from 20 wires. That parameter is also important in the construction of a precision wirewound potentiometer where CRV must be smaller than 1%. The proper choice of materials for the slider and resistive wire is the best way to solve this problem.

11.1.5.4 Smoothness of Regulation Curve of Potentiometer

Exponential curve: In practice, an exponential resistive element consists of two or three linear segments of resistance. As a result, the line is not smooth but has some steps. Rotary noise also increases in that

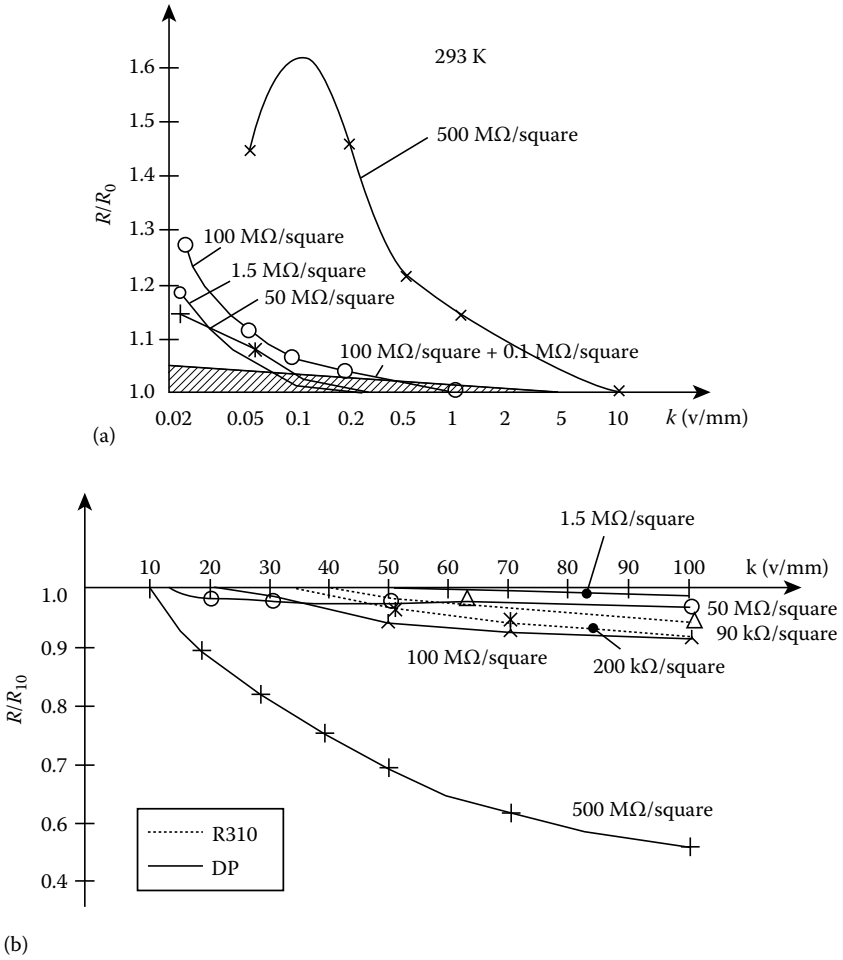


FIGURE 11.30 The change of resistance versus stress voltage for thick film resistors: (a) for low stress and (b) for high stress. (From Nowak, S., *Rozprawy Elektrotechniczne*, 35, 1989. With permission.)

area. More experienced producers use several segments and then the junction is not very sharp but is sawtooth shaped (see Figure 11.31).

Linear curve in a helipot precision potentiometer: In a precision potentiometer, the resistance of the resistive element should be proportional to the distance traveled by wiper.

In Figure 11.32, Δr_1 and Δr_2 are the maximum deviation from a single line. The nonproportionality NP of a precision potentiometer is described by Equation 11.29:

$$\mathrm{NP} = \frac{\Delta r_1 + \Delta r_2}{R_c} \times 100\% \tag{11.29}$$

The value of NP is 0.5%–0.01%. Such a good result is obtained by fluent control of the proportionality over winding of the wire.

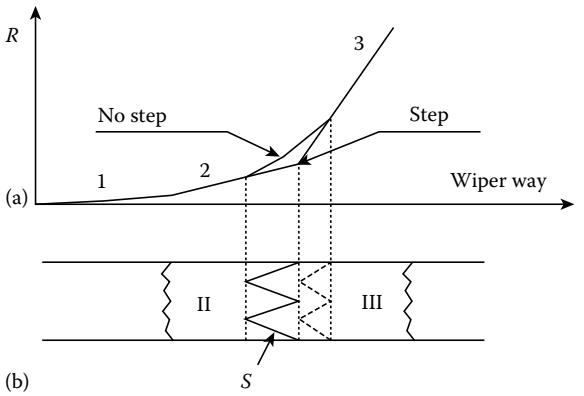


FIGURE 11.31 Shaping of exponential curve: (a) resistance versus wiper movement, the slopes 1, 2, 3 are proportional to resistivity of layer and $\rho_1 < \rho_2 < \rho_3$ and (b) junction of resistive layer area marked by *s* is the saw-tooth-shaped junction of segment II and segment III.

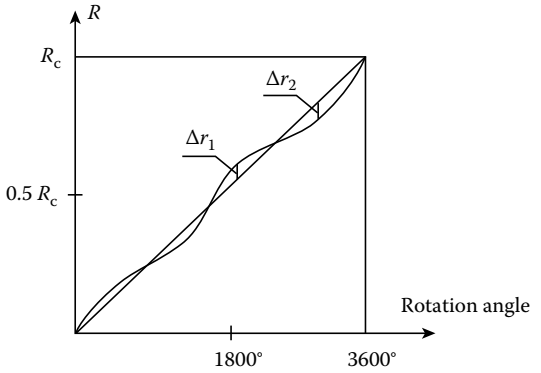


FIGURE 11.32 Resistance versus rotation angle (wiper movement) in precision linear potentiometer.

References

1. G. W. A. Dummer, *Fixed Resistors*, London: Pitman, 1967.
2. L. I. Maissel and R. Glang, Eds., *Handbook of Thin Film Technology*, New York: McGraw-Hill, 1970.
3. S. Nowak, A. Wenta, and E. Kuzma, *Handbuch der Elektronik*, vol. 2 and A. Ambroziak et al., vol. 7, Munich: Franzis-Verlag, 1979.
4. C. A. Neugebauer and M. B. Webb, Electrical conduction mechanism, in ultrathin evaporated metal films, *J. Appl. Phys*., 33, 74, 1962.
5. S. Nowak and D. Lusniak-Wojcicka, Thick film fail-safe resistors, *Electrocompon*, *Sci. Technol*., 10, 4, 255, 1983.
6. S. Nowak, Nonlinearity of thick film resistors, *Rozprawy Elektrotechniczne*, 35, 4, 1989.
7. *Philips Components Catalogue*, Philips Corporation, Inc., Eindhoven, The Netherlands, 1989.
8. Bourns, Inc., *The Potentiometer Handbook*, New York: McGraw-Hill, 1975.
9. Pike, G. E. and C. H. Seager, Electrical properties and conduction mechanism of Ru-based thick film (cermet) resistors, *J. Appl. Phys*, 48, 12, 5152–5168, Dec. 1977.
10. *Siemens Matsushita Components Catalogue*, Siemens AG, Munich, Germany, 1993.
11. R. W. Vest, Conduction mechanism in thick film microcircuits, Final Technical Report, Purdue Univ. Res. Foundat., Grant DAHC-15-70-67, DAHC-15-73-68, ARPA Order: 1642, December 1975.

11.2 Capacitor

11.2.1 Introduction

A capacitor is a container of the electric energy W. This is expressed by the following equation:

$$W = \frac{CV^2}{2} \tag{11.30}$$

where

C is the capacitance expressed in F
V is the voltage on the capacitor plates expressed in V

Capacitance C of the multiplate capacitor can be described by the following equation:

$$C = \frac{x\varepsilon_r(N-1)A}{d}10^{-13} \tag{11.31}$$

where

ε_r is the relative dielectric constant of the insulator
d is the distance between the plates
N is the number of plates

A is the plate area, where $x = 0.0885$ for A and d expressed in centimeters or $x = 0.225$ for A and d in inches.

The relative dielectric constant value ε_r is equal to the ratio of capacity of a capacitor with plates separated by a dielectric to one separated only by vacuum. Dielectric constant values of various materials are presented in Table 11.6.

Fundamental parameters of capacitor are as follows: capacitance C, nominal voltage V_N, testing voltage V_t (note that $V_t = (2\text{–}3)V_N$), temperature coefficient of capacitance (TCC, for class 1 only), insulation resistance R_i (for dc voltage), and power PF (for ac voltage).

TABLE 11.6 Comparison of Capacitor Dielectric Constants

Dielectric	ε_r (Dielectric Constant)
Air or vacuum	1.0
Paper	2.0–6.0
Plastic	2.1–6.0
Mineral oil	2.2–2.3
Silicone oil	2.7–2.8
Quartz	3.8–4.4
Glass	4.8–8.0
Porcelain	5.1–5.9
Mica	5.4–8.7
Aluminum oxide	8.4
Tantalum pentoxide	26
Ceramic	12–400,000

Source: Dorf, R. Ed., *The Electrical Engineering Handbook*, CRC Press, Boca Raton, FL, 1993. With permission.

Power factor (PF) is described by the following equation:

$$\mathrm{PF} = rC\omega = 2\pi frC \tag{11.32}$$

where r is an equivalent series resistance (ESR).

Its inversion is a quality factor Q, given by Equation 11.33. Because the power factor expresses total losses in capacitor, it is a sum of the dissipation factor and losses in electrodes as well as in terminations.

$$Q = \frac{1}{\mathrm{PF}} = \frac{1}{2\pi frC} \tag{11.33}$$

According to IEC Publ 384/1988, capacitors are divided into two salient groups: class 1 and class 2 but also more detailed classifications are commonly used because of the wide range of capacitances and very different applications. The main applications of capacitors include filtering, coupling, tuning, dc blocking, ac passing, bypassing, phase shifting, compensation, through feeding, isolation, energy storage, noise suppressing, motor starting, and so on. Contemporary capacitors cover the 0.1 pF up to 10 F capacity range and the 2.5 V to 100 kV voltage range. Connecting a dc voltage source to the capacitor plates we can observe the given capacitor is gradually charged and current flowing through the capacitor, large at the beginning, decreases in time to negligible small value. On the other hand, ac source causes the current I, given by Equation 11.34, flowing permanently through the capacitor.

$$I = \frac{V}{X_c} = \frac{V}{(1)/(2\pi fC)} = V2\pi fC \tag{11.34}$$

This current increases when capacitance, frequency, or applied voltage increase. The ac current can heat the capacitor, whose temperature would depend on its power factor, capacitor size, and cooling conditions. This phenomenon has to be taken into consideration in energetic 50 Hz equipment or in high-frequency power applications. Miniaturization and integration of electronic sets influence miniaturization of capacitors as well. The index υ', called "own volume" (volume per capacitance) and expressed in $\mathrm{cm}^3/\mu\mathrm{F}$, might be of use during selection of a capacitor for a given circuit. Nominal voltage V_N strongly affects the index υ' value. Table 11.7 presents the index υ' for capacitors with various dielectrics.

TABLE 11.7 υ' Index of Various Capacitors

Capacitor Definition	Main Parameters	υ' ($\mathrm{cm}^3/\mu\mathrm{F}$)
Variable air	500 pF/250 V	200,000
Mica	10 nF/500 V	250
Ceramic (rutile)	1000 pF/500 V	600
Ferroelectronic	40 nF/250 V	50
Ferroelectric multilayer	0.68 μF/50 V	1.5
Polystyrene	2 μF/160 V	300
Polyester (Mylar)	0.1 μF/160 V	12.4
Polycarbonate—metalized	0.15 μF/160 V	5.6
Electrolytic Al (HV)[a]	40 μF/350 V	1.3
Electrolytic Al (LV)[a]	120 μF/7 V	0.008
"Golden" capacitor	1 F/5.5 V	0.00001
Electrolytic Ta (wet)	10 μF/100 V	0.038
Electrolytic Ta (dry)	5.6 μF/10 V	0.0026

Source: Badian, L., *Handbuch der Electronik*., Franzis-Verlag, Munich, 1979. With permission.

[a] HV, High voltage; LV, low voltage.

From the user's point of view, capacitors can be divided as follows: linear and nonlinear, fixed capacitors, adjusting capacitors, power energetic capacitors, start motor capacitors, and interference suppression capacitors.

11.2.2 Linear Capacitor

The linearity of a capacitor depends on the polarity of the dielectric used for its manufacture. Several different polarization mechanisms can contribute to the total polarization. The most important are the following:

1. Electron polarity that exists in the insulator with covalent bonds between atoms; electrical stress causes deformation of the orbital shape but electrons cannot go out of orbit (relaxation time τ is smaller than 10^{-15} s).
2. Ion polarity that occurs in glass and high-quality ceramic; under electrical stress ion centers are displaced ($\tau < 10^{-13}$ s).
3. Dipole polarity that occurs in a polar polymer dielectric. Electric field causes rotation of dipoles in the dielectric. Generally dielectric constant ε_r depends on frequency, temperature, and voltage, but in limited ranges of these factor ε_r is stable.
4. Domain polarity that appears in some insulators, for example, ferrodielectrics. They contain domains that rotate with electric field. This effect is called ferroelectricity because of the analogy to ferromagnetism and gives rise to very high dielectric constant ε_r up to 400,000 (see Table 11.6). It strongly depends on voltage, frequency, and temperature. Ceramic capacitors with a domain polarity mechanism and high dielectric constant are very popular in electronic equipment.

Only capacitors with electron polarity and ion polarity are classified in linear capacitor group 1.

The linear capacitor group consists of fixed capacitors in class 1, adjusting capacitors, energetic capacitors, high-voltage capacitors, and interference suppression capacitors.

11.2.2.1 Fixed Capacitor—Class 1

The main feature of a fixed linear capacitor with class 1 dielectric materials is its stability in time and under temperature. By analogy to resistors we can introduce dynamic tolerance Δ (see Section 11.1.1.2) for capacitance. Production tolerances δp ($\pm$) for class 1 capacitors are $\pm 0.25\%$, $\pm 0.5\%$, $\pm 1\%$, $\pm 2\%$, $\pm 5\%$, $\pm 10\%$, and $\pm 20\%$. The instability δ_{max} expressed by Equation 11.35 after the endurance test is up to 3% (IEC Publ. 384–8).

$$\delta_{max} = \overline{\delta} + 1.64^{*}s \tag{11.35}$$

where $\overline{\delta}$ and s are in accordance with the formula (Equations 11.4 and 11.6), respectively, and

$$\delta_i = \frac{C_i(t) - C_i(0)}{C_i(0)} \times 100\% \tag{11.36}$$

δ_{max} for capacitor is larger than for a stable resistor. This also means that dynamic tolerance Δ for capacitors is larger than for resistors, which should be considered during active rc filter design.

The TCC, given in Table 11.8, describes temperature stability of capacitors. Negative TCC capacitor may be used to compensate positive temperature coefficient of inductance in resonant LC circuit.

The next important parameter of capacitors in class 1 is a power factor that is required to be smaller than 30×10^{-4}. Ceramic capacitors with low ε_r as well as styroflex and mica capacitors meet these conditions, which renders them very suitable for resonant circuits, stable analog filters, integrator circuits, and other circuits, where stable capacitance and small losses are necessary.

TABLE 11.8 TCC and Max Capacitance of Monolithic Ceramic Capacitor

		Maximum Capacitance (pF)				
Dimension $a \times b$ (mm)		NP 0	N 75	N 150	N 750	Ferrodielectric
	TCC ppm/K	0 ± 30	-75 ± 30	-150 ± 30	-750 ± 30	Large
4×4		47	47	47	150	10,000
8×8		680	680	680	1,600	470,000
10×10		4,700	4,700	4,700	6,800	1,000,000[a]

[a] Some producers offer maximum capacitance $C = 4.7$ μF at dimensions of 10×10 mm.

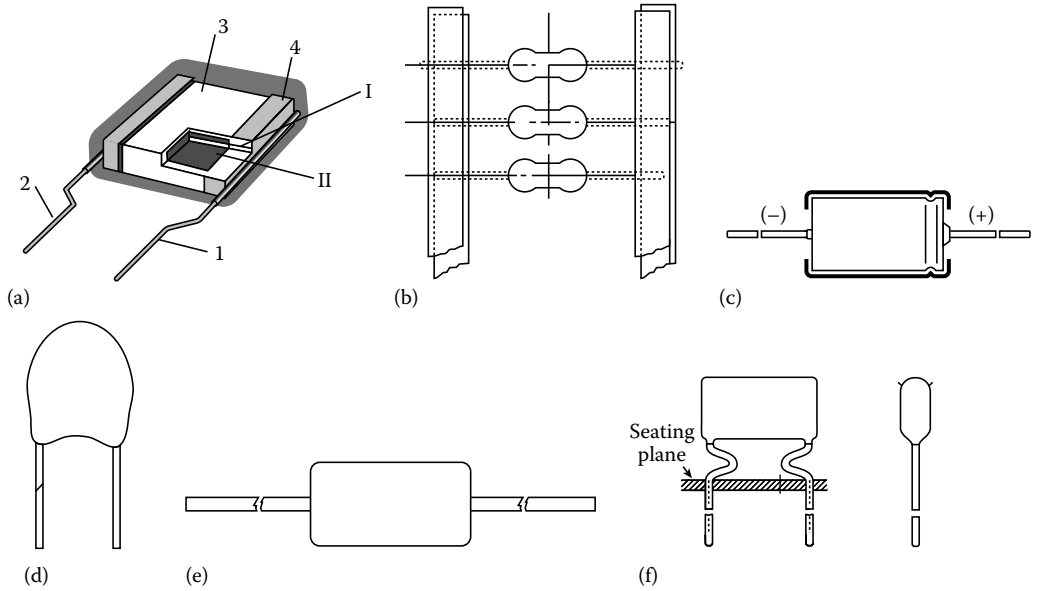


FIGURE 11.33 Typical fixed capacitors. (a) Multilayer ceramic (monolithic): 1,2—termination, 3—dielectric layer, 4—inner termination; I,II—plates (capacitor electrodes). (From Nowak, S., *Rozprawy Elektrotechniczne*, 4, 1989.) (b) Tubular taped (connected in tapes for automatic assembling). (From Taiyo Yuden Co. Ltd., *Tubular Ceramic Capacitors Catalogue*, Tokyo, 1988.) (c) Aluminum electrolytic (wet) with axial terminations. (d) Tantalum electrolytic (dry) with radial terminations. (e) Polystyrene foil with axial terminations. (f) Metalized film. (From Nowak, S., *Rozprawy Elektrotechniczne*, 4, 1989. With permission.)

Ceramic capacitors are produced as tubular, disk, and multilayer (monolithic) capacitors. Disk and tubular ones are inexpensive. Multilayer capacitors are rather expensive but they have small dimensions and low index υ'. Figure 11.33 presents some types of fixed capacitors.

11.2.2.2 Adjustable (Variable) Capacitor

An adjustable capacitor is an electronic component whose capacitance can be mechanically regulated by the user. For example, an AM radio set tuner is adjusted using a variable capacitor of 10 up to 500 pF. The dielectric used in this type of capacitor is either air or plastic foil. The majority of variable capacitors are trimmer capacitors for precision adjustment of reactance in electronic circuits. Their insulator layer is made of air, ceramic of class 1, mica, polystyrene, or Teflon. Typical trimmer capacitors are shown on Figure 11.34.

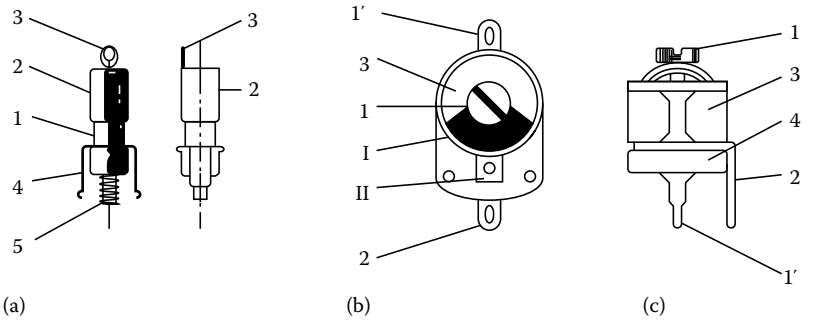


FIGURE 11.34 Typical trimmer capacitors: (a) tubular ceramic: 1—ceramic tube, 2—hot electrode, 3—soldering point, 4—earthed electrode, 5—screw for capacitance regulation; (b) flat ceramic: I—grounded electrode (plate), 1—roller for capacitance regulation (electrically connected with termination 1′); II—hot electrode connected with termination 2, 3—ceramic plate; and (c) Cylindrical air: 1—grounded roller (screw) connected with termination 1′, 2—hot electrode, 3—coaxial cylinders; capacitance regulation is achieved by moving cylinders up and down.

They have the following capacitance ranges: air trimmer, 1–15 pF; tubular ceramic trimmer, 0.1–7 pF; disk trimmer, 10–50 pF; special disk trimmer, 100 pF maximum. The power factor is small $(1\text{–}20) \times 10^{-4}$. Insulating resistance is above 10^{10} Ω. Nominal voltage is 100–500 V.

11.2.2.3 Energetic Power Capacitor

To compensate inductance in electrical engines and other equipment, energetic power capacitors are used. Besides capacitance, it is important to know their resistance power expressed in VAr (between vectors of voltage and of current, there is an angle 90°). They are rather large in size and weight of 10–50 kg per unit. The smaller ones are capacitors used to discharge lamps. They have a capacitance range 2.5–20 μF and ac working voltage of 150 V and effective voltage up to 550 V. Starting motor capacitors also belong to this group of power capacitor. The majority of power capacitors have paper, polypropylene, or mixed dielectrics, and they are impregnated in vacuum against flashover. An impregnant substance, such as mineral or synthetic oil is used. It is very important to match the dielectric constant of the oil with that of the fixed dielectric because it will guarantee uniform distribution of the electric field inside the unit. In some capacitors of smaller size, polypropylene without an impregnant is used as an insulator. Electrodes are made of aluminum foil or deposited with thin metal film on the dielectric. The power factor of these capacitors is required to be low with respect to heat dissipation. According to IEC Publ. 384-17 the maximum power factor is $<50 \times 10^{-4}$ at working frequency and for 1.25 V_N of nominal voltage.

Under overheating, the capacitor reliability falls and lifetime shortens. Polypropylene is chosen as the insulator material because of its good dielectric properties. For example, its PF = (6–7) $\times 10^{-4}$, $\varepsilon_r = 2.1$, TCC = −200 ppm/K and over the temperature range up to 110°C it is stable. The elementary cylindrical section of the power capacitor presented in Figure 11.35a depicts two polypropylene foils on which electrodes (Ia and IIa) were evaporated in vacuum. It is important to provide a fixed, multipoint junction between electrodes (Ia and Ib) and (IIa and IIb). It shortens the path of the current, which at ac voltage is high. That construction reduces the equivalent series inductance (ESI) of the capacitor at high frequencies.

11.2.2.4 High-Voltage Capacitor

A high-voltage capacitor for dc voltage and low frequencies is made of polymer foil, but for high-frequencies ceramic capacitors are preferred. They can be disk shaped up to 6 kV (Figure 11.36) or tube shaped for very high-voltage applications. It is easier to protect a tube-shaped capacitor against flashover.

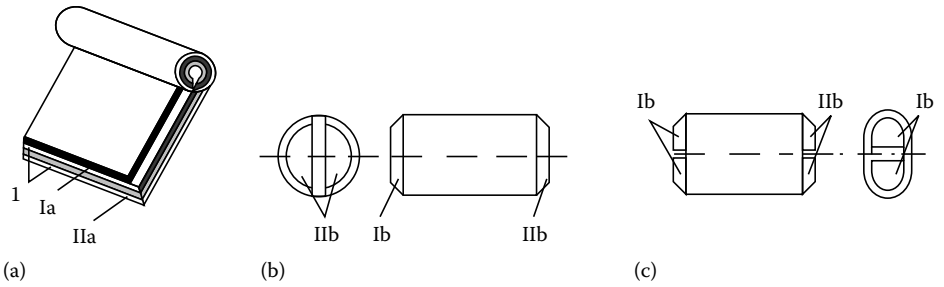


FIGURE 11.35 Metalized polypropylene capacitor. (a) The roll during winding; (b) cylindrical roll; and (c) flat roll: 1—polypropylene foils, Ia, IIa—metalized layer deposited in vacuum, plates (electrodes) of capacitor, Ib, IIb—contacts, deposited by high-pressure airbrush with melted metal.

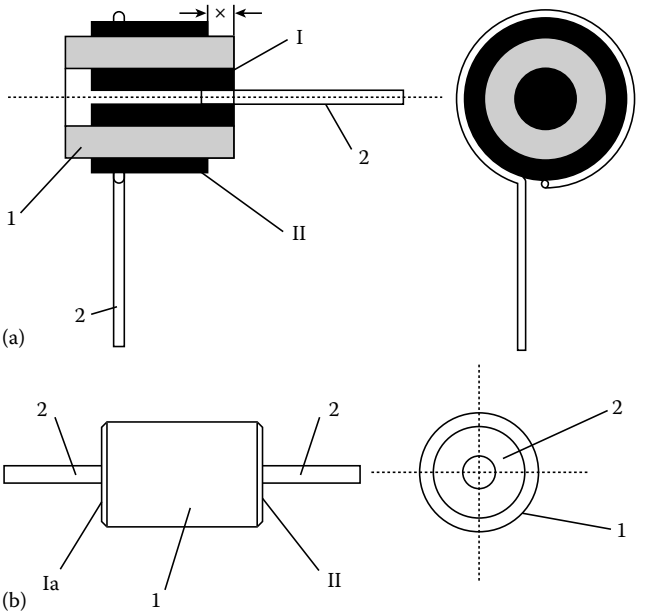


FIGURE 11.36 High-voltage ceramic capacitor. (a) Tubular: 1—inner, high-voltage electrode, II—grounded electrode, 1—ceramic tube, 2—wire terminations and (b) disc (bar) ceramic: I—ceramic bar, 2—wire termination, Ia, IIa—plates of capacitor, made by screen printing; x—distance for high voltage.

Some additional information will be presented in Section 11.2.5.3. To gain a high-voltage level, it is sometimes necessary to connect several capacitors in series. In this case, initially we must select an elementary capacitor. In order to get equal voltage drops on each element, we should choose elementary capacitors having the same capacitance for ac voltage and the same insulating resistance for dc voltage.

Dielectric absorption: According to Equation 11.30, high electrical energy can be stored in power and high-voltage capacitors. Danger to human life may exist even several hours after switching off the circuits that contain such capacitors. Also, capacitors discharging by short circuiting are not nearly safe enough. Because of dielectric absorption, some dangerous charge remains on the capacitor electrodes. To prevent accidents, high-energy capacitors should be shunted by high-ohmic resistors.

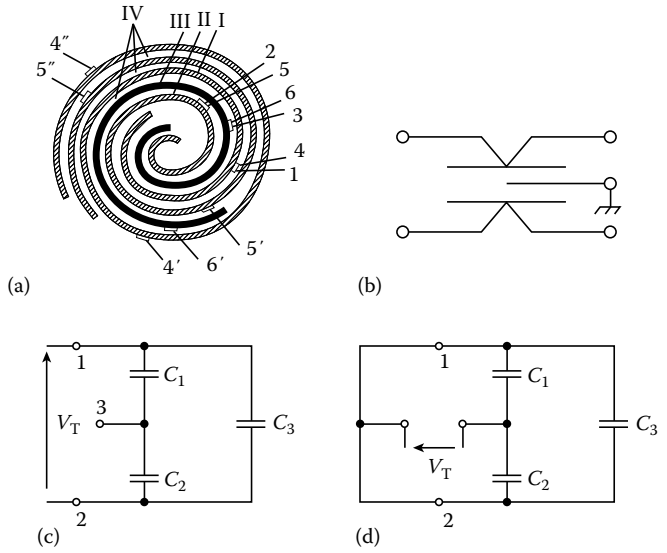


FIGURE 11.37 Capacitor for radio interference suppression: (a) construction principle of three capacitance interference suppression capacitor; I,II,III—Al foil electrode, IV—polymer foil insulator, 1,2,3—wire terminations, 1,2—wire leaded through, 4,5,6,4′,5′,6′,4δ,5δ—pretinned copper foil contact; (b) electrical circuit; (c) circuit for testing in class *X*; and (d) circuit for testing in class *Y*.

11.2.2.5 Interference Suppression Capacitor

This type of capacitor is characterized by low inductance. To accomplish this, a specific construction has been developed, where two or three capacitors are joined in a special way and put into one encapsulation. Figure 11.37 presents an example of this construction. Interference suppression capacitors are divided into two classes: *X* or *Y*. Capacitors of class *X* fulfill lower safety requirements because their breakdown is not dangerous to human life. Capacitors of class *Y* are used in extremely dangerous environments, where, for example, breakdown causes a short circuit between the body of the equipment and an energetic phase of 50 Hz. Properties of these two types of capacitors vary. For example, the 4 μF/250 V capacitor that belongs to the *X* class, has a test voltage on the level of 1100 V; however, one of the *Y* class has to withstand 2250 V. Also, circuits for testing these two types of capacitors (according to IEC Publ. 38414, 1993) differ from each other (see Figure 11.37c and d).

In Figure 11.37b, a lead-through capacitor is shown. The first capacitor is positioned between the two leads. The second and the third leads are arranged between separate leads and the ground. For radio frequencies, polymer capacitors may be used but for very high frequencies, ceramic ones are preferred. A special kind of feed-through ceramic capacitor is shown in Figure 11.38. It should be noted that automotive capacitors have to fulfill some additional requirements to withstand ignition system pulses.

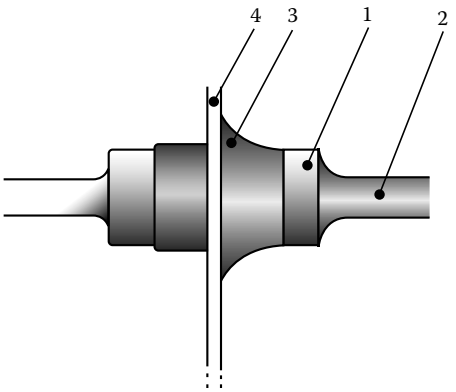


FIGURE 11.38 Feed-through capacitor: 1—ceramic tube, 2—wire termination lead through, connected with the inner electrode of the capacitor, 3—external electrode, 4—metal plate. Connection between 3 and 4 is made by soldering.

11.2.3 Nonlinear Capacitor

As was described in Section 11.2.2.1, the dipole and domain polarity mechanism in a dielectric is related to the nonlinearity of the capacitors. Ferrodielectric capacitors and polar polymer capacitors are nonlinear capacitors.

11.2.3.1 Ferrodielectric Capacitor

Ferrodielectric ceramic capacitors are known for strong dependence of their capacitance on temperature, voltage, and frequency. They are constructed in tubular, disk, and monolithic multilayer versions. Each version fulfills requirements of class 2 ceramics. By mixing different kinds of ceramics, we can get a dielectric composition of ε_r ranging from 1000 up to 30,000. The maximum available ε_r is 400,000 (see Table 11.6). Nevertheless, it should be noted that for production scale, such a dielectric constant is not reached because of its poor stability. Figure 11.39 shows the effect of capacitance instability $\Delta C/C$ versus voltage stress for different ε_r values.

If the 25% of change δ in capacitance is required, the maximum bias dc is 2000 V/mm (for $\varepsilon_r = 2000$); for stronger stress the change will be much greater. Great valve of ε_r, low series inductance, and small dimensions make ferrodielectric capacitors very useful in high-frequency applications up to 1 GHz. We can divide the class 2 ceramics into five subclasses: 2B, 2C, 2D, 2E, and 2F. Maximal changes of capacitance in the maximum category temperature are presented in Table 11.9. This means that the ferromagnetic capacitor must have minimal capacitance but its stability is not as important. Figure 11.40 presents the dependence of the relative change $\Delta\varepsilon/\varepsilon_r$ [3] on temperature for various dielectric constants. Higher values of ε_r show larger inequality of curves. Changes are larger when voltage stress is smaller than what is depicted in Figure 11.40c.

Figure 11.41 illustrates the influence of frequency on capacitance and power factor. It can be noted that up to 1 GHz the changes are acceptable.

Production tolerance δ_p also differs from values in class 1, for example, −40, +80%; −20, +80%; −20, +50%; ±20, ±10%.

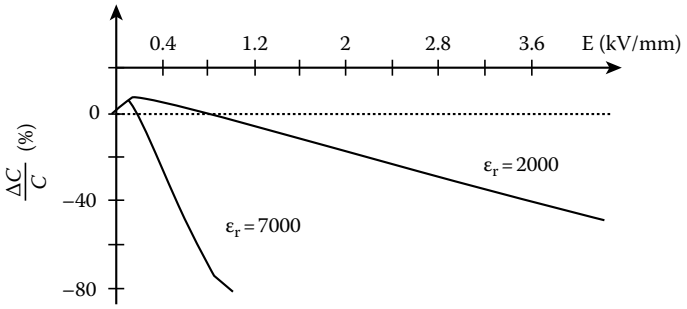


FIGURE 11.39 Dependence of capacitance on stress voltage at different ε_r of ceramic capacitor of class 2.

TABLE 11.9 Maximal Changes of Capacitance in the Maximum Category Temperature

Subclass	Maximal Changes of C_n[%]
2B	+10, −15
2C	+20, −30
2D	+20, −40
2E	+22, −70
2F	+30, −90

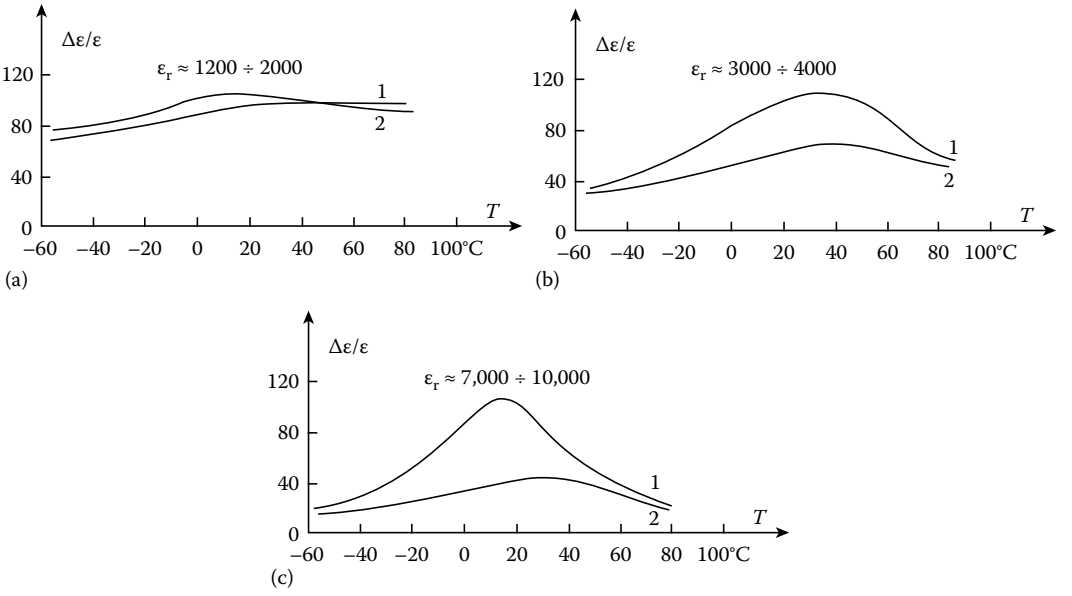


FIGURE 11.40 Normalized change of dielectric constant $\Delta\varepsilon/\varepsilon_r$ vs. temperature for class 2 ceramic. (a) Dielectric constant $\varepsilon_r = 1200–2000$; (b) dielectric constant $\varepsilon_r = 3000–4500$; and (c) dielectric constant $\varepsilon_r = 7{,}000–10{,}000$th; curve 1 at very low voltage; curve 2 at bias 8 kV/mm. (From Nowak, S. et al. *Handbuch der Electronik*, Vols. 2 and 3, Franzis-Verlag, Munich, 1979. With permission.)

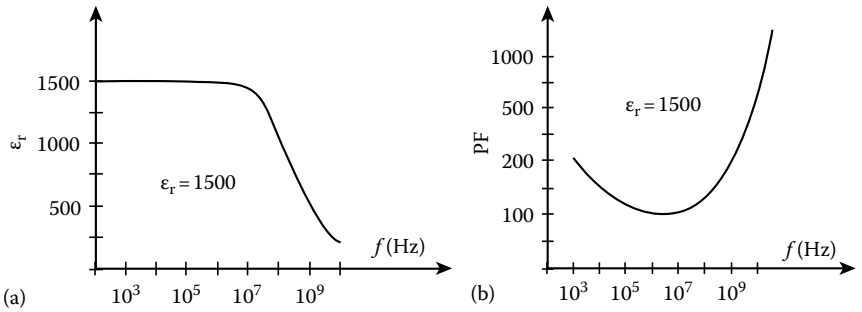


FIGURE 11.41 Dependence of dielectric constant and power factor on frequency. (a) ε_r vs. frequency and (b) PF vs. frequency. (From Nowak, S. et al. *Handbuch der Electronik*, vols. 2 and 3, Franzis-Verlag, Munich, 1979. With permission.)

Series of nominal voltage are 25, 40, 63, 100, 160, 250(200), 400(500), 630, 1000, and 1600 V. The maximal power factor is 350×10^{-4}; after an endurance test, it can increase up to 500×10^{-4} or 700×10^{-4}.

11.2.3.2 Polar Polymer Dielectric Capacitor

Miniaturization of a capacitor is so important in the design of electronic circuits that often a capacitor less stable but smaller in size may be chosen. The polar polymer capacitor exhibits such a feature; as an example, a metalized polycarbonate foil (macrofol) capacitor is used. Its construction is similar to the polypropylene capacitor shown in Figure 11.35. Polycarbonate foil of 3 μm thick is selectively covered by a thin (≤0.5 μm) layer of Al in the PVD process. An aluminum layer (marked by Ia and IIa) functions as the electrode of the capacitor. Very often a flat coil is used (see Figure 11.35c) in capacitor construction; contacts Ib and IIb are deposited by a high pressure airbrush with melted metal; terminations are connected to the contacts by soldering; encapsulation is made by epoxy cover. It is a class 2 capacitor; the voltage series range is 63, 100, 160, and 250 V (for different polycarbonate foil thicknesses).

Capacitances in series E-6 and E-12 range from 4.7 nF up to 10 μF. The power factor is $\leq 30 \times 10^{-4}$ and decreases with increasing temperature. The maximal frequency is about 1 MHz; at frequencies above 100 kHz, the power factor increases up to 200×10^{-4} and the capacitance decreases several percent. The climatic categorization is −55/+100/56; the time constant is $R_i \geq 5000$ s. It operates at both dc and ac voltages (e.g., dc voltage of 630 V corresponds to ac voltage of 220 V_{rms}).

11.2.4 Dependence on Material Properties

Dielectrics have played a fundamental role in capacitor performance (see Tables 11.6 and 11.7). Air capacitors, mineral oil capacitors, and mica capacitors, as well as polystyrene and Teflon capacitors mentioned above, are ranked as class 1 capacitors. Capacitors with natural dielectrics (air, oil, mica) are primarily older types. Perspective dielectrics for capacitors are polymer, ceramics type I and type II, thin film oxide (electrolytic), and electric double layer.

11.2.4.1 Nonorganic Capacitor

Nonorganic insulating materials are often used as dielectrics in capacitors. Class 1 and class 2 ceramics were described earlier. A large group of capacitors is based on oxide thin film. Commonly used electrolytic capacitors are aluminum oxide and tantalum oxide capacitors. Although the ε_r of an oxide is not high, oxide film thickness can be very small, so capacitance per 1 cm^3 is large. Oxide film is made by an electrochemical process. Popular oxides Al_2O_3 and Ta_2O_5 are utilized in aluminum electrolyte and tantalum electrolyte capacitor fabrication. The positive electrode is a metal (Al or Ta) and the negative electrode is a conductive electrolyte. Figure 11.42a depicts a segment of a wet aluminum foil electrolytic capacitor.

Foil is etched to get a large active surface; the growth coefficient of the surface is above 10 and theoretically for low voltage is 100. On the active surface, a thin layer of Al_2O_3 is produced electrochemically. It functions as a dielectric layer of the capacitor and its ε_r equals 8.4. The negative electrode is made as a conductive flux that usually is connected to the Al cover. The dc voltage range is from 6 up to 600 V. The capacitance range is from 4.7 μF to 10 mF in series E-3. In the reverse direction, the dielectric does not withstand high voltages and the maximal admitted voltage is only 10% of nominal voltage. During operation, the capacitors show better parameters than during storage because of the smaller value of the leakage current. A tantalum capacitor is also produced as a foil capacitor, but the dry tantalum capacitor is more popular. Figure 11.42b shows a segment of that capacitor where tantalum balls (0.3 mm diameter) are sintered and function as the positive electrode. A thin film of Ta_2O_5 is produced by an electrochemical process on the surface of each ball. In the next step, a conductive layer (e.g., a colloidal graphite layer) is deposited on Ta_2O_5 film in a chemical process. At the end, a contact for a negative electrode is made. An anode termination is welded to sintered balls and a cathode termination can be soldered to the negative electrode contact. Encapsulation is

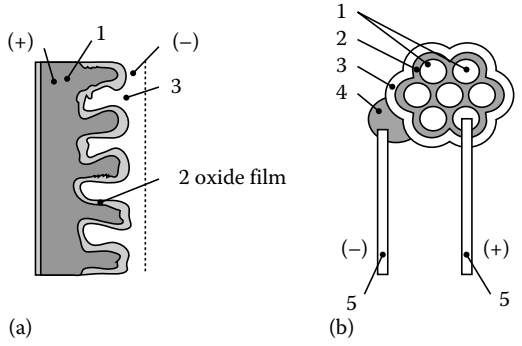


FIGURE 11.42 Segment of electrolytic capacitor. (a) Al_2O_3 wet capacitor 1—Al foil, 2—oxide layer, 3—fluid (negative electrode) and (b) Ta_2O_5 dry capacitor 1—tantalum balls, 2—Ta_2O_5 layer, 3—conductive layer (negative electrode), 4—contact for negative termination.

made using a tixotropic lacquer. Tantalum oxide electrolytic dry capacitors are produced for low dc voltages from 4 to 50 V. The tantalum capacitor shows a smaller leakage current and higher work temperature, up to 125°C. The aluminum capacitor can usually work nominally up to 70°C and up to 85°C maximum. Also, the tantalum capacitor exhibits better high-frequency characteristics. It is really a miniature capacitor; its υ' index is very small (see Table 11.7). Al and Ta electrolytic capacitors are applied to do circuits but can also work at small ac voltage (ac amplitude must be smaller than 10%–15% of dc nominal values). In a wet electrolytic capacitor, self-regeneration exists; that is, after a short breakdown, the capacitor can work further because the layer of oxide is regenerated around the breakdown point. Several firms manufacture bipolar electrolytic capacitors for ac voltages.

11.2.4.2 Super Capacitor (with Electric Double Layer)

The super capacitor, also called the golden capacitor or golden series capacitor, has capacitances up to 10 F and its υ' index is about 0.00001 $cm^3/\mu F$. The origin of that capacitor comes from Helmholtz (1879), who discovered the electric double layer that exists on the interface of two different materials. The layer can store an electric charge. The charge value of the electric double layer increases as the effective contact surface and/or electric field go higher. The right choice of both materials and preparation technology can lead to a very thin (several angstroms) electric double layer for 1.2 V in one call. Cells are connected in series for higher nominal voltage values, which typically achieve 2.5 up to 11 V for the whole super capacitor. In known practical constructions of super capacitors, small particles of activated carbon (with a large effective surface) are used in contact with diluted sulfuric acid; carbon in contact with acid yields the electric double layer. Carbon is the positive pole and acid connected with the metal case is the negative pole. The leakage current of supercap is very small, the time constant of the supercap is very high (over 1 month), and therefore it is very useful for backup purposes, such as maintaining the proper voltage level for reliable data storage of CMOS memories during host power supply failure.

Supercapacitors are sometimes classified as electrolytic capacitors because of their inner wet solution, but the general principles of operation are quite different. Supercaps are manufactured by many companies, such as NEC and Panasonic, and others.

11.2.4.3 Organic Capacitor

This group includes paper and polymer capacitors. The paper capacitor is an obsolete construction and today it is not very popular because paper drying and impregnation are rather expensive. Also, the parameters of paper capacitors are not sufficient. The family of polymer capacitors is very large and some of them were described in Section 11.2.3.

Table 11.10 presents the features of various polymer capacitors. Special attention should be paid to the Teflon capacitor, for instance, that the maximum operating temperature is 280°C and the power factor is ca. 6×10^{-4}; also, instability δ is $<0.5\%/1000$ h.

For dc and low frequency operation, polyethylenetereftalate (Mylar) foil capacitor is very popular. It withstands different fluxes and high temperatures up to 150°C. Its stability and power factor are quite good, but only for low frequencies.

11.2.5 Dependence on Geometry of Material

11.2.5.1 Influence of Capacitor Construction on Equivalent Series Inductance and Equivalent Series Resistance

Equivalent series inductance (ESI) with capacitance C causes self-resonance at resonant frequency f_0. Above frequency f_0, it has an inductance feature so it cannot fulfill its main function. So, reduction of ESI is a very important matter. Figure 11.43 shows the simplest rolling polymer capacitor with two terminations marked 3 and 4.

During charging and discharging, current is flowing through capacitor electrodes that are marked 2 and 5. The electrodes are insulated by polymer foils marked by 6 in Figure 11.43. Such a construction

TABLE 11.10 Features of Various Polymer Capacitors

Capacitor Name	υ' $cm^3/\mu F$	Class	δ_{max} after 1000 h (%)	Smallest $t\delta_p$ (%)	Power Factor $\times 10^{-4}$	TCC (ppm/K)	Maximum Work Temperature (°C)	Remarks	
Polystyrene	300	1	0.5	±0.5	2–5	−100	70	For telecommunications filter	Neutral polymer
Teflon	300	1	0.5	±0.5	6	−150	280	Special applications	
Polyethylene	200	1	1	±1	5	−500	100		
Polypropylene	50	2	5	±5	6–8	−200	110		
Metalized polypropylene	10	2	5	±0.5	6–8	−200	85		
Metalized polyester	5.6	2	10	±10	50 (200 at 1 MHz)	—	—	For ac pulse	Polar polymer
Polyester (polyethylene tereftalate)	12	2	5	±10	50 (200 at 1 MHz)	Large	150		
Polycarbonate	12	2	10	±10	20	Large	100		
Metalized polycarbonate	5.6	2	10	±10	20	Large	100		

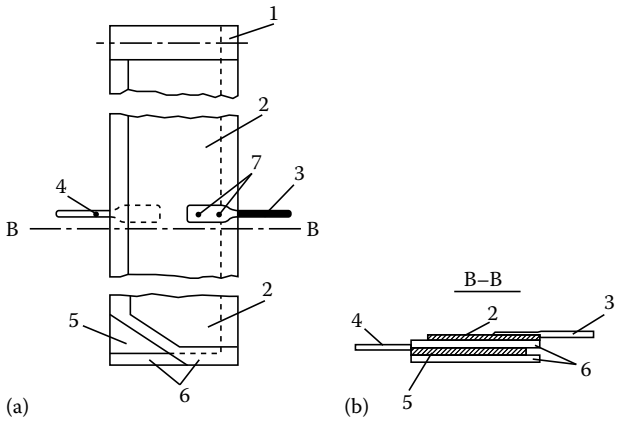


FIGURE 11.43 Simply polymer capacitor construction. (a) 1—start part of roll, 2—Al foil (electrode I), 3,4—wire terminations, 5—Al foil (electrode II), 6—polymer dielectric foil, 7—welding points and (b) cross section B–B.

creates a roll which, in turn, adds significant ESI to the capacitor. The simplest way of decreasing ESI is to supply some additional terminations and connect them in parallel. The maximum number of contacts is shown in Figure 11.35. Contacts are deposited with melted metal on the electrode by using a high-pressure airbrush. In that case, minimal ESI occurs. In this way, equivalent series resistance (ESR) also decreases, because the distance through which current is flowing becomes much shorter. Very small ESI occurs with the feed-through capacitor shown in Figure 11.38.

11.2.5.2 Influence of Plate Shape on Tuning Frequency in Air Variable Capacitors

A variable capacitor consists of standing plates that are insulated usually from the ground and rotary plates that are connected to the ground.

Variable capacitors are used for tuning an LC circuit, for example, to select the proper radio station. As we see from Equation 11.37, resonant frequency f_0 is not proportional to C. Proportionality is achieved by special shaping of the plates as a function of the rotation angle. The modified shape of plate is illustrated in Figure 11.44.

$$f_0 = \frac{1}{2\pi\sqrt{LC}} = \frac{b}{\sqrt{C}} \tag{11.37}$$

Equation 11.37 shows the resonant frequency of the LC circuit.

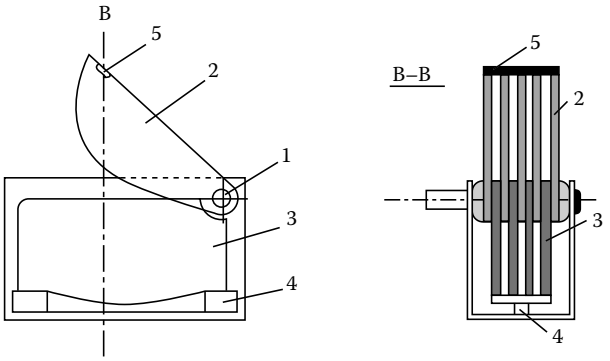


FIGURE 11.44 Rotary capacitor construction. Shape of plates is modified in relation to the circle: 1—shaft, 2—moving plates, 3—standing plates, 4—insulator, 5—insulating traverse.

11.2.5.3 Shape of the High-Voltage Capacitor

The operation of a capacitor at high voltages is always accompanied by many problems, for example, how to achieve the proper distance between capacitor electrodes.

Two drawings in Figure 11.36 help us to overcome this obstacle by using either of the following:

1. A disk capacitor with a high dielectric thickness (actually, it is a bar instead of a disk); when the thickness of the dielectric increases, capacitance decreases.
2. A tubular capacitor when the thickness of the dielectric is satisfactory for high voltage, but the electrodes are far enough from each other with respect to flashover protection.

11.2.5.4 Chip Capacitor

SMT requires electronic components with easily solderable contacts without wire terminations.

Usually those contacts are pretinned during chip production. Figure 11.45a and b shows us a tantalum electrolytic capacitor chip and a ceramic monolythic capacitor chip, respectively. The capacitance of each of these two devices can range from 10 pF up to 10 mF and from 10 pF to 10 nF, respectively, in classes 2 and 1.

11.2.6 Nonideal Linear Capacitor

The main source of capacitance instability of the capacitor is the operating frequency. Its influence on dielectric polarity was described in Section 11.2.2. The change of capacitance with frequency increase in capacitors belonging to the class 1 might be also observed. Furthermore, an ESR decreases the effective capacitance C_{ef} according to the following equation:

$$C_{ef1} = \frac{C}{1 + \omega^2 r^2 C^2} \tag{11.38}$$

where $r = \text{ESR}$ is calculated from the power factor (PF), which is measured at the operating frequency, and C is the capacitance at low frequencies.

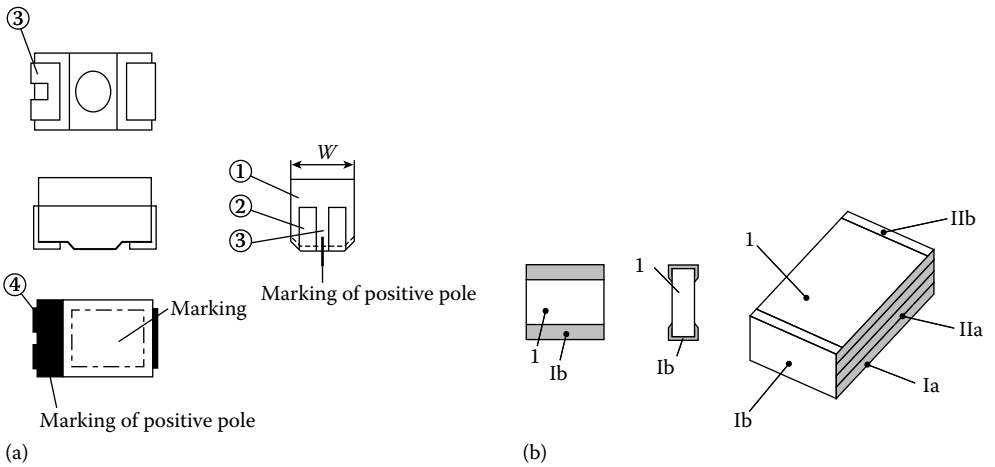


FIGURE 11.45 Chip capacitor. (a) Tantalum pentoxide capacitor. Positive pole is marked by a black bar and double external electrode; 1—plastic encapsulation, 2—external electrode for soldering, 3—cutting in positive electrode, 4—black bar and (b) ceramic multilayer monolithic capacitor, 1—ceramic layers, Ia,IIa—capacitor electrodes (plates), Ib,IIb—external electrodes.

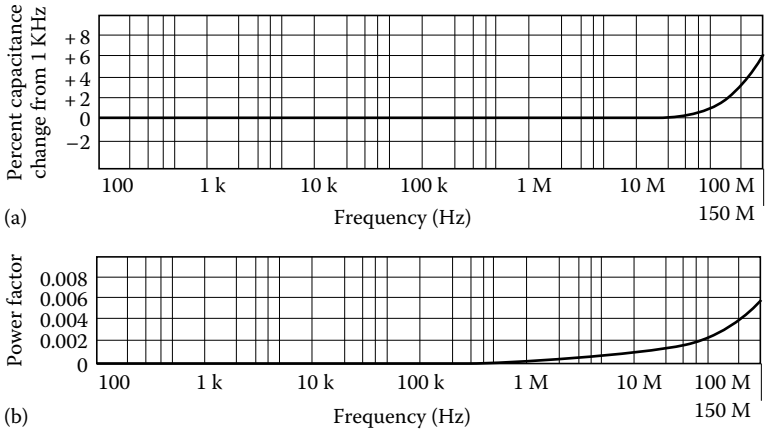


FIGURE 11.46 Dependence of capacitance and power factor vs. frequency for ceramic NPO capacitor—class 1: (a) capacitance and (b) power factor. (From Sprague Electric Co., Monolithic ceramic chip capacitors, *Sprague Eng. Bull.*, Brussels. With permission.)

$$r = \frac{\mathrm{PF}}{\omega C} \tag{11.39}$$

On the other hand, an ESI increases the effective capacitance according to the following Equation 11.40:

$$C_{\mathrm{ef2}} = \frac{C}{1 - \omega^2 \mathrm{LC}} \tag{11.40}$$

where ESI $= L$ can be measured at the self-resonant frequency (see Equation 11.37).

The ESR and ESI Data Cards for a given capacitor can be obtained from its manufacturer. Figure 11.46 shows the dependence of capacitance and power factor on frequency for class 1 ceramic capacitors.

Figure 11.41 presents data for class 2 ceramic capacitors. In a ceramic multilayer capacitor, migration of metal particles through the dielectric is observed, which causes an increase of capacitance and sometimes breakdown. The problem can be solved by roll pressing of two wet ceramic layers into one dielectric during a production process.

Self-regeneration of capacitor: The phenomenon of self-regeneration is exhibited in metalized dielectric capacitors as well as electrolyte capacitors. It is useful in some situations (e.g., in long-life tests), but sometimes it is harmful (e.g., in pulse counting equipment). To solve that problem, we should use a derated capacitor (for 25 V work voltage, 40 V nominal voltage is needed) or change the type of capacitor.

Leakage current of electrolytic capacitor: The insulating resistance R_{i} in an electrolytic capacitor is not as high as in other capacitors. The result of law R_{i} is dc current called leakage current I_{L}. For aluminum electrolytic capacitors, Siemens Matsushita Components Company in their catalog [6] proposes to join I_{L} in μA with capacitance C in μF and working voltage V_{w} in V. According empirical rule (Equation 11.41)

$$I_{\mathrm{L}} \leq 0.03\, CV_{\mathrm{W}} + 20\ \mu\mathrm{A} \tag{11.41}$$

for normal types and

$$I_{\mathrm{L}} \leq 0.006\, CV_{\mathrm{W}} + 4\ \mu\mathrm{A} \tag{11.42}$$

for special long-life capacitors, where $V_{\mathrm{w}} \leq V_{\mathrm{N}}$.

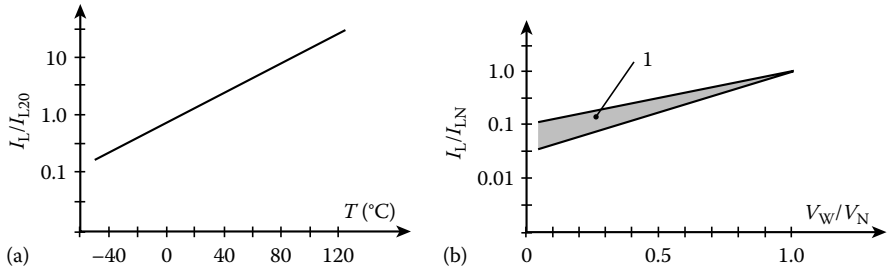


FIGURE 11.47 Leakage current of tantalum electrolytic capacitor. (a) Dependence of leakage current versus temperature and (b) dependence of leakage current vs. working voltage; 1—typical range of I_L. (From AVX Corporation Ltd., *AVX Tantalum Capacitors Catalogue*, Great Britain, 1990. With permission.)

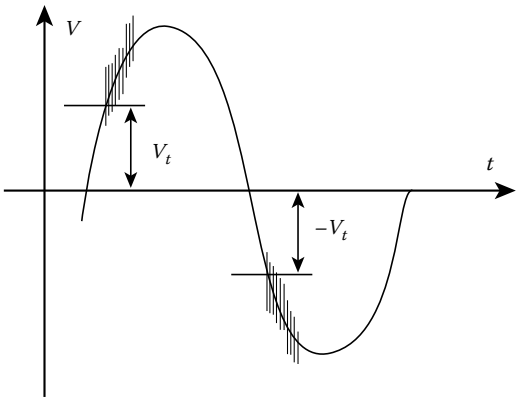


FIGURE 11.48 Ionization threshold in a 50 Hz capacitor. V_t—threshold voltage on which ionization can start.

Tantalum electrolytic capacitors show smaller leakage currents and at a temperature 20°C their value is 0.5 up to 20 μA. That value also depends on C, V_w, and temperature. Dependence I_L on ambient temperature T is shown in Figure 11.47a, and influence of working voltage value V_w is shown in Figure 11.47b. Those data are taken from the 1990 catalogue of the AVX Corporation Ltd. (Great Britain).

It is necessary to know that after long-time storage, the leakage current is going up. To prevent this, one should switch on the testing device containing the capacitors several hours before using.

Noise in capacitor: Foil capacitors can generate noise when contact between termination and electrode plate is not sufficiently fixed. In Figure 11.43, the welding points of contact are introduced (marked 7) to improve the quality of a contact. Some producers do not use welding because at nominal voltages the noise generated by contact is negligible. You should pay attention to the fact that for very low signals poor contacts affect noise in a significant way. Welding removes that effect.

Ionization threshold: In a power capacitor, an ionization of the dielectric is possible; Figure 11.48 shows the effect of this process.

Although breakdown does not occur, the capacitor life shortens when operating voltage approaches maximum.

Threshold is a minimal value V_T of voltage at which ionization starts. This causes heterogeneity of the dielectric, for example, polypropylene and air between plates. Flat rolling of the dielectric layer is strongly suggested for use in a power capacitor, as well as vacuum impregnation of the capacitor.

Insulating resistance R_i: When a capacitor is used as a component of an RC time constant, insulation resistance should be taken into consideration. In that case, foil polymers (particularly polystyrene or Teflon capacitors, where R_i is greater than 100 GΩ) are preferred. Also, protection against dirt and humidity by hermetization improves R_i.

References

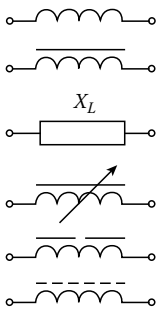
1. AVX Corporation Ltd., *AVX Tantalum Capacitors Catalogue*, Great Britain, 1990.
2. R. Dorf, Ed., *The Electrical Engineering Handbook,* Boca Raton, FL: CRC Press, 1993.
3. L. Badian, *Handbuch der Electronik*, Vol. 3, Munich: Franzis-Verlag, 1979.
4. *IEC Standard Publ.* 384, pp. 1–17, 1988.
5. Capacitors, *Philips Components Catalogue*, Eindhoven, 1989.
6. Aluminium electrolytic capacitors, ceramic capacitors, tantalum capacitors, *Siemens Matsushita Components Catalogue*, Munich, 1993.
7. Sprague Electric Co., Monolithic ceramic chip capacitors, *Sprague Eng. Bull.*, Brussels.
8. Taiyo Yuden Co. Ltd., *Tubular Ceramic Capacitors Catalogue*, Tokyo, 1988.

11.3 Inductor

11.3.1 Basics

An inductor is a device consisting of one or more associate windings, with or without a magnetic core, for introducing inductance into an electric circuit [1]. The origin of the word is from Latin: inducere–excite, induce, incite. Other appellations in use include inductance coil and coil, which are synonyms, and inductance, winding, bobbin, self, and self-inductance, which are jargon, as well as reactor and choke. Reactor and choke are inductors but not every inductor is a reactor or choke. The alphabetical symbol of inductor is *L*. Units of inductance include 1 henry = 1 H = 1 Vs/A and its submultiples nH (10^{-9}), μH (10^{-6}), and mH (10^{-3}).

Inductor graphical symbols:



Generally inductor or ideal inductor;

Inductor with magnetic core, cored inductor;

Inductive reactance $X_L = \omega L$;

Cored inductor with continuously changed inductance; cored inductor with adjuster;

Cored inductor—core with air gap;

Cored inductor, core of bonded magnetic powder.

Working frequency range of inductors (as lumped devices): from acoustic frequencies to roughly 1 GHz.

11.3.1.1 Basic Relationships and Influencing Factors

The main electrical attribute of the inductor is the inductance *L*. The inductance is the property of the electric circuit, whereby an electromotive force is induced in that circuit by a change of current in the circuit [1].

The electromotive force (voltage) *e* induced in the winding of the inductor is given by Faraday's law, which can be written in various equivalent forms:

$$e = -n\frac{\mathrm{d}\Phi}{\mathrm{d}t} = -\frac{d}{\mathrm{d}t}(Li) = -\left(L\frac{\mathrm{d}i}{\mathrm{d}t} + i\frac{\mathrm{d}L}{\mathrm{d}t}\right) \tag{11.43}$$

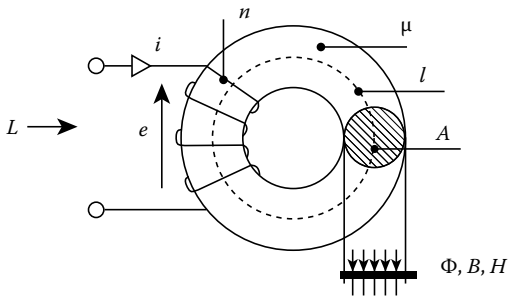


FIGURE 11.49 Inductor of a toroidal shape.

where $n\Phi =$ total magnetic flux linked with the current i flowing through all the n turns forming the winding of inductor having an inductance L.

Equation 11.43 shows that the circuit "senses" the inductance only when the current in the circuit is changing and/or the inductance itself is changing. Generally, the inductance L can depend on the magnitude of current i and inversely a change in the magnitude of L can cause a change in amplitude of i, so e is an implicit function $e[i(t), L(i, t), t]$. The well-known expression $e = -L\mathrm{d}i/\mathrm{d}t$ is fulfilled only if L is a constant independent of current, i, and time t (this restriction is often forgotten).

In terms of the electromagnetic field theory, the inductance L is given by

$$L = \frac{\Phi}{i} = n \oint_A B\mathrm{d}A \Big/ \oint_l H\mathrm{d}l \tag{11.44}$$

where

B is the normal component of the magnetic induction (flux density) through the area A
H is the magnetic field strength along its path l

For more details about various forms of the electromagnetic field equations see courses on the theory of electromagnetic field, for example, Ref. [2].

In case of a radially thin toroid (i.e., in which $A/l \to 0$) and $\mu =$ const, the magnetic induction B, and field strength H, can be assumed as uniform, that is, $\Phi = BA$; $B = \mu_0\mu H$; $H = ni/l$, the inductance L for a shape shown in Figure 11.49 is

$$L = \mu_0\mu n^2 A/l \tag{11.45}$$

where

μ_0 is the $4\pi 10^{-7}$ (H/m) is the magnetic constant
μ is the magnetic permeability of the medium filling up the shape of the toroid

Equation 11.45 can be expressed in different equivalent forms; their use is optional.

$$L = A_L n^2, \qquad A_L = \mu_0\mu\ A/l, \quad A_L = L/n^2 \tag{11.46a}$$

$$L = \mu A_{L0} n^2, \qquad A_{L0} = \mu A/l, \qquad A_{L0} = A_L/\mu \tag{11.46b}$$

$$L = \mu_0 G_m n^2, \qquad G_m = \mu A/l, \qquad G_m = \mu G_{m0} \tag{11.46c}$$

$$L = \mu_0 n^2 R_m, \qquad R_m = l/\mu A, \qquad R_m = 1/G_m \tag{11.46d}$$

$$L = \mu_0\mu G_{m0} n^2, \quad G_{m0} = A/l, \qquad G_{m0} = G_m/\mu \tag{11.46e}$$

$$L = \mu_0\mu n^2/R_{m0}, \quad R_{m0} = l/A, \qquad R_{m0} = 1/G_{m0} \tag{11.46f}$$

where

A_L is the inductance factor of inductor
A_{L0} is the inductance factor of the winding or core shape (at $\mu = 1$)
G_m and R_m are the core permeance and reluctance, respectively
G_{m0} and R_{m0} are the permeance and reluctance of core shape (at $\mu = 1$) inductance L is in H if l is in m and A in m^2

If μ, *B*, and *H* cannot be assumed to be uniform, Equations 11.45 and 11.46 take more complicated forms issuing from Equation 11.44.

The following fundamental observations can be inferred from Equations 11.43 through 11.45:

1. The inductance *L* of the inductor depends on three factors:
   - Square of number of turns n^2 of the winding.
   - Geometrical configuration of the magnetic flux Φ produced by the current *i;* this configuration is represented by the permeance or reluctance of the shape G_{m0} or R_{m0}, containing the magnetic flux.
   - Magnetic properties of the medium or media filling up the shape associated with the carrying current winding, producing a magnetic flux Φ; these properties are represented by the magnetic permeability μ. *Note:* Magnetic permeability, depending on the approach, can be considered as a scalar (number), complex quantity, or a tensor.
2. The voltage induced across the inductor of inductance *L* depends on the following:
   - Rate of change of the current flowing in the inductor winding $\mathrm{d}i/\mathrm{d}t$.
   - Changes in time of the inductance of inductor $\mathrm{d}L/\mathrm{d}t$*;* these changes result from the influence of external factors on inductance *L,* because

$$\frac{\mathrm{d}L}{\mathrm{d}t} = \sum_{\nu} \frac{\partial L}{\partial \xi_\nu} \frac{\partial \xi_\nu}{\partial t} = \sum_{\nu} \left[\frac{\partial L}{\partial \mu} \frac{\partial \mu}{\partial \xi_\nu} \frac{\partial \xi_\nu}{\partial t} + \frac{\partial L}{\partial A} \frac{\partial A}{\partial \xi_\nu} \frac{\partial \xi_\nu}{\partial t} + \frac{\partial L}{\partial !} \frac{\partial l}{\partial \xi_\nu} \frac{\partial \xi_\nu}{\partial t} \right] \tag{11.47}$$

$\nu = 1, 2, 3, \ldots$, ξ = external stressing factors such as time, temperature, mechanical stresses, static and time-varying voltage or magnetic induction, current or magnetic field strength, frequency, moisture, radiation (nuclear, cosmic), and so on.

The term $\mathrm{d}L/\mathrm{d}t$ (Equation 11.47), often forgotten in considerations, is however, of importance because it is responsible for any instability or variability introduced into an electric circuit by the inductors. Examples of various $\mathrm{d}L/\mathrm{d}t$ contributions are listed in Table 11.11.

11.3.1.2 Inductor: Qualifiers and Attributes

Different qualifiers are, in practice, applied to inductors, depending on the approach and related attributes. The most frequently used are listed in Table 11.12. These qualifiers are used separately or in combination, for example, air inductor or linear cored inductor.

TABLE 11.11 Exemplification of Various Stressing Factors ξ Causing Changes in Inductance *L*

	Influencing	
Stressing Factor ξ	Permeability μ, $\mathrm{d}\mu/\mathrm{d}\xi$, for Example	Physical Dimensions *A* and *l* of Winding and/or Core; $\mathrm{d}A/\mathrm{d}\xi$ and $\mathrm{d}l/\mathrm{d}\xi$, for Example
Temperature	Temperature changes of permeability	Thermal changes of dimensions
Mechanical tensile, contracting or torsion stress	Reversible changes of permeability under mechanical stresses, piezomagnetic effects, mechanomagnetic after-effects (relaxations)	Changes of dimensions under mechanical stresses; mechanical relaxations of dimensions of core or winding
Magnetic induction *B* or magnetic field strength *H* (changes of static, time-varying or combined magnitude)	Strong and, generally, nonlinear dependence of μ on *B* and *H*; disturbance of the magnetic state of core material	Ponderometric effects in winding at, for example, high current intensities or pulses; mechanical relaxation of winding or coil formers

TABLE 11.11 (continued) Exemplification of Various Stressing Factors ξ Causing Changes in Inductance *L*

	Influencing	
Stressing Factor ξ	Permeability μ, dμ/dξ, for Example	Physical Dimensions *A* and *l* of Winding and/or Core; d*A*/dξ and d*l*/dξ, for Example
Frequency	Strong dependence of μ on frequency: eddy-current effects (magnetic skin and proximity effects), magnetic resonances (wall resonance, dimensional resonance, spin resonance), relaxation effects	Skin and proximity effects in wires cause change of current density distribution in conductor, thus, the change of effective dimensions of coil
Humidity, moisture, aggressive agents (salts, acids, etc.)	Corrosion effects in metallic cores, therefore structural changes of magnetic material; changes of dielectric properties in ferrites	Various agents' effects as changes of dimensions and electrical and dielectric properties of coil formers, winding and interwinding insulation layers, substrates, etc.
Irradiation	Structural changes in magnetic material	Structural changes in materials of coils, conductors, insulation, dielectric properties of coil formers, substrates, etc.
Time	Changes of μ: reversible effects (disaccommodation and accommodation), irreversible effects (so-called aging), thermal fluctuation effects in nonneutral states	For example, aging and rheological effects affecting materials of coil formers, insulation; diffusion effects between metallic conductors deposited, for example, on ceramic substrate

TABLE 11.12 Inductor Qualifiers and Attributes

Inductor Qualifier	Inductor: Attribute or Quality
Ideal, perfect	Linear inductor having only a ''pure'' inductance, that is, no power loss is related to the flow of time-varying current through the inductor winding. In the ideal inductor, the current of sine wave lags the induced voltage by angle $\varphi = 90^\circ$ ($\pi/2$ rad). The concept of the ideal inductor is used only in idealized or simplified circuit analysis.
Nonideal	Usually a linear inductor in which the power loss in the winding and core is taken into account. The current of sine wave lags the induced voltage by angle $0^\circ \leq \varphi < 90^\circ$ (90° for ideal, power loss-free inductor; 0° for pure resistor). The concept of nonideal inductor is used as a first order approximation of a real inductor.
Linear	Inductor, ideal or nonideal, for which the induced voltage drop across it is proportional to the flowing time-varying current in its steady state. Linear inductor can be described or be used to describe the circuit in terms of transfer function. An air inductor is an example of linear inductor.
Nonlinear	Inductor for which the induced voltage drop is not proportional to the time-varying current flowing by it. As a rule, cored inductors (specifically if a core forms a closed magnetic circuit) are nonlinear. This is a consequence of the strong nonlinear dependence of magnetic induction *B*, proportional to voltage $u = \mathrm{d}L/\mathrm{d}t$, on magnetic field strength *H*, proportional to current *i*.
Real	Inductor with electrically behavioral aspects and characteristics that are all taken into account, for example, magnetic power loss, magnetic flux leakage, self-winding and interwinding capacitances and related dielectric power loss, radiation power loss, parasitic couplings, and so on, and dependences of these factors on frequency, induction, temperature, time, etc.
Air	Inductor not containing magnetic materials as constituents or in its magnetically perceptible vicinity.
Cored	Inductor in which a magnetic material in the form of a core serves intentionally as a path, complete or partial, for guidance of magnetic flux generated by current flowing through inductor winding.
Lumped or discrete	Inductor assumed to be concentrated at a single point.
Distributed	Inductor with inductance and other properties that are distributed over a physical distance(s) which is (are) comparable to a wavelength.

11.3.1.3 Basic Functions of the Inductor

The inductor has to perform four basic functions:

- Impede (oppose) any change in the existing current; as the change is more rapid the opposition is stronger.
- Lag (to shift) the current in respect to induced voltage up to 90° for a sine wave.
- Differentiate the current waveform.
- Store the magnetic energy; this when in combination with electrical energy stored in capacitor, results in electrical resonance (LC—resonance).

The more the real inductor approaches the ideal inductor, the higher the degree in performing these functions.

11.3.1.4 Parameters Characterizing the Inductor

Primary parameters:

- Inductance L (for all excitation levels)
- Power loss at low and middle excitation levels expressed facultatively in terms of
  Resistance R = power loss/(square or rms value of current)
  Tangent of loss angle δ, $\text{tg}\delta = R/2\pi fL$
  Quality factor $Q = 2\pi fL/R = 1/\text{tg}\delta$
- Power loss at high excitation levels is expressed directly in watts or in watts per unit of weight or unit of volume

Secondary Parameters (Selected):

- dc winding resistance
- Self-capacitance
- Self-resonance frequency
- Temperature coefficient or factor of inductance
- Time instability
- Hysteresis loss
- Pulse inductance
- Adjustment range
- Harmonic distortion
- Magnetic radiation
- Influence of the static magnetic field
- Inductance rise factor
- Maximum winding temperature
- Temperature rise
- Immunity from mechanical stresses and environmental exposures
- Acoustic noise

To determine these parameters, the following should be specified: frequency, excitation level (voltage or current), temperature, parameters of measuring coil (winding), type of magnetic conditioning to be used, and other details of measuring procedure, as well as conditions needed to ensure the required accuracy and repeatability of measurement.

The relevant measuring methods are described, for example, in Refs. [3–5] while their methodological basics are summarized in the item Magnetic Materials Measurements: Basic Methodological Principles. See also item International Standards on Inductors, point 4. Measuring Methods and Procedures.

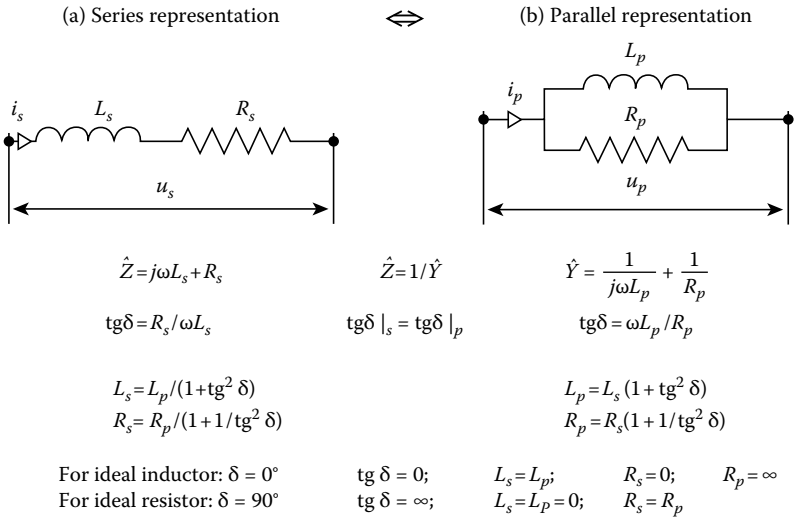


FIGURE 11.50 Equivalent circuit representation of the inductor and the respective expressions. *Note:* When subscript *s* is omitted, symbols *L* and *R* always refer to the series representation.

11.3.1.5 Circuit Representation of Inductor

Circuit representation of the ideal inductor is that as shown by the first inductor graphical symbol. For a nonideal linear inductor, a series or parallel representation is used (Figure 11.50).

The transformation of a nonideal inductor shown in Figure 11.50 is a purely formal operation serving only for the circuit analysis. Transformation results calculated for one specified frequency and specific operating conditions (e.g., *B*, temperature) can be extended over other frequencies and conditions only when *L* and *R* appear to be independent of these factors: $L = \text{const}$ and $R = \text{const}$.

Series and parallel connections of inductors: When individual inductors are connected in series (with no mutual inductive couplings), the total inductance $L_t = L_{ts}$ will be a sum of individual inductances L_i:$L_{ts} = \Sigma_i L_i$; similarly the equivalent loss resistances: $R_{ts} = \Sigma_i R_i$.

When individual inductors are connected in parallel, the total inductance $L_t = L_{tp}$ and resistance R_{tp} will be $L_{tp} = 1/\Sigma_i 1/L_i$ and $R_{tp} = 1/\Sigma_i 1/R_i$, respectively.

Thus, the series connection of inductors always increases a resultant inductance, parallel connection decreases. *Note:* These connections rules (laws) apply only for sine waveforms in the steady state and these rules are applicable neither to the nonlinear inductors nor to transient states.

11.3.2 Air Inductor

This type of inductor is assumed to be linear. This means that at stressing factors that are fixed, the inductance *L* and resistance *R* are characteristic constants, independent of the flowing current or applied voltage.

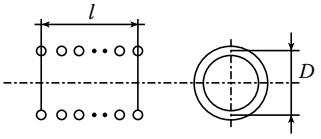
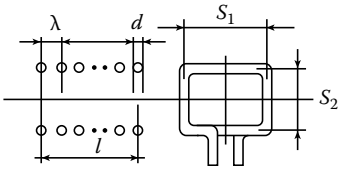
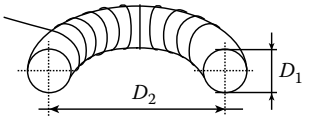
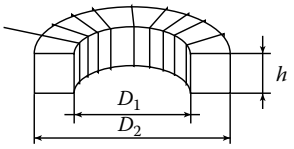
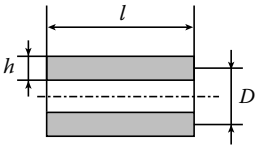
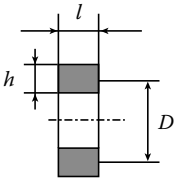
Selected formulas for the inductance of some air inductors are given below. In Table 11.13 corresponding draftings are depicted. All dimensions of coils shown in Table 11.13 are in cm; $n = \text{number of turns}$. For more formulas for calculation of the inductance of air inductors see, for example, Refs. [14,15].

- Single-layer cylindrical coil with a circular conductor, Table 11.13a:

$$L_0 = KDn^2 \times 10^{-9}\ \text{(H)} \tag{11.48a}$$

where $K = \text{coefficient}$ whose values are listed in Appendix 11.A.

TABLE 11.13 Inductance L_0 of Various Air Inductors Dimensionally Similar but Having the Same Number of Turns

	Winding (Coil) Dimensions	When Coil Dimensions Are (cm)	Inductance L_0 for $n = 100$ Turns Is (μH)
(a)		$D_1 = 2$ $l = 10$	19
(b)		$s_1 = 1.5$ $s_2 = 2.5$ λ = 0.05 $l = 10$ $d = 0.05$ $G(0.6;4) = 0.4$ $H(1;100) = 0$	32
(c)		$D_1 = 1$ $D_2 = 3$ $D_1 = 2$ $D_2 = 3$	10.3 41
(d)		$D_1 = 2$ $D_2 = 4$ $h = 1$ $D_1 = 1$ $D_2 = 5$ $h = 2$	13.9 64.4
(e)		$D = 2$ $h = 0.5$ $l = 4$	74
(f)		$D = 2$ $h = 0.5$ $l = 0.2$	245

- Single-layer rectangular coil with a circular conductor, Table 11.13b:

$$L_0 = pn^2(G + H) \times 10^{-9} \text{ (H)} \tag{11.48b}$$

where

$G = f_g(S_1/S_2;\ l/S_2)$

$H = f_h(d/\lambda, n)$ = tabulated coefficients [14,15]

- Single-layer toroidal coil, Table 11.13c:

$$L_0 = 2\pi n^2 \left(D - \sqrt{D^2 - D_1^2} \right) \times 10^{-9} \text{ (mH)} \tag{11.48c}$$

- Single-layer ring coil, Table 11.13d:

$$L_0 = 2hn^2 \ln(D_z/D_w) \times 10^{-9} \text{ (H)} \tag{11.48d}$$

- Multilayer long cylindrical coil, Table 11.13e:

$$L_0 = Dn^2[K - 2\pi h(0.693 + \gamma)/l] \times 10^{-9} \text{ (H)} \tag{11.48e}$$

where
K is the coefficient as for the single-layer cylindrical coil
γ is the coefficient whose values are listed in Appendix 11.B

- Multilayer short cylindrical coil, Table 11.13f:

$$L_0 \cong \frac{25\pi D^2 n^2}{3D + 9l + 10h} \times 10^{-9} \text{ (H)} \tag{11.48f}$$

Design guidance: As can be observed, $L = \psi D n^2$, where ψ is a function of two factors: l/D and h/d; and the inductance of air inductor increase when

- Outer diameter of winding increases
- Ratio of outer to inner winding diameter decreases
- Width of winding decreases
- Diameter of wire decreases

This can also be inferred from the last column of Table 11.13 comparing the values of L_0 for dimensionally similar air inductors.

Inductance L_0 can be readily calculated for a specified shape, dimensions, and number of turns of winding. However, this is not a design approach, because the design problem to be solved is inverse: to determine for a given value of L_0 the dimensions, shape, and number of turns of winding. However, such calculations are impossible to perform. This is a consequence of the given geometry of winding of n turns, which results in one and only one geometry of the magnetic flux Φ, therefore, in only one value of the inductance L_0. Inversely, to a given value of L_0 an infinite number of winding geometries may be attributed at any n value.

Thus, the theoretical problem is to obtain the dimensions of the winding in combination with the suitable number of turns n from the specified value of L_0 is mathematically insolvable.

Therefore, the solutions are determined by experimental or iterative numerical methods.

Quality factor of air inductors: Usually it is required that the inductor would have the maximum Q factor or time constant L/R.

For the following cases in Table 11.14: (a) fixed copper (wire) volume; (b) fixed length of wire; and (c) fixed coil volume, the L/R reaches maximum at following proportions of the coil winding.

TABLE 11.14 Dimensional Proportions of Coil Providing Its Maximum Time Constant L/R

		Case	
Winding		$(a)+(b)$	(c)
Cylindrical, single layer		$D \approx 3l$	$D = (1.3–2)l$
Cylindrical	Long	$D = 5h + 3l$	$D = (1.3–1.6)l$
Multilayer	Short		$D = (4–8)l$
			$D = (5–7)h$
Single layer	Toroidal	$D_2 \approx 3D_1$	—
	Ring	$D_2 \approx 1.6D_1$	—

Quality factor of air inductor extends from several units to several hundred and depends on

- Shape and dimensional proportions of winding (Table 11.14)
- Wire pitch and the wire arrangement
- Operating frequency
- Oil self-capacitance
- Material properties of coil former or substrate, wire insulation, proximity of conducting and/or magnetic parts

11.3.2.1 Typical Frequency Ranges of Air Inductors When in Use

In resonant circuits:

- Cylindrical multilayer 50–1000 kHz
- Cylindrical single layer 1 MHz–2 GHz

As chokes:

- Multilayer (multisectional) 10–1000 kHz
- Single-layer 500 kHz–800 MHz

Conductors (wires) used

- Solid insulated: dc–200 MHz
- Solid uninsulated: 100 MHz–1 GHz
- Stranded (litz wire): 50 kHz–10 MHz
- Tubular: 3 MHz–3 GHz
- Cu-coated metal: 30 MHz–3 GHz
- Sn- or Ag-coated metal: 0.2–3 GHz
- Sn, Ag, Au conducting path: 0.2–3 GHz

11.3.2.2 Advantages of Air Inductors

- High stability of properties versus temperature, time, and magnetic excitation levels (when compared with cored inductors).
- Very low dependence of inductance on frequency (related mainly to changes of effective dimensions of winding due to skin and proximity effects in wires), therefore, a practically linear dependence of $X_L = \omega L$ on frequency.
- Insaturable dependence of B on i (as opposed to cored inductors) allows the air inductors to produce very high pulses of induction B.

11.3.2.3 Disadvantages of Air Inductors

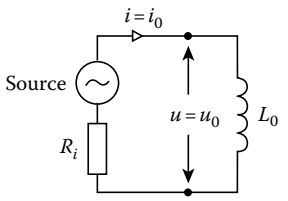
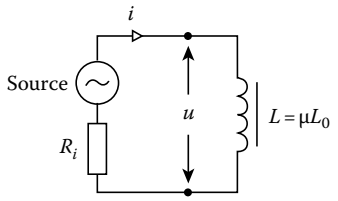
- Practical inability to obtain dimensionally small inductors of high inductance L, therefore, inherent inability to miniaturization.
- Air inductor acts both as an emitting and receiving antenna, therefore, it is (a) a source of unwanted electromagnetic radiation to surrounding circuitry and environment; (b) a receiver of electromagnetic signals introducing them into circuit as disturbance (usually)—electromagnetic interference (EMI). These strong coupling features make the air inductor a highly incompatible component in electronic circuitry, especially miniaturized, and in electronic and electrical environments.
- Air inductors are made, as a rule, as inductors of fixed value of inductance L. If a change of L is required, connections to some wires (taps) positioned between winding extremities, or a sliding contact to wires for single-layer windings, enable the inductance to be changed.

11.3.3 Cored Inductor

Reason for using the magnetic material in inductors: If a winding of inductance L_0 produces in vacuum (in air) a magnetic flux Φ_0 distributed over some volume and that this volume is completely filled up with a magnetic material of permeability μ, then the inductance L of the so combined winding with the magnetic material is $L = \mu L_0$.

Cored inductor and its parameters in electric circuit: In Table 11.15 the parameters of the inductor are compared for cases when it is immersed in air ($\mu = 1$) and in magnetic material of permeability μ. It is

TABLE 11.15 Inductor-Related Parameters

	Magnetic Material of Permeability, μ		
	Absent	Present, Fully Coupled with Winding	
Parameter	Any Source	Case (a): Constant emf Source ($R_i = 0$)	Case (b): Constant Current Source ($R_i = \infty$)
u =	u_0	u_0	μu_0
i =	i_0	i_0/μ	i_0
B =	B_0	B_0	μB_0
Φ =	Φ_0	Φ_0	$\mu\Phi_0$
H =	H_0	H_0/μ	H_0
L = [a]	L_0	μL_0	μL_0
[a]$L = \Phi/i$ =	$\Phi_0/i_0 = L_0$	$\Phi_0/(i_0/\mu) = \mu L_0$	$\mu\ \Phi_0/i_0 = \mu L_0$
			

Note: Voltage drop u, current i, induction B, magnetic flux Φ, field strength H, inductance L at the absence and presence of magnetic material fully coupled with inductor winding is shown for two limit cases: (a) inductor operates in circuit of constant emf ($R_i = 0$) and (b) in circuit of constant current ($R_i = \infty$).

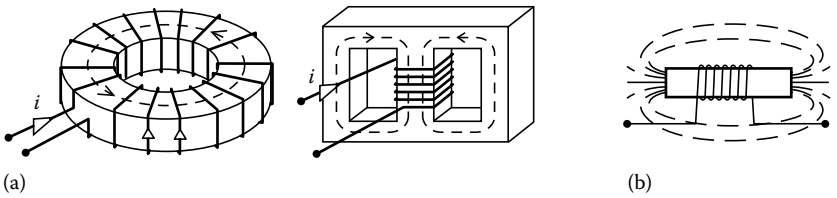


FIGURE 11.51 Examples of (a) closed magnetic circuits and (b) open magnetic circuit.

supposed that the magnetic flux Φ_0 produced by the inductor winding runs completely through the magnetic material (the case of the complete coupling of winding with the magnetic material).

In case (a), the voltage drop across inductor, magnetic induction, and flux remain unchanged while the current through inductor and magnetic field decreases μ times; in case (b), the current and magnetic field remain unchanged, whereas the voltage drop, magnetic induction, and flux increases μ times. The only parameter that increases μ times in all the cases, limit and intermediate ones, $0 < R_i < \infty$, is the inductance $L = \mu L_0$.

The main reason for applying the magnetic material as a core in inductor is to increase its inductance.

Inductor, magnetic core, closed and open magnetic circuit: If the magnetic flux Φ generated by a winding runs for the most of its part through the core being in the possibly strongest coupling with that winding, it is said that this core forms a closed or nearly closed magnetic circuit.

If at the possibly strongest coupling of a winding with a core, the flux Φ runs for its nonnegligible part also through the outside of the core, it is said that the core forms an open magnetic circuit.

The ring core and EE-core illustrate the closed magnetic circuits (Figure 11.51a), whereas the rod core illustrates the open magnetic circuit (Figure 11.51b).

11.3.3.1 Closed Magnetic Circuit

Equivalent dimensions and parameters: The permeance G_m of a toroidal core having a very small radial thickness, magnetic path length l, a uniform cross-section area A, and permeability μ, is given by Equation 11.46c.

If the shape of a core differs from the above radially thin toroid, the calculation of permeance becomes very complex, including the problem formulation and mathematics. To overcome these difficulties, a simplified standard procedure is recommended [3]: a core having the shape different from radially thin toroid, nonuniform cross-sections along magnetic path and often, nonuniform permeability, is replaced by a hypothetic equivalent toroid or ring having the same permeance as the nonuniform core (Figure 11.52).

The parameters of equivalent toroid: the equivalent magnetic path length l_e, equivalent area A_e, equivalent volume V_e, and equivalent permeability μ_e are calculated from the following formulas:

$$l_e = C_1^2/C_2; \quad A_e = C_1/C_2; \quad V_e = l_e A_e = C_1^3/C_2^2 \tag{11.49}$$

and equivalent magnetic parameters:

$$\begin{aligned} &\mu_e = C_1 \Big/ \sum_{i=1}^{v} \frac{l_i}{\mu_i A_i}; \quad B_e = \frac{u}{\omega n} \cdot \frac{C_2}{C_1}; \\ &H_e = ni\frac{C_2}{C_1^2}; \quad \Phi_e = \frac{u}{\omega n} \cdot \frac{C_1^2}{C_2^2} \end{aligned} \tag{11.50}$$

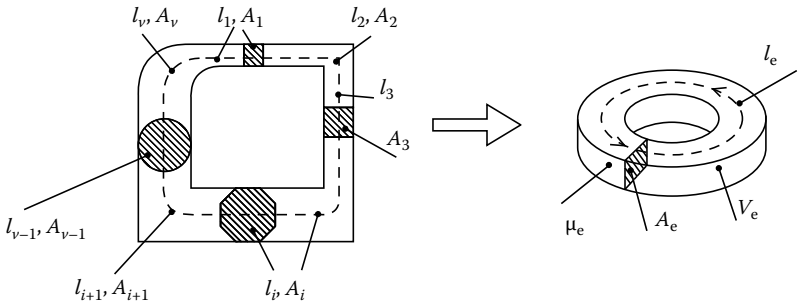


FIGURE 11.52 Nonuniform core and its equivalent toroid or ring having the same permeance.

The core coefficients C_1 and C_2 are calculated as

$$C_1 = \sum_{i=1}^{\nu} \frac{l_i}{A_i}; \quad C_2 = \sum_{i=1}^{n} \frac{l_i}{A_i^2} \tag{11.51}$$

Therefore, the permeances G_m and G_{m0} and reluctances R_m and R_{m0} are

$$G_m = \mu/C_1; \quad G_{m0} = 1/C_1; \quad R_m = C_1/\mu; \quad R_{m0} = C_1 \tag{11.52}$$

Inductance L is

$$L = A_L n^2 = \mu_e A_{L0} n^2 = \mu_0 G_m n^2 = \mu_0 \mu_e G_{m0} n^2 = \mu_0 \mu_e n^2 / C_1 \tag{11.53}$$

The core coefficients C_1, C_2, as well as equivalent dimensions l_e, A_e, and V_e for standardized cores are specified in relevant IEC standards and in core manufacturers' catalogs.

The previously described procedure is a standard averaging procedure (line and surface integrals are replaced by summation) of internationally accepted applicability. This procedure is commonly used in engineering, design, and measuring practice.

Air gap: Reason for introduction: An intentional air gap is introduced into a magnetic circuit of inductors to improve its performance, namely

- To decrease the magnetic material nonlinearity and/or lessen the effect of static magnetic field (bias field) on inductance
- To lessen the magnetic power loss
- To lessen excessive instabilities of the core magnetic material (e.g., thermal)
- To make possible a smooth adjustment of inductance to a very precise value by a special adjusting device or a specially shaped air gap

A closed magnetic circuit of a complicated shape (the EE core—by way of example) made of a material of magnetic permeability μ, and with a small air-gap of length l_g, is shown in Figure 11.53(a). A notional transformation of that circuit into equivalent successive ring cores is presented in Figure 11.53(b). Related permeabilities μ_e and μ_{eg} are also indicated there.

The length of intentional air-gap l_g is usually small compared with magnetic path length l. A ratio l_g/l usually does not exceed a fraction of a percent.

Effect of an air gap on core permeability: The inductance of an inductor having a core with air-gap (at the complete magnetic coupling between winding(s) and core) is given by

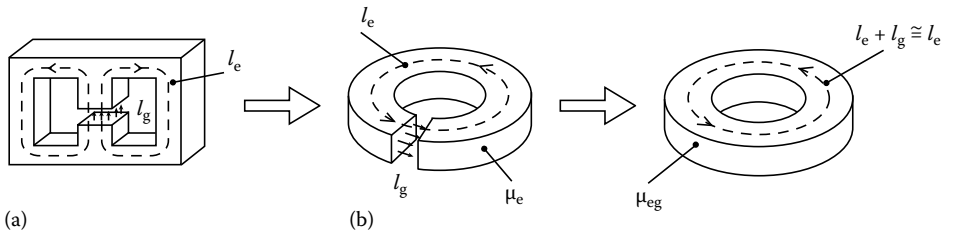


FIGURE 11.53 Ring cores (b) equivalent to the EE gapped core (a).

$$L = \frac{\mu_0 \mu_e n^2}{C_1 \left[1 + (\mu_e - 1)\frac{l_g}{l_e}\right]} = \frac{\mu_0 \mu_{eg} n^2}{C_1} \tag{11.54}$$

where

$$\mu_{eg} = \mu_e \left[1 + (\mu_e - 1)\frac{l_g}{l_e}\right]^{-1} \tag{11.55}$$

μ_{eg} = equivalent permeability of the core with air-gap. The dependence (Equation 11.55) is shown in Figure 11.54.

Equation 11.55 and Figure 11.52 ignore a bulging of the magnetic flux traversing the air gap, Figure 11.55.

Roughly, the bulging is greater as the term $(\mu_e - 1)l_g/l_e = p$ is larger. The bulging, expressed as an apparent increase of the air-gap area $A'_g/A_e = \kappa_g$, becomes perceptible at $p > 1$. The ratio κ_g, known as the bulging factor, can attain an amount of 2 for $p = 30$–60.

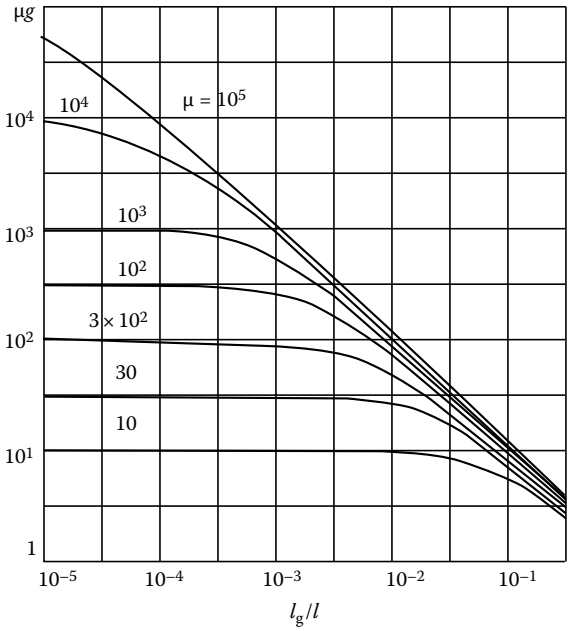


FIGURE 11.54 Equivalent permeability of the core with air-gap versus l_g/l.

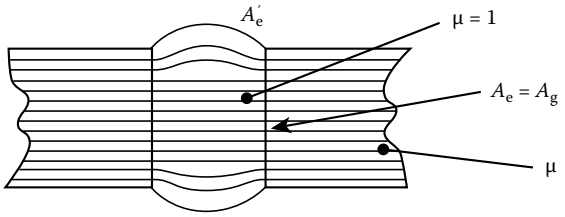


FIGURE 11.55 Bulging of the magnetic flux traversing the air-gap.

A change in flux distribution within the core, resulting from the occurrence of a large reluctance of the air gap in the magnetic circuit, is also ignored in Equation 11.55. In effect, the value of inductance L for larger air gaps is usually higher than that resulting from Equation 11.54.

As the air gap increases, the inductance L becomes less and less dependent on any variation of the magnitude of permeability μ_e. This effect is known as the linearization of cored inductor by the air-gap.

The air-gap effect can be summarized as follows:

The cored inductor winding perceives not the core with material permeability μ, but a core with reduced permeability μ_g lower than μ (or μ_{eg} when dealing with μ_e). Reduction of permeability of gapped core follows (Equation 11.55) depicted in Figure 11.54.

The higher the permeability, the greater the reduction of μ, for example, the air-gap of the same (relative) length $l_g/l = 0.01$ reduces material permeability $\mu = 10{,}000$ by a factor of 100 and that of $\mu = 10$ only by a factor of about 1.1. The use of high permeability materials as cores with larger air gaps is not effective.

Inductor with incomplete coupling: In real cases, the magnetic material forming the core of the inductor does not occupy the whole space where the magnetic flux Φ_0, emanating from the winding, is distributed. The following simplified quantitative approach is in use.

The flux Φ_0 is a sum of a leakage flux Φ_l, which runs completely outside the core and a hypothetical flux Φ_h, which runs completely inside the hypothetical equivalent toroid of permeability μ_e (or μ_{eg}): $\Phi_0 = \Phi_h + \Phi_l$. These three fluxes are related with three respective inductances: $L_0 =$ inductance of winding without core, $L_h =$ inductance of the shape of the hypothetical equivalent toroid, and $L_l =$ leakage inductance whose flux Φ_l does not penetrate into the core:

$$L_0 = L_h + L_l \tag{11.56}$$

Therefore, the inductance of the cored inductor with leakage is

$$L = \mu_e L_h + L_l = (\mu_e - 1)L_h + L_0 \tag{11.57}$$

This formula is very convenient to determine the equivalent permeability from an actual measurement. Because $L_h = \mu_0 n^2/C_1$, L and L_0 are taken from measurement, the equivalent permeability μ_e (or μ_{eg} for the gapped core) can be determined:

$$\mu_e = \frac{L - L_0}{\mu_0 n^2} C_1 + 1 \quad \text{or} \quad \mu_e = \frac{A_L - A_{L0}}{\mu_0} C_1 + 1 \tag{11.58}$$

For phenomenal considerations a so-called winding coupling coefficient $L_h/L_0 = k_w$ introduced. Then,

$$L = L_0[(\mu_e - 1)k_w + 1] \tag{11.59}$$

Apparent permeability: A commonly used parameter quickly characterizing the cored inductor is the apparent permeability μ_{app}:

$$\mu_{app} = L/L_0 = (\mu_e - 1)k_w + 1 \tag{11.60}$$

Equation 11.60 demonstrates whether the winding is completely coupled with the core; then $k_w = 1$, $\mu_{app} = \mu_e$; if $k_w = 0$, $\mu_{app} = 1$: the inductor is coreless; for $0 < k_w < 1$ always $\mu_{app} < \mu_e$ as in most of the real cases. The coupling coefficient is readily calculated from the formula:

$$k_w = (\mu_{app} - 1rrb)/(\mu_e - 1) \tag{11.61}$$

μ_{app} shows directly how many times the presence of the core, being in a given coupling with the winding, increases its inductance L_0.

Low winding coupling strongly lowers the effect of high permeability on the inductance of cored inductor. The high permeability materials are effective only for windings strongly coupled with the core. Such are monolayer windings of the thinnest possible wire, tightly fitting the core along its whole length, with no spacing between neighboring turns. This is in distinct opposition to other practical requirements, such as current load, low self-capacitance, interwire insulation and electrical breakdown, and ease of manufacturing.

Leakage coefficient: Also for the phenomenal considerations, the inductor leakage coefficient $\sigma = L_l/L$ is used. It shows what part of the total inductance L is not related with the core and its properties. The leakage coefficient is calculated as

$$\sigma = (\mu_e - \mu_{app})/\mu_{app}(\mu_e - 1) \quad \text{or} \quad \sigma = (1 - k_w)/\mu_{app} \tag{11.62}$$

If $k_w = 1$ or equivalently $\mu_e = \mu_{app}$, $\sigma = 0$, the inductor is leakage-free; if $k = 0$, then $\mu_{app} = 1$ and $\sigma = 1$, which results in $L = L_0 = L_l$— the leakage inductance is the inductance L_0 of the coreless winding.

Summarizing:

- Coupling coefficient k_w shows the part of the winding flux that is instrumental in carrying the core properties into the inductor properties.
- Leakage coefficient σ shows what part of total inductor properties has no relation with the core properties.
- Core, when not completely coupled with the winding, increases its inductance by amount of only μ_{app}.
- Apparent permeability μ_{app} of the real cored inductor is always lower than the core permeability μ, μ_e, or μ_{eg}.

Joint effect of the air gap and coupling on the cored inductor inductance: Combining the effect of air-gap (Equation 11.55) and that of coupling (Equation 11.59), the formula for the inductance of cored inductor is

$$L = \frac{\mu_e L_0 k_w}{1 + (\mu_e - 1)l_g/\kappa_g l_e} - \left(L_0 - \frac{\mu_0 n^2}{C_1}\right) \tag{11.63}$$

In this formula, two terms are to be distinguished: the first term, being roughly proportional to μ_e, is generally of nonlinear character and carries in itself all magnetic material properties including its instabilities. The second term, between parentheses, being effectively an air coreless inductance, is of purely linear character. Magnetic material properties ''enter'' inductor properties only by the first term.

As the air-gap and leakage become larger, the share of the first term decreases for the benefit of the second term: the inductor linearizes or stabilizes. For the sake of example: for an inductor with an EE42/20 ferrite core of material permeability $\mu \cong 1800$, the ratio of the first to the second term for ungapped core is about 450. If an air-gap of length $l_g = 1$ mm is cut in that core, this ratio drops to about 30. Thus, the linearization here reaches an amount of about 15. This effect is called the linearization or stabilization by the air-gap and/or by the leakage (or coupling).

A cored inductor with coupling $k_w = 1$ and with no air-gap behaves as a pure magnetic material itself, whereas for $k_w = 0$ it becomes a coreless air inductor having the inductance L_0.

11.3.3.2 Open Magnetic Circuit

The most commonly used open magnetic circuit is a cylindrical rod core. The inductor is built by axially placing the rod core inside a cylindrical coil as it is illustrated in Figure 11.56.

Inductors of this type are built as variable and fixed inductance inductors. To vary the inductance, the rod core is moved axially inside the coil. The central axial position of the rod in relation to the coil corresponds to the maximum inductance. In fixed inductance inductors, the movement between the core and coil is not provided.

If the rod core is enclosed by a centrally placed winding, as shown in Figure 11.56, the inductance of such an inductor can be calculated from the following empirical formula:

$$L = L_0[(\mu_r mq - 1)k_{wr} + 1] = L_0\mu_{app} \tag{11.64}$$

where

L_0 is the inductance of the winding without core
μ_r is the equivalent (effective) magnetic permeability of the rod, of similar significance as μ_e for the closed magnetic circuit
m and q are experimentally determined coefficients
k_{wr} is the coupling coefficient

The permeability of the rod μ_r is a complicated function of the material permeability μ and of the ratio of the rod length to the rod diameter l_r/d_r [13]. This function is shown in Figure 11.57a.

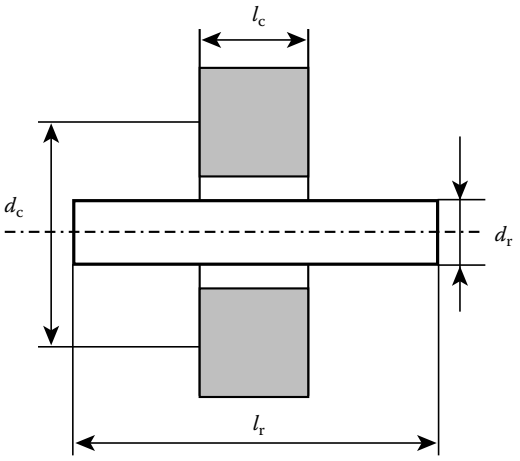


FIGURE 11.56 Open magnetic circuit having a rod core.

The experimental coefficients $m = f(l_c/l_r)$ and $q = f(d_c/d_r)$ are given in Figure 11.57b. The coupling coefficient k_{wr} is

$$k_{wr} = A_{Lh}/A_{L0}[\text{nH}]/[\text{nH}] \tag{11.65}$$

where

$A_{Lh} = Kd_r$, K is the coefficient listed in Appendix 11.A
$A_{L0} = L_0/n^2$, and $L_0 =$ winding inductance

The inaccuracy of the above calculations of L and μ_{app} does not exceed several percent.

As can be inferred from Figure 11.57b, the maximum inductance L is obtained for $mq = 1$, that is, for $l_c \cong 0.83l_r$ and $d_c = d_r$: that is, for the winding of width nearly equal to the rod length and closely fitting the rod diameter. As the winding diameter d_c is larger and winding width is narrower, the core contributes less to the inductor inductance L.

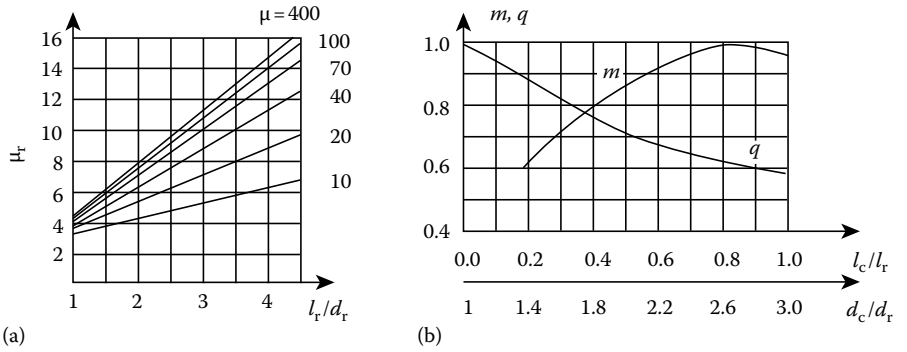


FIGURE 11.57 (a) Equivalent permeability of the rod μ_r versus ratio l_r/d_r for different material permeabilities μ and (b) experimental coefficients m vs. l_c/l_r and q vs. d_c/d_r.

It is often of great importance to estimate the material permeability of the rod core. The calculation inverse to that of Equation 11.64 can allow it: first one calculates $\mu_r = (\mu_{app} - 1 + k_{wr})/mqk_{wr}$, next for the so determined μ_r and given $l/_r/d_r$, the material permeability μ can be found from Figure 11.57a. This method, however, can practically distinguish $\mu < 400$ only and needs to gather very accurate input data and may provide with only approximate μ-values providing that $\mu < 400$.

11.3.3.3 Emerging Inductor-Related Advances

- Ongoing adaption of inductor core's shapes and windings to:
- surface mount technologies (SMT),
- Denser packing (per volume unit) of more and more miniaturized components
- Automatic assembling

Example: low-profile type of ferrite cores.

- Expanding applications of "lumped" inductors over:
- UHF range (300 MHz–3 GHz)

Example: Planar spiral-type inductors and microinductors:

- Frequency range of 300 MHz–1 GHz

Example: "quasi-microwave-range" inductors (air-, cored-, wound, and planar ones).

- Nanocrystalline-core inductors.
- Inductor-based probes, sensors, and microsensors.

11.3.3.4 Power Loss in Inductor

Power loss (in an inductor) is a power that is converted in an inductor into heat and radiation. The total power loss in the inductor is admitted to be a sum of the following contributory losses:

- Resistive loss in winding R_{res}.
- Magnetic loss in core magnetic material R_μ.
- Dielectric loss related to self-capacitance and other parasitic capacitances of the inductor R_d.
- Other losses, for example, radiation loss due to the electromagnetic radiation leaving inductor, loss due to dimensional resonance in the core, loss due to piezomagnetic effects. These losses are not discussed here.

For linear inductors and in the linear range of cored inductors, the power loss P_L is expressed as a tangent of the loss angle δ, tgδ, (Figure 11.50), or equivalently, as the Q-factor ($Q = 1/\text{tg}\delta$). The inductor power loss P_L and tgδ are related by the expressions

$$P_L = \frac{u_{rms}^2}{\omega L_p}\text{tg}\delta = i_{rms}^2 \omega L_s \text{tg}\delta \tag{11.66}$$

where

u_{rms} is the rms value of voltage drop across the inductor
i_{rms} is the rms value of current flowing through the inductor

At high excitation levels, especially in a nonlinear range, the power loss of inductors is expressed directly in watts

Resistive loss in winding: A loss due to a dc resistance (ohmic resistance) of a winding conductor increased by increments due to skin and proximity effects R_{res} referred to the inductive reactance of inductor ωL is

$$\text{tg}\delta_{res} = R_{res}/\omega L \tag{11.67}$$

The precise determination of R_{res}, and especially, a determination of skin and proximity effect contributions, is rather difficult. If an optimization of such winding parameters as a type of conductor (solid, stranded), its diameter, etc., is sought, then a necessary calculation needs a theoretical background [13]. R_{res} has a linear voltage–current character.

Magnetic loss: Depending on whether the excitation level is (1) low, (2) middle, (3) or high, three representations of magnetic loss are accustomed. They are presented next.

(1) Low B (or H) amplitude excitation: A range of B or H amplitudes exists at which μ (B or H) = const.

This corresponds to the case of cored inductors operating at small-signal levels. For this range the notions of complex inductance $\hat{L}$ and complex permeability $\hat{\mu}$ are used.

In series representation—$\hat{Z} = j\omega L + R_s = j\omega\hat{L}_s$:

$$\hat{L}_s = \hat{\mu}_s L_0, \quad \hat{\mu}_s = \mu_s' - j\mu_s'', \quad L_s = \mu_s' L_0, \quad \text{and} \quad R_s = \omega L_0 \mu_s'' \tag{11.68}$$

In parallel representation—$\hat{Y} = 1/j\omega L_p + 1/R_p = 1/\omega\hat{L}_p$:

$$\hat{L}_p = \hat{\mu}_p L_0, \quad 1/\hat{\mu}_p = 1/\mu_p' - 1/j\mu_p'', \quad L_p = \mu_p' L_0, \quad \text{and} \quad R_p = \omega L_0 \mu_p'' \tag{11.69}$$

Magnetic loss tangent tgδμ:

$$\text{tg}\delta_\mu = \mu_s''/\mu_s' = \mu_p'/\mu_p'' \tag{11.70}$$

$$\mu_p' = \mu_s'\left(1 + \text{tg}^2\delta_\mu\right); \quad \mu_p'' = \mu_s''\left(1 + 1/\text{tg}^2\delta_\mu\right) \tag{11.71}$$

In these formulas, $\mu'(=\mu_s'$ or $\mu_p')$ and μδ ($=\mu\delta_s$ or $\mu\delta_p$) are real and imaginary components of the complex magnetic permeability $\hat{\mu}(=\hat{\mu}_s$ or $\hat{\mu}_p)$ in series or parallel representation, respectively.

Note: When the subscript at μ is deleted, the symbol always refers to the series representation.

The complex permeability $\hat{\mu}$ and related parameters are restricted to linear voltage–current (or *B*–*H*) dependences of permeability and loss. It means that μ' represents the dynamic initial permeability and $\mu\delta$ the power loss, which are related solely to only reversible magnetization processes in magnetic material. In that convention the magnetic loss at low excitations is presented usually in form of $\text{tg}\delta\ \mu(f)$ or relative (magnetic) loss factor $\text{tg}\delta\ \mu/\mu_{i(st)}$ ($\mu_{i(st)}$ = static initial permeability).

All previous considerations concerning the contribution of magnetic permeability to the cored inductor properties are extendable to deal with the magnetic loss. With that intent, the permeability μ or μ_e appearing in the formulas of interest should be replaced by the complex permeability $\hat{\mu}$. This simple formal operation results, however, in a serious complication of calculation because of the dealing with complex numbers.

One of the most important results is that the air-gap reduces the loss resistance R_s and $\text{tg}\delta\ \mu$ by the ratio μ'_{eg}/μ':

$$R_{sg} = R_s(\mu'_{eg}/\mu'); \quad \text{tg}\delta_{\mu g} = \text{tg}\delta_{\mu}(\mu'_{eg}/\mu') \tag{11.72}$$

On the other hand, the power loss in magnetic material P_μ remains unchanged for the same core, gapped or ungapped, if it operates at the same induction B; this is because

$$P_\mu = \left(\omega B_p^2 V_e/2\mu_0\right) \times (\text{tg}\delta_\mu/\mu') = \left(\omega B_p^2 V_e/2\mu_0\right) \times \left(\text{tg}\delta_{\mu eg}/\mu'_{eg}\right) \tag{11.73}$$

where

B_P is the peak value of B

V_e is the core equivalent volume

(2) Middle B (or H) amplitude excitations: A range of B or H amplitudes exists where magnetic permeability increases linearly with H: $\mu = \mu_i + \nu H$, magnetic induction B is a parabolic function of H and the hysteresis effects occur (Rayleigh range). The magnetic loss $\text{tg}\delta_\mu$ is here expressed as a sum of eddy-current loss $\text{tg}\delta_F$, residual loss $\text{tg}\delta_n$, and hysteresis loss $\text{tg}\delta_h$:

$$\text{tg}\delta_\mu = \text{tg}\delta_h + \text{tg}\delta_F + \text{tg}\delta_n \tag{11.74}$$

$\text{tg}\delta_h$ is proportional to H or B, $\text{tg}\delta_F$ to frequency, $\text{tg}\delta_n$ is assumed to be a constant. This is visible when other notation (given by Legg) is used:

$$R_s/\mu fL = aB_P + ef + c = 2\pi\text{tg}\delta_\mu/\mu' \tag{11.75}$$

where

a, e, and *c* are the hysteresis, eddy current, and residual loss coefficients, respectively

B_P is the peak value of induction B

The particular loss tangents can be determined as shown in Figure 11.58.

(3) High B (or H) amplitude excitations: A range of the strongest dependence of the permeability μ on induction B (or field strength H) exists, therefore, of the strongest nonlinearity of μ versus excitation. As a rule, the power inductors and transformers operate in that range. In this range, the magnetic power loss P_μ is being expressed by the heuristic Steinmetz expression:

$$P_\mu = \eta f^m B_p^n \tag{11.76}$$

where η = numerical coefficient m and n are experimentally determined Steinmetz exponents, which are approximately constant for a given magnetic material, for example, for power ferrites

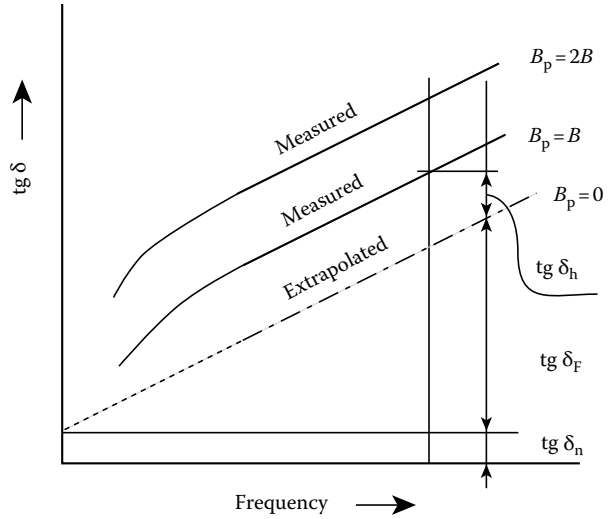


FIGURE 11.58 Determination of the magnetic loss tangents according to (Equation 11.74).

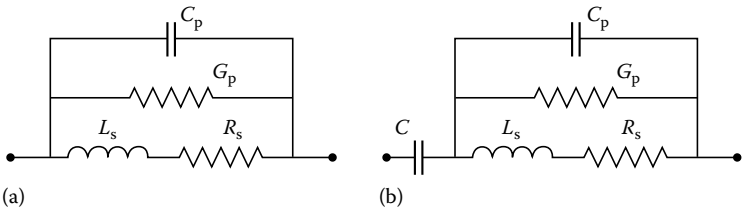


FIGURE 11.59 Self-capacitance C_P and conductance G_P of an inductor L_s, R_s in (a) intentionally nonresonant circuit; (b) in an intentionally series resonant circuit.

$1.2 < m < 1.4$ and $2 < n < 3$ at frequencies between 10 and 100 kHz. For Fe-Si alloys: $1.3 < n < 1.4$ and $1.6 < m < 1.8$ for $f < 500$ Hz. The plots log P_μ versus log f or versus log B_P are therefore straight lines. If these plots deviate from the straight lines, the constancy of exponents m and n deteriorates and the applicability of Equation 11.76 becomes less useful.

Dielectric loss: Every real inductor is inevitably associated with a shunting parasitic capacitance, called self-capacitance C_P, and accompanied conductance $G_P \supseteq C_P$ and G_P result from various capacitive and conductive couplings distributed over a real geometrical assembly of various materials (conducting, insulating) forming the physical body of the inductor. In practice, C_P and G_P are assumed to be as lumped, connected in parallel inductor elements of loss tangent $\text{tg}\delta_d = G_P/\omega C_P$. It is advisable to consider the contribution of $\text{tg}\delta_d$ to overall inductor loss for two cases (1) and (2) [13].

1. Inductor (L_s of $\text{tg}\delta = R_s/\omega L_s$) operating in an intentionally nonresonant circuit, Figure 11.59a

$$\text{tg}\delta_p = \omega^2 L_s C_P \text{tg}\delta_d = \text{tg}\delta_d (f/f_0)^2 \tag{11.77}$$

   where $f_0 = (1/2\Pi)\sqrt{L_s C_P}$ = inductor self-resonant frequency.
2. The same inductor operating in a series LC intentional resonant circuit, Figure 11.57b and, due to the resonant circumstances, $\text{tg}\delta_p$ contributes in that case as $\text{tg}\delta_{pr}$:

TABLE 11.16 Summary of Power-Loss Expressions for the Sinusoidally Excited Inductor

Loss Representation	Low Excitation ($\mu' = \text{const}$)	Middle Excitation ($\mu = \mu_i + \nu H$)	High Excitation (μ—strongly nonlinear)
$\text{tg}\delta_t =$	$\text{tg}\delta_{tl} = \underbrace{\text{tg}\delta_{res} + \overbrace{\mu''/\mu'}^{\text{tg}\delta_\mu^{(b)}}}_{\text{tg}\delta} + \text{tg}\delta_c^{(a)}$	$\text{tg}\delta_{tm} = \underbrace{\text{tg}\delta_{res} + \overbrace{\text{tg}\delta_h + \text{tg}\delta_F + \text{tg}\delta_n}^{\text{tg}\delta_\mu^{(b)}}}_{\text{tg}\delta} + \text{tg}\delta_c^{(a)}$	Not applicable
$R_s =$	$\omega L_s \cdot \text{tg}\ \delta_{tl}$	$\omega L_s \cdot \text{tg}\ \delta_{tm}$	Not applicable
$R_p =$	$R_s(1 + 1/\text{tg}^2\ \delta_{tl})$	$R_s(1 + 1/\text{tg}^2\ \delta_{tm})$	Not applicable
$P_L =$	$i_{rms}^2 \cdot \omega L_s \text{tg}\delta_{tl}$ or $u_{rms}{}^2 \cdot \text{tg}\delta_{tl}/\omega L_p$	$i_{rms}^2 \cdot \omega L_s \text{tg}\delta_{tm}$ or $u_{rms}{}^2 \cdot \text{tg}\delta_{tm}/\omega L_p$	$u_{rms}^2 \left(\frac{R_{res}}{R_{res}^2 + \omega^2 L_0^2} + \omega C_p \text{tg}\delta_d\right)^c$
$P_\mu =$	$\left(\omega B_p^2 V_e/2\mu_0\right) \cdot \left(\mu''/\mu'^2\right)$	$\left(\omega B_p^2 V_e/2\mu_0\right) \cdot \left(\text{tg}\delta_\mu/\mu'\right)$	$\eta f^m B_p^n$

Distinction: low, middle, and high excitation apply only to the cored inductors.

$\text{tg}\delta_t$ = total loss tangent; $\text{tg}\delta_{tl}$ = total loss tangent at low excitation level; $\text{tg}\ \delta_{tm}$ = total loss tangent at middle excitation level.

[a] $\text{tg}\delta_c$ { $\text{tg}\delta_p = \text{tg}\delta_d \cdot (f/f_0)^2$ for inductor operating in a nonresonant circuit, Figure 11.59a; $\text{tg}\delta_{pr} = (2\text{tg}\delta + \text{tg}\delta_d) \cdot (f/f_0)^2$ for inductor operating in a series resonant circuit, Figure 11.59b

[b] For air inductors: $\text{tg}\delta_\mu = 0$ because $\mu' = 1$ and μ'', $\text{tg}\delta_h$, $\text{tg}\delta_F$, $\text{tg}\delta_n$, P_μ equal zero.

[c] For air inductor. For cored inductor P_L is practically not calculable because of the distorted signal.

$$\text{tg}\delta_{pr} = \omega^2 L_s C_p (2\text{tg}\delta + \text{tg}\delta_d) = (2\text{tg}\delta + \text{tg}\delta_d)(f/f_0)^2 \tag{11.78}$$

Because the value of $\text{tg}\delta_d$ may come up even to 0.1, the product $L_s C_p$ should be as low as possible to make the self-resonant frequency $f_0 \gg f$. As an example, for an inductor having $\text{tg}\delta = 0.004$ (Q-factor = 250) and $\text{tg}\delta_d = 0.02$, $(f/f_0)^2 = 0.1$, the degraded Q-factor attains 170 in case (1) and 150 and in case (2). Additionally, $\text{tg}\delta_d$ is very sensible to any climatic factors and different agents; the case is known that a low value $\text{tg}\delta_d$ increased by nearly 200 times in a hot humidity environment.

Inductor total power loss: Different representations of total power loss in inductor are summarized in Table 11.16.

11.3.4 Nonlinearity in Inductors

The cored inductors are nonlinear except when the excitation does not exceed the level where $\mu(B$ or $H) = \text{const}$. The most current three characteristics of inductor expose the extent of nonlinearity:

- Inductance versus ac current or applied voltage
- Inductance versus dc bias current or voltage with superimposed ac current or voltage
- Harmonic content versus specified excitation

The nonlinearity of a cored inductor is due to hysteretic properties of the core material, which result in coercive and remanence effects. That peculiarity of magnetic material causes both multivalent and multivalued characters of nonlinearity in cored inductors. In addition, in many circumstances the nonlinearity can also be time dependent, for example, the harmonic content can vary with time.

Generally, the nonlinearity occurs when the core permeability becomes excitation dependent. Such excitation-dependent permeability is termed as amplitude permeability μ_a.

Numerous phenomenal expressions are used for dependence of amplitude permeability μ_a on excitation. The most well known are expressions for the Rayleigh range of excitations given by

$$\text{Rayleigh:} \quad \mu_a = \mu_i + \nu H; \tag{11.79}$$

$$\text{Peterson:} \quad \mu_a = \alpha_0 + \alpha_1 H + \alpha_2 H^2 + \cdots \tag{11.80}$$

where

μ_i is the initial permeability
ν is the Rayleigh constant
α_0, α_1, α_2, and so on are the Peterson coefficients

Today, these expressions are of lesser importance because any experimentally obtained dependence of an inductor on excitation can be numerically processed and used for overall nonlinear analysis (e.g., of signal distortion) in a given circuit. All previously given relationships between the permeability and inductance are to be observed in that analysis.

A practical manner of lowering the nonlinearity of an inductor (for a given core material) is to increase the air-gap length and/or to decrease the coupling coefficient k_w, or to lower the induction B in the core.

For the sake of an example, the third harmonic content, $\text{THC} = \text{emf}_{3f}/\text{emf}_f$ of the ungapped core is related to tangent of material hysteresis loss $\text{tg}\delta_h$ in the Rayleigh range by the formula

$$\text{THC} = 0.6\text{tg}\delta_h \tag{11.81}$$

For the gapped core, this formula transforms into

$$\text{THC}_{\text{gapped}} = \text{THC}_{\text{ungapped}} \times (\mu_{eg}/\mu) \tag{11.82}$$

Generally, for a given core material, the most nonlinear inductor is that equipped with a core having an ungapped magnetic circuit (e.g., a toroid); the least nonlinear is the inductor having an open short core (e.g., a rod core of low length to diameter ratio).

The nonlinear core related relationships are strongly frequency dependent and, nearly as a rule, decrease as the frequency increases.

11.3.5 Magnetic Core Materials

For inductor core applications, three families of soft magnetic materials are concerned:

- Ferrites, mainly Mn–Zn and Ni–Zn, being solid polycrystalline ceramics in whatever form they may be
- Iron and/or nickel based alloys (Fe, Fe–Si, Fe–Ni, Fe–Co, Ni–Mn, Ni–Mn–Mo, etc.) in the form of laminations, strips, and ribbons
- Pure iron and Fe–Ni or Ni–Mn, etc., alloys in the form of fine powders (2–150 μm) with individual particles that are bonded and electrically insulated by polymers

The most distinctive features of these three families are listed in Table 11.17.

Recently developed amorphous and nanocrystalline magnetic materials gained as of yet a moderate use in inductor applications. Some ring power chokes used as core-wound amorphous ribbons have advantageous properties at the kHz range.

The basic characteristics of magnetic materials for inductor applications are depicted in Figure 11.60.

11.3.6 Magnetic Materials Measurements: Basic Methodological Principles

For the measurements of magnetic properties and characteristics of a magnetic material as, for example, its magnetic permeability, quality factor or power loss, the sample (core) being made of that material shall have a ring or toroidal shape in which:

TABLE 11.17 Three Main Families of Magnetic Materials Used as Cores in Inductors

Family	B_s (T)	μ_i	ρ (Ωm)	f_w
Ferrites	Low: <0.6	8–20,000	High: 10^0–10^9	1 kHz–2 GHz
Fe, Ni alloys	High: 0.8–2.4	200–100,000	Low: (1–50)10^{-9}	dc—400 Hz (up to 50 kHz)[a]
Bonded powders	Medium: <1.2	4–150	Low: $\sim 10^{-7}$	dc—100 kHz (up to 10 MHz)[a]

B_s = saturation induction; μ_i = initial permeability; ρ = dc resistivity; f_w = typical working frequency range.
[a] For very small material thickness.

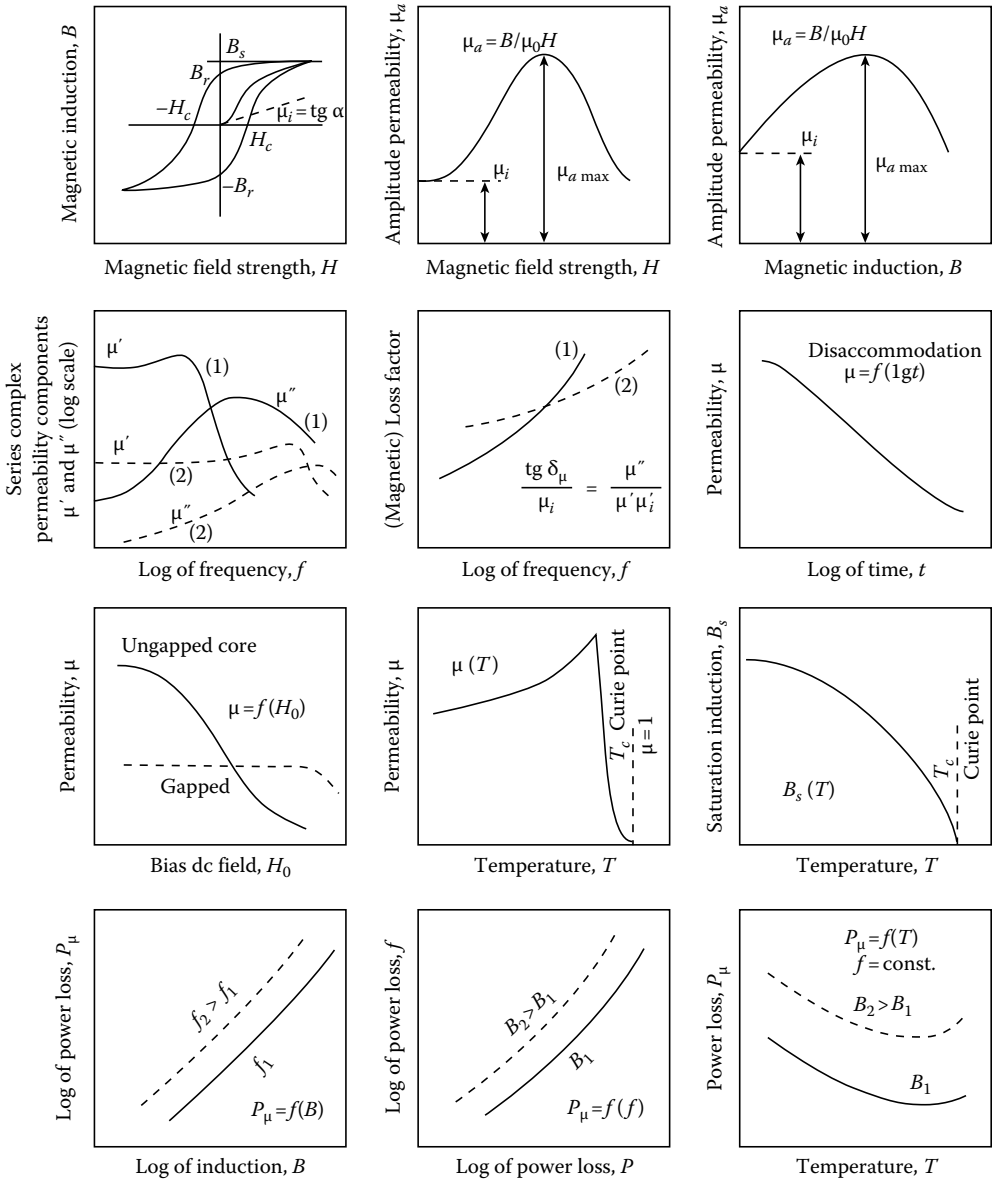


FIGURE 11.60 Basic characteristics of magnetic materials essential for inductor applications.

1. There are no transversal (to the magnetic flux path) air gaps.
2. The ratio of outer to inner diameter shall not be greater than 1.4.
3. The measuring winding(s) shall be distributed uniformly along the circumference of the core, close to its surface, so as to get the inductive coupling coefficient between the core and the winding(s) approaching one. (This stipulation becomes more critical as the magnetic permeability of the core is low.)

Besides, the core to be measured shall be magnetically or thermally conditioned (past terms: neutralized, demagnetized) to arrive at a well-defined and reproducible magnetic state of the material.

This operation shall take place at a determined period of time before the start of each measurement or before a sequence of measurements, depending on a prescribed measuring procedure.

11.3.7 Physical Irregularities

Physical irregularity stands for inconsistency of the state or quality of the core surface, bulk or shape with their intended regularity.

The irregularities and their locations give, as a rule, a detrimental or critical effect on functional properties of magnetic cores, components and devices and, in consequence, on their applicative usefulness.

A detailed categorization, related nomenclature, and graphical examplification of most physical irregularities occurring in ferrite cores are provided in the International Standard IEC 60401-1: *Terms and nomenclature for cores made of magnetically soft ferrit—Part 1: Terms used for physical irregularities.*

11.3.8 International Standards on Inductors

International standardization activities on inductors, their attributes, applications and testing are managed by the Technical Committee 51 (TC51) ''Magnetic Components and Ferrite Materials'' of the International Electrotechnical Commission (IEC).

The International Standards (Publications) issued by the IEC/TC51 in years 1972-2007, in a number of 62, can be classified into 6 groups as below:

1. Terminology and specific nomenclature (6 Stds)
2. Shapes and dimensions (19 Stds)
3. Design and calculations: guidelines and formulas (3 Stds)
4. Measuring methods and procedures (13 Stds)
5. Specific features and characteristics (19 Stds)
6. Marketing (2 Stds)

The list of the current IEC/TC51 Standards is available at http://www.iec.ch. Some of them are given, for example, in Refs. [6–12].

Defining Terms

Amplitude permeability: The permeability obtained from the peak value of the induction and the peak value of the applied field strength when the field is varying periodically with time, the average of the field over the time is zero, and the material is initially in a neutral state.

Choke: A device, being an inductor, used in a special application to impede the current in a circuit over a specified frequency range while allowing relatively free passage of the current at lower frequencies.

Complex permeability: The complex quotient of the induction in the material and the product of magnetic constant and field strength, at the excitation range, where both the induction and field vary sinusoidally with time.

Inductor: A device consisting of one or more associate windings, with or without magnetic core, for introducing inductance into an electric circuit.

Initial permeability: The limiting value of the amplitude permeability when the field strength is vanishingly small.

Loss angle of inductor: The angle by which the fundamental component of passing current lags the fundamental component of the fundamental voltage.

Magnetic core: A configuration of magnetic material, which is placed in a specific geometrical position in relation to current-carrying conductors and whose magnetic properties are essential to its use.

Magnetic power loss: The power loss in a magnetic material when it is magnetized.

Physical irregularity: The inconsistency of the state or quality of the magnetic core surface, bulk, or shape with their intended regularity.

Planar (or chip) inductor: An inductor consisting of a small piece of chip-like, nonconductive magnetic or nonmagnetic material—substrate—onto which is deposited, etched, or printed a conductive pattern in the form of a circular or rectangular spiral. Planar inductors may be structured as single or multilayered devices.

Power loss: The power expended without useful work and converted into heat and/or nonintended radiation.

Quality factor: The magnitude of the ratio of the inductor effective reactance to its effective resistance at a given frequency (reciprocal of the tangent of the loss angle of the inductor).

Tangent of the loss angle (of inductor): The magnitude of the ratio of the inductor effective resistance to its effective reactance at a given frequency (reciprocal of the quality factor of the inductor).

Tangent of the magnetic loss angle: The ratio of the imaginary component of the complex permeability (of a material) to its real component.

Winding: A conductive path, usually of wire, consisting of one (single-loop winding) or more turns, usually forming a spiral.

References

1. ANSI-IEEE Standard 100-1988, *IEEE Standard Dictionary of Electrical and Electronic Terms*, 4th edn., New York: IEEE, 1988.
2. M. Zahn, *Electromagnetic Field Theory: A Problem Solving Approach*, New York: John Wiley & Sons, 1979.
3. IEC 60205: *Calculation of the Effective Parameters of Magnetic Piece Parts*.
4. IEC 60723-1 to 5: *Inductor and Transformer Cores for Telecommunications*.
5. IEC 61007: *Transformers and Inductors for Use in Electronic and Telecommunication Equipment—Measuring Methods and Test Procedures*.
6. IEC 61248-1 to 7: *Transformers and Inductors for Use in Electronic and Telecommunication Equipment—Parts 1 to 7: Generic and Sectional Specifications*.
7. IEC 61332: *Soft Ferrite Material Classification*.
8. IEC 62044–3: *Measuring Methods—Part 3: Magnetic Properties at High Excitation Level*.
9. IEC 62024–1: *High-Frequency Inductive Components—Electrical Characteristics and Measuring Methods—Part 1: Nanohenry Range Chip Inductor*.
10. IEC 62025–1 to 2: *High-frequency inductive components—Non-Electrical Characteristics and Measuring Methods—Part 1: Fixed, Surface Mounted Inductors for Use in Electric and Telecommunication Equipment*, Part 2: Test methods for non-electrical characteristics.
11. IEC 62211: *Inductive Components—Reliability Management*.
12. The complete catalogue of IEC Standards (including those from [3] to [11]) issued by the Technical Committee 51 (IEC/TC51): Magnetic Components and Ferrite Materials, Available at: http://www.iec.ch.

13. E. C. Snelling, *Soft Ferrites*, *Properties and Applications*, 2nd edn., London: Butterworths, 1988.
14. H. Hartwig, *Induktivitäten*, Berlin: Verlag für Radio-Foto-Kinotechnik, GmbH, 1954.
15. F. E. Terman, *Radio Engineering Handbook*, 3rd edn., New York: McGraw-Hill, 1947.

Further Information

Several "books" provide comprehensive and in-depth information on inductors, choice of suitable magnetic material for core, design methods, and various applications: E. C. Snelling, 1988, *Soft Ferrites. Properties and Applications*, London: Butterworths, 1988; W. Kampczyk and E. Roess, *Ferritkerne*, Berlin, München: Siemens AG, A. Goldman, *Modern Ferrite Technology,* New York: Van Nostrand Reinhold, 1990; L. Michalowsky u.a., *Magnettechnik,* Fachbuchverlag Leipzig GmbH, 1993; R. Boll, *Weichmagnetische Werkstoffe*, Hanau, Berlin, München: Vacuumschmelze GmbH—Siemens AG, 1990; G. E. Fish, 1989, *Soft Magnetic Materials, Proc. IEEE*, vol. 78, no. 6, p. 947, 1989, review paper; McCurrie, R. A., *Ferromagnetic Materials,* Academic Press, 1994; Niknejad, A. M. and Mayer, R. G., *Design, Simulation and Application of Inductors and Transformers for Si RF ICs,* Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000; R. C. Dorf, Ed., *Electrical Engineering Handbook*, Boca Raton, FL: CRC Press, 1993; W. T. McLyman, *Transformer and Inductor Design Handbook*, 2nd ed., New York: Marcel Dekker, 1999; C. J. Kaiser, *Inductor Handbook*, CJ Publishing, 1996; Y. I. Ismail and E. G. Friedman, *On-Chip Inductance in High Speed Integrated Circuits*, Dordrecht, The Netherlands: Kluwer Academic, 2001.

The following journals and conference proceedings publish articles on inductors and related issues. "Journals" *IEEE Transactions on Magnetism; Journal of Magnetism and Magnetic Materials; Coil and Winding International.* "Conferences" *INTERMAG*; (American) *Conference on Magnetism and Magnetic Materials* (MMM); *International Conference on Magnetism* (ICM); *International Conference on Ferrites* (ICF); *Soft Magnetic Materials Conference* (SMM); *European Magnetic Materials and Application Conference* (EMMA); *Intertech Business Conferences on Magnetic Materials*, (American) Annual Applied Power Electronics Conference (APEC).

Because of the enormous diversity of inductor applications, properties of various magnetic materials, inductors, and core shapes, it is suggested that one consult the relevant data published by inductor manufacturers in their catalogs and application notes.

Chip inductors are described in Chapter 1 of *Analog and VLSI circuits*.

The measuring methods concerning the inductors and magnetic core properties are given in "Standards" prepared by Technical Committee No. 51 of the International Electrotechnical Commission (IEC). Copies of these Standards may be obtained from the IEC Central Office: 1, rue de Varembé, Geneva, Switzerland or from the IEC National Committees.

Appendices

APPENDIX 11.A Coefficient $K = f(d/l)$ Figuring in Equation 11.48a

D/l	K	D/l	K	D/l	K	D/l	K
0.02	0.1957	0.32	2.769	0.80	5.803	2.20	10.93
0.04	0.3882	0.34	2.919	0.85	6.063	2.40	11.41
0.06	0.5776	0.36	3.067	0.90	6.171	2.60	12.01
0.08	0.7643	0.38	3.212	0.95	6.559	2.80	12.30
0.10	0.9465	0.40	3.355	1.00	6.795	3.00	12.71
0.12	1.126	0.42	3.497	1.10	7.244	3.50	13.63
0.14	1.303	0.44	3.635	1.20	7.670	4.00	14.43
0.16	1.477	0.46	3.771	1.30	8.060	4.50	15.14
0.18	1.648	0.48	3.905	1.40	8.453	5.00	15.78

(*continued*)

APPENDIX 11.A (continued)

D/l	K	D/l	K	D/l	K	D/l	K
0.20	1.817	0.50	4.039	1.50	8.811	6.00	16.90
0.22	1.982	0.55	4.358	1.60	9.154	7.00	17.85
0.24	2.144	0.60	4.668	1.70	9.480	8.00	18.68
0.26	2.305	0.65	4.969	1.80	9.569	9.00	19.41
0.28	2.406	0.70	5.256	1.90	10.09	10.00	20.07
0.30	2.616	0.75	5.535	2.00	10.37	12.00	21.21

Source: Hartwig, H., *Induktivitäten*, Verlag für Radio-Foto-Kinotechnik, GmbH, Berlin, 1954. With permission.

APPENDIX 11.B Coefficient $\gamma = f(l/h)$ Figuring in Equation 11.48e

l/h	γ	l/h	γ	l/h	γ	l/h	γ
1	0.0000	9	0.2730	17	0.3041	25	0.3169
2	0.1202	10	0.2792	18	0.3062	26	0.3180
3	0.1753	11	0.2844	19	0.3082	27	0.3190
4	0.2076	12	0.2888	20	0.3099	28	0.3200
5	0.2292	13	0.2927	21	0.3116	29	0.3209
6	0.2446	14	0.2961	22	0.3131	30	0.3218
7	0.2563	15	0.2991	23	0.3145		
8	0.2656	16	0.3017	24	0.3157		

Source: Hartwig, H., *Induktivitäten*, Verlag für Radio-Foto-Kinotechnik, GmbH, Berlin, 1954. With permission.

11.4 Transformer

11.4.1 Introduction

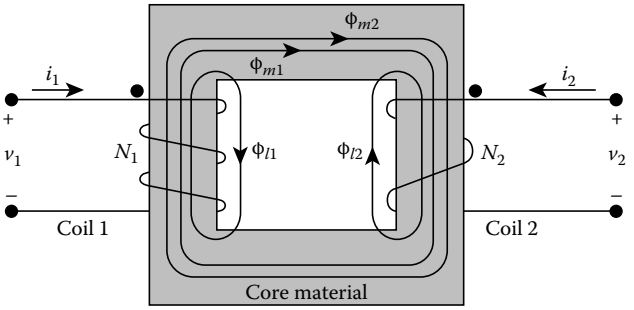
The transformer is a two-port passive circuit element that consists of two coils that are coupled magnetically, but have no conductive coupling. It is shown diagrammatically in Figure 11.61, where the dots by one end of each coil indicate that the magnetic fluxes, Φ_{m1} and Φ_{m2} are in the same direction when both currents either enter or leave by the dot marked terminal. Coil 1 is connected to the transformer input terminals and is called the primary winding. Coil 2 is called the secondary winding and is connected to the transformer output terminals.

A transformer can be used to connect a source to a load and comes in a wide range of sizes. The sizes include very large power-distribution transformers and very small transformers used in electronic equipment. The coils of some transformers used in electronic equipment are wound on a nonmagnetic core such as plastic. These transformers are called air-core transformers. All transformers used in power-distribution systems and some transformers used in electric equipment use an iron core, which produces a coupling coefficient of nearly unity. The coupling coefficient for an air-core transformer seldom exceeds 0.5.

Four major characteristics of transformers are as follows:

1. Accept energy at one voltage and deliver it at a different voltage.
2. Change the load impedance as seen by the source.
3. Provide conductive isolation between two portions of a circuit.
4. Produce bandpass signal filters when combined with capacitors.

The first characteristic indicated is commonly used in electric power-distribution systems since higher voltages are desired for electric energy transmission than can be safely used by the customer. The higher



N_1 = number of turns in coil 1

N_2 = number of turns in coil 2

Φ_{l1} = leakage flux due to current i_1

Φ_{l2} = leakage flux due to current i_2

Φ_{m1} = magnetizing flux due to current i_1

Φ_{m2} = magnetizing flux due to current i_2

$\Phi_1 = \Phi_{l1} + \Phi_{m1}$
= total flux due to current i_1

$\Phi_2 = \Phi_{l1} + \Phi_{m2}$
= total flux due to current i_2

$k_2 = \dfrac{\Phi_{m1}}{\Phi_1}$
= coupling factor for coil 1

$k_2 = \dfrac{\Phi_{m2}}{\Phi_2}$
= coupling factor for coil 2

$k_1 = \sqrt{k_1 k_2}$ = transformer coupling coefficient < 1

FIGURE 11.61 Transformer diagram and definitions.

transmission voltage produces lower transmission current, which requires smaller transmission line conductors.

The output stage of an audio amplifier may include a transformer to provide the second and third characteristics listed. In this way, the low impedance of the speaker is matched to the higher output impedance of the power amplifier to yield maximum power transfer for the signal. The isolation property permits the isolation of dc biasing voltages in the amplifier from the speaker coil to avoid magnetic flux saturation in the coil.

Finally, a transformer may be used with capacitors for interstage coupling in a radio frequency amplifier. The inductance of the transformer coils and the capacitance of the capacitors can be adjusted to produce bandpass filtering for signals.

This section presents linear mathematical models for transformers. The models are in terms of phasors and impedances; that is, they are frequency-domain models. The section begins with a first-order model known as the ideal transformer. A nonideal linear transformer model that includes transformer inductive and resistive effects is then shown.

11.4.2 Ideal Transformer

The ideal transformer models the first three transformer characteristics listed above with an ideal, lossless, circuit element. It is a reasonably good model for a transformer with a nearly unity coefficient

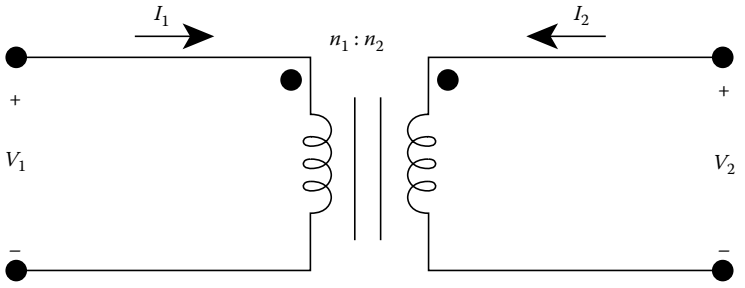


FIGURE 11.62 Circuit symbol for the ideal transformer.

of coupling and primary and secondary coil inductive impedances that are very large with respect to source and load impedances. Well-designed, iron-core transformers have approximately these characteristics over a reasonable range of frequencies and terminating impedances. A circuit symbol that is often used for an ideal transformer is shown in Figure 11.62, where $a = n_1/n_2 \cong N_1/N_2$ is the effective turns ratio. In place of the lines between the coils, which are intended to indicate the similarity to an iron-core transformer, the word ideal or the equation $k = 1$ may be written below the coils to identify an ideal transformer.

The equations that represent the ideal transformer model are $\boldsymbol{V}_2 = \boldsymbol{V}_1/a$ and $\boldsymbol{I}_2 = -a\boldsymbol{I}_1$. Thus, the voltages are in the same ratio as the effective turns ratio. Also, the ideal transformer is lossless since the complex power supplied by the source $\boldsymbol{V}_1\boldsymbol{I}_1^*$ equals the complex power absorbed by the load $-\boldsymbol{V}_2\boldsymbol{I}_2^*$. The hybrid-$h$-parameter two-port equation that represents the ideal transformer model is

$$\begin{bmatrix} V_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \end{bmatrix} \tag{11.83}$$

If the load impedance $\boldsymbol{Z}_L$ is connected to the output terminals of an ideal transformer, then $\boldsymbol{V}_2 = -\boldsymbol{I}_2\boldsymbol{Z}_L$. In this case, the impedance seen by a source connected to the input terminals is

$$Z_{eq} = \frac{V_1}{I_1} = \frac{(aV_2)}{\left(-\frac{1}{a}I_2\right)} = a^2 Z_L \tag{11.84}$$

Thus, the impedance seen by the source is the square of the effective turns ratio times the load impedance.

11.4.3 Nonideal Transformer

The ideal transformer is not an adequate transformer model when the coefficient of coupling is not near unity and/or the load and source impedances are not negligible with respect to the transformer-coil inductive impedances. Also, it cannot be used to investigate the signal filtering that can be performed with a transformer and capacitors. In these cases, a more detailed model is required.

11.4.3.1 Linear Transformer

The linear transformer model is shown in Figure 11.63. R_1 and R_2 are the resistances and L_1 and L_2 are the self-inductances of the two transformer coils. M is the mutual inductance corresponding to the magnetic coupling of the coils.

The linear transformer is a reasonable model for an air-core transformer since the magnetic flux is proportional to the current. The self-inductances are

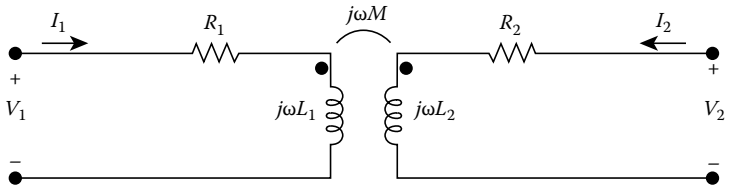


FIGURE 11.63 Linear transformer model.

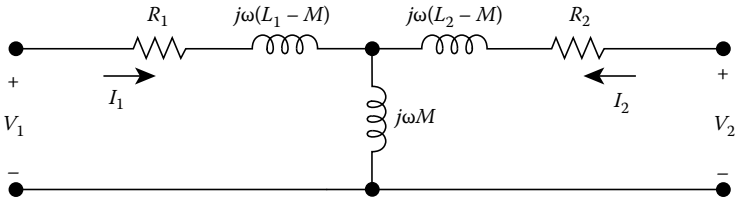


FIGURE 11.64 Equivalent *T*-network representation of a linear transformer.

$$L_1 = N_1 \frac{\Phi_1}{\mathbf{i}_1}; \quad L_2 = N_2 \frac{\Phi_2}{\mathbf{i}_2} \tag{11.85}$$

and the mutual inductance is

$$M = k\sqrt{L_1 L_2} \tag{11.86}$$

From Figure 11.63, the impedance-parameter two-port equations for the linear-transformer model are

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} R_1 + j\omega L_1 & j\omega M \\ j\omega M & R + j\omega L_2 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{11.87}$$

Except for its isolation characteristics, the linear transformer in Figure 11.63 can be represented by the equivalent *T*-network shown in Figure 11.64, since this network produces Equation 11.87. The *T*-network is completely equivalent to the nonisolating linear transformer that is produced when the lower ends of the two coils are conductively connected.

11.4.3.2 Leakage and Magnetizing Inductances

The nonideal linear-transformer circuit model can be changed, so it is expressed in terms of leakage inductance and magnetizing inductance instead of self-inductance and mutual inductance. This is convenient because the effects produced by the coil resistance and leakage inductance are nearly linear. Thus, the approximation required to produce a linear model is primarily contained in the magnetizing inductance.

From Figure 11.61 and Equation 11.85, the magnetizing and leakage inductances for coil 1 are, respectively,

$$L_{m1} = \frac{N_1 \Phi_{m1}}{i_1} = k_1 L_1 \tag{11.88}$$

and

$$L_{\ell 1} = \frac{N_1 \phi_{\ell 1}}{i_1} = (1 - k_1) L_1 \tag{11.89}$$

Therefore,

$$L_1 = L_{\ell 1} + L_{m1} \tag{11.90}$$

Similarly, for coil 2,

$$L_{m2} = \frac{N_2 \phi_{m2}}{i_2} = k_2 L_2 \tag{11.91}$$

$$L_{\ell 2} = \frac{N_2 \phi_{\ell 2}}{i_2} = (1 - k_2) L_2 \tag{11.92}$$

and

$$L_2 = L_{\ell 2} + L_{m2} \tag{11.93}$$

are the magnetizing, leakage, and self-inductances.

Using Figure 11.61 and Equation 11.86, we can write the mutual inductance as

$$\begin{aligned} M &= \sqrt{k_1 k_2}\sqrt{L_1 L_2} = \sqrt{k_1 L_1 k_2 L_2} \\ &= \sqrt{L_{m1} L_{m2}} \\ &= a L_{m2} = \left(\frac{1}{a}\right) L_{m1} \end{aligned} \tag{11.94}$$

where

$$\sqrt{\frac{L_{m1}}{L_{m2}}} \equiv \frac{n_1}{n_2} = a \cong \frac{N_1}{N_2} \tag{11.95}$$

is the effective turns ratio. How closely the effective turns ratio n_1/n_2 approximates the actual turns ratio N_1/N_2 depends on how completely all magnetic flux links all coil turns. This is a function of the coil and core geometry.

11.4.3.3 Circuit Model

Substitution of the leakage inductances, magnetizing inductances, and effective turns ratio into Equation 11.87 produces an alternate form for the impedance-parameter two-port equations for the linear transformer model. These equations are

$$\begin{aligned} \mathrm{V}_1 &= [R_1 + j\omega(L_{\ell 1} + L_{m1})] I_1 + j\omega L_{m1} \left(\frac{1}{a}\right) I_2 \\ &= (R_1 + j\omega L_{\ell 1}) I_1 + j\omega L_{m1} (I_1 - I_{\mathrm{a}}) \end{aligned} \tag{11.96}$$

and

$$\begin{aligned} V_2 &= j\omega L_{m1} \left(\frac{1}{a}\right) I_1 + [R_2 + j\omega(L_{\ell 2} + L_{m2})] I_2 \\ &= \left(\frac{1}{a}\right) V_{\mathrm{a}} + (R_2 + j\omega L_{\ell 2}) I_2 \end{aligned} \tag{11.97}$$

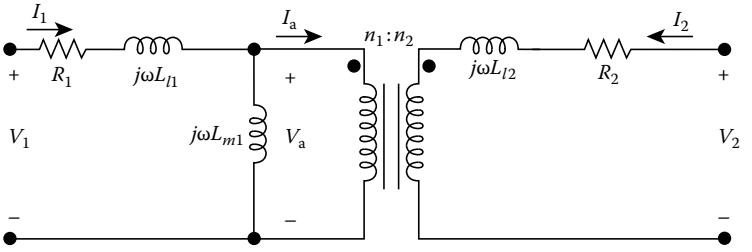


FIGURE 11.65 Linear circuit model for a nonideal transformer.

where

$$I_a = -\left(\frac{1}{a}\right) I_2 \tag{11.98}$$

and

$$\begin{aligned} \mathrm{V_a} &= j\omega L_{m1}\mathrm{I_1} + j\omega a L_{m2}\mathrm{I_2} \\ &= j\omega L_{m1}(\mathrm{I_1} - \mathrm{I_a}) \end{aligned} \tag{11.99}$$

The transformer circuit model that produces Equations 11.96 through 11.99 is depicted in Figure 11.65.

The only energy losses in the transformer that are accounted for by the linear model depicted in Figure 11.65 are the heating losses in the coils (called copper losses). An iron-core transformer also has heating losses in the core material (called core losses). The core losses have two components. The first component is hysteresis losses. Hysteresis is the nonlinear phenomenon that causes the magnetic flux response to increasing current to be different from the response to decreasing current. The plot of flux as a function of current traces out a closed curve called a hysteresis loop. The area inside this loop is proportional to the energy that produces core heat.

The second component of transformer core losses is eddy-current losses. This is a heating loss caused by currents (called eddy currents) that flow in the transformer core due to the voltage induced in the core material by the changing flux. The eddy-current losses can be decreased by laminating the core material to reduce the voltage induced in an eddy-current path and thus the eddy-current value.

Core losses are caused by nonlinear effects in the magnetic circuit formed by the transformer core. However, they can be approximately included in a linear transformer circuit model by introducing a resistance in parallel with the magnetizing inductance. This improves the linear model accuracy in modeling terminal-characteristics of the transformer. The resulting circuit model is shown in Figure 11.66, where I_{el} is

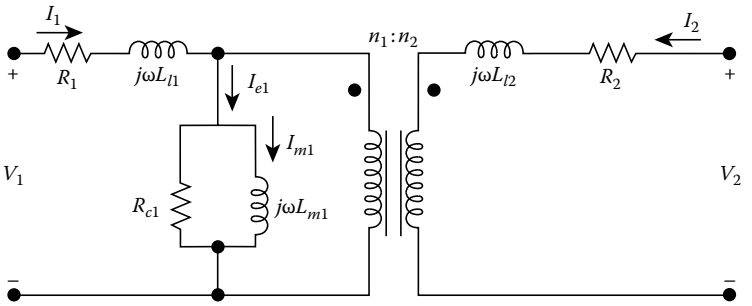


FIGURE 11.66 Linear circuit model for a nonideal transformer—including core losses.

called the excitation current, I_{ml} is called the magnetization current, and R_{cl} is the resistance used to model the core losses.

Defining Terms

Transformer: A two-port passive circuit element consisting of two magnetically coupled coils that are not conductively connected.
Air-core transformer: A transformer with a nonmagnetic core.
Source: Signal generator that supplies energy to a network.
Load: Device that converts electrical energy supplied to a useful output. Modeled as an impedance.
Magnetic flux: A magnetic field descriptor. It is thought of as lines indicating the direction of force that the magnetic field produces on a moving charge. The density of the lines indicates the force.
Primary winding: Transformer input coil.
Secondary winding: Transformer output coil.
Leakage inductance: Magnetic flux that links only one coil.
Magnetizing inductance: Magnetic flux that links both coils.

References

1. D. R. Cunningham and J. A. Stuller, *Basic Circuit Analysis*, Boston: Houghton Mifflin, 1991.
2. A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric Machinery*, 4th edn., New York: McGraw-Hill, 1983.
3. W. H. Hayt, Jr., and J. E. Kemmerly, *Engineering Circuit Analysis*, 4th edn., New York: McGraw-Hill, 1986.
4. P. Horowitz and W. Hill, *The Art of Electronics*, 2nd edn., Cambridge: Cambridge University Press, 1989.
5. J. G. Kassakian, M. F. Schlecht, and G. C. Verghese, *Principles of Power Electronics*, Reading, MA: Addison-Wesley, 1991.
6. G. McPherson and R. D. Laramore, *An Introduction to Electrical Machines and Transformers*, 2nd edn., New York: John Wiley & Sons, 1990.
7. J. W. Nillson, *Electric Circuits*, 3rd edn., Reading, MA: Addison-Wesley, 1990.
8. D. L. Schilling and C. Belove, *Electronic Circuits*, 3rd edn., New York: McGraw-Hill, 1989.

Further Information

More detailed developments of linear transformer circuit models can be found in most circuit analysis texts. Two example texts are Refs. [1,3].

Power transformers and their models are presented in considerable detail in electrical machinery texts such as Ref. [6]. These presentations also extend to multiple-coil, three-phase transformers, and tapped transformers. Included in most electrical machinery texts are in-depth discussions of hysteresis and eddy currents, and of methods for reducing losses due to these effects. Also included are procedures for measuring the parameters for transformer circuit models.

The use of transformers in electronic circuits is considered in electronic circuit design texts and reference books such as Refs. [4,8]. Amplifier circuit models that include transformer circuit models are developed in some texts for tuned transformer coupled amplifier stages and transformer coupled load impedances. It is demonstrated how these models can be used to determine amplifier frequency response and power transfer to the load. Other uses indicated for transformers in electronic circuits occur in power supplies and isolation amplifiers.

11.5 Semiconductor Diode

Semiconductor diodes are made out of p–n semiconductor junctions. Nonlinear current–voltage characteristics of such junctions are used to rectify and shape electrical signals. Exponential current–voltage characteristics are sometimes used to build logarithmic amplifiers. The variations of junction capacitances with applied voltages are used to tune high-frequency electronic circuits. The semiconductor p–n junction illuminated by light will generate a voltage on its terminals. Such a diode is known as a solar battery. Also, the reverse diode current is proportional to the light intensity at the junction. This phenomenon is used in photodiodes. If a diode is biased in the forward direction, it can generate a light. In order to obtain high emission efficiency the light emitting diode (LED) should be made out of a semiconductor material with a direct energy band structure. This way electrons and holes can recombine directly between valence and conduction bands. Typically, LEDs are fabricated using various compositions of $Ga_yAl_{1-y}As_xP_{1-x}$. The wavelength of generated light is inversely proportional to the potential gap of a junction material. When a light intensity is enhanced by additional micromirrors, then laser action occurs. The silicon diodes are not emitting light because the silicon has an *indirect* band structure and the probability of direct band-to-band recombination is very small.

When both sides of the junction are very heavily doped, then for small forward-biasing voltages (0.1–0.3 V), a large tunneling current may occur. For larger forward voltages (0.4–0.5 V), this tunneling current vanishes. This way the current–voltage characteristic has a negative resistance region somewhere between 0.2 and 0.4 V (Figure 11.68d). Germanium and other silicon semiconductors are used to fabricate tunnel diodes. The backward diode has slightly lower impurity concentrations than the tunnel diode and the tunneling current in the forward direction does not occur (Figure 11.68e). The backward diode is characterized by very sharp knee near zero voltage, and it is used for detection (rectifications) of signals with very small magnitude.

Diodes with high breakdown voltage have a p–i–n structure with an impurity profile shown in Figure 11.67d. Similar p–i–n structure is also used in microwave circuits as a switch or as an attenuating resistor. For reverse biasing, such a microwave p–i–n diode represents an open circuit with a small parasitic junction capacitance. In the forward direction this diode operates as a resistor whose conductance is proportional to the biasing current. At very high frequencies, electrons and holes will oscillate rather than flow. Therefore, the microwave p–i–n diode exhibits linear characteristics even for large modulating voltages. Another interesting ''diode'' structure has the impurity profile shown in Figure 11.67f. When reverse biasing exceeds the breakdown voltage, this element generates a microwave signal with a frequency related to the electron transient time through structure. Such a diode is known as an IMPATT (IMPact Avalanche Transit Time) diode.

The switching time of a p–n junction from the forward to the reverse direction is limited by the storage time of minority carriers injected into the vicinity of the junction. Much faster operation is possible in the Schottky diode, where minority carrier injection does not exist. Another advantage of the Schottky diode is that the forward voltage drop is smaller than in the silicon p–n junction. This diode uses the metal–semiconductor contact for its operation. Schottky diodes are characterized by relatively small reverse breakdown voltage, rarely exceeding 30 V.

11.5.1 Nonlinear Static *I–V* Characteristics

Semiconductor diodes are characterized by nonlinear current–voltage characteristics. Typical *I–V* diode characteristics are shown in Figure 11.68. In the case of a common silicon diode, the forward-direction current increases exponentially at first, and then is limited by an ohmic resistance of the structure. A very small reverse current at first increases slightly with applied voltage and then starts to multiply near the breakdown voltage (Figure 11.69). The current at the breakdown is limited by ohmic resistances of the structure.

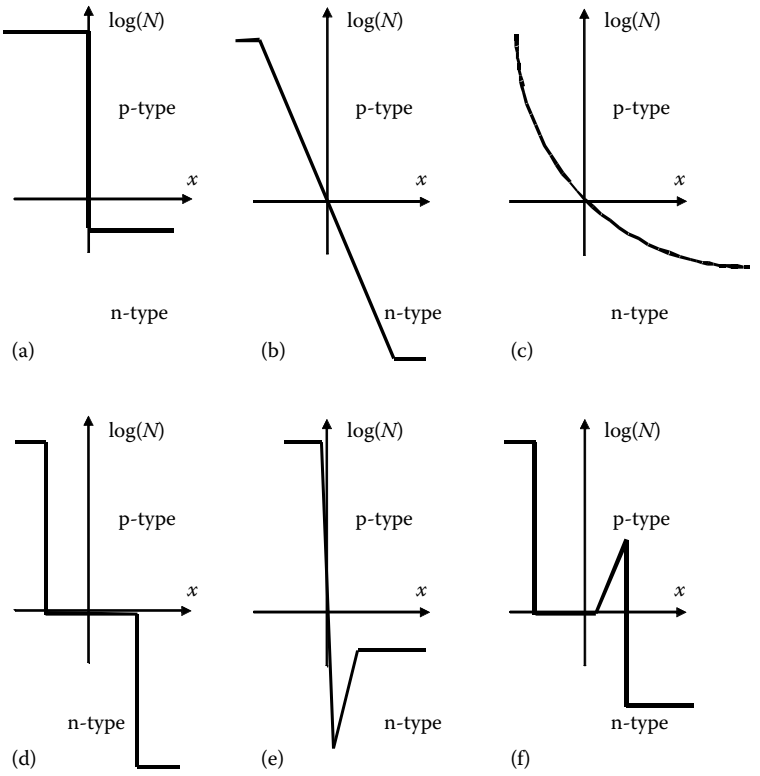


FIGURE 11.67 Impurity profiles for various diodes: (a) step junction; (b) linear junction; (c) diffusion junction; (d) p–i–n junction; (e) p–n^+–n junction; (f) p–i–p–n junction.

11.5.1.1 p–n Junction Equation

The n-type semiconductor material has a positive impurity charge attached to the crystal lattice structure. This fixed positive charge is compensated by free moving electrons with negative charges. Similarly, the p-type semiconductor material has a lattice with a negative charge that is compensated by free moving holes, as Figure 11.70 shows. The number of majority carriers (electrons in p-type and holes in n-type materials) is approximately equal to the donor or acceptor impurity concentrations, that is, $n_n = N_D$ and $p_p = N_A$. The number of minority carriers (electrons in p-type and holes in n-type) can be found using the equations

$$n_p = \frac{n_i^2}{p_p} \approx \frac{n_i^2}{N_A} \quad p_n = \frac{n_i^2}{n_n} \approx \frac{n_i^2}{N_D} \tag{11.100}$$

The intrinsic carrier concentration n_i is given by

$$n_i^2 = \xi T^3 \exp\left(-\frac{V_g}{V_T}\right); \quad V_T = \frac{kT}{q} \tag{11.101}$$

where

$V_T = kT/q$ is the thermal potential ($V_T = 25.9$ mV at 300 K)
T is the absolute temperature in K
$q = 1.6 \times 10^{-16}$ C is the electron charge

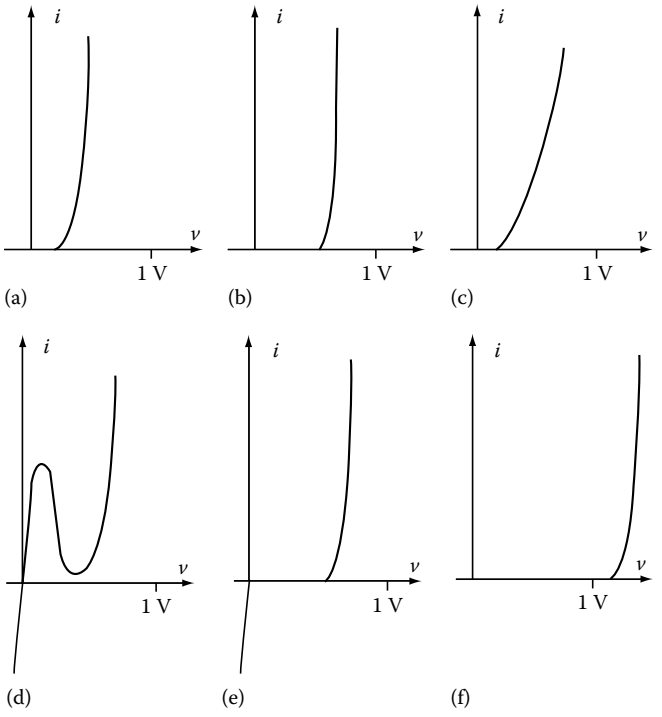


FIGURE 11.68 Forward current–voltage characteristics of various types of diodes: (a) germanium diode; (b) silicon diode; (c) Schottky diode; (d) tunnel diode; (e) backward diode; and (f) LED diode.

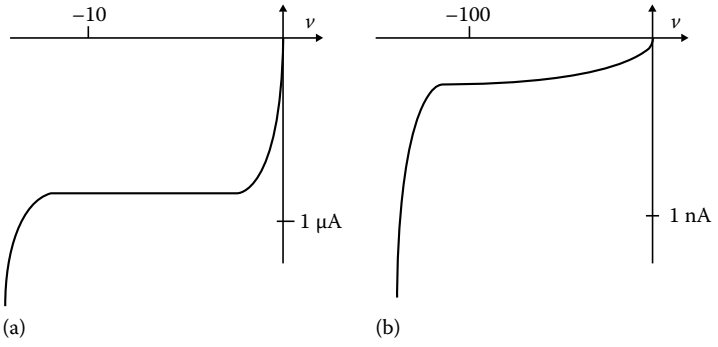


FIGURE 11.69 Reverse current–voltage characteristics: (a) germanium diode and (b) silicon diode.

$k = 8.62 \times 10^{-5}$ eV/K is the Boltzmann constant
V_g is potential gap ($V_g = 1.124$ V for silicon)
ξ is a material constant

For silicon, intrinsic concentration n_i is given by

$$n_i = 7.98 \times 10^{15} T^{\frac{3}{2}} \exp\left(-\frac{6522}{T}\right) \tag{11.102}$$

For silicon at 300 K, $n_i = 1.5 \times 10^{10}$ cm^{-2}.

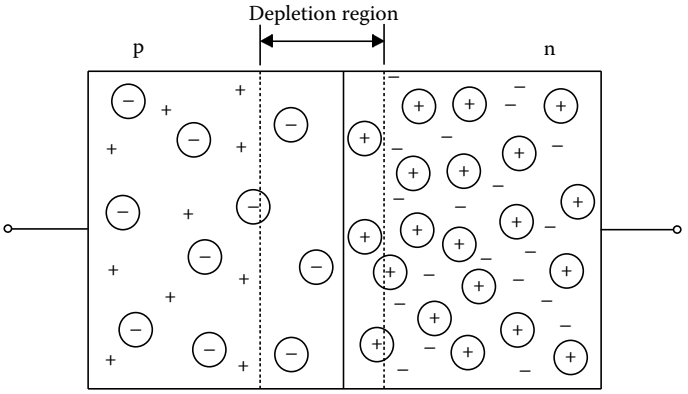


FIGURE 11.70 Illustration of the p–n junction.

When a p–n junction is formed, the fixed electrostatic lattice charges form an electrical field at the junction. Electrons are pushed by electrostatic forces deeper into the n-type region and holes into the p-type region, as illustrated in Figure 11.70. Between n-type and p-type regions, a depletion layer exists with a built-in potential, which is a function of impurity doping level and intrinsic concentration n_i:

$$V_{pn} = V_T \ln\left(\frac{N_A N_D}{n_i^2}\right) = V_T \ln\left(\frac{n_n p_p}{n_i^2}\right) = V_T \ln\left(\frac{n_n}{n_p}\right) = V_T \ln\left(\frac{p_p}{p_n}\right) \tag{11.103}$$

The junction current as a function of biasing voltage is described by the diode equation:

$$i = I_s\left[\exp\left(\frac{v}{V_T}\right) - 1\right] \tag{11.104}$$

where

$$I_s = Aqn_i^2 V_T\left(\frac{\mu_p}{\int_0^{L_p} n_n dx} + \frac{\mu_n}{\int_0^{L_n} p_p dx}\right) \tag{11.105}$$

where

$n_n \approx N_D$

$p_P \approx N_A$

μ_n and μ_p are the mobility of electrons and holes

L_n and L_p are the diffusion length for electrons and holes, and A is the device area

In the case of diodes made of silicon or other semiconductor materials with a high energy gap, the reverse-biasing current cannot be calculated from the diode Equation 11.104. This is due to the carrier generation-recombination phenomenon. Lattice imperfection and most impurities are acting as generation-recombination centers. Therefore, the more imperfections there are in the structure, the larger the deviation from ideal characteristics.

11.5.1.2 Forward *I–V* Diode Characteristics

The diode Equation 11.104 was derived with an assumption that injected carriers are recombining on the other side of the junction. The recombination within the depletion layer was neglected. In real forward-biased diodes, electrons and holes are injected through the depletion region and they may recombine there. The recombination component of the forward-biased diode is given by

$$i_{\text{rec}} = qwA\frac{n_{\text{i}}}{2\tau_0}\exp\left(\frac{v}{2V_T}\right) = I_{r0}\exp\left(\frac{v}{2V_T}\right) \tag{11.106}$$

where

w is the depletion layer thickness

τ_0 is the carrier lifetime in the depletion region

The total diode current $i_T = i + i_{\text{rec}}$ where i and i_{rec} are defined by Equations 11.104 and 11.106. The recombination component dominates at low current levels, as Figure 11.71 illustrates.

Also in very high current levels, the diode Equation 11.104 is not valid. Two phenomena cause this deviation. First, an ohmic resistance always plays an important role for large current values. The second deviation is due to high concentration of injected minority carriers. For very high current levels, the injected minority carrier concentrations may approach, or even become larger than the impurity concentration. An assumption of the quasi-charge neutrality leads to an increase of the majority carrier concentration. Therefore, the effective diode current is lower, as can be seen in Equation 11.105. The high current level in the diode follows the relation

$$i_{\text{h}} = I_{\text{h0}}\exp\left(\frac{v}{2V_T}\right) \tag{11.107}$$

Figure 11.71 shows the diode *I–V* characteristics, which include generation-recombination, diffusion, and high current phenomena. For modeling purposes, the forward diode current can be approximated by

$$i_{\text{D}} = I_0\exp\left(\frac{v}{\eta V_T}\right) \tag{11.108}$$

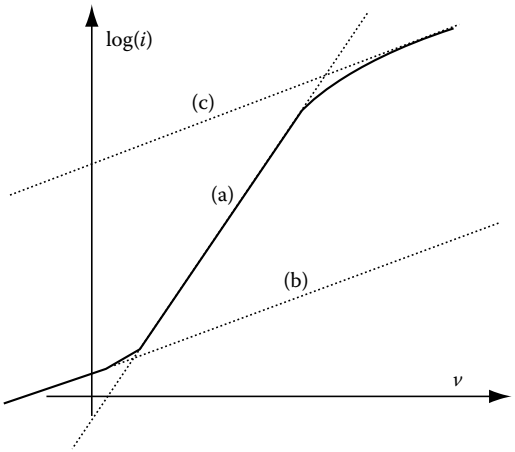


FIGURE 11.71 Current–voltage characteristics of the p–n junction in the forward direction: (a) diffusion current; (b) recombination current; and (c) high-level injection current.

where η has a value between 1.0 and 2.0. Note, that the η coefficient is a function of current, as can be seen in Figure 11.71. It has a larger value for small and large current regions and it is close to unity in the medium current region.

11.5.1.3 Reverse *I–V* Characteristics

The reverse leakage current in silicon diodes is mainly caused by the electron–hole generation in the depletion layer. This current is proportional to the number of generation-recombination centers. These centers are formed either by a crystal imperfection or deep impurities, which create energy states near the center of the energy gap. Once the reverse voltage is applied, the size of the depletion region and the number of generation-recombination centers increase. Thus, the leakage current is proportional to the thickness of the depletion layer $w(v)$. For a step-abrupt junction

$$w = \sqrt{\frac{2\varepsilon\varepsilon_0\left(V_{\text{pn}} - v\right)}{qN_{\text{eff}}}} \tag{11.109}$$

For other impurity profiles, w can be approximated by

$$w = K\left(V_{\text{pn}} - v\right)^{\frac{1}{m}} \tag{11.110}$$

The reverse-diode current for small and medium voltages can therefore be approximated by

$$i_{\text{rev}} = Aw(v)\frac{qn_{\text{i}}}{2\tau_0} \tag{11.111}$$

where n_i is given by Equation 11.101 and w by Equations 11.109 or 11.110. The reverse current increases rapidly near the breakdown voltage. This is due to the avalanche multiplication phenomenon. The multiplication factor is often approximated by

$$m = \frac{1}{1 - \left(\frac{v}{BV}\right)^m} \tag{11.112}$$

where BV stands for the breakdown voltage and m is an exponent chosen experimentally. Note that, for the reverse biasing, both v and BV have negative values and the multiplication factor M reaches an infinite value for $v = \text{BV}$.

11.5.2 Diode Capacitances

Two types of capacitances are associated with a diode junction. One capacitance, known as diffusion capacitance, is proportional to the diode current. This capacitance exists only for the forward-biased condition and has the dominant effect there. The second capacitance, known as the depletion capacitance, is a weak function of the applied voltage.

11.5.2.1 Diffusion Capacitance

In a forward-biased diode, minority carriers are injected into opposite sides of the junction. Those minority carriers diffuse from the junction and recombine with the majority carriers. Figure 11.72 shows the distribution of minority carriers in the vicinity of the junction of uniformly doped n-type and p-type regions. The electron charge stored in the p-region corresponds to the area under the curve, and it is equal to $Q_{\text{n}} = qn_0L_{\text{n}}$. Similarly, the charge of stored holes $Q_{\text{p}} = qp_0\ L_{\text{p}}$. The storage charge can also be expressed as $Q_{\text{n}} = I_{\text{n}}\ \tau_{\text{n}}$ and $Q_{\text{p}} = I_{\text{p}}\ \tau_{\text{p}}$, where I_{n} and I_{p} are electron and hole currents at the junction and

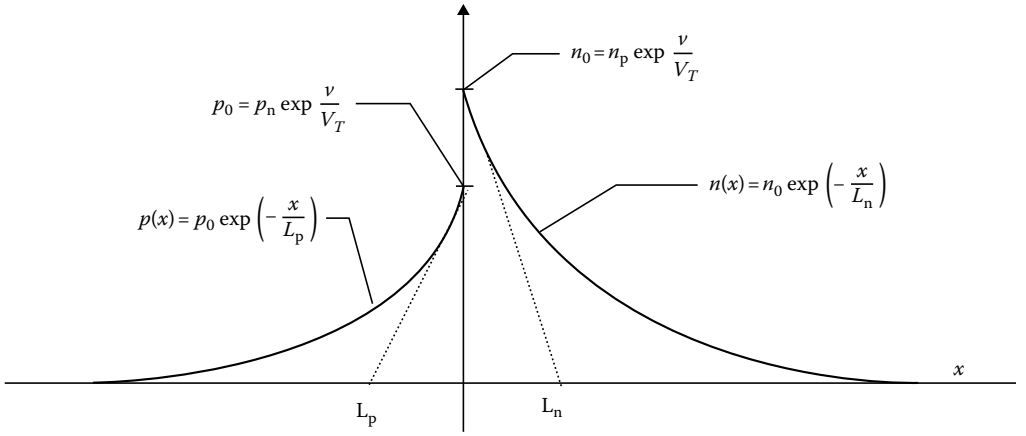


FIGURE 11.72 Minority carrier distribution in the vicinity of the p–n junction biased in the forward direction.

τ_n and τ_p are the lifetimes for minority carriers. Assuming $\tau = \tau_n = \tau_p$ and knowing that $I = I_p + I_n$ the total storage charge at the junction is $Q = I\tau$. The diffusion capacitance can be then computed as

$$C_{dif} = \frac{dQ}{dv} = \frac{d}{dv}\left[\tau I_0 \exp\left(\frac{v}{\eta V_T}\right)\right] = \frac{\tau I_B}{\eta V_T} \tag{11.113}$$

As one can see, the diffusion capacitance C_{dif} is proportional to the storage time τ and to the diode biasing current I_B. Note that the diffusion capacitance does not depend on the junction area, but only on the diode current. The diffusion capacitances may have very large values. For example, for 100 mA current and $\tau = 1$ μs, the junction diffusion capacitance is about 4 μF. Fortunately, this diffusion capacitance is connected in parallel to the small-signal junction resistance $r = \eta V_T / I_B$, and the time constant r. C_{dif} is equal to the storage time τ.

11.5.3 Depletion Capacitance

The reversed-biased diode looks like a capacitor with two ''plates'' formed of p-type and n-type regions and a dielectric layer (depletion region) between them. The capacitance of a reversed-biased junction can then be written as

$$C_{dep} = A\frac{\varepsilon}{w} \tag{11.114}$$

where

A is a junction area
ε is the dielectric permittivity of semiconductor material
w is the thickness of the depletion layer

The depletion layer thickness w is a weak function of the applied reverse-biasing voltage. In the simplest case, with a step-abrupt junction, the depletion capacitance is

$$C_j = \sqrt{\frac{qN_{eff}\varepsilon\varepsilon_0}{2(V_{pn} - v)}}; \quad \frac{1}{N_{eff}} = \frac{1}{N_D} + \frac{1}{N_A} \tag{11.115}$$

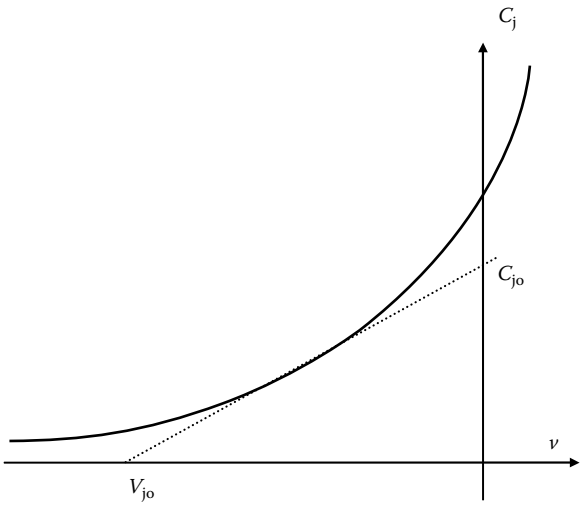


FIGURE 11.73 Capacitance–voltage characteristics for reverse-biased junction.

The steepest capacitance–voltage characteristics are in $p^+ - i - p - n^+$ diodes with the impurity profiles shown in Figure 11.67f. In general, for various impurity profiles at the junction, the depletion capacitance C_j can be approximated by

$$C_j = C_{j0}\left(1 - \frac{v}{V_{pn}}\right)^{\frac{1}{m}} \tag{11.116}$$

or using linear approximation as shown in Figure 11.73:

$$C_j = C_{j0}\left(1 - \frac{v}{V_{j0}}\right) \tag{11.117}$$

11.5.4 Diode as a Switch

The switching time of the p–n junction is limited mainly by the storage charge of injected minority carriers into the vicinity of the junction (electrons injected in p-type region and holes injected in n-type region). When a diode is switched from the forward to the reverse direction, these carriers may move freely through the junction. Some of the minority carriers recombine with time. Others are moved away to the other side of the junction. The diode cannot recover its blocking capability as long as a large number of the minority carriers exist and can flow through the junction. An example of the current–time characteristics of a diode switching from the forward to the reverse direction is shown in Figure 11.74. A few characteristics that are shown in the figure are for the same forward current and different reverse currents. Just after switching, these reverse currents are limited only by external circuitry. In this example, shown in Figure 11.74, most of the minority carriers are moved to the other side of the junction by the reverse current, and the recombination mechanism is negligible. Note that the larger the reverse current flowing after switching, the shorter the time required to recover the blocking capability. This type of behavior is typical for commonly used high-voltage diodes.

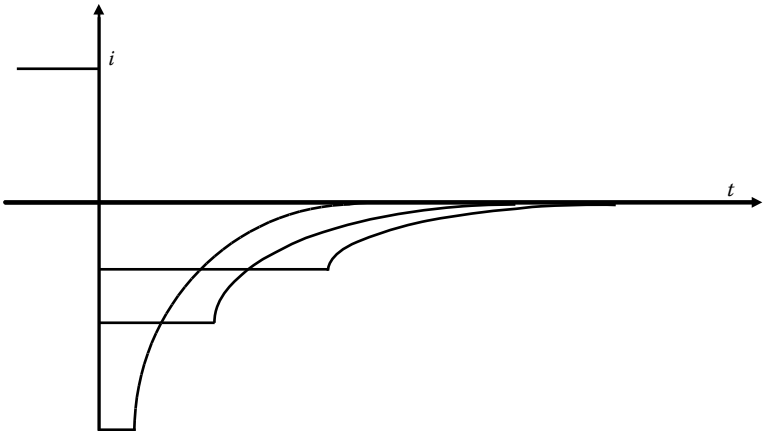


FIGURE 11.74 Currents in diode with large minority carrier lifetimes after switching from the forward to the reverse direction.

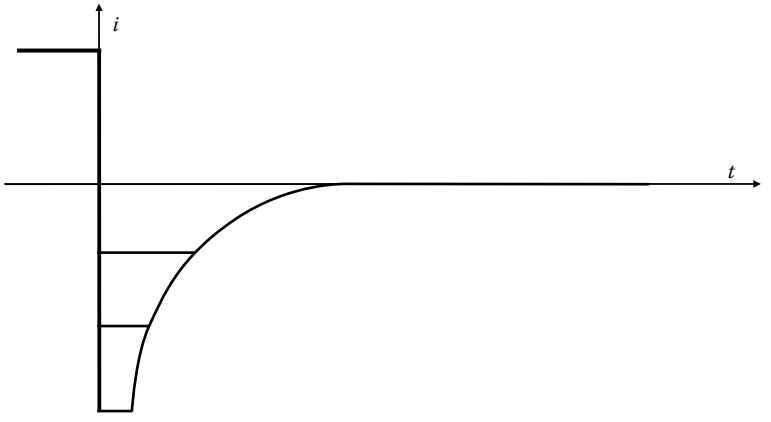


FIGURE 11.75 Currents in diode with small minority carrier lifetimes after switching from the forward to the reverse direction.

In order to shorten the switching time, diodes sometimes are doped with gold or other deep-level impurities to create more generation centers and to increase the carrier recombination. This way, the minority carrier lifetimes of such switching diodes are significantly reduced. The switching time is significantly shorter, but it is almost independent of the reverse-diode current after switching, as shown in Figure 11.75. This method of artificially increasing recombination rates has some severe disadvantages. Such switching diodes are characterized by very large reverse leakage currents and small breakdown voltages.

The best switching diodes utilize metal–semiconductor contacts. They are known as Schottky diodes. In such diodes there is no minority carrier injection phenomenon; therefore, these diodes recover the blocking capability instantaneously. Schottky diodes are also characterized by a relatively small (0.2–0.3 V) voltage drop in the forward direction. However, their reverse leakage current is larger, and the breakdown voltage rarely exceeds 20–30 V. Lowering the impurity concentration in the

semiconductor material leads to slightly larger breakdown voltages, but at the same time, the series diode resistances increase significantly.

11.5.5 Temperature Properties

Both forward and reverse diode characteristics are temperature dependent. These temperature properties are very important for correct circuit design. The temperature properties of the diode can be used to compensate for the thermal effects of electronic circuits. Diodes can be used also as accurate temperature sensors. The major temperature effect in a diode is caused by the strong temperature dependence of the intrinsic concentration n_i (Equations 11.101 and 11.102) and by the exponential temperature relationship of the diode Equation 11.104. By combining Equations 11.101 and 11.104 and assuming the temperature dependence of carrier mobilities, the voltage drop on the forward-biased diode can be written as

$$v = \eta\left[V_T \ln\left(\frac{i}{\xi T^a}\right) + V_g\right] \tag{11.118}$$

or diode current

$$i = I_0\left(\frac{T}{T_0}\right)^{\alpha} \exp\left(\frac{(v/\eta) - V_g}{V_{T_0}}\frac{T_0}{T}\right) \tag{11.119}$$

where

V_g is the potential gap in semiconductor material, $V_g = 1.124$ V for silicon and $V_g = 1.424$ V for GaAs
α is a material coefficient ranging between 2.5 and 4.0

The temperature dependence of the diode voltage drop dv/dT can be obtained by calculating the derivative of Equation 11.118

$$\frac{dv}{dT} = \frac{v - \eta\left(V_g + \alpha V_T\right)}{T} \tag{11.120}$$

For example, in the case of the silicon diode with a 0.6 voltage drop and assuming $\eta = 1.1$, $\alpha = 3.0$, and $T = 300$ K, the $dV/dT = 1.87$ mV/°C.

The reverse-diode current is a very strong function of the temperature. For diodes made of semiconductor material with a small potential gap, such as germanium, the diffusion component dominates. In this case, the reverse current is proportional to

$$i_{rev} \propto T^{\alpha} \exp\left(-\frac{qV_g}{kT}\right) \tag{11.121}$$

For diodes made of silicon and semiconductors with a higher energy gap, the recombination is the dominant mechanism. In this case, reverse leakage current is proportional to

$$i_{rev} \propto T^{\alpha} \exp\left(-\frac{qV_g}{2kT}\right) \tag{11.122}$$

Using Equation 11.122, one may calculate that for silicon diodes at room temperatures, the reverse leakage current doubles for about every 10°C.

The breakdown voltage is also temperature dependent. The tunneling effect dominates in diodes with small breakdown voltages. This effect is often known in literature as the Zener breakdown. In such diodes the breakdown voltage decreases with the temperature. The avalanche breakdown dominates in diodes with large breakdown voltages. When the avalanche mechanism prevails, then the breakdown voltage increases 0.06%/°C–0.1%/°C. For medium-range breakdown voltages, one phenomenon compensates the other, and temperature-independent breakdown voltage can be observed. This zero temperature coefficient exists for diodes with breakdown voltages equal to about $5V_g$. In the case of the silicon diode, this breakdown voltage, with a zero temperature coefficient, is equal to about 5.6 V.

11.5.6 Piecewise Linear Model

Nonlinear diode characteristics are often approximated by a piecewise linear model. A few possible approaches can be used to linearize the diode characteristics (see Figure 11.76). The parameters of the most accurate linearized diode model are shown in Figure 11.77a, and the linearized diode equivalent circuit is shown in Figure 11.77b.

The modified diode Equation 11.108 also can be written as

$$v = \eta V_T \ln\left(\frac{i}{I_0}\right) \tag{11.123}$$

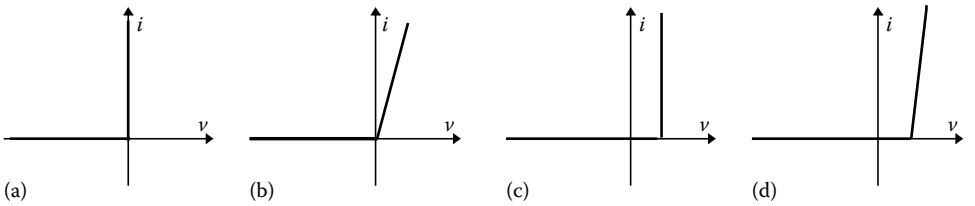


FIGURE 11.76 Various ways to linearize diode characteristics.

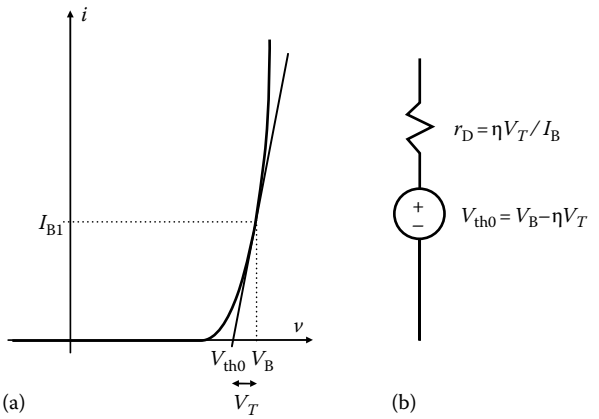


FIGURE 11.77 Linearization of the diode: (a) diode characteristics and (b) equivalent diagram.

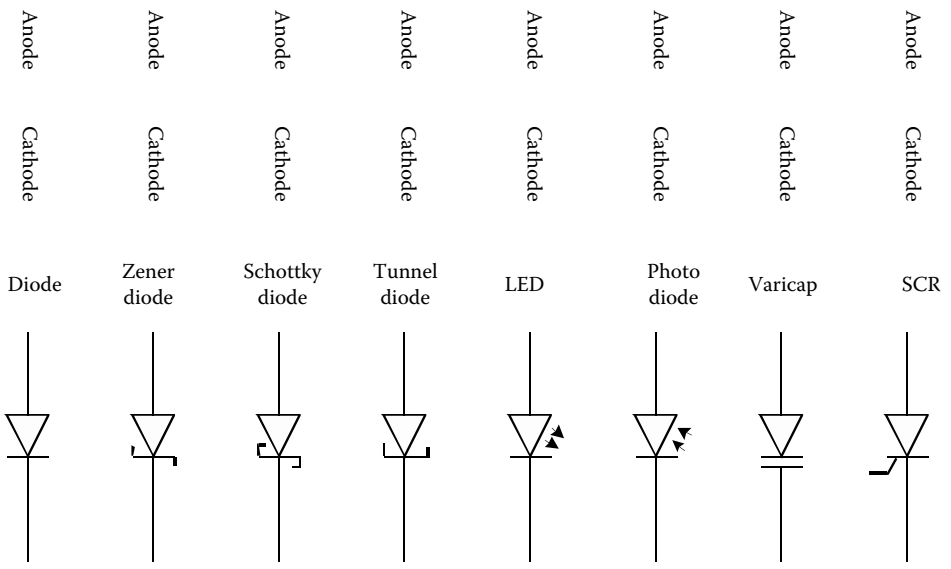


FIGURE 11.78 Symbols used for various types of diodes.

For the biasing points V_B and I_B, the small-signal diode resistance dv/di can be computed from Equation 11.123 as

$$r = \frac{dv}{di} = \frac{\eta V_T}{I_B}; \quad V_{th0} = V_B - \eta V_T \tag{11.124}$$

and it is only the function of the thermal potential V_T and the biasing current I_B. Note that the small-signal diode resistance is almost independent of the diode construction or semiconductor material used. If one requires that this linearized diode have I_B current for V_B voltage, then the piecewise diode characteristics should be as in Figure 11.77a. The equivalent Thevenin and Norton circuits are shown in Figure 11.77b. In the case of large-signal operation, the diode can be approximated by shifting the characteristics to the left by ΔV. In this case, the threshold voltage becomes $V_{th0} = V_B - 2V_T$ instead of $V_{th0} = V_B - \eta^{VT}$.

References

1. A. S. Grove, *Physics and Technology of Semiconductor Devices*, New York: John Wiley & Sons, 1967.
2. S. M. Sze, *Physics of Semiconductor Devices*, 2nd edn., New York: John Wiley & Sons, 1981.
3. G. W. Neudeck, *The PN Junction Diode*, Vol. II: Modular Series on Solid-State Devices, Reading, MA: Addison-Wesley, 1983.
4. R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, 2nd edn., New York: John Wiley & Sons, 1986.
5. E. S. Yang, *Microelectronic Devices*, New York: McGraw-Hill, 1988.
6. B. G. Streetman, *Solid State Electronic Devices*, 3rd edn., Englewood Cliffs, NJ: Prentice Hall, 1990.
7. D. A. Neamen, *Semiconductor Physics and Devices*, Homewood, IL: Irwin, 1992.

12

RF Passive IC Components

Tomas H. Lee
Stanford University

Maria del Mar Hershenson
Stanford University

Sunderarajan S. Mohan
Stanford University

Hirad Samavati
Stanford University

C. Patrick Yue
Stanford University

12.1 Introduction **12**-1
12.2 Fractal Capacitors **12**-1
Lateral Flux Capacitors • Fractals • Fractal Capacitor Structures
12.3 Spiral Inductors **12**-8
Understanding Substrate Effects • Simple, Accurate Expressions for Planar Spiral Inductances
12.4 On-Chip Transformers **12**-14
Monolithic Transformer Realizations • Analytical Transformer Models
References **12**-19

12.1 Introduction

Passive energy storage elements are widely used in radio-frequency (RF) circuits. Although their impedance behavior often can be mimicked by compact active circuitry, it remains true that passive elements offer the largest dynamic range and the lowest power consumption. Hence, the highest performance will always be obtained with passive inductors and capacitors. Unfortunately, standard integrated circuit technology has not evolved with a focus on providing good passive elements. This chapter describes the limited palette of options available, as well as means to make the most use out of what is available.

12.2 Fractal Capacitors

Of capacitors, the most commonly used are parallel-plate and MOS structures. Because of the thin gate oxides now in use, capacitors made out of MOSFETs have the highest capacitance density of any standard IC option, with a typical value of approximately 7 fF/μm^2 for a gate oxide thickness of 5 nm. A drawback, however, is that the capacitance is voltage dependent. The applied potential must be well in excess of a threshold voltage in order to remain substantially constant. The relatively low breakdown voltage (on the order of 0.5 V/nm of oxide) also imposes an unwelcome constraint on allowable signal amplitudes. An additional drawback is the effective series resistance of such structures, due to the MOS channel resistance. This resistance is particularly objectionable at radio frequencies, since the impedance of the combination may be dominated by this resistive portion.

Capacitors that are free of bias restrictions (and that have much lower series resistance) may be formed out of two (or more) layers of standard interconnect metal. Such parallel-plate capacitors are quite linear and possess high breakdown voltage, but generally offer two orders of magnitude lower capacitance density than the MOSFET structure. This inferior density is the consequence of a conscious and continuing effort by technologists to keep low the capacitance between interconnect layers. Indeed, the vertical spacing between such layers generally does not scale from generation to generation. As a result, the disparity between MOSFET capacitance density and that of the parallel-plate structure continues to grow as technology scales.

A secondary consequence of the low density is an objectionably high capacitance between the bottom plate of the capacitor and the substrate. This bottom-plate capacitance is often a large fraction of the main capacitance. Needless to say, this level of parasitic capacitance is highly undesirable.

In many circuits, capacitors can occupy considerable area, and an area-efficient capacitor is therefore highly desirable. Recently, a high-density capacitor structure using lateral fringing and fractal geometries has been introduced [1]. It requires no additional processing steps, and so it can be built in standard digital processes. The linearity of this structure is similar to that of the conventional parallel-plate capacitor. Furthermore, the bottom-plate parasitic capacitance of the structure is small, which makes it appealing for many circuit applications. In addition, unlike conventional metal-to-metal capacitors, the density of a fractal capacitor increases with scaling.

12.2.1 Lateral Flux Capacitors

Figure 12.1a shows a lateral flux capacitor. In this capacitor, the two terminals of the device are built using a single layer of metal, unlike a vertical flux capacitor, where two different metal layers must be used. As process technologies continue to scale, lateral fringing becomes more important. The lateral spacing of the metal layers, s, shrinks with scaling, yet the thickness of the metal layers, t, and the vertical spacing of the metal layers, t_{ox}, stay relatively constant. This means that structures utilizing lateral flux enjoy a significant improvement with process scaling, unlike conventional structures that depend on vertical flux. Figure 12.1b shows a scaled lateral flux capacitor. It is obvious that the capacitance of the structure of Figure 12.1b is larger than that of Figure 12.1a.

Lateral flux can be used to increase the total capacitance obtained in a given area. Figure 12.2a is a standard parallel-plate capacitor. In Figure 12.2b, the plates are broken into cross-connected sections [2]. As can be seen, a higher capacitance density can be achieved by using lateral flux as well as vertical flux. To emphasize that the metal layers are cross connected, the two terminals of the capacitors in Figure 12.2b are identified with two different shadings. The idea can be extended to multiple metal layers as well.

Figure 12.3 shows the ratio of metal thickness to minimum lateral spacing, t/s, versus channel length for various technologies [3–5]. The trend suggests that lateral flux will have a crucial role in the design of capacitors in future technologies.

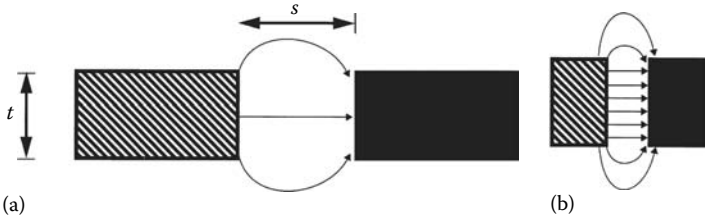


FIGURE 12.1 Effect of scaling on lateral flux capacitors: (a) before scaling and (b) after scaling.

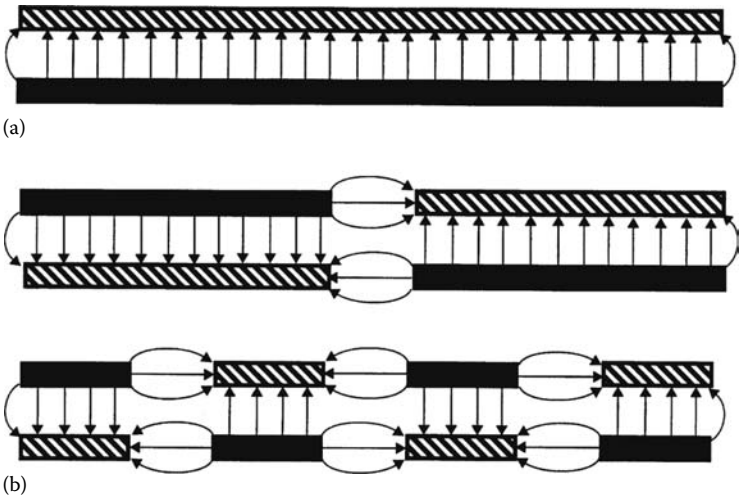


FIGURE 12.2 Vertical flux vs. lateral flux: (a) a standard parallel-plate structure and (b) cross-connected metal layers.

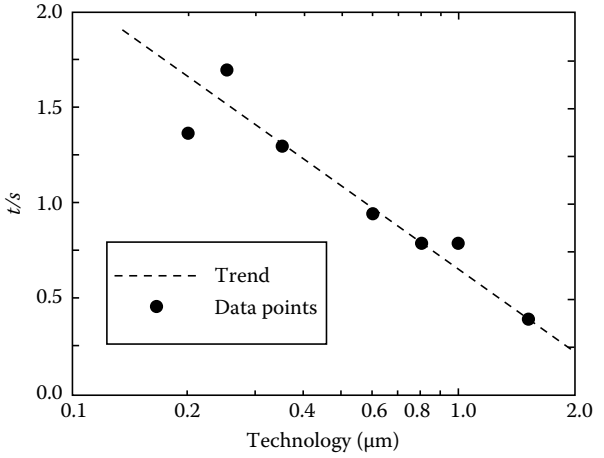


FIGURE 12.3 Ratio of metal thickness to horizontal metal spacing vs. technology (channel length).

The increase in capacitance due to fringing is proportional to the periphery of the structure; therefore, structures with large periphery per unit area are desirable. Methods for increasing this periphery are the subject of the following sections.

12.2.2 Fractals

A fractal is a mathematical abstract [6]. Some fractals are visualizations of mathematical formulas, while others are the result of the repeated application of an algorithm, or a *rule*, to a *seed*. Many natural phenomena can be described by fractals. Examples include the shapes of mountain ranges, clouds, coastlines, etc.

Some ideal fractals have finite area but infinite perimeter. The concept can be better understood with the help of an example. *Koch islands* are a family of fractals first introduced as a crude model for the shape of a

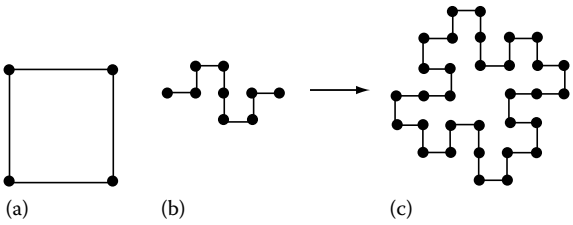


FIGURE 12.4 Construction of a Koch curve: (a) an initiator, (b) a generator, and (c) first step of the process.

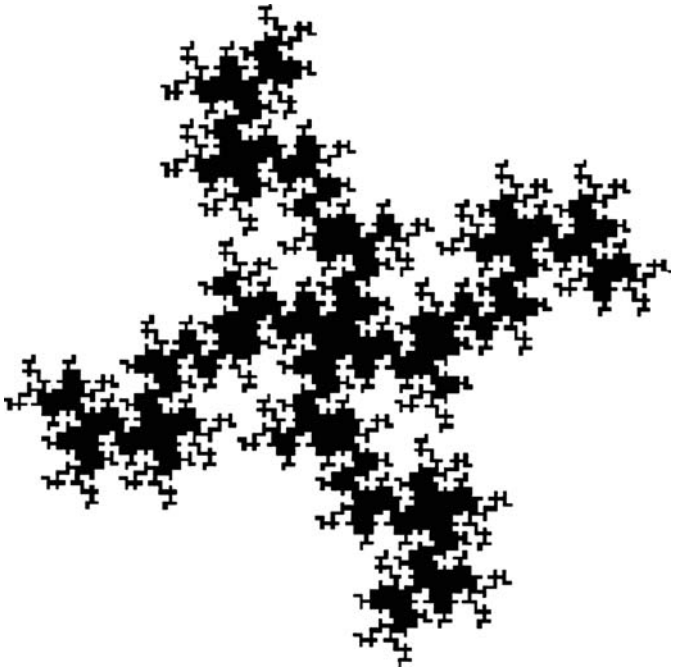


FIGURE 12.5 A Koch island with $M = 4$, $N = 32$, and $r = 1/8$.

coastline. The construction of a Koch curve begins with an *initiator*, as shown in the example of Figure 12.4a. A square is a simple initiator with $M = 4$ sides. The construction continues by replacing each segment of the initiator with a curve called a *generator*, an example of which is shown in Figure 12.4b that has $N = 8$ segments. The size of each segment of the generator is $r = 1/4$ of the initiator. By recursively replacing each segment of the resulting curve with the generator, a fractal border is formed. The first step of this process is depicted in Figure 12.4c. The total area occupied remains constant throughout the succession of stages because of the particular shape of the generator. A more complicated Koch island can be seen in Figure 12.5. The associated initiator of this fractal has four sides and its generator has 32 segments. It can be noted that the curve is self-similar, that is, each section of it looks like the entire fractal. As we zoom in on Figure 12.5, more detail becomes visible, and this is the essence of a fractal.

Fractal dimension, D, is a mathematical concept that is a measure of the complexity of a fractal. The dimension of a flat curve is a number between 1 and 2, which is given by

$$D = \frac{\log(N)}{\log\left(\frac{1}{r}\right)} \tag{12.1}$$

where

N is the number of segments of the generator

r is the ratio of the generator segment size to the initiator segment size

The dimension of a fractal curve is not restricted to integer values, hence, the term "fractal." In particular, it exceeds 1, which is the intuitive dimension of curves. A curve that has a high degree of complexity, or D, fills out a two-dimensional flat surface more efficiently. The fractal in Figure 12.4c has a dimension of 1.5, whereas for the border line of Figure 12.5, $D = 1.667$.

For the general case where the initiator has M sides, the periphery of the initiator is proportional to the square root of the area:

$$P_0 = k \cdot \sqrt{A} \tag{12.2}$$

where k is a proportionality constant that depends on the geometry of the initiator. For example, for a square initiator, $k = 4$; and for an equilateral triangle, $k = 2 \cdot \sqrt[4]{27}$. After n successive applications of the generation rule, the total periphery is

$$P = k\sqrt{A} \cdot (Nr)^n \tag{12.3}$$

and the minimum feature size (the resolution) is

$$l = \frac{k\sqrt{A}}{M} \cdot r^n \tag{12.4}$$

Eliminating n from Equations 12.3 and 12.4 and combining the result with Equation 12.1, we have

$$P = \frac{k^D}{M^{D-1}} \cdot \frac{(\sqrt{A})^D}{l^{D-1}} \tag{12.5}$$

Equation 12.5 demonstrates the dependence of the periphery on parameters such as the area and the resolution of the fractal border. It can be seen from Equation 12.5 that as l tend toward zero, the periphery goes to infinity; therefore, it is possible to generate fractal structures with very large perimeters in any given area. However, the total periphery of a fractal curve is limited by the attainable resolution in practical realizations.

12.2.3 Fractal Capacitor Structures

The final shape of a fractal can be tailored to almost any form. The flexibility arises from the fact that a wide variety of geometries can be used as the initiator and generator. It is also possible to use different generators during each step. This is an advantage for integrated circuits where flexibility in the shape of the layout is desired.

Figure 12.6 is a three-dimensional representation of a fractal capacitor. This capacitor uses only one metal layer with a fractal border. For a better visualization of the overall picture, the terminals of this square-shaped capacitor have been identified using two different shadings. As was discussed before, multiple cross-connected metal layers may be used to improve capacitance density further.

One advantage of using lateral flux capacitors in general, and fractal capacitors in particular, is the reduction of the bottom-plate capacitance. This reduction is due to two reasons. First, the higher density of the fractal capacitor (compared to a standard parallel-plate structure) results in a smaller area. Second, some of the field lines originating from one of the bottom plates terminate on the adjacent plate, instead of the substrate, which further reduces the bottom-plate capacitance as shown in Figure 12.7. Because of

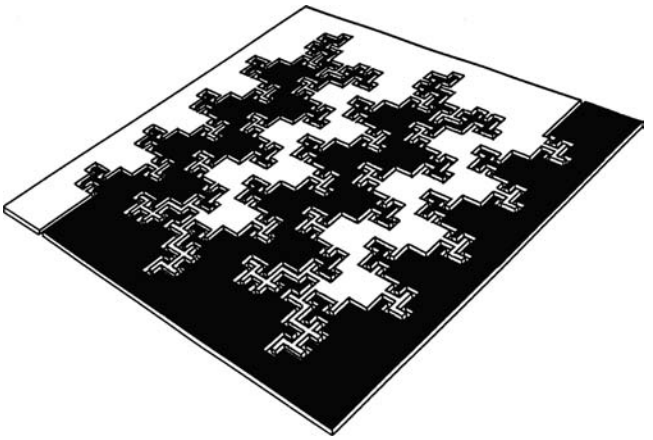


FIGURE 12.6 Three-dimensional representation of a fractal capacitor using a single metal layer.

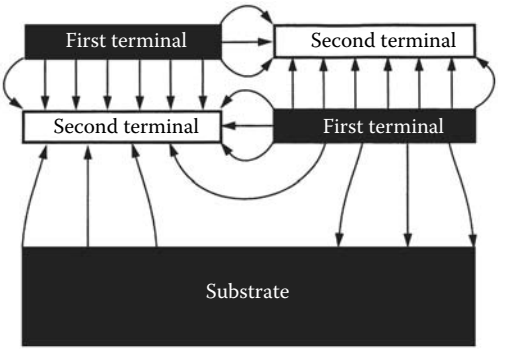


FIGURE 12.7 Reduction of the bottom-plate parasitic capacitance.

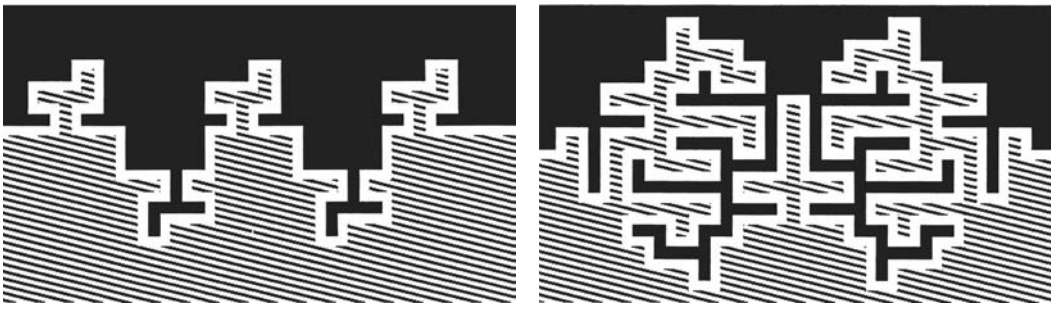


FIGURE 12.8 Fractal dimension of (a) is smaller than (b).

this property, some portion of the parasitic bottom-plate capacitor is converted into the more useful plate-to-plate capacitance.

The capacitance per unit area of a fractal structure depends on the dimension of the fractal. To improve the density of the layout, fractals with large dimensions should be used. The concept of fractal dimension is demonstrated in Figure 12.8. The structure in Figure 12.8a has a lower dimension compared to the one in Figure 12.8b, so the density (capacitance per unit area) of the latter is higher.

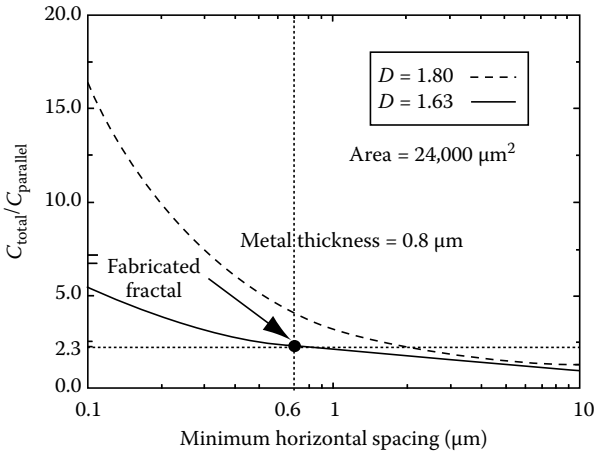


FIGURE 12.9 Boost factor vs. horizontal spacing.

To demonstrate the dependence of capacitance density on dimension and lateral spacing of the metal layers, a first-order electromagnetic simulation was performed on two families of fractal structures. In Figure 12.9, the boost factor vs. horizontal spacing of the metal layers is plotted. The *boost factor* is defined as the ratio of the total capacitance of the fractal structure to the capacitance of a standard parallel-plate structure with the same area. The solid line corresponds to a family of fractals with a moderate fractal dimension of 1.63, while the dashed line represents another family of fractals with $D = 1.80$, which is a relatively large value for the dimension. In this first-order simulation, it is assumed that the vertical spacing and the thickness of the metal layers are kept constant at a 0.8 μm level. As can be seen in Figure 12.9, the amount of boost is a strong function of the fractal dimension as well as scaling.

In addition to the capacitance density, the quality factor, Q, is important in RF applications. Here, the degradation in quality factor is minimal because the fractal structure automatically limits the length of the thin metal sections to a few microns, keeping the series resistance reasonably small. For applications that require low series resistance, lower dimension fractals may be used. Fractals thus add one more degree of freedom to the design of capacitors, allowing the capacitance density to be traded for a lower series resistance.

In current IC technologies, there is usually tighter control over the lateral spacing of metal layers compared with the vertical thickness of the oxide layers, from wafer to wafer and across the same wafer. Lateral flux capacitors shift the burden of matching away from oxide thickness to lithography. Therefore, by using lateral flux, matching characteristics can improve. Furthermore, the pseudorandom nature of the structure can also compensate, to some extent, the effects of nonuniformity of the etching process. To achieve accurate ratio matching, multiple copies of a unit cell should be used, as is standard practice in high-precision analog circuit design.

Another simple way of increasing capacitance density is to use an interdigitated capacitor depicted in Figure 12.10 [2,7]. One disadvantage of such a structure compared to fractals is its inherent parasitic inductance. Most of the fractal geometries randomize the direction of the current flow and thus reduce the effective series inductance; whereas for interdigitated capacitors, the current flow is in the same direction for all the parallel stubs. In addition, fractals usually have lots of rough edges that accumulate electrostatic energy more efficiently compared to interdigitated capacitors, causing a boost in capacitance (generally of the order of 15%). Furthermore, interdigitated structures are more vulnerable to nonuniformity of the etching process. However, the relative simplicity of the interdigitated capacitor does make it useful in some applications.

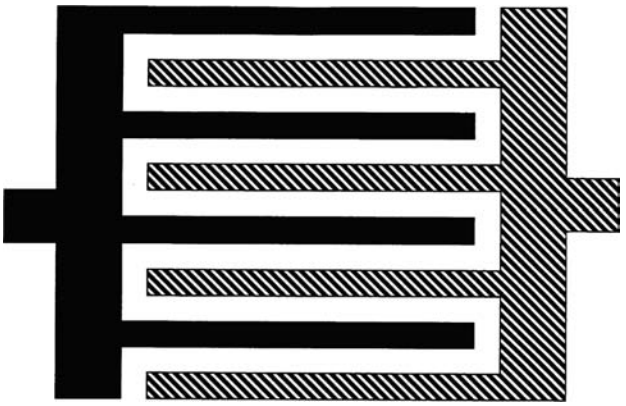


FIGURE 12.10 An interdigitated capacitor.

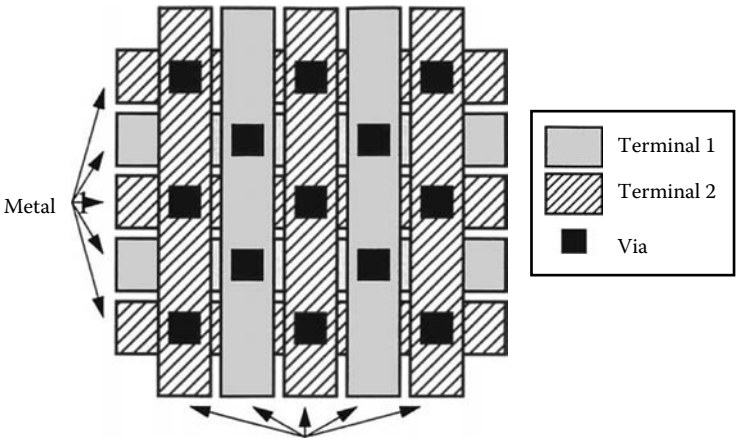


FIGURE 12.11 A woven structure.

The woven structure shown in Figure 12.11 may also be used to achieve high capacitance density. The vertical lines are in metal-2 and horizontal lines are in metal-1. The two terminals of the capacitor are identified using different shades. Compared to an interdigitated capacitor, a woven structure has much less inherent series inductance. The current flowing in different directions results in a higher self-resonant frequency. In addition, the series resistance contributed by vias is smaller than that of an interdigitated capacitor, because cross-connecting the metal layers can be done with greater ease. However, the capacitance density of a woven structure is smaller compared to an interdigitated capacitor with the same metal pitch, because the capacitance contributed by the vertical fields is smaller.

12.3 Spiral Inductors

More than is so with capacitors, on-chip inductor options are particularly limited and unsatisfactory. Nevertheless, it is possible to build practical spiral inductors with values up to perhaps 20 nH and with Q values of approximately 10. For silicon-based RF ICs, Q degrades at high frequencies due to energy dissipation in the semiconducting substrate [8]. Additionally, noise coupling via the substrate at GHz frequencies has been reported [9]. As inductors occupy substantial chip area, they can potentially

be the source and receptor of detrimental noise coupling. Furthermore, the physical phenomena underlying the substrate effects are complicated to characterize. Therefore, decoupling the inductor from the substrate can enhance the overall performance by increasing Q, improving isolation, and simplifying modeling.

Some approaches have been proposed to address the substrate issues; however, they are accompanied by drawbacks. Some [10] have suggested the use of high-resistivity (150–200 Ω-cm) silicon substrates to mimic the low-loss, semi-insulating GaAs substrate, but this is rarely a practical option. Another approach selectively removes the substrate by etching a pit under the inductor [11]. However, the etch adds extra processing cost and is not readily available. Moreover, it raises reliability concerns such as packaging yield and long-term mechanical stability. For low-cost integration of inductors, the solution to substrate problems should avoid increasing process complexity.

In this section, we present the *patterned ground shield* (PGS) [12], which is compatible with standard silicon technologies, and which reduces the unwanted substrate effects. The great improvement provided by the PGS reduces the disparity in quality between spiral inductors made in silicon and GaAs IC technologies.

12.3.1 Understanding Substrate Effects

To understand why the PGS should be effective, consider first the physical model of an ordinary inductor on silicon, with one port and the substrate grounded, as shown in Figure 12.12 [8]. An on-chip inductor is physically a three-port element including the substrate. The one-port connection shown in Figure 12.12 avoids unnecessary complexity in the following discussion and at the same time preserves the inductor characteristics. In the model, the series branch consists of L_s, R_s, and C_s. L_s represents the spiral inductance, which can be computed using the Greenhouse method [13] or well approximated by simple analytical formulas to be presented later. R_s is the metal series resistance whose behavior at RF is governed by the eddy current effect. This resistance accounts for the energy loss due to the skin effect in the spiral interconnect structure as well as the induced eddy current in any conductive media close to the inductor. The series feedforward capacitance, C_s, accounts for the capacitance due to the overlaps between the spiral and the center-tap underpass [14]. The effect of the interturn fringing capacitance is usually small because the adjacent turns are almost at equal potentials, and therefore it is neglected in this model. The overlap capacitance is more significant because of the relatively large potential difference between the spiral and the center-tap underpass. The parasitics in the shunt branch are modeled by C_{ox}, C_{Si}, and R_{Si}. C_{ox} represents the oxide capacitance between the spiral and the substrate. The silicon substrate capacitance and resistance are modeled by C_{Si} and R_{Si}, respectively [15,16]. The element R_{Si} accounts for the energy dissipation in the silicon substrate.

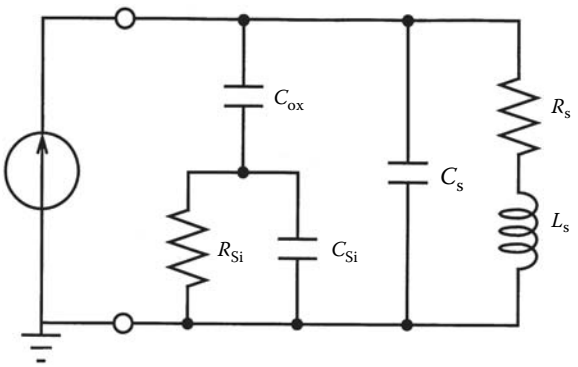


FIGURE 12.12 Lumped physical model of a spiral inductor on silicon.

Expressions for the model element values are as follows:

$$R_s = \frac{\rho l}{\delta w\left(1 - e^{-\frac{t}{\delta}}\right)} \tag{12.6}$$

$$C_s = nw^2 \cdot \frac{\varepsilon_{ox}}{t_{oxM1-M2}} \tag{12.7}$$

$$C_{ox} = \frac{\varepsilon_{ox}}{2t_{ox}} \cdot l \cdot w \tag{12.8}$$

$$C_{Si} = \frac{1}{2} \cdot l \cdot w \cdot C_{sub} \tag{12.9}$$

$$R_{Si} = \frac{2}{l \cdot w \cdot G_{sub}} \tag{12.10}$$

where

ρ is the DC resistivity of the spiral
t is the overall length of the spiral windings
w is the line width
δ is the skin depth
n is the number of crossovers between the spiral and center-tap (and thus $n = N - 1$, where N is the number of turns)
$t_{oxM1-M2}$ is the oxide thickness between the spiral and substrate
C_{sub} is the substrate capacitance per unit area
G_{sub} is the substrate conductance per unit area

In general, one treats C_{sub} and G_{sub} as fitting parameters.

Exploration with the model reveals that the substrate loss stems primarily from the penetration of the electric field into the lossy silicon substrate. As the potential drop in the semiconductor (i.e., across R_{Si} in Figure 12.12) increases with frequency, the energy dissipation in the substrate becomes more severe. It can be seen that increasing R_p to infinity reduces the substrate loss. It can be demonstrated that R_p approaches infinity as R_{Si} goes either to zero or infinity. This observation implies that Q can be improved by making the silicon substrate either a perfect insulator or a perfect conductor. Using high-resistivity silicon (or etching it away) is equivalent to making the substrate an open circuit. In the absence of the freedom to do so, the next best option is to convert the substrate into a better conductor. The approach is to insert a ground plane to block the inductor electric field from entering the silicon. In effect, this ground plane becomes a pseudosubstrate with the desired characteristics.

The ground shield cannot be a solid conductor, however, because image currents would be induced in it. These image currents tend to cancel the magnetic field of the inductor proper, decreasing the inductance. To solve this problem, the ground shield is patterned with slots orthogonal to the spiral as illustrated in Figure 12.13. The slots act as an open circuit to cut off the path of the induced loop current. The slots should be sufficiently narrow such that the vertical electric field cannot leak through the patterned ground shield into the underlying silicon substrate. With the slots etched away, the ground strips serve as the termination for the electric field. The ground strips are merged together around the four outer edges of the spiral. The separation between the merged area and the edges is not critical. However, it is crucial that the merged area does not form a closed ring around the spiral since it can potentially support the unwanted loop current. The shield should be strapped with the top layer metal to provide a low-impedance path to ground. The general rule is to prevent negative mutual coupling while minimizing the impedance to ground.

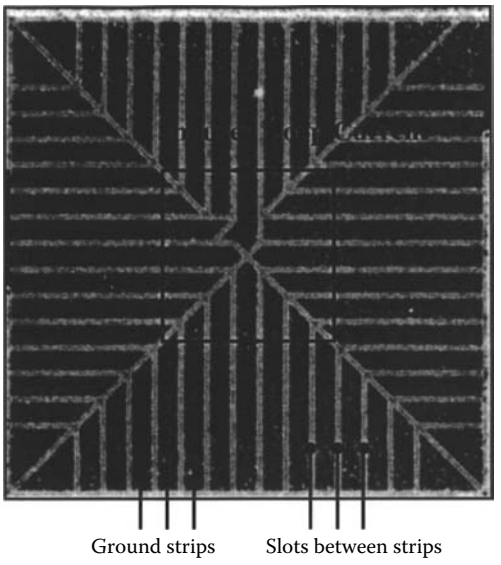


FIGURE 12.13 A close-up photo of the patterned ground shield.

The shield resistance is another critical design parameter. The purpose of the patterned ground shield is to provide a good short to ground for the electric field. Because the finite shield resistance contributes to energy loss of the inductor, it must be kept small. Specifically, by keeping the shield resistance small compared to the reactance of the oxide capacitance, the voltage drop that can develop across the shield resistance is very small. As a result, the energy loss due to the shield resistance is insignificant compared to other losses. A typical on-chip spiral inductor has parasitic oxide capacitance between 0.25 and 1 pF, depending on the size and the oxide thickness. The corresponding reactance due to the oxide capacitance at 1–2 GHz is of the order of 100 Ω and hence a shield resistance of a few ohms is sufficiently small not to cause any noticeable loss.

With the PGS, one can expect typical improvements in Q ranging from 10% to 33%, in the frequency range of 1–2 GHz. Note that the inclusion of the ground shields increases C_p, which causes a fast roll-off in Q above the peak-Q frequency and a reduction in the self-resonant frequency. This modest improvement in inductor Q is certainly welcome, but is hardly spectacular by itself. However, a more dramatic improvement is evident when evaluating inductor–capacitor resonant circuits. Such LC tank circuits can absorb the parasitic capacitance of the ground shield. The energy stored in such parasitic elements is now part of the circuit; therefore, the overall circuit Q is greatly increased. Improvements of factors of approximately two are not unusual, so that tank circuits realized with PGS inductors possess roughly the same Q as those built in GaAs technologies.

As stated earlier, substrate noise coupling can be an issue of great concern owing to the relatively large size of typical inductors. Shielding by the PGS improves isolation by 25 dB or more at GHz frequencies. It should be noted that, as with any other isolation structure (such as a guard ring), the efficacy of the PGS is highly dependent on the integrity of the ground connection. One must often make a trade-off between the desired isolation level and the chip area that is required to provide a low-impedance ground connection.

12.3.2 Simple, Accurate Expressions for Planar Spiral Inductances

In Section 12.3.1, a physically based model for planar spiral inductors was offered, and reference was made to the Greenhouse method as a means for computing the inductance value. This method uses as computational atoms the self- and mutual inductances of parallel current strips. It is relatively straightforward to apply, and yields accurate results. Nevertheless, simpler analytic formulas are generally preferred for design since important insights are usually more readily obtained.

As a specific example, square spirals are popular mainly because of their ease of layout. Other polygonal spirals have also been used to improve performance by more closely approximating a circular spiral. However, a quantitative evaluation of possible improvements is cumbersome without analytical formulas for inductance.

Among alternative shapes, hexagonal and octagonal inductors are used widely. Figures 12.14 through 12.16 show the layout for square, hexagonal, and octagonal inductors, respectively. For a given shape, an inductor is completely specified by the number of turns n, the turn width w, the turn spacing s,

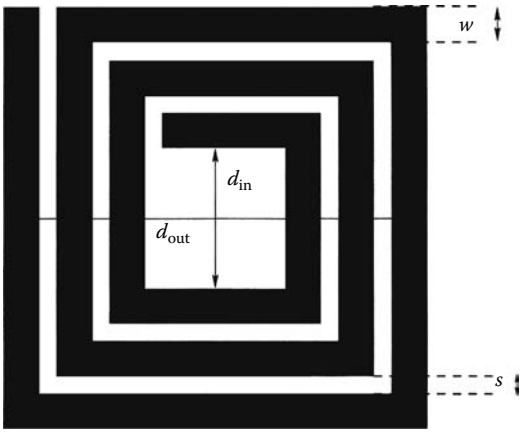


FIGURE 12.14 Square inductor.

and any one of the following: the outer diameter d_{out}, the inner diameter d_{in}, the average diameter $d_{avg} = 0.5(d_{out} + d_{in})$, or the fill ratio, defined as $\rho = (d_{out} - d_{in})/(d_{out} + d_{in})$. The thickness of the inductor has only a very small effect on inductance and will therefore be ignored here.

We now present three approximate expressions for the inductance of square, hexagonal, and octagonal planar inductors. The first approximation is based on a modification of an expression developed by Wheeler [17]; the second is derived from electromagnetic principles by approximating the sides of the spirals as current sheets; and the third is a monomial expression derived from fitting to a large database of inductors (whose exact inductance values are obtained from a three-dimensional electromagnetic field solver). All three expressions are accurate, with typical errors of 2% to 3%, and very simple, and are therefore excellent candidates for use in design and optimization.

12.3.2.1 Modified Wheeler Formula

Wheeler [17] presented several formulas for planar spiral inductors, which were intended for discrete inductors. A simple modification of the original Wheeler formula allows us to obtain an expression that is valid for planar spiral integrated inductors:

$$L_{mw} = K_1 \mu_0 \frac{n^2 d_{avg}}{1 + K_2 \rho} \tag{12.11}$$

where

ρ is the fill ratio defined previously

K_1 and K_2 are coefficients and are layout dependent, shown in Table 12.1

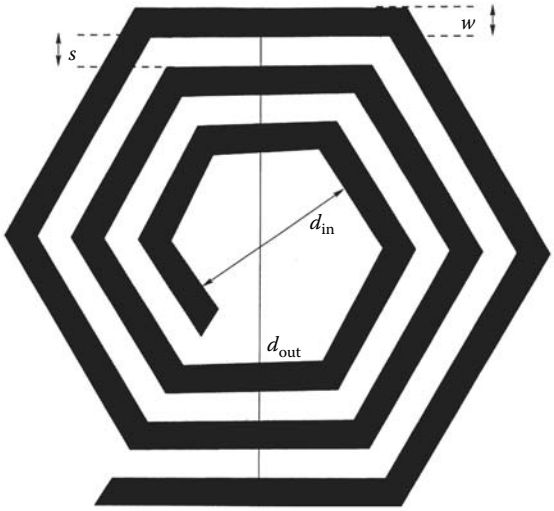


FIGURE 12.15 Hexagonal inductor.

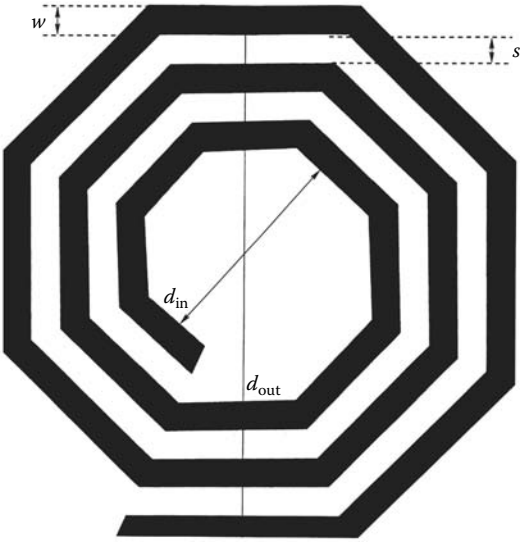


FIGURE 12.16 Octagonal inductor.

TABLE 12.1 Coefficients for Modified Wheeler Formula

Layout	K_1	K_2
Square	2.34	2.75
Hexagonal	2.33	3.82
Octagonal	2.25	3.55

The fill factor ρ represents how hollow the inductor is: for small ρ we have a hollow inductor ($d_{out} \cong d_{in}$), and for a large ρ we have a filled inductor ($d_{out} \gg d_{in}$). Two inductors with the same average diameter but different fill ratios will, of course, have different inductance values; the filled one has a smaller inductance because its inner turns are closer to the center of the spiral, and so contribute less positive mutual inductance and more negative mutual inductance. Some degree of hollowness is generally desired because the innermost turns contribute little overall inductance, but significant resistance.

12.3.2.2 Expression Based on Current Sheet Approximation

Another simple and accurate expression for the inductance of a planar spiral can be obtained by approximating the sides of the spirals by symmetrical current sheets of equivalent current densities [18]. For example, in the case of the square, we obtain four identical current sheets: the current sheets on opposite sides are parallel to one another, whereas the adjacent ones are orthogonal. Using symmetry and the fact that sheets with orthogonal current sheets have zero mutual inductance, the computation of the inductance is now reduced to evaluating the self-inductance of one sheet and the mutual inductance between opposite current sheets. These self- and mutual inductances are evaluated using the concepts of geometric mean distance (GMD) and arithmetic mean distance (AMD) [18,19]. The resulting expression is

$$L_{GMD} = \frac{\mu n^2 d_{avg}}{\pi}(c_1(\log c_2/\rho) + c_3\rho) \tag{12.12}$$

where the coefficients c_i are layout dependent and are listed in Table 12.2.

TABLE 12.2 Coefficients for Current-Sheet Inductance Formula

Layout	c_1	c_2	c_3
Square	2.00	2.00	0.54
Hexagonal	1.83	1.71	0.45
Octagonal	1.87	1.68	0.60

TABLE 12.3 Coefficients for Monomial Inductance Formula

Layout	b	α_1	α_2	α_3	α_4	α_5
Square	1.66×10^{-3}	−1.33	−0.13	2.50	1.83	−0.022
Hexagonal	1.33×10^{-3}	−1.46	−0.16	2.67	1.80	−0.030
Octagonal	1.34×10^{-3}	−1.35	−0.15	2.56	1.77	−0.032

A detailed derivation of these formulas can be found in Ref. [20]. Because this formula is based on a current sheet approximation, its accuracy worsens as the ratio s/w becomes large. In practice, this is not a problem because practical integrated spiral inductors are built with $s < w$. The reason is that a smaller spacing improves the interwinding magnetic coupling and reduces the area consumed by the spiral. A large spacing is only desired to reduce the interwinding capacitance. This is rarely a concern as this capacitance is always dwarfed by the underpass capacitance [8].

12.3.2.3 Data-Fitted Monomial Expression

Our final expression is based on a data-fitting technique, in which a population of thousands of inductors are simulated with an electromagnetic field solver. The inductors span the entire range of values of relevance to RF circuits. A monomial expression is then fitted to the data, which ultimately yields

$$L_{\text{mon}} = \beta d_{\text{avg}}^{\alpha 1} w^{\alpha 2} d_{\text{avg}}^{\alpha 3} n^{\alpha 4} s^{\alpha 5} \tag{12.13}$$

where the coefficients β and α_i are layout dependent, and given in Table 12.3.

Of course, it is also possible to use other data-fitting techniques; for example, one which minimizes the maximum error of the fit, or one in which the coefficients must satisfy given inequalities or bounds. The monomial expression is useful because, similar to the other expressions, it is very accurate and very simple. Its real value, however, is that it can be used for the optimal design of inductors and circuits containing inductors, using geometric programming, which is a type of optimization method that requires monomial models [21,22].

Figure 12.17 shows the absolute error distributions of these expressions. The plots show that typical errors are in the 1%–2% range, and most of the errors are below 3%. These expressions for inductance, while quite simple, are thus sufficiently accurate that field solvers are rarely necessary.

These expressions can be included in a physical, scalable lumped-circuit model for spiral inductors where, in addition to providing design insight, they allow efficient optimization schemes to be employed.

12.4 On-Chip Transformers

Transformers are important elements in RF circuits for impedance conversion, impedance matching, and bandwidth enhancement. Here, we present an analytical model for monolithic transformers that is suitable for circuit simulation and design optimization. We also provide simple expressions for calculating the mutual coupling coefficient (k).

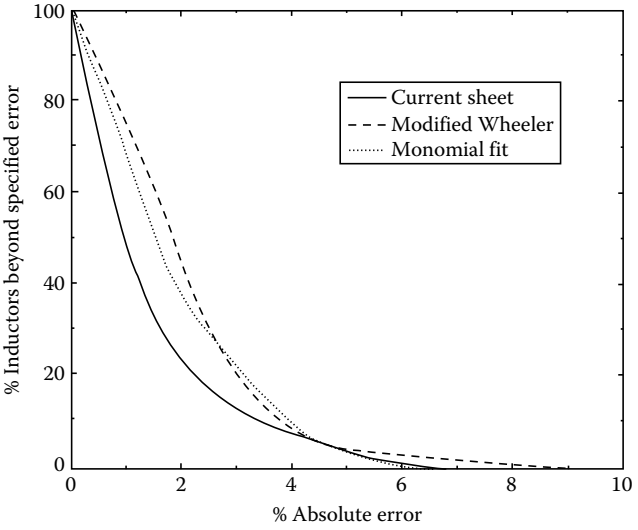


FIGURE 12.17 Error distribution for three formulas, compared with field solver simulations.

We first discuss different on-chip transformers and their advantages and disadvantages. We then present an analytical model along with expressions for the elements in it and the mutual coupling coefficient.

12.4.1 Monolithic Transformer Realizations

Figures 12.18 through 12.23 illustrate common configurations of monolithic transformers. The different realizations offer varying trade-offs among the self-inductance and series resistance of each port, the

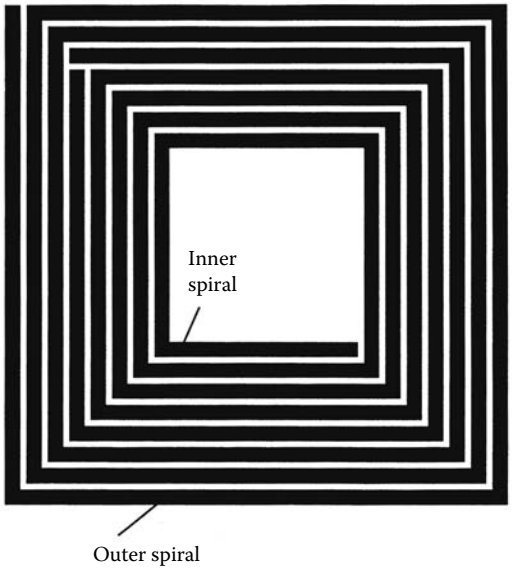


FIGURE 12.18 Tapped transformer.

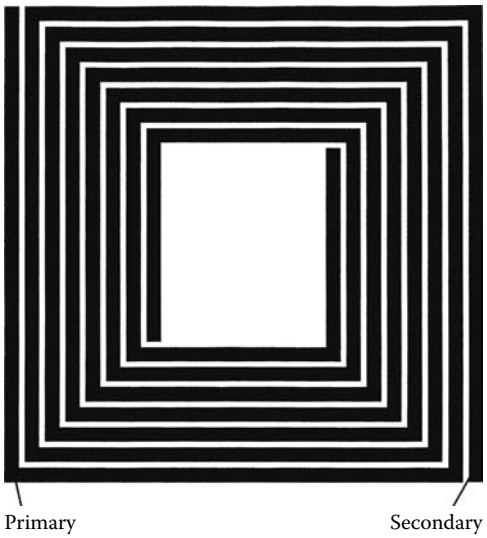


FIGURE 12.19 Interleaved transformer.

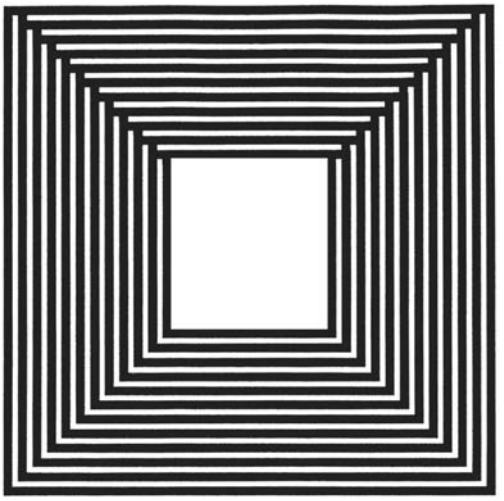


FIGURE 12.20 Stacked transformer with top spiral overlapping the bottom one.

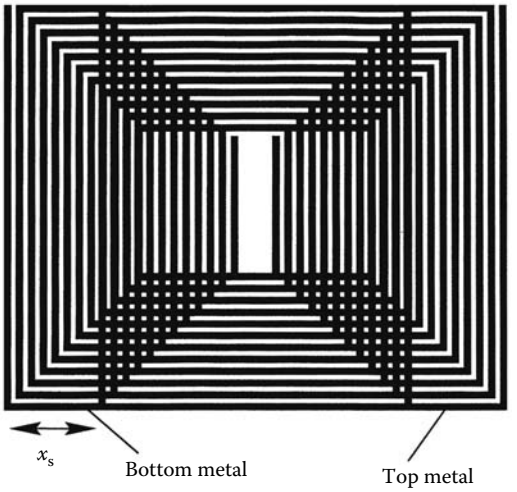


FIGURE 12.21 Stacked transformer with top and bottom spirals laterally shifted.

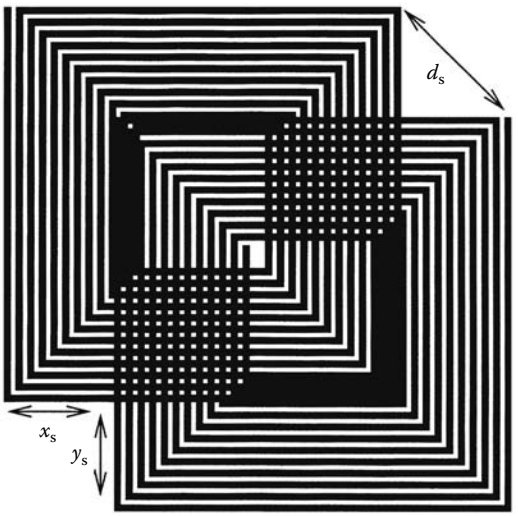


FIGURE 12.22 Stacked transformer with top and bottom spirals diagonally shifted.

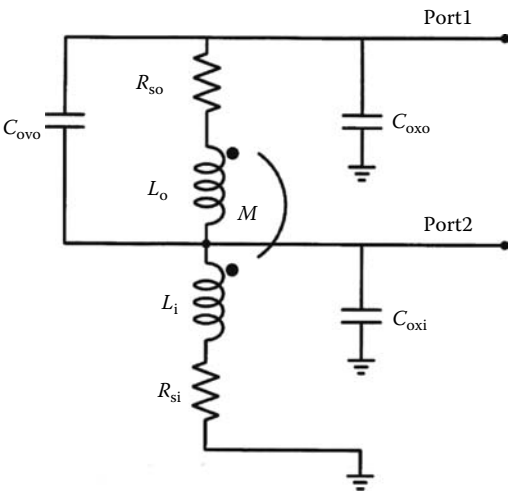


FIGURE 12.23 Tapped transformer model.

mutual coupling coefficient, the port-to-port and port-to-substrate capacitances, resonant frequencies, symmetry, and area. The models and coupling expressions allow these trade-offs to be systematically explored, thereby permitting transformers to be customized for a variety of circuit design requirements.

The characteristics desired of a transformer are application dependent. Transformers can be configured as three- or four-terminal devices. They may be used for narrowband or broadband applications. For example, in single-sided to differential conversion, the transformer might be used as a four-terminal narrowband device. In this case, a high mutual coupling coefficient and high self-inductance are desired, along with low series resistance. On the other hand, for bandwidth extension applications,

the transformer is used as a broadband three-terminal device. In this case, a small mutual coupling coefficient and high series resistance are acceptable, while all capacitances need to be minimized [23].

The tapped transformer (Figure 12.18) is best suited for three-port applications. It permits a variety of tapping ratios to be realized. This transformer relies only on lateral magnetic coupling. All windings can be implemented with the top metal layer, thereby minimizing port-to-substrate capacitances. Because the two inductors occupy separate regions, the self-inductance is maximized while the port-to-port capacitance is minimized. Unfortunately, this spatial separation also leads to low mutual coupling ($k = 0.3$–0.5).

The interleaved transformer (Figure 12.19) is best suited for four-port applications that demand symmetry. Once again, capacitances can be minimized by implementing the spirals with top level metal so that high resonant frequencies may be realized. The interleaving of the two inductances permits moderate coupling ($k = 0.7$) to be achieved at the cost of reduced self-inductance. This coupling may be increased at the cost of higher series resistance by reducing the turn width (w) and spacing (s).

The stacked transformer (Figure 12.20) uses multiple metal layers and exploits both vertical and lateral magnetic coupling to provide the best area efficiency, the highest self-inductance, and highest coupling ($k = 0.9$). This configuration is suitable for both three- and four-terminal configurations. The main drawback is the high port-to-port capacitance, or equivalently a low self-resonant frequency. In some cases, such as narrowband impedance transformers, this capacitance may be incorporated as part of the resonant circuit. Also, in multilevel processes, the capacitance can be reduced by increasing the oxide thickness between spirals. For example, in a five-metal process, 50%–70% reductions in port-to-port capacitance can be achieved by implementing the spirals on layers five and three instead of five and four. The increased vertical separation will reduce k by less than 5%. One can also trade off reduced coupling for reduced capacitance by displacing the centers of the stacked inductors (Figures 12.21 and 12.22).

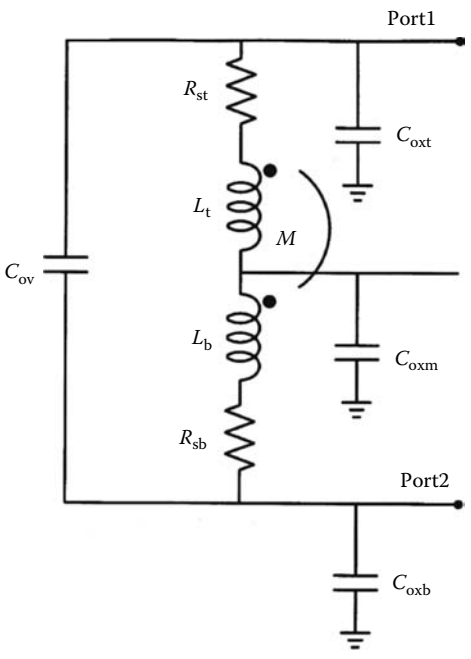


FIGURE 12.24 Stacked transformer model.

12.4.2 Analytical Transformer Models

Figures 12.23 and 12.24 present the circuit models for tapped and stacked transformers, respectively. The corresponding element values for the tapped transformer model are given by the following equations (subscript o refers to the outer spiral, i to the inner spiral, and T to the whole spiral):

$$L_T = \frac{9.375\mu_0 n_T^2 AD_T{}^2}{11OD_T - 7AD_T} \tag{12.14}$$

$$L_o = \frac{9.375\mu_0 n_o^2 AD_o{}^2}{11OD_o - 7AD_o} \tag{12.15}$$

$$L_i = \frac{9.375\mu_0 n_i^2 AD_i{}^2}{11OD_i - 7AD_i} \tag{12.16}$$

$$M = \frac{L_T - L_o - L_i}{2\sqrt{L_o L_i}} \tag{12.17}$$

$$R_{so} = \frac{\rho l_o}{\delta w\left(1 - e^{-\frac{t}{\delta}}\right)} \tag{12.18}$$

$$R_{si} = \frac{\rho l_i}{\delta w\left(1 - e^{-\frac{t}{\delta}}\right)} \tag{12.19}$$

$$C_{ovo} = \frac{\varepsilon_{ox}}{t_{ox,t-b}} \cdot (n_o - 1)w^2 \tag{12.20}$$

$$C_{oxo} = \frac{\varepsilon_{ox}}{2t_{ox}} \cdot l_o w \tag{12.21}$$

$$C_{oxi} = \frac{\varepsilon_{ox}}{2t_{ox}} \cdot (l_o + l_i)w \tag{12.22}$$

where

ρ is the DC metal resistivity
δ is the skin depth
$t_{ox,t-b}$ is the oxide thickness from top level metal to bottom metal
n is the number of turns
OD, AD, and ID are the outer, average, and inner diameters, respectively
l is the length of the spiral
w is the turn width
t is the metal thickness
A is the area

Expressions for the stacked transformer model are as follows (subscript ''t'' refers to the top spiral and b to the bottom spiral):

$$L_t = \frac{9.375\mu_0 n^2 \mathrm{AD}^2}{11\mathrm{OD_T} - 7\mathrm{AD_T}} \tag{12.23}$$

$$L_b = L_t \tag{12.24}$$

$$k = 0.9 - \frac{d_s}{\mathrm{AD}} \tag{12.25}$$

$$M = k\sqrt{L_t L_b} \tag{12.26}$$

$$R_{st} = \frac{\rho_t l}{\delta_t w\left(1 - e^{-\frac{t_t}{\delta_t}}\right)} \tag{12.27}$$

$$R_{sb} = \frac{\rho_b l}{\delta_b w\left(1 - e^{-\frac{t_b}{\delta_b}}\right)} \tag{12.28}$$

$$C_{ov} = \frac{\varepsilon_{ox}}{2t_{ox,t-b}} \cdot l \cdot w \cdot \frac{A_{ov}}{A} \tag{12.29}$$

$$C_{oxt} = \frac{\varepsilon_{ox}}{2t_{oxt}} \cdot l \cdot w \cdot \frac{A - A_{ov}}{A} \tag{12.30}$$

$$C_{oxb} = \frac{\varepsilon_{ox}}{2t_{ox}} \cdot l \cdot w \tag{12.31}$$

$$C_{oxm} = C_{oxt} + C_{oxb} \tag{12.32}$$

where

t_{oxt} is the oxide thickness from top metal to the substrate
t_{oxb} is the oxide thickness from bottom metal to substrate
k is the coupling coefficient
A_{ov} is the overlap area of the two spirals
d_s is the center-to-center spiral distance

The expressions for the series resistances (R_{so}, R_{si}, R_{st}, and R_{sb}), the port-substrate capacitances (C_{oxo}, C_{oxi}, C_{oxt}, C_{oxb}, and C_{oxm}), and the crossover capacitances (C_{ovo}, C_{ovi}, and C_{ov}) are taken from Ref. [8]. Note that the model accounts for the increase in series resistance with frequency due to skin effect. Patterned ground shields (PGS) are placed beneath the transformers to isolate them from resistive and capacitive coupling to the substrate [12]. As a result, the substrate parasitics can be neglected.

The inductance expressions in the foregoing are based on the modified Wheeler formula discussed earlier [24]. This formula does not take into account the variation in inductance due to conductor thickness and frequency. However, in practical inductor and transformer realizations, the thickness is small compared to the lateral dimensions of the coil and has only a small impact on the inductance. For typical conductor thickness variations (0.5–2.0 μm), the change in inductance is within a few percent for practical inductor geometries. The inductance also changes with frequency due to changes in current distribution within the conductor. However, over the useful frequency range of a spiral, this variation is negligible [12]. When compared with field solver simulations, the inductance expression exhibits a maximum error of 8% over a broad design space (outer diameter OD varying from 100 to 480 μm, L varying from 0.5 to 100 nH, w varying from 2 μm to 0.3 OD, s varying from 2 μm to w, and inner diameter ID varying from 0.2 to 0.8OD).

For the tapped transformer, the mutual inductance is determined by first calculating the inductance of the whole spiral (L_T), the inductance of the outer spiral (L_o), the inductance of the inner spiral (L_i), and then using the expression $M = (L_T - L_o - L_i)/2$. For the stacked transformer, the spirals have identical lateral geometries and therefore identical inductances. In this case, the mutual inductance is determined by first calculating the inductance of one spiral (L_T), the coupling coefficient (k) and then using the expression $M = kL_T$. In this last case, the coupling coefficient is given by $k = 0.9 - d_s/(\text{AD})$ for $d_s < 0.7\text{AD}$, where d_s is the center-to-center spiral distance and AD is the average diameter of the spirals. As d_s increases beyond 0.7AD, the mutual coupling coefficient becomes harder to model. Eventually, k crosses zero and reaches a minimum value of approximately -0.1 at $d_s = \text{AD}$. As d_s increases further, k asymptotically approaches zero. At $d_s = 2\text{AD}$, $k = -0.02$, indicating that the magnetic coupling between closely spaced spirals is negligible.

The self-inductances, series resistances, and mutual inductances are independent of whether a transformer is used as a three- or four-terminal device. The only elements that require recomputation are the port-to-port and port-to-substrate capacitances. This situation is analogous to that of a spiral inductor being used as a single- or dual-terminal device.

As with the inductance formulas, the transformer models obviate the need for full field solutions in all but very rare instances, allowing rapid design and optimization.

References

1. Samavati, H. et al., Fractal capacitors, *1998 IEEE ISSCC Digest of Technical Papers*, Feb. 1998.
2. Akcasu, O. E., High capacitance structures in a semiconductor device, U.S. Patent 5208725, May 1993.
3. Bohr, M., Interconnect scaling—The real limiter to high performance VLSI, *International Electron Devices Meeting Technical Digest*, 241–244, 1995.
4. Bohr, M. et al., A high performance 0.25 μm logic technology optimized for 1.8V operation, *International Electron Devices Meeting Technical Digest*, 847–850, 1996.

5. Venkatesan, S. et al., A high performance 1.8 V, 0.20 μm CMOS technology with copper metallization, *International Electron Devices Meeting Technical Digest*, 769–772, 1997.
6. Mandelbrot, B. B., *The Fractal Geometry of Nature*, W.H. Freeman, New York, 1983.
7. Pettenpaul, E. et al., Models of lumped elements on GaAs up to 18 GHz, *IEEE Transactions on Microwave Theory and Techniques*, 36(2), 294–304, Feb. 1988.
8. Yue, C. P., Ryu, C., Lau, J., Lee, T. H., and Wong, S. S., A physical model for planar spiral inductors on silicon, *International Electron Devices Meeting Technical Digest*, 155–158, Dec. 1996.
9. Pfost, M., Rein, H.-M., and Holzwarth, T., Modeling substrate effects in the design of high speed Si-bipolar IC's, *IEEE Journal of Solid-State Circuits*, 31(10), 1493–1501, Oct. 1996.
10. Ashby, K. B., Koullias, I. A., Finley, W. C., Bastek, J. J., and Moinian, S., High Q inductors for wireless applications in a complementary silicon bipolar process, *IEEE Journal of Solid-State Circuits*, 31(1), 4–9, Jan. 1996.
11. Chang, J. Y.-C., Abidi, A. A., and Gaitan, M., Large suspended inductors on silicon and their use in a 2 μm CMOS RF amplifier, *IEEE Electron Device Letters*, 14(5), 246–248, May 1993.
12. Yue, C. P. et al., On-chip spiral inductors with patterned ground shields for Si-based RF ICs, *IEEE Journal of Solid-State Circuits*, 33, 743–752, May 1998.
13. Greenhouse, H. M., Design of planar rectangular microelectronic inductors, *IEEE Transactions on Parts, Hybrids, and Packing*, PHP-10(2), 101–109, June 1974.
14. Wiemer, L. and Jansen, R. H., Determination of coupling capacitance of underpasses, air bridges and crossings in MICs and MMICs, *Electronics Letters*, 23(7), 344–346, Mar. 1987.
15. Ho, I. T. and Mullick, S. K., Analysis of transmission lines on integrated-circuit chips, *IEEE Journal of Solid-State Circuits*, SC-2(4), 201–208, Dec. 1967.
16. Hasegawa, H., Furukawa, M., and Yanai, H., Properties of microstrip line on Si-SiO_2 system, *IEEE Transactions on Microwave Theory and Techniques*, MTT-19(11), 869–881, Nov. 1971.
17. Wheeler, H. A., Simple inductance formulas for radio coils, *Proceedings of the IRE*, 16(10), 1398–1400, October 1928.
18. Rosa, E. B., Calculation of the self-inductances of single-layer coils, *Bulletin of the Bureau of Standards*, 2(2), 161–187, 1906.
19. Maxwell, J. C., *A Treatise on Electricity and Magnetism*, 3rd edn., Mineola, New York, 1967.
20. Mohan, S. S., Formulas for planar spiral inductances, *Technical Report*, *IC Laboratory*, Stanford University, Palo Alto, California, Aug. 1998, http://www-smirc.stanford.edu.
21. Boyd, S. and Vandenberghe, L., Introduction to convex optimization with engineering applications, Course Notes, 1997, http://www-leland.stanford.edu/class/ee364/.
22. Hershenson, M., Boyd, S. P., and Lee, T. H., GPCAD: A tool for CMOS op-amp synthesis, in *Digest of Technical Papers, IEEE International Conference on Computer-Aided Design*, San Jose, California, Nov. 1998.
23. Lee, T. H., *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, Cambridge, 1998.
24. Wheeler, H. A., Simple inductance formulas for radio coils, *Proceedings of the IRE*, 16(10), 1398–1400, Oct. 1928.

13

Circuit Elements, Modeling, and Equation Formulation

13.1 Lumped Circuit Approximation **13**-1
13.2 Circuit Elements and Connecting Multiport **13**-1
13.3 Characterizations of Circuit Elements **13**-3
Formal Methods of Characterization • Resistive Elements • Reactive Elements • Memristive Elements • Dynamic Models
13.4 Connecting Multiport **13**-10
13.5 Tableau Formulation **13**-11
References **13**-11

Josef A. Nossek
Technical University of Munich

13.1 Lumped Circuit Approximation

Most texts on circuits, whether they deal with linear or nonlinear circuits, consider only lumped circuits. If this is not the case, such is normally stated explicitly (see, e.g., Section III, in *Feedback, Nonlinear, and Distributed Circuits*). A physical circuit is considered to be a lumped circuit, if its size is small enough that for the situation under discussion, electromagnetic waves propagate across the circuit virtually instantaneously. If this is satisfied, voltages across ports and currents through terminals are well defined and, therefore, well suited to describe and analyze the behavior of a circuit.

To check whether an actual circuit is lumped or not, the largest extension d of the circuit in any spatial dimension is compared to the shortest wavelength λ of interest or with the shortest time interval τ of interest. If

$$d \ll \lambda = c/f, \quad d \ll \tau c \tag{13.1}$$

is fulfilled, the circuit is lumped. In Equation 13.1, c is the propagation velocity of electromagnetic waves in the medium under consideration, and f is the frequency corresponding to the wavelength λ as well as to the period τ.

13.2 Circuit Elements and Connecting Multiport

It is obvious that a nonlinear circuit is described by a set of nonlinear equations, which, generally speaking, can be solved only approximately. Moreover, we may not find a unique solution, but a set of different solutions. Because of this complicated situation (compared to the simple solution of a linear

circuit), it is even more important here to exploit the structure of the equations that reflect the structural properties of the circuit. The most important step in this direction is to partition the circuit into two parts: one containing all circuits elements separately and one containing the interconnections thereof only. The latter is called the connecting multiport (CMP). This partitioning is demonstrated with a real circuit in Figure 13.1; it is obvious that this partitioning is completely independent of the nature of the utilized circuit elements: linear or nonlinear, two-terminal or multiterminal, time-variant, time-invariant, passive or active, and so forth.

The equations that describe the CMP are merely Kirchhoff's current and voltage laws (KCL, KVL) (Section 13.4), which are, of course, linear, while nonlinearities show up in the description of the circuit elements. First, the circuit element is described, and then some details of the CMP are discussed.

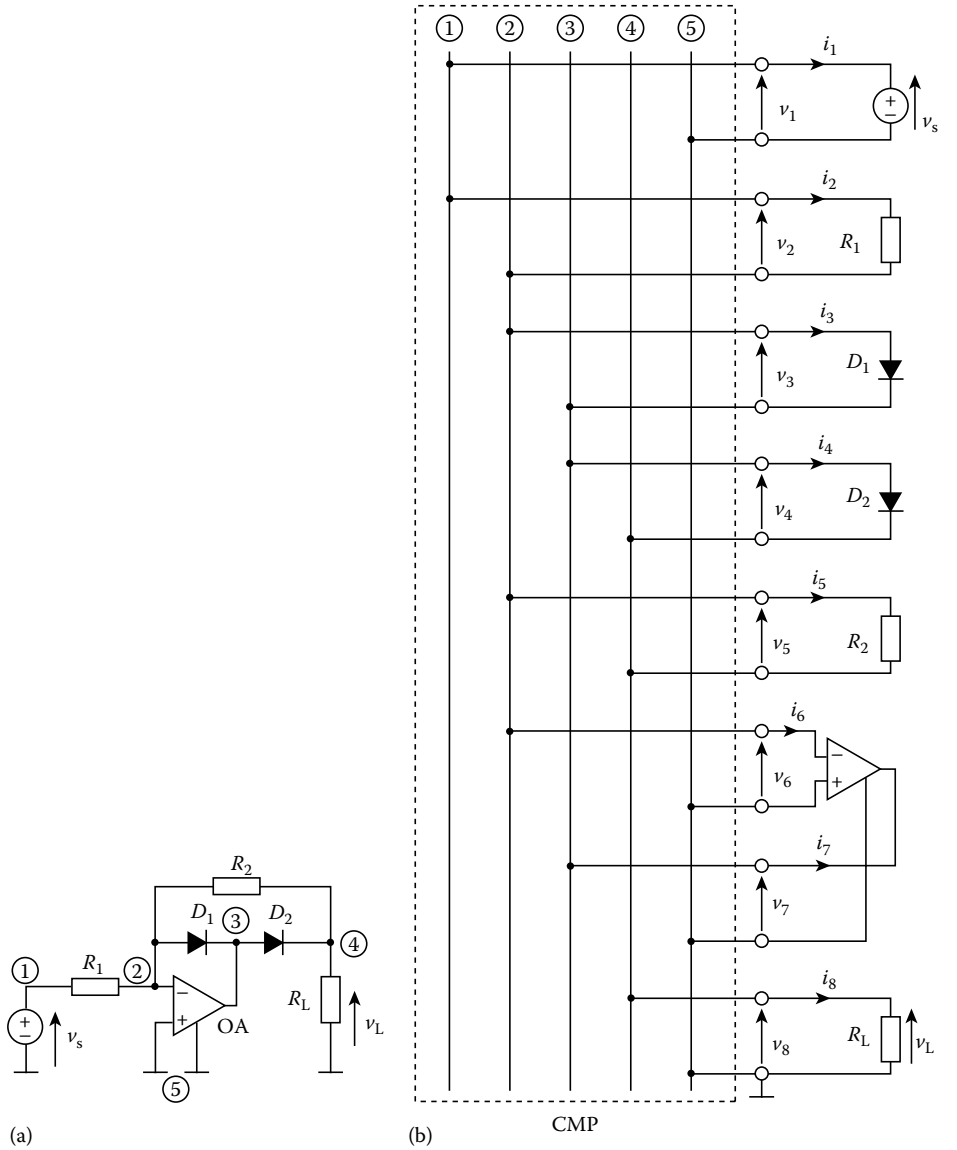


FIGURE 13.1 (a) A nonlinear circuit conventionally drawn. (b) Partitioning into CMP and circuit elements.

13.3 Characterizations of Circuit Elements

This section characterizes a circuit element, two-terminal or multiterminal, with algebraic equations. This necessitates a proper choice of variables. Because of the algebraic nature of this discussion, differential or integral operators cannot be used. Let us begin with a formal discussion, using variables x and y without elaborating on their physical meaning. Later, these methods are applied to specific circuit elements of practical relevance, and voltages, currents, charges, and fluxes are utilized instead of x and y.

13.3.1 Formal Methods of Characterization

A relation between variables x and y can be given in an implicit form

$$\mathcal{F} \subset R \times R,\ x \in R,\ y \in R \quad \mathcal{F} = \{(x,y) | f(x,y) = 0\} \tag{13.2}$$

Here, $\mathcal{F}$ is the characteristic of a two-terminal device which is described by a single implicit equation, $f(x, y) = 0$. Note that this equation is not unique. Various equivalent forms of $f(x, y) = 0$, exist which look quite different, but define the same one-port:

$$g(x,y) = e^{f(x,y)} - 1 = 0 \Leftrightarrow f(x,y) = 0 \tag{13.3}$$

The two functions f and g are quite different, but the tuples (x, y) defined by them constitute the same set $\mathcal{F}$.

A simple example for such an implicit description is

$$f(x,y) = (y/y_0) - \arctan(x/x_0) = 0 \tag{13.4}$$

the plot of which in the (x, y)-plane is shown in Figure 13.2.

An alternative to the aforementioned implicit form is a parametric description, in which we use an additional parameter $\lambda \in R$ to express the port variables as functions of this parameter:

$$x = f_x(\lambda), \quad y = f_y(\lambda) \tag{13.5a}$$

with every tuple

$$(x,y) = \big(f_x(\lambda), f_y(\lambda)\big) \in \mathcal{F} \tag{13.5b}$$

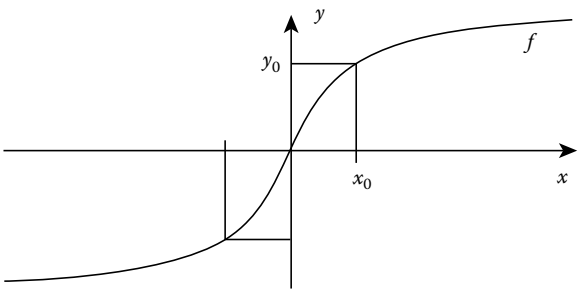


FIGURE 13.2 Plot of the one-port characteristic (Figure 13.1).

being an admissible element of $\mathcal{F}$. Using Equation 13.4 with $\lambda = y/y_0$ we have $x/x_0 = \tan\lambda$ and, therefore,

$$x = f_x(\lambda) = x_0 \tan\lambda, \quad y = f_y(\lambda) = y_0\lambda, \quad \lambda \in (-\pi/2,\pi/2) \tag{13.6}$$

Parameterized descriptions are also not unique, but if the parameter is properly chosen (so that f_x and f_y are continuous and differentiable), they are quite advantageous to work with.

The most favorable description in practical applications is explicit, but it exists only if the relation (Equation 13.2) is unique in at least one of the variables x or y. If y is a function of x (or vice versa), we write

$$y = f(x), \quad x = g(y) \tag{13.7}$$

For this example, both explicit versions do exist:

$$y = y_0 \arctan(x/x_0), \quad x = x_0 \tan(y/y_0), \quad y \in (-y_0\pi/2, y_0\pi/2) \tag{13.8}$$

All these descriptive methods can be extended in a straightforward manner to the multiterminal case by simply replacing the scalars x and y by vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ and the functions f and g by vectors of functions $\boldsymbol{f}$ and $\boldsymbol{g}$.

The formal approach is applied to actual circuit elements in the following section.

13.3.2 Resistive Elements

A resistive element is, by definition, uniquely characterized by one of the aforementioned algebraic descriptions, where x and y are replaced by voltage, v, and current, i. This relation may depend on time, t (time-variant circuit element), but not on the history of the variables v and i.

Many important circuit elements can be modeled resistively, as far as their main property is concerned. This is true for most semiconductor devices such as diodes, bipolar transistors, field effect transistors, operational amplifiers (op-amps), and so forth. This section concentrates on the main effect, leaving a more detailed description, including parasitics, to later sections.

An example of a time-variant (nonautonomous) resistive one-port is given in Figure 13.3

$$\mathcal{F}(t) = \{(v(t), i(t)) | i(t) = I_s(\exp(v(t)/V_T) - 1) - i_L(t)\} \tag{13.9}$$

Using a reverse saturation current $I_s = 10$ μA and a thermal voltage $V_T = 25$ mV the individual characteristics in Figure 13.3b are obtained, illustrating the nonautonomous nature of the device with

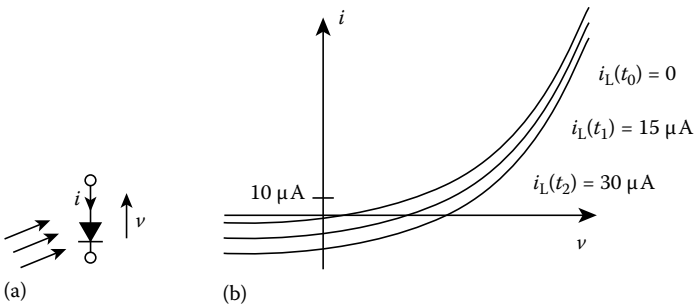


FIGURE 13.3 (a) Device symbol of a photodiode. (b) Device i–v characteristic.

the photocurrent i_L as the controlling parameter, which itself depends on the light intensity, which is assumed to be time dependent.

An ordinary *pn*-junction diode is nothing more than a special case of Equation 13.9, i.e., $i_L = 0$.

Given the device characteristic in graphical form such as Figure 13.3b, which may be the summary of a set of measurements, it is easy to check as to whether explicit descriptions $i = g(u)$ or $u = f(i)$ exist or not. A simple example of a device for which $i = g(u)$ does exist, but g does not have an inverse, is the tunnel diode (Figure 13.4).

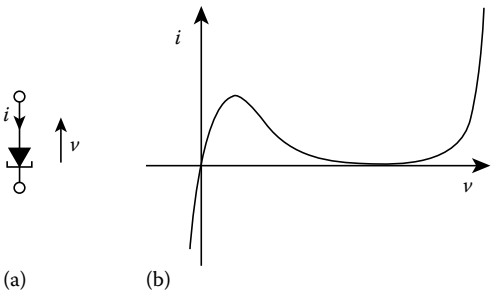


FIGURE 13.4 (a) Symbol and (b) v–i characteristic of a tunnel diode.

Many more nonlinear resistive one-ports or models thereof can be found, but the basic concept is always the same. Therefore, we proceed with an important example of a multiterminal device, the transistor. Again, we have a multitude of various transistors [bipolar *npn* and *pnp*, unipolar field-effect insulated gate transistors (MOSFET) and junction type (JFET), *n*- and *p*-channel, enhancement and depletion, etc.]. Here, we demonstrate only the basic idea of a resistive two-port (or three-terminal *e*mitter, collector, base) model of a bipolar *npn* transistor. Many more details are given in Section II of *Feedback, Nonlinear, and Distributed Circuits*.

The so-called Ebers–Moll equations describe a bipolar *npn* transistor:

$$\begin{aligned} i_e &= -I_{es}(\exp(-v_{eb}/V_T) - 1) + \alpha_R I_{cs}(\exp(-v_{cb}/V_T) - 1) = -i_1 + \alpha_R i_2 \\ i_c &= \alpha_F I_{es}(\exp(-v_{eb}/V_T) - 1) - I_{cs}(\exp(-v_{cb}/V_T) - 1) = \alpha_F i_1 - i_2 \end{aligned} \tag{13.10}$$

which is an explicit two-port description

$$\mathbf{i} = \mathbf{f}(\mathbf{v}) \tag{13.11}$$

with

$$\mathbf{i} = \begin{bmatrix} i_e \\ i_c \end{bmatrix} \quad v = \begin{bmatrix} v_{eb} \\ v_{cb} \end{bmatrix}$$

using the base b as the common terminal. α_F and α_R are the forward and reverse current gain of the transistor in common base configuration as shown in Figure 13.5.

As with any device having more than two terminals, the device characteristics cannot be represented simply as a curve in a plane. In general it is a hypersurface in a multidimensional space. Especially in the

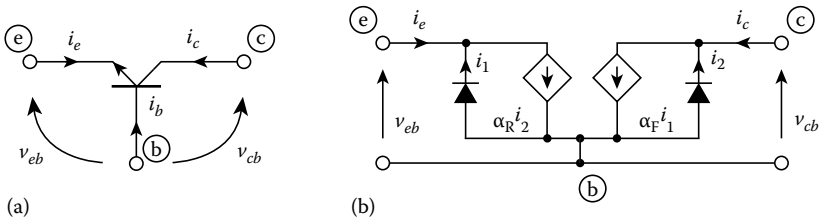


FIGURE 13.5 (a) Symbol and (b) equivalent circuit with two linear CCCSs and two diodes exactly representing Ebers–Moll equation (Equation 13.10).

case of the three-terminal transistor, it is a two-dimensional surface in four-dimensional space. Because this is not easy to visualize, it is normally split into two three-dimensional representations, which commonly are given as follows:

$$\begin{aligned} i_b &= f_1(v_{be}, v_{ce}) \approx f_1(v_{be}) \\ i_c &= f_2(v_{be}, v_{ce}) \approx f_2'(v_{ce}, i_b) \end{aligned} \tag{13.12}$$

To obtain Equation 13.12 from Equation 13.10, we must use

$$i_b = -i_e - i_c, \quad v_{ce} = v_{cb} - v_{eb}, \quad v_{be} = -v_{eb}$$

and i_b is almost independent of v_{ce}. The first of the two equations (Equation 13.12) is already well suited for having a v–i characteristic plotted in a v_{eb}–i_b-plane. The second equation is normally plotted in the v_{ce}–i_c-plane, with i_b as a parameter.

The device characteristics of this bipolar *npn* transistor, as well as those of many other multiterminal semiconductor devices, are today quite standard; they are given in data sheets and used by designers (discussed further in Section II of *Feedback, Nonlinear, and Distributed Circuits*). These nonlinear models are the basis for deriving linearized small signal models, where they are needed.

Finally, we look at a higher level model of a multiterminal device, and model a complete op-amp (containing a multitude of transistors) using a very simple, but nevertheless very powerful resistive model. An op-amp, at a rather high level of abstraction, is a four-terminal device, as depicted in Figure 13.6. In this figure, everything dealing with power supply, biasing, and offset compensation has been hidden. In this very simple model, we assume the input currents to be zero and the output voltage to depend only on the difference of the two input voltages (common mode gain is zero):

$$v_d = v_+ - v_- \tag{13.13}$$

in the following way:

$$v_0 = \begin{cases} V_{\text{sat}} & v_d & \geq & V_{\text{sat}}/A_0 \\ A_0 v_d & |v_d| & \leq & V_{\text{sat}}/A_0 \\ -V_{\text{sat}} & v_d & \leq & -V_{\text{sat}}/A_0 \end{cases} \tag{13.14}$$

Therefore, we have a piecewise linear transfer characteristic, consisting of three pieces (Figure 13.7). According to the three pieces (I, II, and III) we have three equivalent circuits (Figure 13.8). Even if we increase our idealization to $A_0 \to \infty$, the equivalent circuit in Figure 13.8b reduces from the voltage-controlled voltage source (VCVS) to a nullor. (It is worth emphasizing that this surprisingly simple model of such a complex nonlinear functional unit as an op-amp is capable of capturing all of the main effects of such a multitransistor circuit. For many practically important applications, it provides an accurate prediction of the behavior of real circuits.)

If we are not satisfied with resistive models because of the bandwidth of signals to be processed, the model must be refined by including elements with memory. First, we need to set up a way to describe memory-possessing elements (Section 13.3.3) and then combine this with the resistive model into a dynamic model (Section 13.3.5).

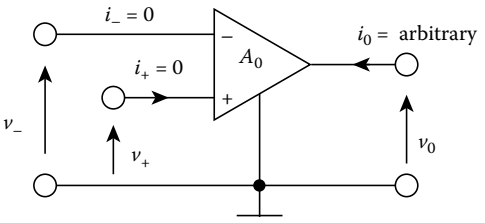


FIGURE 13.6 Symbolic representation of an op-amp.

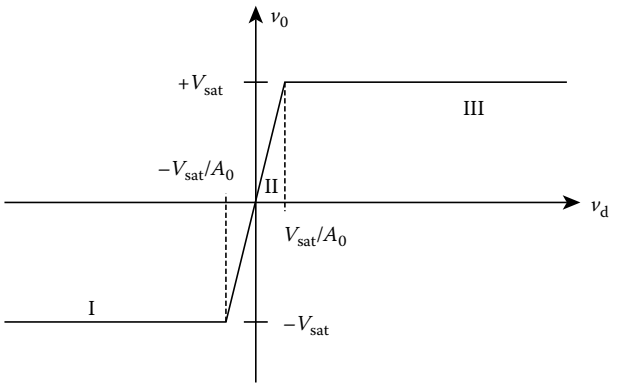


FIGURE 13.7 PWL transfer characteristic of an op-amp.

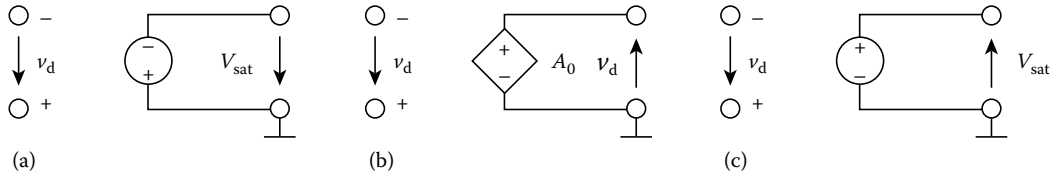


FIGURE 13.8 Equivalent circuit for an op-amp (a) in the negative saturation region I ($v_d \leq -V_{sat}/A_0$), (b) in the linear region II ($v_d \leq V_{sat}/A_0$), and (c) in the positive saturation region III ($v_d \geq V_{sat}/A_0$).

13.3.3 Reactive Elements

To use algebraic description and to plot a characteristic such as a curve in an x–y-plane, we have to extend the set of variables from v and i (resistive case) to charge q and flux ϕ

$$q(t) = q(t_0) + \int_{t_0}^{t} i(\tau)\mathrm{d}\tau \quad \phi(t) = \phi(t_0) + \int_{t_0}^{t} v(\tau)\mathrm{d}\tau \tag{13.15}$$

If the integrals exist for $t_0 \to -\infty$ and if $q(-\infty) = 0$ and $\phi(-\infty) = 0$, we can write

$$q(t) = \int_{-\infty}^{t} i(\tau)\mathrm{d}\tau \quad \phi(t) = \int_{-\infty}^{t} v(\tau)\mathrm{d}\tau$$

which simply means to ignore the initial conditions of charge q and flux ϕ, which are unimportant for the electrical behavior of the component and the circuit.

With this in mind, we define a capacitive (inductive) one-port in the following way:

$$\mathcal{F}_C = \{(v, q) | f_C(v, q) = 0\}, \quad \mathcal{F}_L = \{(i, \phi) | f_L(i, \phi) = 0\} \tag{13.16}$$

Other than this implicit algebraic description, parameterized or even explicit descriptions may exist, similar to the resistive case dealt with previously.

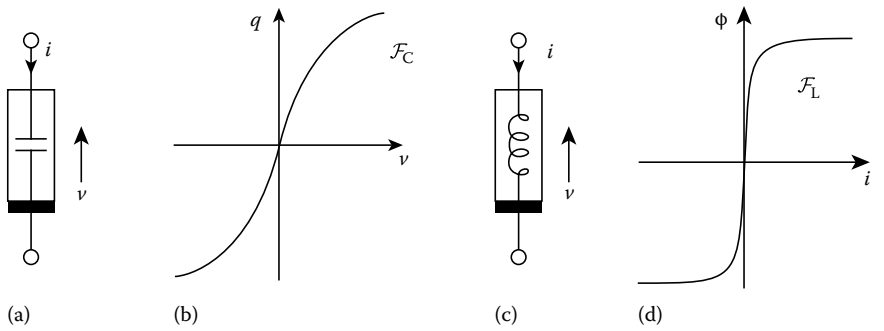


FIGURE 13.9 (a) Symbol and (b) characteristic of capacitor with dielectric material and (c) symbol and (d) characteristic of an inductor with ferromagnetic material.

Figure 13.9 gives examples of some nonlinear characteristics of a capacitive and an inductive reactance. It is obvious that this concept can be extended to the multiterminal case by replacing the scalars v and q or i and ϕ by vectors $\boldsymbol{v}$ and $\boldsymbol{q}$ or $\mathbf{i}$ and ϕ, respectively. This is useful when creating a first-order model of a multiport transformer.

13.3.4 Memristive Elements

After having dealt with resistive and reactive elements and considering all the variables and their interrelations that we used (Figure 13.10), an interesting question remains: What about an element with an algebraic characterization in the q–ϕ-plane? Because this missing element is characterized by an algebraic relation between the integral of current and the integral of voltage, it is a resistive element with memory, and is therefore called a memristor:

$$\mathcal{F}_M = \{(q, \phi) | f_M(q, \phi) = 0\} \tag{13.17}$$

A real-world example of a memristor is the so-called Coulomb cell, which consists of a gold anode immersed in an electrolyte in a silver can cathode. Memristive descriptions, although not widely utilized, are very useful for modeling the behavior of electrochemical elements.

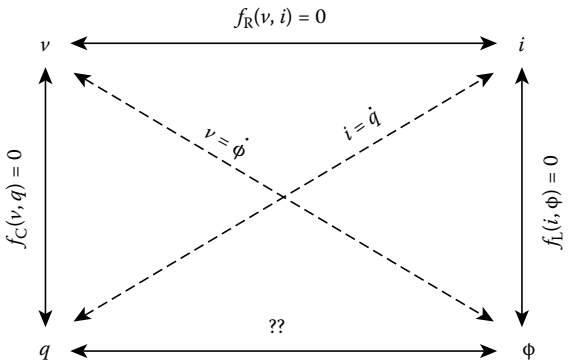


FIGURE 13.10 Interrelation between variables.

13.3.5 Dynamic Models

Only a few examples are illustrated in this section, comprising resistive and reactive elements, in order to achieve a realistic description of electronic devices, including dynamical effects.

A *pn*-junction was described resistively earlier in the chapter, with an exponential *v*–*i* characteristic, neglecting dynamical effects. To remove this shortcoming, we use the more elaborate dynamic model depicted in Figure 13.11, which consists of a resistive *pn*-junction in parallel with a nonlinear capacitor. This extension can, of course, be carried on for the (multiterminal) transistor case, where we use dynamic diode models (Figure 13.12).

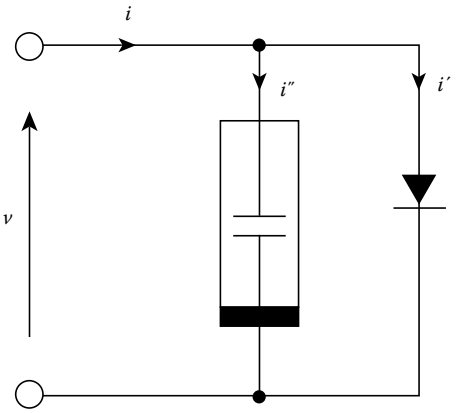


FIGURE 13.11 Dynamic *pn*-junction model.

The combination of the resistive and reactive models in one circuit leads to a description that makes use of differential and integral operators. It is almost always possible to reduce this to a set of nonlinear ordinary differential and algebraic equations. We return to this point later, but first we conclude this section with a dynamic model of an op-amp (Figure 13.13), making use of nonlinear controlled sources (VCVS) and voltage-controlled circuit sources (VCCS). This simple model accounts for the following important practical properties of a real-world op-amp:

- First-order, low-pass behavior in the linear region of the controlled sources.
- Slew rate limitation is incorporated by the nonlinearity of $g_m(v_d)$ of the VCCS (Figure 13.14).
- Output voltage saturation is modeled with the aid of the nonlinearity of $\mu(v_c)$ (Figure 13.15).

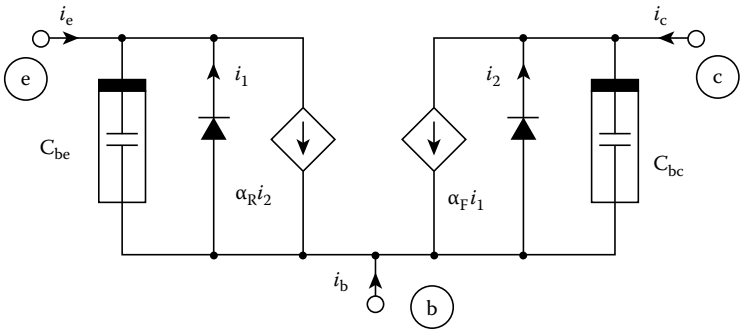


FIGURE 13.12 Dynamic transistor model.

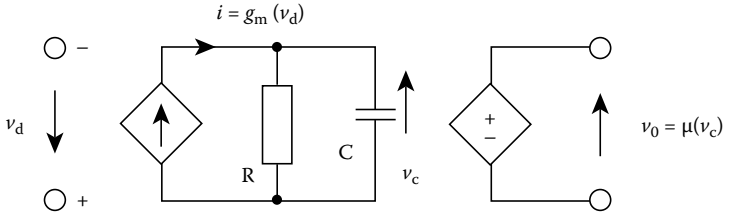


FIGURE 13.13 Dynamic op-amp model.

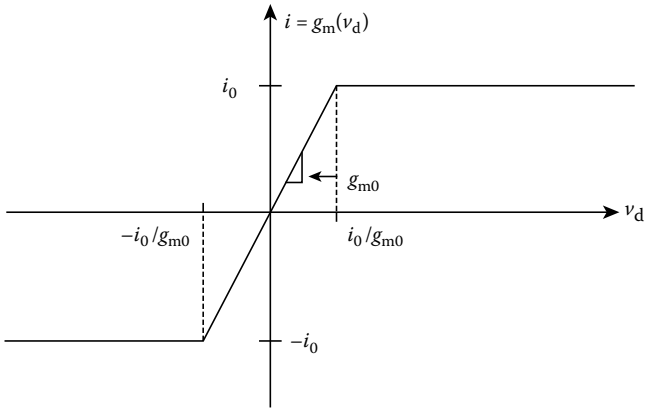


FIGURE 13.14 Nonlinear VCCS in the input stage of the op-amp.

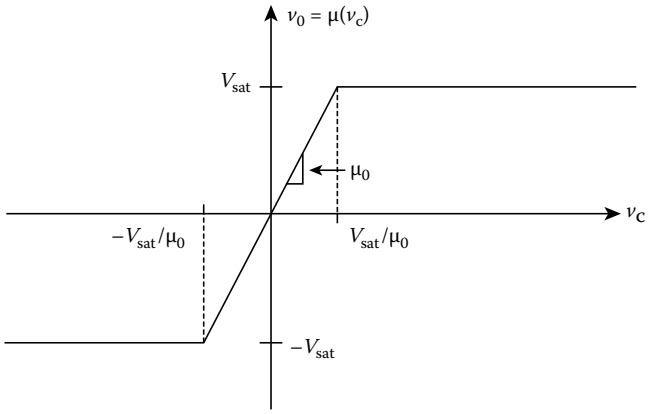


FIGURE 13.15 Nonlinear VCVS in the output stage of the op-amp.

The 3 dB bandwidth of this op-amp is given by $\omega_{3\ dB} = 1/(RC)$ with a 20 dB/decade roll-off. The slew rate is given by $SR = i_0/C$, while the DC open-loop gain $A_0 = g_{m0}\, R\mu_0$.

Using this simple model, not all, but some, of the most important nonlinear dynamical effects are properly described.

13.4 Connecting Multiport

After having dealt with the description of the circuit elements, we return to the interconnection structure, which is summarized in the CMP (Figure 13.1b). This multiport is linear, lossless, and reciprocal and its description stated in implicit form is simply KCL and KVL equations

$$\begin{bmatrix} \mathbf{B} & 0 \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ \mathbf{A} \end{bmatrix} \mathbf{i} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \mathbf{Bv} = 0 \quad \text{and} \quad \mathbf{Ai} = 0 \tag{13.18}$$

From Equation 13.18, the linearity is obvious, while the losslessness and reciprocity can be proven easily by making use of

$$\mathbf{BA}^{\mathrm{T}} = 0 \quad \mathbf{AB}^{\mathrm{T}} = 0 \quad \text{rank } \mathbf{A} + \text{rank } \mathbf{B} = b \tag{13.19}$$

(b is the number of ports of the CMP).

In addition to the perfect wires, which are the ingredients of CMP, ideal transformers (which are also linear, lossless, and reciprocal) can be accommodated in the CMP without changing the structure of Equations 13.18 and 13.19. **A** is an $(n-1) \times b$ incidence matrix containing the coefficients of any $(n-1)$ linearly independent nodal equations (or supernodal or cutset equations), while **B** is a $(b-(n-1)) \times b$ incidence matrix, the entries of which are the coefficients of any $(b-(n-1))$ linearly independent loop equations (or fundamental-loop equations).

13.5 Tableau Formulation

Combining the description of the CMP and the description of all circuit elements into one tableau, all information about the circuit under consideration is at our fingertips:

$$\begin{aligned} &\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad b \text{ linear algebraic equations} \\ &\mathbf{f}(\dot{\mathbf{v}}, \mathbf{v}, \dot{\mathbf{i}}, \mathbf{i}, t) = \mathbf{0}, \quad b \text{ nonlinear differential equations} \end{aligned} \tag{13.20}$$

This set of equations is not unique, although their solution is unique for properly modeled circuits. Instead of using derivatives $\dot{\mathbf{v}} = \mathrm{d}\mathbf{v}/\mathrm{d}t$, $\mathbf{i} = \mathrm{d}\mathbf{i}/\mathrm{d}t$, we could have used integrals, or we could have formulated the equations with $\mathbf{q}$, ϕ,$\dot{\mathbf{q}}$, $\dot{\phi}$ as variables.

It is important to note that at least half of the equations are linear. To solve the nonlinear equations numerical techniques are commonly used, however, this is beyond the scope of this chapter.

References

1. L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*, New York: McGraw-Hill, 1987.
2. J. E. Solomon, The monolithic opamp: A tutorial study, *IEEE Journal of Solid-State Circuits*, SC-9, 314–332, December 1974.
3. W. Mathis, *Theorie Nichtlinearer Netzwerke,* Berlin: Springer-Verlag, 1987.
4. R. K. Brayton, L. O. Chua, J. D. Rhodes, and R. Spence, *Modern Network Theory—An Introduction,* Saphorin, Switzerland: Georgi Publishing, 1978.

14

Controlled Circuit Elements

14.1 Controlled Sources **14**-1
Introduction • Voltage-Controlled Current Source • Voltage-Controlled Voltage Source • Current-Controlled Voltage Source • Current-Controlled Current Source
References **14**-7
14.2 Signal Converters **14**-7
Gyrator • Voltage Negative Impedance Converter • Current Negative Impedance Converter • Circulator
Defining Terms **14**-15
References **14**-15

Edwin W. Greeneich
Arizona State University

James F. Delansky
Pennsylvania State University

14.1 Controlled Sources

14.1.1 Introduction

Controlled sources generate a voltage or current whose value depends on, or is controlled by, a voltage or current that exists at some other point in the circuit. Four such sources exist: (1) voltage-controlled current source (VCCS), (2) voltage-controlled voltage source (VCVS), (3) current-controlled current source (CCCS), and (4) current-controlled voltage source (CCVS). In an ideal controlled source, the generated voltage or current does not vary with the load to which it is connected; this implies a zero output impedance for a voltage source and an infinite output impedance for a current source. In practice, actual controlled sources have finite output impedance, which causes the generated source to vary somewhat with the load. Circuit representations of the four ideal controlled sources are given in Figure 14.1. The input terminals on the left represent the controlling voltage or current, and the output terminals on the right represent the controlled voltage or current; the value of the controlled source is proportional to the controlling input through the constants g, μ, β, and r.

14.1.2 Voltage-Controlled Current Source

A VCCS produces an output current that is proportional to an input control voltage. The idealized small-signal low-frequency behavior of a field-effect transistor (FET) can be characterized by a VCCS, as illustrated in the equivalent circuit for an n-channel metal-oxide-semiconductor field-effect transistor (MOSFET) in Figure 14.2. In the circuit model, the small-signal drain current i_d of the transistor is proportional to the small-signal gate-to-source voltage v_{gs} through the transconductance parameter g_m.

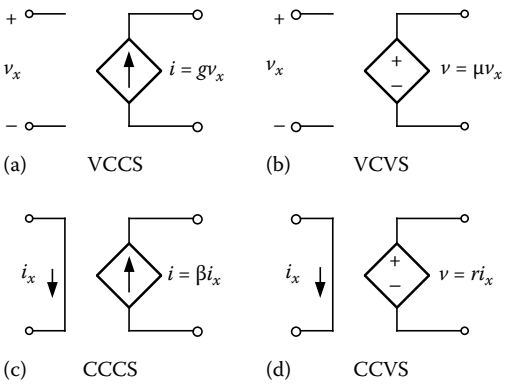


FIGURE 14.1 Circuit representations of ideal controlled sources.

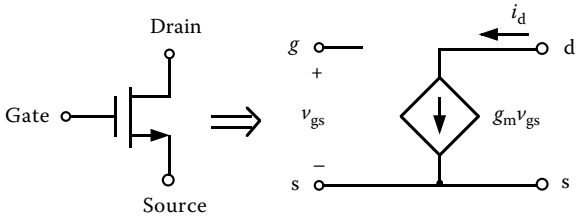


FIGURE 14.2 Small-signal equivalent circuit of a MOSFET.

14.1.2.1 SPICE Format

The format for a VCCS in the circuit simulation program SPICE [1] is illustrated in Figure 14.3. In the data statement, GXXX represents the source name containing up to eight alphanumeric characters (the first character G signifying a VCCS), N+ and N– are the positive and negative nodes of the source, NC+ and NC– are the positive and negative nodes between which the controlling voltage is measured, and VALUE is the multiplicative constant giving the value of the current source. The convention used is that positive current flows from the N+ node through the source to the N– node.

Figure 14.4 is an example of a circuit using a VCCS. The SPICE data specification for the source is: GS1 8 6 3 6 3.4.

14.1.2.2 Circuit Implementation

A circuit implementing a VCCS using an operational amplifier (op-amp) is depicted in Figure 14.5. Assuming the op-amp to be ideal, it is easy to show that $I_o = gV_x$, where $g = 1/R_1$, as indicated in the figure.

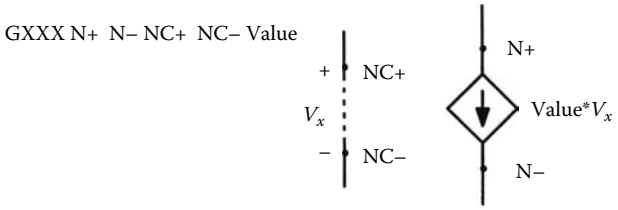


FIGURE 14.3 SPICE format for a VCCS.

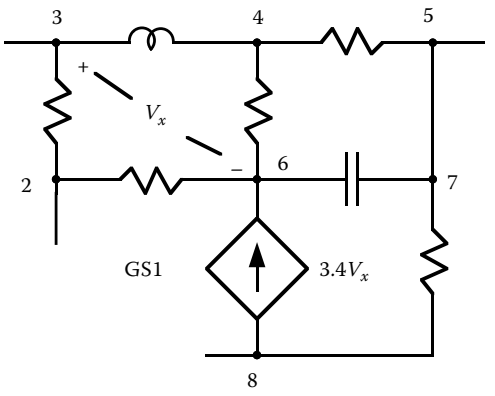


FIGURE 14.4 Example showing a portion of a circuit using a VCCS.

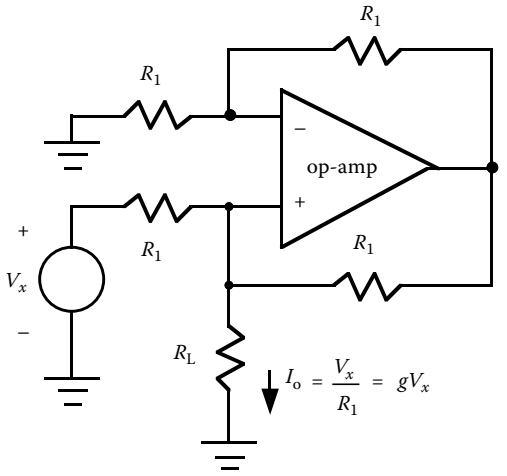


FIGURE 14.5 Op-amp implementation of a VCCS.

An alternate form of the VCCS circuit that uses fewer components is illustrated in Figure 14.6. In the circuit, Q_1 and Q_2 form a current mirror which, due to their equal base–emitter voltages, have the same collector currents, which are reflected in the collector currents of Q_3 and Q_4, causing them to have the same base–emitter voltages; the voltage at the emitter of Q_4 is thus equal to the input control voltage V_x. The emitter current of Q_4 is thus equal to V_x/R_1, which is then (neglecting base currents) equal to the collector current of Q_2. This current is mirrored by Q_5 giving the voltage-controlled output current I_o. This circuit derives from a general form of this configuration called a current conveyor [2]. With this circuit (and as with all practical current sources), the output current is not totally independent of the output voltage across the source, but instead, I_o shows a slight increase with increasing voltage. This is due to the finite output resistance of the source; in the circuit of Figure 14.6, this resistance is equal to the collector-to-emitter resistance of transistor Q_5. For an integrated circuit transistor, this resistance may be of the order of 50 kΩ or so.

14.1.3 Voltage-Controlled Voltage Source

A VCVS produces an output voltage that is proportional to an input control voltage. A voltage amplifier can be thought of as a VCVS; the output voltage is equal to the input voltage multiplied by the voltage gain, μ, of the amplifier.

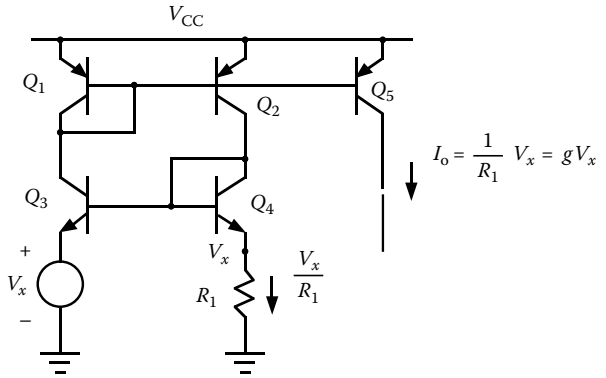


FIGURE 14.6 Circuit that implements a VCCS.

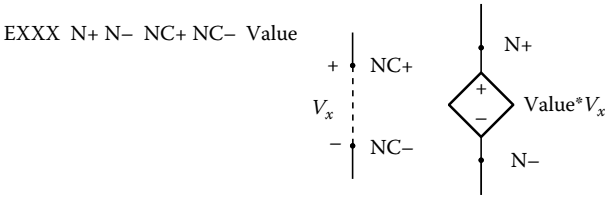


FIGURE 14.7 SPICE format for a VCVS.

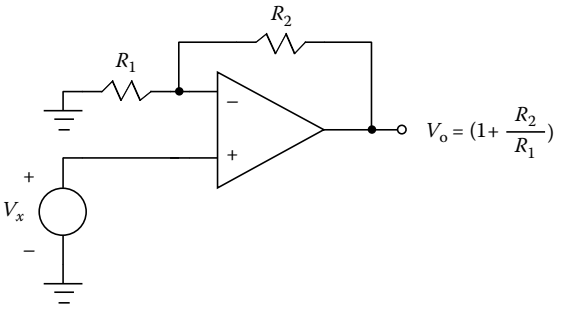


FIGURE 14.8 Circuit implementation of a VCVS using an op-amp.

14.1.3.1 SPICE Format

The SPICE format for a VCVS is given in Figure 14.7. The first character E in the source name signifies a VCVS. The output voltage of the source is given by the product of the VALUE constant and the control voltage V_x.

14.1.3.2 Circuit Implementation

Figure 14.8 is an implementation of the VCVS using an op-amp. Here $V_o = \mu V_x$ where $\mu = 1 + R_2/R_1$. The VCCS circuit of Figure 14.6 can be modified to produce a VCVS. In Figure 14.9, current sensing resistor R_2 develops a voltage drop equal to $V_x R_2/R_1$, which is buffered by a unity-gain stage to reduce loading effects on R_2. The output voltage V_o is thus proportional to the input control voltage V_x. The buffer should have a low output impedance to minimize variations in the output voltage, V_o, with load current drawn by the source.

14.1.4 Current-Controlled Voltage Source

A CCVS produces an output voltage that is proportional to an input control current. In this context, a CCVS may be thought of as a current-to-voltage transducer; the output voltage is equal to the input voltage multiplied by the transresistance, r, of the transducer.

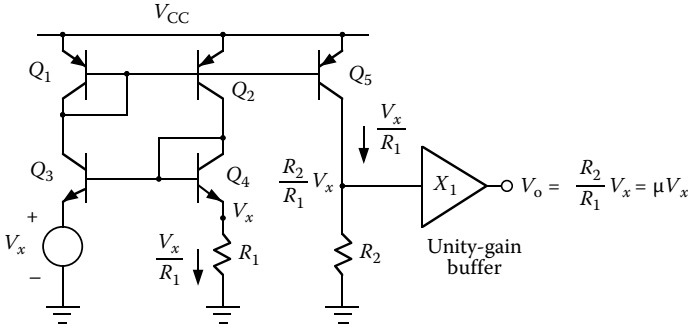


FIGURE 14.9 Circuit that implements a VCVS.

14.1.4.1 SPICE Format

The SPICE format for a CCVS is given in Figure 14.10. The first character *H* in the source name signifies a CCVS. There are no ammeters in SPICE, so currents are measured through voltage sources. VNAME is the voltage source through which the control current I_x is measured. The output voltage of the source is given by the product of VALUE and I_x. If the point in the circuit at which the control current is to be measured does not contain a voltage source, a test voltage source of zero value can be inserted in the circuit.

Figure 14.11 is an example of a circuit using a CCVS. The data statement for the source is HVS2 3 2 VTEST 0.2.

VTEST is a zero-valued voltage source inserted into the circuit to measure the control current I_x. Its data statement is VTEST 6 7 0.

14.1.4.2 Circuit Implementation

An op-amp implementation of a CCVS is given in Figure 14.12. Here, $V_o = rI_x$ where $r = R_1 + R_2$. A simple circuit implementing a CCVS is shown in Figure 14.13. In the current mirror comprising

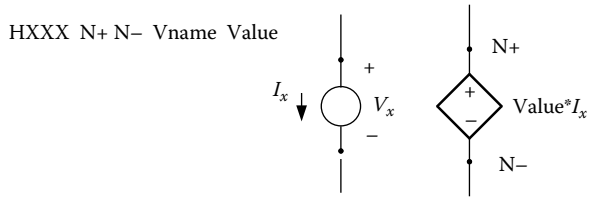


FIGURE 14.10 SPICE format for a CCVS.

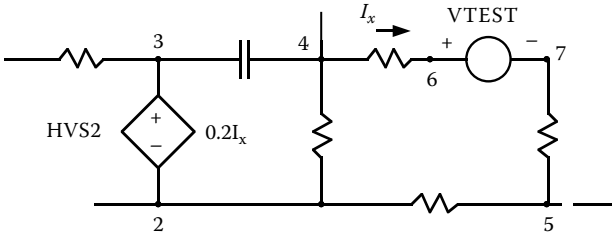


FIGURE 14.11 Example showing a portion of a circuit using a CCVS.

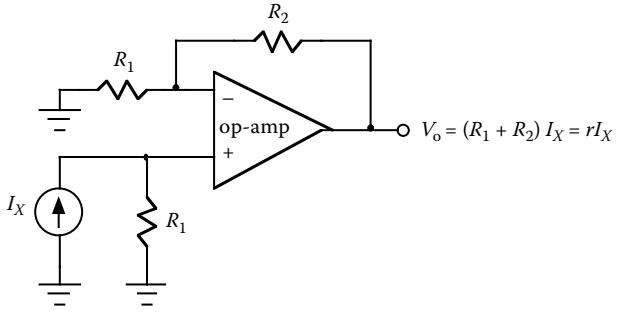


FIGURE 14.12 op-amp implementation of a CCVS.

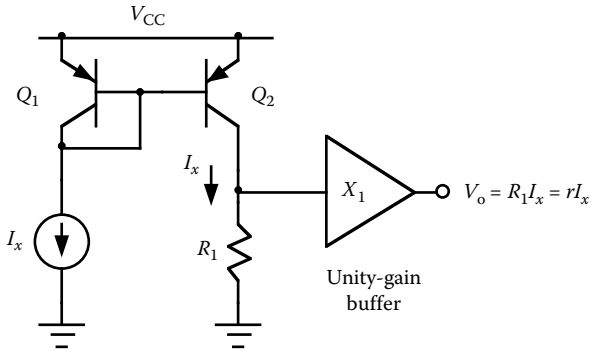


FIGURE 14.13 Circuit that implements a CCVS.

transistors Q_1 and Q_2, the collector current of Q_2 is equal (neglecting base currents) to the control current I_x. The voltage across R_1 is then equal to $R_1 I_x$, which after the buffer, is the current-controlled output voltage.

14.1.5 Current-Controlled Current Source

A CCCS produces an output current that is proportional to the input control current. The idealized large-signal (and small-signal as well) low-frequency behavior of a bipolar transistor can be characterized by a CCCS, as illustrated in the equivalent circuit for an NPN transistor in Figure 14.14. In the circuit model, the collector current I_C is proportional to the base current I_B through the current gain parameter β_F.

14.1.5.1 SPICE Format

The SPICE format for a CCCS is given in Figure 14.15. The first character F in the source name signifies a CCCS. The output current is given by the product of the VALUE constant and the control current I_x. As with the CCVS, the controlling current is measured through an independent voltage source.

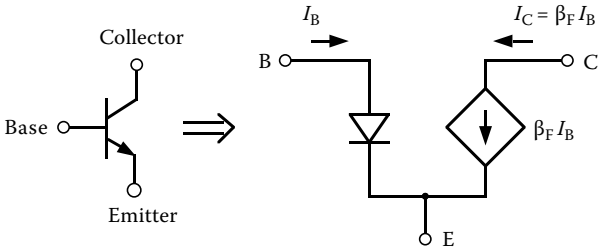


FIGURE 14.14 Large-signal equivalent circuit of a bipolar transistor.

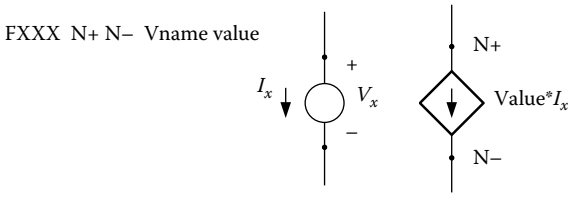


FIGURE 14.15 SPICE format for a CCCS.

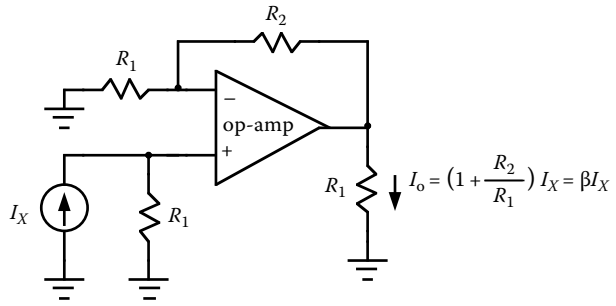


FIGURE 14.16 op-amp implementation of a CCCS.

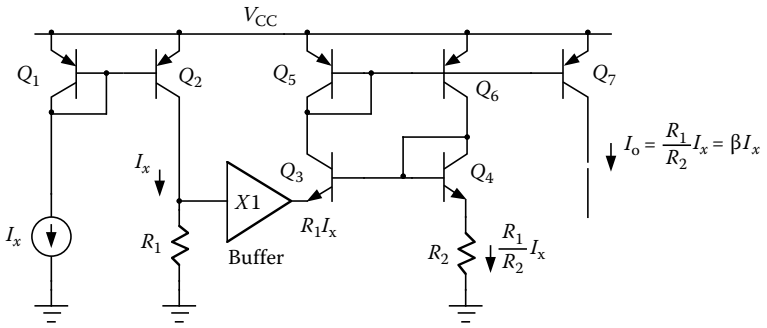


FIGURE 14.17 Circuit that implements a CCCS.

14.1.5.2 Circuit Implementation

Figure 14.16 is an op-amp implementation of a CCCS. Here, $I_o = \beta I_x$ where $\beta = 1 + R_2/R_1$.

The CCVS circuit in Figure 14.13 can be combined with the VCCS circuit in Figure 14.6 to produce a CCCS, as illustrated in Figure 14.17. The output voltage of the CCVS stage is equal to $R_1 I_x$, which, applied at the emitter of Q_3, is the input voltage of the VCCS stage; the output current is equal to $(R_1/R_2)I_x$, and thus proportional to the input control current.

References

1. Nagel, L.W., 1975. SPICE2: A computer program to simulate semiconductor circuits, Electronics Research Laboratory Report No. ERL-M520, University of California, Berkeley.
2. Ioumazou, C., Lidgey, F.J., and Haigh, D.G., eds., 1990. *Analogue IC Design: The Current-Mode Approach*, London: Peter Peregrinus Ltd.

14.2 Signal Converters

An accessible terminal pair of a network, regarded as a single entity to which an independent 2-terminal signal generator (source or input) is to be connected, is called a port of the network. An equivalent view of a port of a network is an accessible terminal pair such that the current entering one of the terminals of the pair is the same current leaving the other terminal of the pair. Thus, an accessible terminal pair of a network is a port when this terminal pair is terminated by (connected to) a 1-port network. For a given port of a network, the port voltage (the voltage drop between the two terminals) and the port current

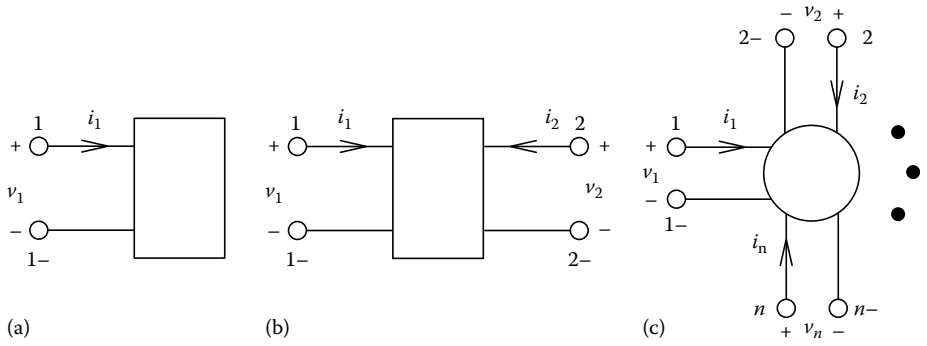


FIGURE 14.18 (a) A 1-port. (b) A 2-port. (c) An *n*-port.

(the current entering one of the terminals) are to be associated as follows: the positive sign for the voltage (drop) is always assumed at the terminal at which the current (positive charges) enters the network. A 1-port, 2-port, and general *n*-port are illustrated in Figure 14.18.

In linear network theory, a class of *n*-ports (particularly $n = 2$), known as signal converters, has become salient as crucial building blocks. These 2-port signal converters include the various versions of the ''transformer,'' ''controlled source,'' and ''operational amplifier'' found elsewhere in Section II. This section will introduce the 2-port signal converters known as the ''gyrator'' and ''negative impedance converter (NIC).'' As extensions of these, the ($n \geq 2$)-port ''circulator'' is developed and, as special cases, the 1-ports known as ''nullator'' and ''norator'' (properly viewed as degenerate cases) and the 2-port known as ''nullor'' are given.

One of the most general external descriptions of a linear *n*-port network is the ''scattering'' matrix. In many cases, a linear *n*-port network may be externally described by one or more of the ''open-circuit impedance,'' ''short-circuit admittance,'' ''hybrid,'' ''inverse hybrid,'' ''chain (transmission),'' or ''inverse chain (inverse transmission)'' matrix. For details of these *n*-port descriptions, see [1]. As necessary or convenient, any of the above *n*-port descriptions will be utilized in this section.

14.2.1 Gyrator

The concept of reciprocity satisfied by a linear *n*-port network ($n = 2$, in particular) will be useful in the present context.

Definition: A linear *n*-port network is said to be a reciprocal linear *n*-port network if the port voltages and currents satisfy

$$\sum_{k=1}^{n} \left[v_k^{(1)} i_k^{(2)} - v_k^{(2)} i_k^{(1)} \right] = 0 \tag{14.1}$$

where $v_k^{(1)}$, $i_k^{(1)}$ and $v_k^{(2)}$, $i_k^{(2)}$ are any two distinct sets of port voltages and currents that satisfy Kirchhoff's laws for the linear *n*-port. If Equation 14.1 is not satisfied, the linear *n*-port is said to be nonreciprocal. [Note: for many useful linear *n*-port networks, Equation 14.1 can be derived from Tellegen's theorem in Chapter 7].

The idea of isolating the nonreciprocity of a linear passive *n*-port in a single network building block was first advanced by Tellegen [2]. The linear passive 2-port developed there, christened the gyrator, was

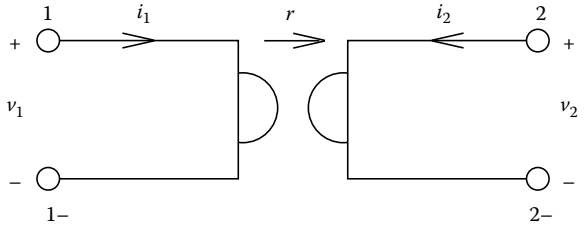


FIGURE 14.19 The ideal 2-port gyrator.

shown to be necessary and sufficient for this purpose. This (ideal) 2-port gyrator is described by the skew-symmetric open-circuit impedance matrix

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} = \begin{bmatrix} 0 & -r \\ r & 0 \end{bmatrix} \tag{14.2}$$

where r (the gyrator transfer impedance parameter) is a real positive number, and is depicted in Figure 14.19.

It can be observed that $v_1 = -ri_2$ and $v_2 = ri_1$ (so the signal conversion is clear), and by multiplying these together to obtain $rv_1i_1 = -rv_2i_2$ or $v_1i_1 + v_2i_2 = 0$, then no energy is generated, dissipated, or stored. Thus, the ideal gyrator is a lossless passive nonreciprocal 2-port. Also easily shown is that, if port 2 is terminated with the 1-port (driving-point) impedance Z, then the driving-point impedance seen at port 1 is $Z_{11} = -z_{12}z_{21}/Z = r^2/Z$ so that the ideal 2-port gyrator is also an ideal impedance invertor. This leads to another attribute of the ideal 2-port gyrator as a fundamental building block for the synthesis of linear networks. If Z in the above is the impedance of an ideal capacitor C, i.e., $Z = 1/sC$, then

$$Z_{11'} = \frac{r^2}{Z} = r^2sC = L_{11'}s \tag{14.3}$$

which means that the driving-point impedance $Z_{11'}$ is exactly equivalent to the impedance of an ideal inductor $L_{11'} = r^2C$. This opens a viable avenue to the inductorless synthesis of linear passive *RLC* networks.

The ideal 2-port gyrator is a 4-terminal passive network (device or element). For network (and physical realization) purposes, however, it is usually implemented with active elements (and thus considered as an "active" building block) and results in a 3-terminal or "common ground" 2-port. For this reason, the simulation of a nongrounded ("floating") inductor in a network is not straightforward. The active realization of the ideal gyrator also results in a nonideal gyrator (i.e., in general $z_{11} \neq 0$, $z_{22} \neq 0$, $z_{12} \neq -z_{21}$). These concerns are treated elsewhere [3].

A circuit using two operational amplifiers and *R*'s and a single capacitor for inductor simulation (and thus a gyrator simulation) was proposed by Antoniou [4]. This circuit has subsequently become very widely used. Assuming ideal operational amplifiers and general *RC* 1-ports, this circuit is illustrated in Figure 14.20.

Analysis yields

$$Z_{11} = \frac{V_1}{I_1} = \left(\frac{Z_1Z_3}{Z_2Z_4}\right)Z_5 \tag{14.4}$$

For Equation 14.4 to appear as Equation 14.3, two choices exist because either Z_2 or Z_4 must be the impedance of the capacitor. The simplest choices then are

1. $Z_1 = R_1$, $Z_2 = \dfrac{1}{C_2s}$, $Z_3 = R_3$, $Z_4 = R_4$, $Z_5 = R_5$ results in

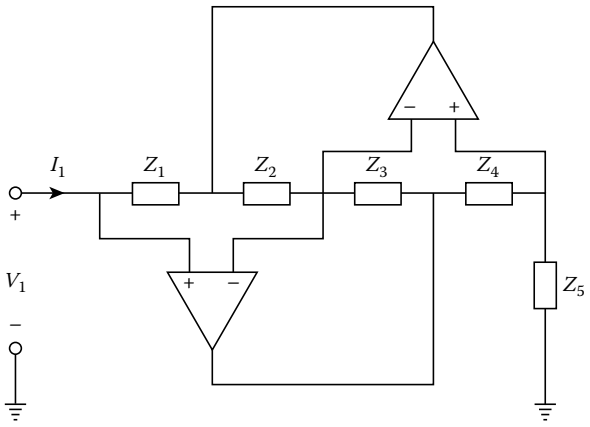


FIGURE 14.20 Antoniou's circuit.

$$Z_{11} = \left(\frac{R_1 R_3 R_5}{R_4}\right) C_2 s = L_{11} s$$

2. $Z_1 = R_1, \quad Z_2 = R_2, \quad Z_3 = R_3, \quad Z_4 = \dfrac{1}{C_4 s}, \quad Z_5 = R_5$ results in

$$Z_{11} = \left(\frac{R_1 R_3 R_5}{R_2}\right) C_4 s = L_{11} s$$

These are known as type-*A* and type-*B* simulations, respectively. Assume the capacitor in *A* and *B* above is removed (pliers-type entry) from Figure 14.20, thus forming a 2-port. With the proper choice of numbering port 2 terminals, the resulting 2-port is a 4-terminal ideal gyrator. On the other hand, forming the second port by removing Z_5 from Figure 14.20, and the obvious choice of numbering port 2 terminals, results in a grounded 2-port impedance converter with

$$Z_{11} = \left(\frac{Z_1 Z_3}{Z_2 Z_4}\right) Z_5 = K(s) Z_5 \tag{14.5}$$

This is Antoniou's generalized impedance convertor (GIC).

The rational function $K(s)$ in Equation 14.5 can, in general, be any function subject to Z_1, Z_2, Z_3, and Z_4 being the driving-point impedances of *RC* 1-port networks. So if the GIC of Equation 14.5 has $Z_1 = 1/C_1 s$, $Z_2 = R_2$, $Z_3 = R_3$, but $Z_4 = R_4$ and $Z_5 = 1/C_5 s$, then

$$Z_{11}(s) = \frac{R_3}{C_1 C_5 R_2 R_4 s^2} = \frac{1}{D s^2} \tag{14.6}$$

so that for sinusoidal excitations ($s = j\omega$)

$$Z_{11}(j\omega) = \frac{-1}{D\omega^2} = R_{11}\left(\omega^2\right) \tag{14.7}$$

Thus, Z_{11} of Equation 14.7 is a frequency-dependent negative resistance (FDNR). Alternate realizations of FDNR are possible. The FDNR element plays an important role in the design of active *RC* networks.

14.2.2 Voltage Negative Impedance Converter

Suppose a 2-port signal converter such that, with one of its ports terminated with an impedance, the driving-point impedance at the other port is proportional to the negative of the terminating impedance. Such a 2-port is known as an NIC [5]. For a general 2-port terminated at port 2 with the impedance $Z = 1/Y$, the driving-point impedance seen at port 1 is, in terms of the hybrid parameters of the 2-port,

$$Z_{11} = \frac{h_{11} - h_{12}h_{21}}{h_{22} + Y} \tag{14.8}$$

Hence, necessary and sufficient conditions for a 2-port to be an NIC is that $h_{11} = 0$, $h_{22} = 0$, and $h_{12}h_{21} > 0$, so Equation 14.8 becomes

$$Z_{11} = -h_{12}h_{21}Z = -kZ \tag{14.9}$$

where $k > 0$ is called the negative impedance parameter. Now $h_{12}h_{21} > 0$ holds in two cases for h_{12} and h_{21} real:

$$1.\quad h_{12} < 0 \quad \text{and} \quad h_{21} < 0 \tag{14.10}$$

$$2.\quad h_{12} > 0 \quad \text{and} \quad h_{21} > 0 \tag{14.11}$$

Consider an NIC satisfying Equation 14.10. From the hybrid description, this 2-port signal converter has $v_1 = h_{12}v_2$ and since $h_{12} < 0$ this implies a voltage reversal between the ports, while $i_2 = h_{21}i_1$ and because $h_{21} < 0$ this implies the current direction remains the same. For this reason, Equation 14.10 defines a voltage inversion negative impedance converter (VNIC).

14.2.3 Current Negative Impedance Converter

Consider an NIC satisfying Equation 14.11. From the hybrid description, this 2-port signal converter has $v_1 = h_{12}v_2$ and since $h_{12} > 0$, this implies no voltage reversal between the ports, while $i_2 = h_{21}i_1$ and because $h_{21} > 0$ this implies the current directions have reversed. For this reason, Equation 14.11 defines the current inversion negative impedance converter (INIC).

As with the gyrator, for network (and physical realization) purposes, the NIC is implemented with active devices but for a more fundamental reason. Consider k in Equation 14.9 to be 1 with $h_{12} = h_{21}$ to be either 1 or -1. Then for VNIC or INIC this implies $v_1i_1 - v_2i_2 = 0$ so the NIC is inherently active. Any physical realization of the NIC will result in a nonideal NIC (i.e., in general $h_{11} \neq 0$, $h_{22} \neq 0$, $h_{12}h_{21} \neq 1$). These concerns are treated elsewhere, e.g., Ref. [3].

An obvious use of an NIC in active network synthesis is to obtain negative elements, i.e., negative resistor or inductor or capacitor from positive resistor or inductor or capacitor, respectively. Another way an NIC is used in active network synthesis is the partitioning of a network using an NIC. Consider two cascaded 2-ports as shown in Figure 14.21 to form an overall 2-port.

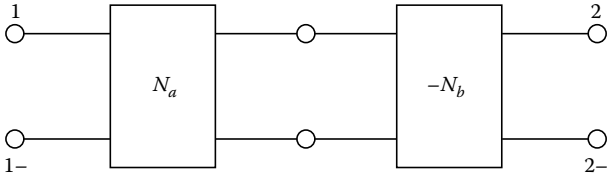


FIGURE 14.21 Cascade of 2-ports designated N_a and $-N_b$.

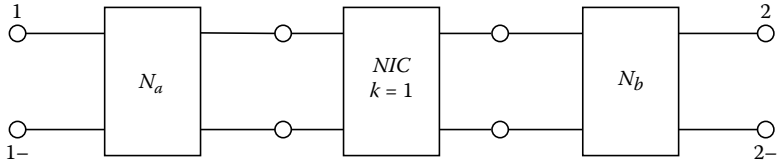


FIGURE 14.22 Equivalent to Figure 14.21.

Suppose in Figure 14.21, N_a is a 2-port with all positive elements and $-N_b$ is a 2-port with all negative elements. Then it can be shown that the overall 2-port shown in Figure 14.22 with an NIC of $k = 1$, and with N_b being the same as $-N_b$ in Figure 14.21, except it is now composed of all positive elements, is equivalent to the overall 2-port shown in Figure 14.21. These and other methods of using NIC's in active *RC* synthesis may be found elsewhere, e.g., Ref. [3].

14.2.4 Circulator

As discussed in Section 14.2.1, the gyrator is the basic representation of nonreciprocity in linear networks. It is quite natural to believe this property can also be exploited to derive a means of controlling the power flow in n-port ($n \geq 2$) linear networks from the input port to the remaining ports in a prescribed manner. This is indeed the situation and the most important of such n-ports are known as circulators.

Circulators are best described in terms of the scattering matrix, since this makes their function very clear. A brief discussion of the scattering description for a 2-port network (extended in a natural way for n-port ($n > 2$) networks) follows, see Ref. [6]. Consider the terminated 2-port in Figure 14.23.

Assume $r_{01} > 0$ and $r_{02} > 0$. Define incident and reflected power waves $\boldsymbol{a} = [a_1\ a_2]'$ and $\boldsymbol{b} = [b_1\ b_2]'$, respectively, where the prime denotes matrix transpose, as

$$\boldsymbol{a} = (1/2)\boldsymbol{R}_0^{-1/2}(\boldsymbol{V} + \boldsymbol{R}_0\boldsymbol{I}) \tag{14.12}$$

$$\boldsymbol{b} = (1/2)\boldsymbol{R}_0^{-1/2}(\boldsymbol{V} - \boldsymbol{R}_0\boldsymbol{I}) \tag{14.13}$$

where

$\boldsymbol{R}_0 = \text{diag}(r_{01}, r_{02})$
$\boldsymbol{V} = [V_1\ V_2]'$
$\boldsymbol{I} = [I_1\ I_2]'$

Assume that $\boldsymbol{V} = \boldsymbol{Z}\boldsymbol{I}$ for the 2-port and because Equations 14.12 and 14.13 are also linear relations between $\boldsymbol{V}$ and $\boldsymbol{I}$, then $\boldsymbol{a}$ and $\boldsymbol{b}$ are also linearly related as

$$\boldsymbol{b} = \boldsymbol{S}\boldsymbol{a} \tag{14.14}$$

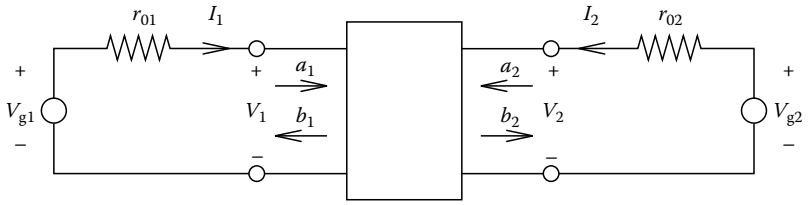


FIGURE 14.23 General terminated 2-port network.

where $\boldsymbol{S}$ is the scattering matrix with respect to $\boldsymbol{R}_0$. For the 2-port network in Figure 14.23, Equation 14.14 is explicitly

$$\begin{aligned} b_1 &= S_{11}a_1 + S_{12}a_2 \\ b_2 &= S_{21}a_1 + S_{22}a_2 \end{aligned} \tag{14.15}$$

Hence, the scattering parameters are determined as

$$S_{jj} = \frac{b_j}{a_j}, \quad \text{for } a_k = 0 \text{ and } k \neq j \tag{14.16}$$

and

$$S_{kj} = \frac{b_k}{a_j}, \quad \text{for } a_k = 0 \text{ and } k \neq j \tag{14.17}$$

The parameter S_{jj} is called the reflection coefficient at the jth port and S_{kj} is called the transmission coefficient from port j to port k. The conditions in Equations 14.16 and 14.17, together with Equations 14.12 and 14.13, imply.

$$S_{jj} = \frac{(Z_{jj} - r_{0j})}{(Z_{jj} + r_{0j})} \tag{14.18}$$

and

$$S_{kj} = -2\sqrt{r_{0j}r_{0k}}\,\frac{I_k}{V_{gj}} \tag{14.19}$$

The network is said to be matched at port j if $S_{jj} = 0$, or from Equation 14.18, $r_{0j} = Z_{jj}$. Also from Equation 14.19 and $|S_{kj}|^2 = |b_k|^2/|a_j|^2$, it is clear that $|S_{kj}|^2$ is the power gain of the terminated n-port from the source at port j to the load resistor r_{0k} at port k. For example, consider the ideal 2-port gyrator with gyration transfer parameter $r > 0$ and $r_{01} = r_{02} = r$. Here,

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix} \tag{14.20}$$

The simplest type of circulator is the 2-port circulator. Consider the 2-port in Figure 14.24, which has (for $r_{01} = r_{02} = 1$)

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{14.21}$$

This 2-port circulator has unity power transmission from port 1 to port 2, but zero power transmission from port 2 to port 1, and is also called an isolator or one-way line.

Now a 3-port circulator may be defined by the scattering matrix

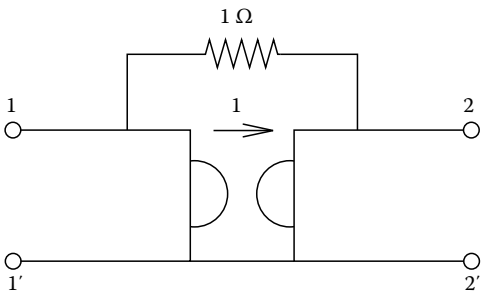


FIGURE 14.24 2-Port circulator.

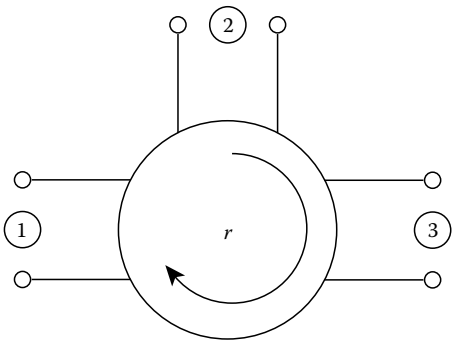


FIGURE 14.25 Matched 3-port circulator.

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & S_{13} \\ S_{21} & 0 & 0 \\ 0 & S_{32} & 0 \end{bmatrix} \tag{14.22}$$

with each nonzero $S_{kj} = \pm 1$, so it has unity power transfer from port 1 to 2, 2 to 3, and 3 to 1. This 3-port circulator is usually depicted as shown in Figure 14.25, where the + and − terminals of each port depend on the selection of each nonzero S_{kj} in Equation 14.22. Likewise, a 4-port circulator may be defined by the scattering matrix

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & S_{14} \\ S_{21} & 0 & 0 & 0 \\ 0 & S_{32} & 0 & 0 \\ 0 & 0 & S_{43} & 0 \end{bmatrix} \tag{14.23}$$

with each nonzero $S_{kj} = \pm 1$, and depicted as suggested in Figure 14.25. Similarly, the n-port ($n > 4$) circulator may be defined and depicted as suggested in Figure 14.25. See Ref. [6] for details.

14.2.4.1 Nullator

It is a fact that every linear passive n-port network has a scattering description. However, to complete the description of general linear n-port networks (which may not have a scattering description), a few "pathological" or "degenerate" 1- and 2-ports must be included [7]. The first of these is the 1-port linear network denoted as the "nullator." For the exact configuration obtained by terminating Figure 14.25 (with normalization parameters equal to unity) at port 2 with $-1\ \Omega$ and at port 3 with $+1\ \Omega$, the driving-point relations seen at port 1 are

$$V_1 = I_1 = 0 \tag{14.24}$$

so the resulting 1-port is at once a short and open circuit! This linear 1-port has been designated as the nullator. The circuit symbol for the nullator is depicted in Figure 14.26.

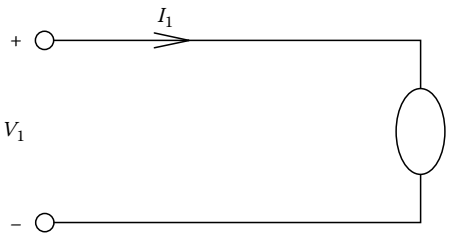


FIGURE 14.26 Nullator ($V_1 = I_1 = 0$).

14.2.4.2 Norator

Another degenerate 1-port can be obtained from the exact configuration obtained by terminating Figure 14.25 (with unity normalization parameters) at port 2 with $+1\ \Omega$ and at port 3 with $-1\ \Omega$. The resulting driving-point relations seen at port 1 are that

$$V_1 \text{ and } I_1 \text{ are completely independent} \tag{14.25}$$

This (nonreciprocal) linear 1-port has been designated as the norator. The circuit symbol for the norator is shown in Figure 14.27.

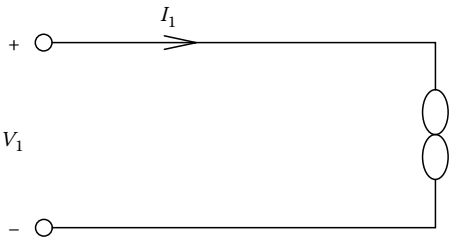


FIGURE 14.27 The norator (V_1 and I_1 are arbitrary).

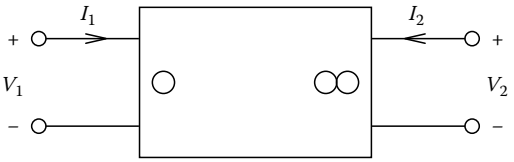


FIGURE 14.28 The nullor ($V_1 = I_1 = 0$, V_2, and I_2 arbitrary).

14.2.4.3 Nullor

The final building block for the most general n-port linear network is the 2-port designated as the nullor. It is defined as a 2-port which, at port 1, demands $V_1 = I_1 = 0$ while simultaneously at port 2, V_2 and I_2 are arbitrary! These relations can be obtained from a 4-port circulator with two of its ports appropriately terminated with negative and positive resistors. The symbol for the nullor is given in Figure 14.28. Both 4 and 3 terminal equivalent circuits exist for the nullor.

These singular network elements, the nullator, the norator, and the nullor, have been used to generate realizable circuits for the various signal converters discussed in this section. They have also been used to obtain realizable circuits for other widely used circuit elements such as the family of controlled sources, op-amps, transistors, etc. See Refs. [3,7] for the applications mentioned above.

Of course, many references are possible for the use of the network elements discussed in this section. Among others, Refs. [8,9] contain extensive bibliographies.

Defining Terms

Reciprocal linear *n*-port: The port voltages and currents satisfy Equation 14.1 and its restrictions.
Ideal gyrator: The basic nonreciprocal 2-port satisfying Equation 14.2.
GIC: A 2-port that satisfies Equation 14.5 and its conditions.
FDNR: A 1-port that has the driving-point impedance of Equation 14.7.
VNIC: A 2-port satisfying Equation 14.10 and foregoing conditions.
INIC: A 2-port satisfying Equation 14.11 and foregoing conditions.
The *n*-port ($n \geq 3$) circulator: Described by its n-port scattering matrix with main-diagonal elements zero and off-diagonal elements restricted to only one per row or column of unity magnitude.
Nullator: A 1-port such that $V_1 = I_1 = 0$.
Norator: A 1-port such that V_1 and I_1 and arbitrary.
Nullor: A 2-port such that $V_1 = I_1 = 0$, while V_2 and I_2 are arbitrary.

References

1. R. W. Newcomb, *Linear Multiport Synthesis*, New York: McGraw-Hill, 1966.
2. B. D. H. Tellegen, The gyrator, a new electric network element, *Phillips Res. Rep.*, 3, 81–101, 1948.
3. S. K. Mitra, *Analysis and Synthesis of Linear Active Networks*, New York: John Wiley & Sons, 1969.
4. A. Antoniou, Realization of gyrators using operational amplifiers and their use in RC-active network synthesis, *Proc. IEE*, 116, 1838–1850, 1969.
5. J. L. Merill Jr., Theory of the negative impedance converter, *Bell System Tech. J.*, 30, 88–109, 1951.
6. H. J. Carlin and A. B. Giordano, *Network Theory*, Englewood Cliffs, NJ: Prentice Hall, 1964.
7. H. J. Carlin, Singluar network elements, *IEEE Trans. Circuit Theory*, CT-11, 67–72, 1964.
8. S. K. Mitra, *Active Inductorless Filters*, New York: IEEE, 1971.
9. M. Herpy and J.-C. Berka, *Active RC Filter Design*, New York: Elsevier Science, 1986.

15

Bipolar Junction Transistor Amplifiers

15.1 Introduction **15**-1
Need for Amplification • Brief History of Amplifying Devices
15.2 Amplifier Types and Applications **15**-2
Input and Output Variables • Frequency Range • Output Power • Distortion in Power Stages • Operational Amplifiers
15.3 Differences between Discrete and IC Amplifier Design **15**-6
Component Differences • Parasitic Differences
15.4 Building Blocks for IC Amplifiers **15**-8
Small-Signal Models • Active Loads • Common Emitter Stage • Common-Base Stage • Emitter Follower
15.5 Compound Stage Building Blocks for IC Amplifiers **15**-15
Cascode Amplifier Stage • Differential Stage • BJT Differential Pair
15.6 Operational Amplifiers **15**-21
Classical IC Op-Amp Architecture • High-Gain Differential Stage • Second Amplifier Stage • Op-Amp Specifications
15.7 Wideband Amplifiers **15**-27
Composite or Compound Stages • Feedback Cascades
15.8 IC Power Output Stages **15**-29
Thermal Resistance • Circuit or Conversion Efficiency • Class-B Output Stage • Class-D Output Stages
References **15**-36

David J. Comer
Brigham Young University

Donald T. Comer
Brigham Young University

15.1 Introduction

15.1.1 Need for Amplification

The field of electronics includes many applications wherein important information is generated in the form of a voltage or current waveform. Often this voltage is too small to perform the function for which it is intended. Examples of this situation are the audio microphone in a sound amplification signal, the infrared diode detector of a heat-seeking missile, the output of the light intensity sensor of a CD player,

Much of the material from Sections 4, 5, and 6 comes from D. J. Comer and D. T. Comer, *Fundamentals of Electronic Circuit Design*. Copyright 2003 by John Wiley & Sons, Inc.

the output of a receiving antenna for an amplitude modulation (AM) or frequency modulation (FM) signal, the cell phone signal received by the base station before retransmission, and the output of a receiving dish antenna in a satellite TV system. Each of these signals have peak voltages in the range of hundreds of microvolts to hundreds of millivolts. To transmit a signal over phone lines or through the atmosphere and then activate a speaker or to illuminate a cathode-ray tube or other type of TV display may require a signal of several volts rather than millivolts.

An audio amplification system may require peak voltages of tens of volts to drive the speaker. The speakers in a CD system or in radio receivers may also require several volts to produce appropriate levels of volume. The video information for a TV signal may need to be amplified to a value of thousands of volts to drive the picture tube or display. Even in computer systems, the signals recovered from the floppy or hard disk drives must be amplified to logic levels to become useful. Amplification of each of these signals is imperative to make the signal become useful in its intended application. Consequently, amplifiers make up a part of almost every electronic system designed today.

At the present time, the bipolar junction transistor (BJT) and the metal-oxide semiconductor field-effect transistor (MOSFET) are the major devices used in amplification of electronic signals.

15.1.2 Brief History of Amplifying Devices

Amplification is so important that the so-called ''era of electronics'' was ushered in only after the development of the triode vacuum tube that made amplification possible in 1906 [1]. This device enabled public address systems and extended distances over which telephone communications could take place. Later, this element allowed the small signal generated by a receiving antenna located miles away from a transmitter to be amplified and demodulated by a radio receiver circuit. Without amplifiers, communications over long distances would be difficult if not impossible.

Improvement of the vacuum tube continued for several years with the tetrode and pentode tubes emerging in the late 1920s. These devices enabled the rapid development of AM radio and television. Vacuum tubes dominated the electronics field well into the 1950s serving as the basic element in radio, television, instrumentation, and communication circuits.

The forerunner of the BJT, the point-contact transistor, was invented in 1947 [2]. The BJT followed shortly and grew to dominance in amplifier applications by the mid 1960s. From then until the 1990s, this device had no peer in the electronics field. In the 1990s, the MOSFET became important as an amplifying device; however, the silicon BJT along with the newer heterojunction BJT (HBT), continue to be used in many amplifier applications.

15.2 Amplifier Types and Applications

Amplifiers can be classified in various ways. The input and output variables can be used to describe the amplifier. The frequency range of the output variable or the power delivered to a load can also describe a major characteristic of an amplifier. In the case of an operational amplifier (op-amp), the mathematical operations that can be performed by the circuit suggest the name of the amplifier.

15.2.1 Input and Output Variables

Electronic transducers generally produce either voltage or current as the quantity to be amplified. After amplification, the signal variable of interest could again be voltage or current, depending on the circuit or device driven. A given amplifier can then have either current or voltage as the input signal and current or voltage as the output signal. As the input variable changes, the amplifier produces a corresponding output variable change. Gain of an amplifier is defined as the ratio of output variable change to the input variable change as shown in Figure 15.1.

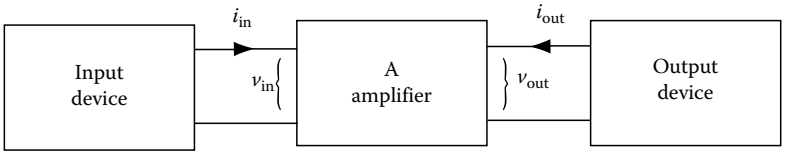


FIGURE 15.1 Amplifier circuit.

TABLE 15.1 Amplifiers Based on Output/Input Ratio

Input Variable	Output Variable	Gain	Amplifier Type
Voltage	Voltage	v_{out}/v_{in}	Voltage amplifier
Current	Current	i_{out}/i_{in}	Current amplifier
Voltage	Current	i_{out}/v_{in}	Transadmittance amplifier
Current	Voltage	v_{out}/i_{in}	Transimpedance amplifier

There are four possible ratios for the gain. These are summarized in Table 15.1 along with the corresponding amplifier type.

The voltage amplifier is used more often than the others, but each has specific applications. Some photo detectors generate output current proportional to the light intensity and this current is the input variable to the amplifier. The output variable may be voltage for this application. There are other occasions when an amplifier must generate an output current that is proportional to the input variable, thus, a given application may require any one of the four possible types listed in Table 15.1.

15.2.2 Frequency Range

The frequency range over which an amplifier exhibits useful gain is an important specification. An audio amplifier may have a relatively constant gain from tens of hertz up to nearly 20 kHz. The magnitude of this constant gain is often called the midband gain. Those frequencies at which the magnitude of gain falls by 3 dB from the midband value are called 3-dB frequencies. For a high-fidelity audio amplifier, the lower 3-dB frequency may be 20 Hz while the upper 3-dB frequency may be 20 kHz. The bandwidth of this amplifier extends from 20 Hz to 20 kHz. In spite of the relatively small absolute bandwidth, this type of amplifier is referred to as a wideband amplifier. The ratio of the upper 3-dB frequency to the lower 3-dB frequency is 20,000/20 which is 1000 in this case.

Another important type of amplifier is that used in radio receiver circuits. The receiver for an AM signal consists of several stages that amplify over a frequency range from about 450 kHz to about 460 kHz. The gain may be maximum at 455 kHz, but drops very rapidly toward zero when the frequency is below 450 kHz or above 460 kHz. This type of amplifier is called a narrowband amplifier because the ratio of upper 3-dB frequency to lower 3-dB frequency has a near-unity value of 460/450 in this case.

15.2.3 Output Power

Some audio speakers may require hundreds of watts of power to operate at the desired level. An LED display may only require milliwatts of power. The amplifiers that drive these devices can be classified as high power or low power, respectively. While there are no fixed power values to differentiate, amplifiers that generate outputs in the milliwatt range are called low-power amplifiers or simply amplifiers. Amplifiers that produce outputs in the watt to several watt range are called power amplifiers or high-power amplifiers. The bulk of integrated circuit (IC) amplifiers are used in low-power applications although pulse-width modulation (PWM) IC amplifier stages can deliver several watts of power to a load.

15.2.4 Distortion in Power Stages

There are two major causes of nonlinear distortion in BJT stages [3]. The first is the change of current gain β with changes in I_C or V_{CE}. If an undistorted base current enters the device, the output current is distorted as a result of the change in current gain as the output quantities vary. This effect can also be explained for a common-emitter stage in terms of the output characteristics which show unequal spacing as I_C or V_{CE} is changed. Obviously, if the output signal amplitude is limited, less distortion occurs. In power stages, very large output-current changes may be prevalent, and higher distortion levels are to be expected.

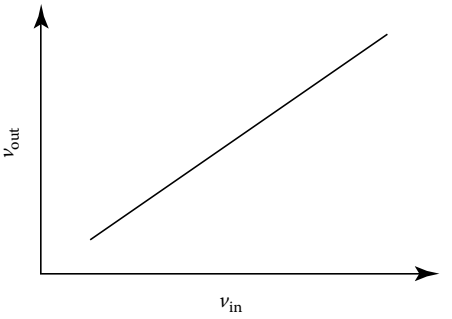


FIGURE 15.2 Transfer characteristics of a distortion-free amplifier.

The second important cause of distortion arises from the nonlinearity of the base–emitter characteristics of the transistor. When a voltage source drives a common-emitter stage with little series resistance, the base current can be quite distorted. This current varies with voltage in the same way that diode current varies with diode voltage. It is possible to use this input distortion to partially offset the output distortion; however, feedback techniques are most often used in distortion reduction. A perfect, distortion-free amplifier would have transfer characteristics that form a straight line as shown in Figure 15.2.

The BJT with no emitter degeneration has a collector current that relates to base–emitter voltage given by

$$I_C = \beta K_1 e^{qV_{BE}/kT}$$

where k is Boltzmann's constant. This function is highly nonlinear, and only small variations of V_{BE} can be applied to approximate linear behavior. An emitter resistance or emitter degeneration can be added to make the circuit more linear at the expense of reduced gain. This amounts to feedback to improve the nonlinear distortion. Another method of reducing distortion is to use several cascaded stages to achieve a very high gain; then feedback is applied around the amplifier to decrease the distortion.

15.2.4.1 Total Harmonic Distortion

In an amplifier circuit, the output signal should ideally contain only those frequency components that appear in the input signal. The nonlinear nature of amplifying devices introduces extraneous frequencies in the output signal that are not contained in the input signal. These unwanted signals are referred to as harmonic distortion. One measure of the amplifier's performance is called the total harmonic distortion (THD).

If a sinusoidal signal is applied to the amplifier input, the output will also contain a major component of this signal at the fundamental frequency, with amplitude designated v_f. In addition, there will be smaller harmonic components in the output signal. These amplitude values will be designated v_2, v_3, $v_4, \ldots, v_n, \ldots$ Fortunately, in engineering applications, the harmonics decrease in amplitude as frequency increases. Thus, perhaps only the second or third harmonic is large enough to affect the THD. The THD can defined as a percentage by

$$\text{THD} = \frac{\sqrt{v_2^2 + v_3^2 + v_4^2 + \cdots}}{v_f} \times 100 \tag{15.1}$$

In dB, this parameter becomes

$$\text{THD} = 10 \log\left(\frac{v_2^2 + v_3^2 + v_4^2 + \cdots}{v_f^2}\right) \tag{15.2}$$

The THD of an amplifier is a major specification used to characterize the amplifier performance.

15.2.4.2 Intercept Points

The unwanted harmonic distortion of an amplifier is a function of output signal size. As the signal becomes larger, the linear output will increase, and so will the distortion. In fact, unwanted components due to harmonic distortion may increase more rapidly than the linear component.

Typically, an amplifier with a small output signal, compared to the size of the active region, will have a large linear output component or fundamental component relative to the harmonic components. As signal size increases, the size of the second or third harmonic may become as large as the fundamental component. Generally, this will not occur before the output signal reaches the extremes of the active region; consequently it would be impossible to measure the size of the output signal that results in a harmonic component that is equal in amplitude to the fundamental component.

The concept of the intercept point allows this output to be approximated. A plot, often in dB, is made of the input power versus the output power of the fundamental component. This plot is a straight line with a slope of unity for smaller output signals. As the output approaches the edges of the active region, the output power increases much more slowly than does the input power. Finally, the output power is limited to some fixed value as the output signal size is limited by the active region boundaries. Figure 15.3 shows this variation.

For small signals, each harmonic component is small. The third harmonic is plotted in Figure 15.3 in addition to the fundamental component. As input power is increased, the third harmonic component may increase at three times the rate of increase in the fundamental component. Before the input power is high enough to significantly limit the fundamental power, a straight-line extension is added to the fundamental power plot and the third harmonic power plot. The point where these two extensions intersect is called the third-order intercept point (TOI). Note that this point often falls beyond the actual output power that can be achieved by the amplifier. The TOI is specified in terms of the input power required at this point. For example, in Figure 15.3, the TOI is −10 dB. The higher the intercept point, the more linear is the amplifier.

A second-order intercept can also be defined in much the same way as the TOI is defined. For communication amplifiers, the TOI is generally more important than the second-order intercept because the third-order term can result in a frequency that falls within the desired passband of the amplifier. The TOI is an oft-used specification for power stages as well as high-frequency amplifiers.

15.2.5 Operational Amplifiers

The op-amp was developed long before the IC was invented. It was used in analog computers that were designed to solve differential equations by simulation. This amplifier formed the basis of circuits that

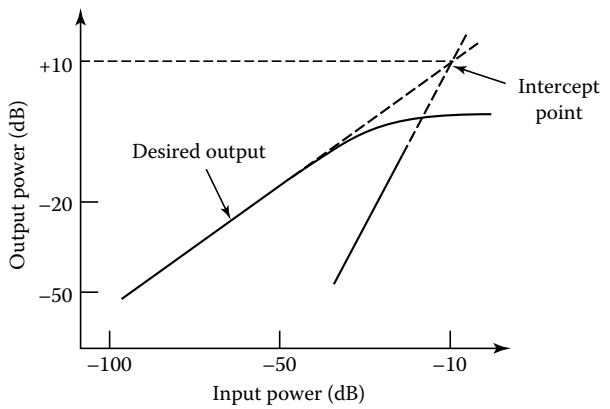


FIGURE 15.3 Amplifier output power versus input power.

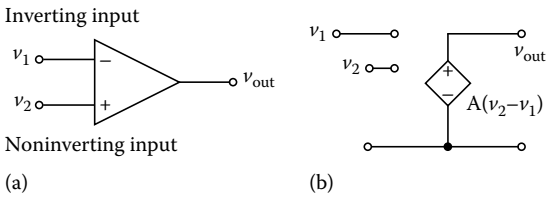


FIGURE 15.4 (a) Symbol for the op-amp. (b) Ideal equivalent circuit.

performed mathematical operations such as weighting, summing, subtracting, and integrating electrical signals. The op-amp has two features that allow these operations to be accomplished. The first feature is a differential input to allow subtraction. The second feature is that a virtual ground can be created when a feedback resistor is connected between the amplifier output and the inverting or negative input terminal. This virtual ground allows perfect summation of currents into the inverting terminal and also allows perfect integration of an input signal.

Before the IC op-amp was developed, the discrete circuit op-amp was quite expensive and large, perhaps costing $200 and occupying a volume of 250 cm^3. With its low cost and small size, the IC op-amp has now made a rarely used component one of the most popular IC chips in the electronics field. The major limitation of the op-amp is its frequency response that limits its use in very high frequency applications.

The near-ideal op-amp has a very high gain, a high input impedance, and a low output impedance. The symbol for this device is shown in Figure 15.4a along with the ideal equivalent circuit of Figure 15.4b.

15.3 Differences between Discrete and IC Amplifier Design

Prior to the early 1960s, electronic circuits were constructed from discrete component circuits. Discrete resistors, capacitors, inductors, and transistors are packaged individually to be connected with wires or printed circuit board (PCB) conductors to create a functioning electronic circuit. The discrete component circuit is still required in certain applications in the twenty-first century, but the IC is now the dominant form of implementation for an electronic system. There are major differences in the design principles used for the IC and the discrete circuit as the following section will explain [4].

15.3.1 Component Differences

Discrete resistors, capacitors, and inductors are available in a very large range of values. Although many amplifier circuits are designed with element values of 5%–10% tolerances, very precise values can be obtained at a higher cost. The range of resistor values extends from a few ohms to many megohms and capacitor values range from femtofarads to hundreds of microfarads. Metal core or air core inductors are available over a wide range of values as also are transformers with a wide choice of turns ratios.

For most standard fabrication processes, the IC chip becomes too large to be useful when the total resistance of resistors on the chip exceed some value such as 100 kΩ. The same applies if the total capacitance exceeds perhaps 50 to 100 pF. It is only in the last decade that inductors could be fabricated on a chip. Even now, integrated inductors with limited values in the nanohertz range are possible and losses may limit Q-values to the range of 2–8 [5].

A major problem with IC components is the lack of precise control of values. Resistors or capacitors are typically fabricated with an absolute accuracy of around 20%. It is possible, however, to create resistive ratios or capacitive ratios that approach a 0.1% accuracy, even though absolute value control is very poor in standard processes.

The cost of a discrete circuit is determined by different variables than that of the IC. This results in a design philosophy for ICs that differs from that of discrete component circuits. Before ICs were available, a designer attempted to minimize the number of transistors in a circuit. In fact, some companies

estimated the overall component cost of a circuit by multiplying the number of transistors in the circuit by some predetermined cost per stage. This cost per stage was determined mainly by the transistor cost plus a small amount added to account for the passive component cost. The least expensive item in the discrete circuit is generally the resistor which may cost a few cents.

In an IC, the greatest cost is often associated with the component that requires the most space. The BJT typically requires much less space than resistors or capacitors and is the least expensive component on the chip. An IC designer's attempt to minimize space or simplify the fabrication process for a given circuit generally results in much larger numbers of transistors and smaller numbers of resistors and capacitors than the discrete circuit version would contain. For example, a discrete circuit bistable flip-flop of 1960 vintage contained two transistors, six resistors, and three capacitors. A corresponding IC design had 18 transistors, 2 resistors, and no capacitors.

Matching of components in IC design can be considerably better than discrete design, but also presents unique problems as well. Components that are made from identical photographic masks can vary in size and value because of uneven etching rates influenced by nearby structures or by asymmetrical processes. Dummy components must often be included to achieve similar physical environments for all matched components. Certain processes must be modified to result in symmetrical results. Metal conductors deposited on the IC chip can introduce parasitics or modify performance of the circuit if placed in certain areas of the chip, thus care must be taken in placing the metal conductors. Of course, these kinds of considerations are unnecessary in discrete circuit design.

While discrete circuits can use matched devices such as differential stages, often single-ended stages can be designed to meet relatively demanding specifications. Discrete components can be produced with very tight tolerances to lead to circuit performance that falls within the accuracy of the specifications. Circuit design can be based on the absolute accuracy of key components such as resistors or capacitors. Although it may add to the price of a finished circuit, simple component tuning methods can be incorporated into the production of critical discrete circuits.

Since the absolute values of resistors and capacitors created by standard processes for ICs cannot be determined with great accuracy, matching of components and devices is used to achieve acceptable performance. Differential stages are very common in IC design since matched transistors and components are easy to create even if absolute values cannot be controlled accurately.

Some analog circuits do not require high component densities while others may pack a great deal of circuitry into a small chip space. For low density circuits, more resistors and small capacitors may be used, but for high density circuits, these elements must be minimized. Section 15.4 considers circuits that replace resistors and capacitors with additional BJTs in IC amplifier building blocks. The BJT current mirror, which is quite popular in biasing of IC amplifiers is discussed in [4]. This circuit is also used to replace the load resistance of a conventional amplifier to make it easier to integrate on a chip.

15.3.2 Parasitic Differences

One advantage of an IC is the smaller internal capacitance parasitics of each device. A proper layout of an IC leads to small parasitic device capacitances that can be much less than corresponding values for discrete devices. Although the discrete and IC device may be fabricated with similar processes, the collector and emitter of the IC device are typically much smaller and usually connect to fine pitch external leads/connectors. The discrete device must make external connections to the base, collector, and emitter and these external connections can add a few picofarads of capacitance. An on-chip IC device generally connects to another device adding minimal parasitic capacitance. However, external connects from the chip to the package must be carefully designed to minimize parasitic capacitance as signals are brought off chip.

Another difference relates to the parasitic capacitance associated with IC resistors and inductors. These elements have capacitance as a result of the oxide dielectric between metal layers and the chip substrate. The frequency response of IC resistors and inductors must be considered in critical designs.

At higher frequencies (above approximately 1 GHz) other considerations influence the IC design. In order to maintain the advantage of low cost in IC circuits over discrete realizations, the devices on the chip are made much smaller than their discrete counterparts. Current and power levels must be limited in the smaller devices to avoid gain roll-off due to high-level injection effects or overheating. The lower currents necessitate higher impedance levels in most bipolar chip designs.

For many analog circuits, terminations of 50 Ω are used at I/O ports. This requires large current magnitudes to reach the specified voltage levels. For example, a 20 mA peak current would be required to develop a peak voltage of 1 V. This is not a difficult problem in discrete circuit design using PC boards. The limitation on current and power in the chip design leads to less flexibility with higher impedance interconnects to develop acceptable voltages. High frequency chip designs must also deal with bond wire or other package inductance in the range of 0.1–0.5 nH and the package capacitance that may reach 5 pF. A buffer is often needed to generate the off-chip signals with sufficient current to develop the specified voltages, but IC chip heating due to power dissipation of the buffer stage must be carefully considered. Common thermal centroid techniques are used to prevent unbalanced heating of matched devices.

15.4 Building Blocks for IC Amplifiers

This section discusses several single-stage amplifying circuits that may be combined with other stages to create a high-performance amplifier.

15.4.1 Small-Signal Models

Most amplifier circuits constrain the BJT stages to active region operation. In such cases, the models can be assumed to have constant elements that do not change with signal swing. These models are linear models. The quiescent or bias currents are first found using a nonlinear model to allow the calculation of current-dependent element values for the small-signal model. For example, the dynamic resistance of the base–emitter diode, r_e, is given by

$$r_e = \frac{kT}{qI_E} \tag{15.3}$$

where I_E is the dc emitter current. Once I_E is found from a nonlinear dc equivalent circuit, the value of r_e is calculated and assumed to remain constant as the input signal is amplified.

The equivalent circuit [6] of Figure 15.5 shows the small-signal hybrid-π model that is closely related to the Gummel-Poon model and is used for ac analysis in the Spice program.

The capacitance, C_π, accounts for the diffusion capacitance and the emitter–base junction capacitance. The collector–base junction capacitance is designated C_μ. The resistance, r_π, is equal to $(\beta + 1)r_e$. The transconductance, g_m, is given by

$$g_m = \frac{\alpha}{r_d} \tag{15.4}$$

The impedance, r_o, is related to V_A, the Early voltage, and I_C, collector current at $V_{CE} = 0$ by

$$r_o = \frac{V_A}{I_C} \tag{15.5}$$

R_B, R_E, and R_C are the base, emitter, and collector resistances, respectively.

In many high-frequency, discrete amplifier stages and in some high-frequency IC stages, a small value of load resistance will be used. Approximate hand analysis can then be done by neglecting

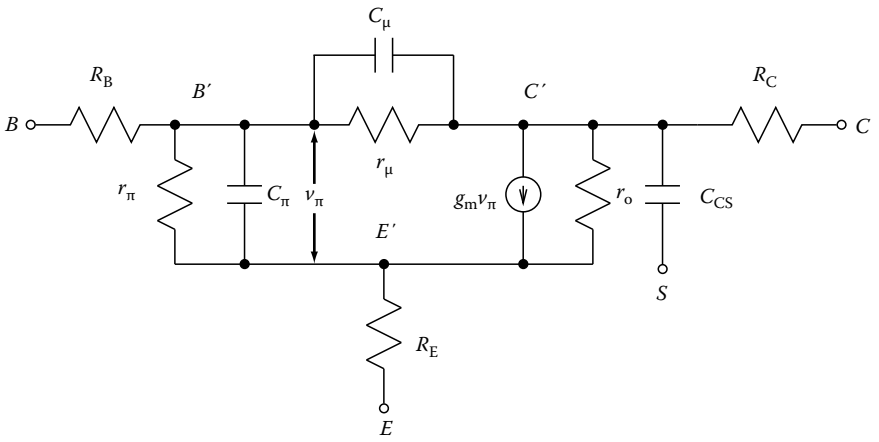


FIGURE 15.5 The hybrid-π small signal model for the BJT.

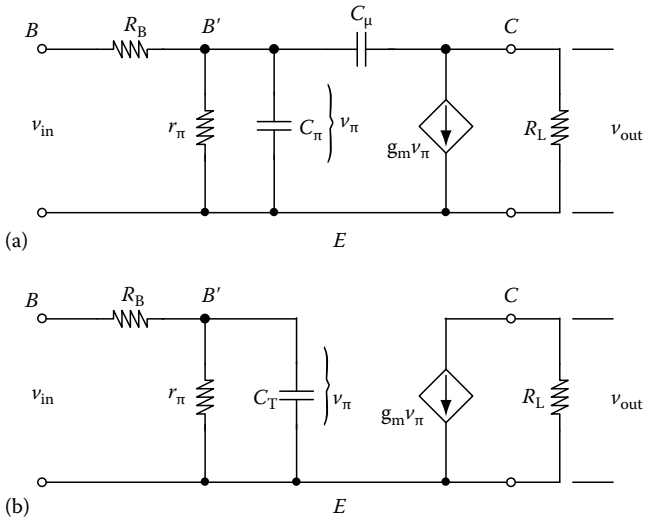


FIGURE 15.6 (a) Approximate equivalent circuit. (b) Unilateral equivalent.

the ohmic resistances, R_E and R_C along with C_{CS}, the collector to substrate capacitance. If the impedance r_o is much larger than the load resistance, R_L, of the high-frequency amplifying stage it can be considered an open circuit. The resulting model that allows relatively viable hand analysis is shown in Figure 15.6a.

For small values of load resistances, the effect of the capacitance can be represented by an input Miller capacitance that appears in parallel with C_π. This simplified circuit now includes two separate loops as shown in Figure 15.6b and is referred to as the unilateral equivalent circuit. Typically, the input loop limits the bandwidth of the amplifier stage. When driven with a signal generator having a source resistance of R_s, the upper 3-dB frequency can be approximated as

$$f_{high} = \frac{1}{2\pi R_{eq}(C_\pi + C_M)} \tag{15.6}$$

In this equation, the values of R_{eq} and C_M are

$$R_{eq} = r_\pi \parallel (r_b + R_s) \tag{15.7}$$

and

$$C_M = C_\mu(1 + |A|) = C_\mu(1 + g_m R_L) \tag{15.8}$$

where A is the gain of the stage from point B′ to the collector of the stage. Although C_μ is small, the resulting Miller effect increases this value to the point that it can dominate the expression for upper 3-dB frequency. In any case, it often has a major effect on frequency performance.

When higher impedance loads are used in a BJT as is often the case for IC amplifier design, the Miller effect approach becomes less accurate and the performance must be simulated to achieve more accurate results. However, the unilateral circuit that includes C_μ and C_{CS} from the collector to ground after reflecting the effects of C_μ to the input loop can be used to approximate the amplifier stage performance. Manually calculated results are generally within 5%–10% of simulation results.

15.4.2 Active Loads

In order to achieve high voltage gains and eliminate load resistors, active loads are used in BJT IC amplifier stages [4]. In a conventional common-emitter stage, the gain is limited by the size of the collector resistance. The midband voltage gain from base to collector of a common-emitter stage is given by

$$A_v = -\frac{\alpha R_C}{(r_e + R_E)}$$

It would be possible to increase this voltage gain by increasing R_C, however, making R_C large can lead to some serious problems. A large collector load requires a low quiescent collector current to result in proper bias. This may lead to lower values of β, since current gain in a silicon transistor typically falls at low levels of emitter current. In order to achieve a voltage gain of 1000 V/V, a collector load of perhaps 100–200 kΩ might be required. The low collector current needed for proper bias, a few microamps, would lead to a low value of β and a very high value of r_e. The desired high voltage gain may not be achievable under these conditions.

It would be desirable if the collector load presented a low resistance to dc signals, but presented a high incremental resistance. This combination of impedances would result in a stable operating point along with a high gain. A perfect current source with infinite incremental resistance along with a finite dc current flow satisfies the requirements, but is not a practical solution. On the other hand, circuits that approximate current sources are relatively easy to construct. A good approximation to the current source is obtained by using another transistor for the collector load of an amplifying transistor. Not only can this device present a low dc and high incremental impedance, it is a simple element to implement on a chip. This transistor that replaces the resistive load is referred to as an active load.

The circuit of Figure 15.7 demonstrates one type of BJT active load. The transistor $Q1$ is the amplifying element with $Q2$ acting as the load. Transistor $Q1$ looks into the collector of $Q2$. The incremental output impedance at the collector of a transistor having an emitter resistance in the low kiloohm range can easily exceed 500 kΩ. With such a high impedance, $Q2$ approximates a current source.

The dc collector currents of both transistors are equal in magnitude. This magnitude can be set to a value that leads to a reasonable value of β. Since $Q2$ has a very high output impedance, the midband

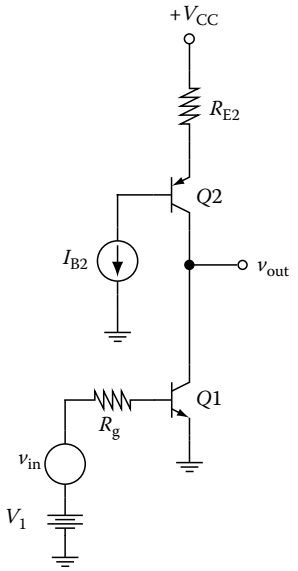


FIGURE 15.7 Active load amplifier.

voltage gain will be determined primarily by the collector-to-emitter resistance of $Q1$and can be calculated from

$$A_{MB} = \frac{-\beta_1 r_{ce1}}{R_g + r_{\pi 1}} \tag{15.9}$$

where r_{ce1} is the output impedance of $Q1$. If the generator resistance, R_g is negligible, this equation reduces to

$$A_{MB} = -\frac{r_{ce1}}{r_{e1}} = -\frac{(V_A + V_{CQ1})/I_C}{V_T/I_E} \approx -\frac{V_A + V_{CQ1}}{V_T} \tag{15.10}$$

For an Early voltage of $V_A = 80$ V and $V_T = 0.026$ V, a small-signal voltage gain exceeding -3000 V/V could result. In a normal application, this stage would drive a second stage. The input impedance of the second stage will load the output impedance of the first stage, further lowering the gain. Depending on the input impedance of the second stage and the impedance of the active load stage, the gain magnitude may still exceed 1000 V/V.

The concept of an active load that presents a large incremental resistance while allowing a large dc quiescent current is important in IC design. In addition to the current source load just considered, the current mirror stage can also be used to provide the active load of a differential stage as discussed in Section 15.4.3.

15.4.3 Common Emitter Stage

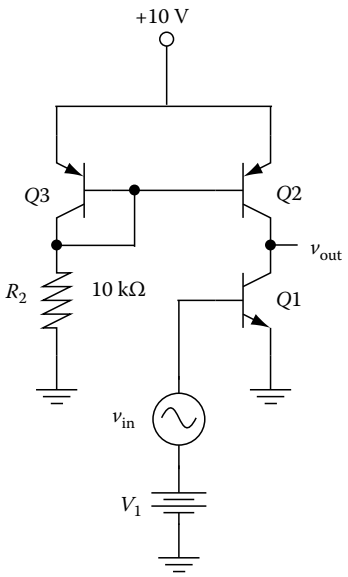


FIGURE 15.8 Common-emitter amplifier with current mirror active load.

A simple configuration for an IC amplifying stage is shown in Figure 15.8. In this stage, the output impedance of the current mirror is not large enough to be considered an open circuit as it was in the circuit of Figure 15.7. Thus, the analysis will have to account for this element.

The high frequency response of discrete BJT stages is often determined by the input circuit, including the Miller effect capacitance. The collector load resistance in a discrete stage is usually small enough that the output circuit does not affect the upper corner frequency. In the circuit of Figure 15.8, as in many IC amplifier stages, the output impedance is very high compared to the discrete stage. For this circuit, the output impedance of the amplifier consists of the output impedance of $Q2$ in parallel with that of $Q1$. This value will generally be several tens of kiloohms.

The equivalent circuit of the amplifier of Figure 15.8 is indicated in Figure 15.9. The value of R_{out} is

$$R_{out} = r_{o1} \,\|r_{o2} = r_{ce1}\| \, r_{ce2} \tag{15.11}$$

The capacitance in parallel with R_{out} is approximately

$$C_{out} = C_{\mu 1} + C_{\mu 2} + C_{cs1} + C_{cs2} \tag{15.12}$$

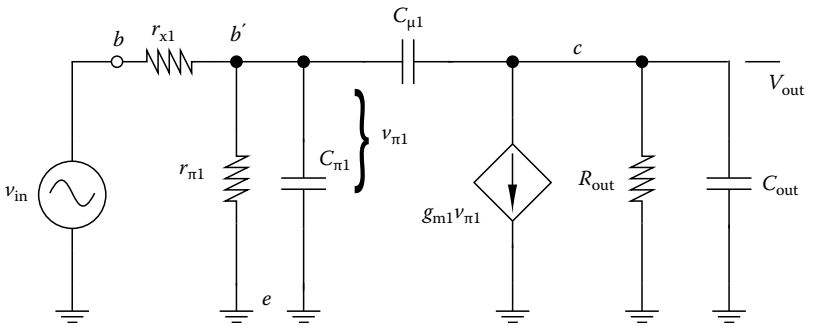


FIGURE 15.9 Equivalent circuit for the amplifier of Figure 15.8.

In this equation, $C_{\mu 1}$ and $C_{\mu 2}$ are the collector-to-base junction capacitances and C_{cs1} and C_{cs2} are the collector-to-substrate capacitances of the respective transistors. If no generator resistance is present, $C_{\mu 1}$ will also appear in parallel with the output terminal and ground. When R_g is present, we will still approximate the output capacitance with the same equation, although feedback effects between the output and the bases of $Q1$and $Q2$ actually modify the value slightly.

The midband gain is easy to evaluate as

$$A_{MB} = \frac{-\beta_1 R_{out}}{R_g + r_{x1} + r_{\pi 1}} \tag{15.13}$$

The upper corner frequency is now more difficult to evaluate than that of the discrete circuit with its low value of collector load resistance. In the discrete circuit, the input loop generally determines the overall upper corner frequency of the circuit. Although the Miller effect will be much larger in the IC stage, lowering the upper corner frequency of the input loop, the corner frequency of the output loop will also be smaller due to the large value of R_{out}. Both frequencies may influence the overall upper corner frequency of the amplifier.

As mentioned previously, it is difficult to manually calculate the upper corner frequency of the stage and an accurate value is generally found by simulation. An approximation of the upper corner frequency can be manually found by reflecting the bridging capacitance, C_μ, to both the input and the output. The value reflected to the input side, across terminals b' and e, is

$$(1 + |A_{b'c1}|)C_{\mu 1} \tag{15.14}$$

as in the discrete circuit amplifier. Thus, the total input capacitance in parallel with $r_{\pi 1}$ is

$$C_{in} = C_{b'e1} + (1 + |A_{b'c1}|)C_{\mu 1} \tag{15.15}$$

The major component of $C_{b'e1}$ is the diffusion capacitance, C_π, of $Q1$.

The upper corner frequency resulting from the input circuit of this stage is

$$f_{in\text{-}high} = \frac{1}{2\pi C_{in} R_{eq}} \tag{15.16}$$

where $R_{eq} = (R_g + r_{x1}) \| r_{\pi 1}$.

The upper corner frequency resulting from the output side of the stage is

$$f_{out\text{-}high} = \frac{1}{2\pi C_{out} R_{out}} \tag{15.17}$$

The approximate overall upper corner frequency, f_{2o}, must be found by considering a two-pole response. The overall response is expressed as

$$A(\omega) = \frac{A_{MB}}{\left(1 + jf/f_{\text{in-high}}\right)\left(1 + ff/f_{\text{out-high}}\right)} \tag{15.18}$$

It is easy to show that the overall upper corner frequency can be found by solving the equation

$$\left(\frac{f^2}{f_{\text{in-high}}^2 + 1}\right)\left(\frac{f^2}{f_{\text{out-high}}^2 + 1}\right) = 2 \tag{15.19}$$

Although this method is not as accurate as the simulation, results will often be within 5%–10%.

15.4.4 Common-Base Stage

The common-base stage shown in Figure 15.10 has an advantage and a disadvantage when compared to the common-emitter stage. The advantage is that the Miller effect, that is, the multiplication of apparent capacitance at the input, is essentially eliminated. The noninverting nature of the gain does not lead to an increased input capacitance. Furthermore, the capacitance between input (emitter) and output (collector) is generally negligible. The upper corner frequency is then higher than that of a comparable common-emitter stage.

The disadvantage is the low current and power gain of the common-base stage compared to the common-emitter stage. The low-frequency current gain for the common-base stage equals α and is slightly less than unity. In the common-emitter stage, the current gain equals β and may be over 200. Since voltage gain is similar for the two stages, the power gain is also much lower for the common-base stage. The input resistance at low frequencies is also much lower than that of the common-emitter stage and can load the previous stage.

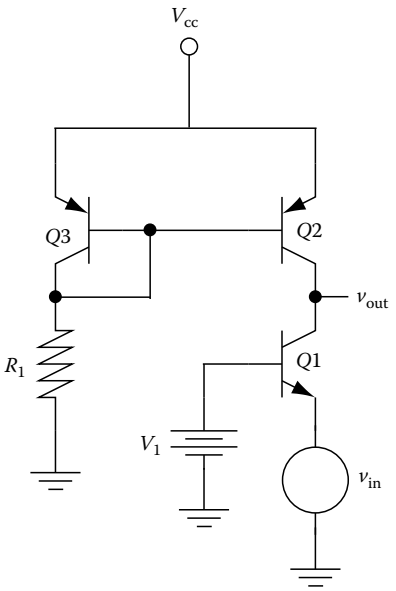


FIGURE 15.10 Common-base amplifier.

The equation for voltage gain is

$$A_{MB} = \frac{\alpha_1 R_{out}}{\frac{r_{x1}}{\beta+1} + r_{e1} + R_g} \tag{15.20}$$

where R_{out} is the parallel combination of r_{out1} and r_{ce2}. The output impedance of $Q1$ depends on the generator resistance, R_g, and ranges from r_{ce1} for $R_g = 0$ up to βr_{ce1} when R_g approaches infinity. The midband voltage gain is similar to that of the common-emitter stage.

The expression for voltage gain as a function of frequency is

$$A = \frac{A_{MB}}{\left(1 + j\dfrac{f}{f_{\text{in-high}}}\right)\left(1 + j\dfrac{f}{f_{\text{out-high}}}\right)} \tag{15.21}$$

where $f_{\text{in-high}}$ is the corner frequency of the input circuit and $f_{\text{out-high}}$ is the corner frequency of the output circuit. These values are

$$f_{\text{in-high}} = \frac{1}{2\pi C_{\pi 1}\left(r_{e1} \| R_g\right)} \tag{15.22}$$

$$f_{\text{out-high}} = \frac{1}{2\pi C_{\text{out}} r_{\text{out}}} \tag{15.23}$$

The output capacitance consists of the collector to base capacitances and the collector to substrate capacitances of both $Q1$ and $Q2$, that is,

$$C_{\text{out}} = C_{\mu 1} + C_{\mu 2} + C_{\text{cs1}} + C_{\text{cs2}} \tag{15.24}$$

Generally, the corner frequency of the input circuit is considerably higher than the corner frequency of the output circuit for high-gain IC stages and the overall upper corner frequency is approximated by $f_{\text{out-high}}$.

The midband voltage gain of the common-base and common-emitter stages decreases with increasing generator resistance. In multistage amplifiers, the generator resistance of one stage is the output resistance of the previous stage. If this value is large, the gain of the following stage may be small. Furthermore, the upper corner frequency of the common-emitter stage is affected by the generator resistance. Large values of R_{g} can lead to small values of upper corner frequency.

15.4.5 Emitter Follower

A stage that can be used to minimize the adverse effect on frequency response caused by a generator resistance is the emitter follower. Although this stage has a voltage gain near unity, it can be driven by a higher voltage gain stage while the emitter follower can drive a low impedance load. A typical stage is shown in Figure 15.11.

The output stage of the *npn* current mirror, $Q2$, serves as a high impedance load for the emitter follower, $Q1$. An equivalent circuit that represents the emitter follower of Figure 15.11 is indicated in Figure 15.12. For this circuit, $g_{\text{m1}} = \alpha_1 / r_{\text{e}} \approx 1/r_{\text{e}}$.

This circuit can be analyzed to result in a voltage gain of

$$A = \frac{\frac{C_{\pi 1}}{\Pi C\left(R_{\text{g}} + r_{\text{x1}}\right)}\left(j\omega + \frac{g_{\text{m1}} r_{\pi 1} + 1}{r_{\pi 1} C_{\pi 1}}\right)}{-\omega^2 + bj\omega + d} \tag{15.25}$$

where

$$\Pi C = C_{\text{out2}} C_{\mu 1} + C_{\text{out2}} C_{\pi 1} + C_{\pi 1} C_{\mu 1} \tag{15.26}$$

$$b = \frac{1}{\Pi C}\left(\frac{C_{\pi 1} + C_{\text{out2}}}{R_{\text{g}} + r_{\text{x1}}} + \frac{C_{\pi 1} + C_{\mu 1}\left(1 + g_{\text{m1}} r_{\text{ce2}}\right)}{r_{\text{ce2}}} + \frac{C_{\text{out2}} + C_{\mu 1}}{r_{\pi 1}}\right) \tag{15.27}$$

and

$$d = \frac{R_{\text{g}} + r_{\text{x1}} + r_{\pi 1} + r_{\text{ce2}}\left(g_{\text{m1}} r_{\pi 1} + 1\right)}{\Pi C\left(R_{\text{g}} + r_{\text{x1}}\right) r_{\pi 1} r_{\text{ce2}}} \tag{15.28}$$

Note that the output capacitance of $Q2$ can be approximated as the sum of $C_{\mu 2}$ and C_{cs2}.

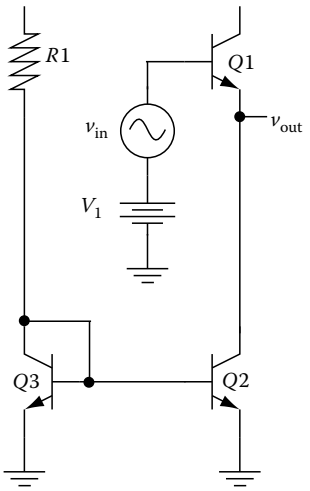


FIGURE 15.11 Emitter follower.

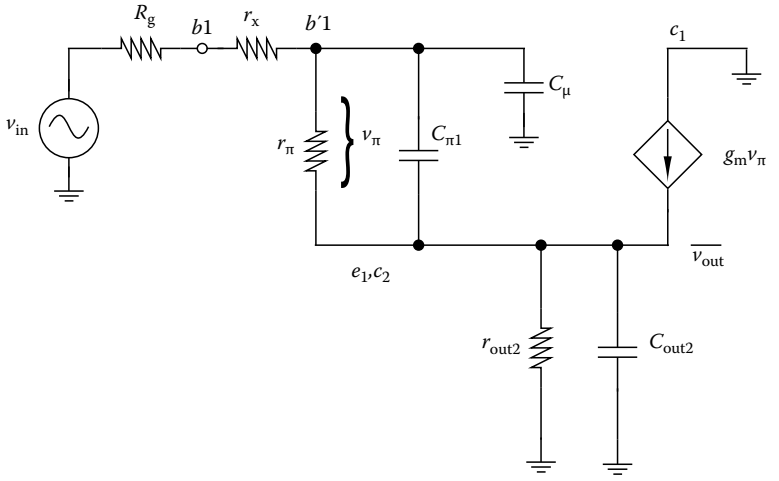


FIGURE 15.12 Equivalent circuit for the emitter follower.

The midband voltage gain is found from Equation 15.25 by letting $\omega \to 0$. This value is

$$A_{MB} = \frac{(1 + g_{m1} r_{\pi 1}) r_{ce2}}{R_g + r_{x1} + r_{\pi 1} + (1 + g_{m1} r_{\pi 1}) r_{ce2}} \tag{15.29}$$

This gain is very near unity for typical element values.

The bandwidth is more difficult to calculate since the response has one zero and two poles. The zero for the circuit of Figure 15.12 is typically larger than the lowest frequency pole. If these frequencies canceled, the larger pole would determine the corner frequency. Since they do not cancel, the overall upper corner frequency is expected to be smaller than the larger pole frequency. An accurate calculation can be made from Equation 15.25 when the parameters are known.

In high-frequency design, it must be recognized that the output impedance of the emitter follower can become inductive [7].

15.5 Compound Stage Building Blocks for IC Amplifiers

Several of the single stages of Section 15.4 can be combined to construct high-performance two-stage building blocks.

15.5.1 Cascode Amplifier Stage

One of the problems with the common-emitter stage using an active load is the Miller effect. This stage has a high voltage gain from base to collector. The circuit of Figure 15.8 has an inverting voltage gain, A_{MB}, that can be quite high, perhaps -700 V/V. The base-collector junction capacitance is multiplied by $(1 + |A_{MB}|)$ and reflected to the input loop. This capacitance adds to the diffusion capacitance from point b' to point e and can lower the upper corner frequency to a relatively small value.

On the other hand, the common-base stage of Figure 15.10 has essentially no Miller effect, but has a low input impedance and can load the impedance of the driving source. The cascade stage combines the best features of both the common-emitter and common-base stages. The cascode amplifier stage of Figure 15.13 minimizes the capacitance reflected to the input. The input impedance to the circuit is that

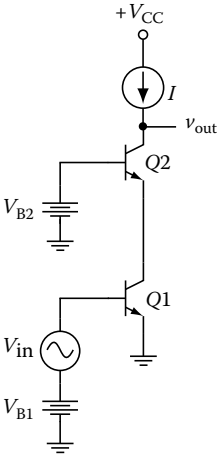


FIGURE 15.13 Cascode amplifier.

of the common-emitter stage and is two orders of magnitude higher than the common-base stage. In this circuit, the input capacitance is primarily composed of the diffusion capacitance of *Q*1. The voltage gain from base-to-collector of *Q*1 is quite low since the collector load of this device consists of the impedance looking into the emitter of *Q*2. This impedance is approximately equal to the base-emitter diode resistance of *Q*2 which is

$$r_{e2} = \frac{26}{1_{E2}}$$

With low voltage gain, the Miller effect of the first stage is minimized. The upper device passes the incremental signal current of *Q*1 to its collector and develops a large voltage across the current source impedance. There is no Miller multiplication of capacitance from the input of *Q*2 (emitter) to the output (collector) since the gain is noninverting and negligible capacitance exists between emitter and collector. Thus, the cascode stage essentially eliminates Miller effect capacitance and its resulting effect on upper corner frequency.

A high-frequency equivalent circuit of this stage is shown in Figure 15.14. The resistance R includes any generator resistance and the base resistance, r_{x1} of *Q*1. The resistance r_{cs} is the output resistance of the current source. The output capacitance is the sum of $C_{\mu 2}$, C_{cs2}, and any capacitance at the current source output. The resistance r_{out2} can be quite large since *Q*2 sees a large emitter resistance looking into the collector of *Q*1. This emitter load leads to negative feedback that increases the output resistance of *Q*2.

The midband voltage gain is calculated from the equivalent circuit of Figure 15.14 after eliminating the capacitors. This gain is found rather easily by noting that the input current to *Q*1is

$$i_{b1} = \frac{v_{in}}{R + r_{\pi 1}} \tag{15.30}$$

This current will be multiplied by β_1 to become collector current in *Q*1. This current also equals the emitter current of *Q*2. The emitter current of *Q*2 is multiplied by α_2 to become collector current of *Q*2. The output voltage is then

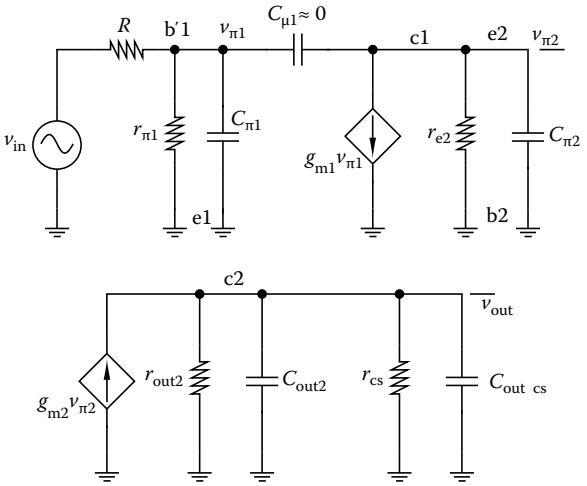


FIGURE 15.14 Equivalent circuit for the cascode amplifier.

$$v_{out} = i_{c2} \times R_3 \tag{15.31}$$

where $R_3 = r_{out2} || r_{cs}$. This resistance could be very large if the current source resistance, r_{cs}, is large. The value of r_{out2} will be high since the emitter of $Q2$ sees a resistance of r_{ce1}. Combining this information results in a midband voltage gain of

$$A_{MB} = \frac{-\beta_1 \alpha_2 R_3}{R + r_{\pi 1}} \tag{15.32}$$

If no generator resistance is present and if $R_3 = 100$ kΩ, this midband voltage gain might exceed 5000 V/V.

The gain as a function of frequency can be found as

$$A = A_{MB} \frac{1}{[1 + j\omega C_{\pi 1}(r_{\pi 1} \parallel R)]} \frac{1}{[1 + j\omega r_{e2} C_{\pi 2}]} \frac{1}{[1 + j\omega R_3 C_{out}]} \tag{15.33}$$

Typically, the corner frequency of the second frequency term in Equation 15.33, that is,

$$f_2 = \frac{1}{2\pi r_{e2} C_{\pi 2}}$$

is much higher than that of the first term,

$$f_{in\text{-}high} = \frac{1}{2\pi (r_{\pi 1} \parallel R) C_{\pi 1}}$$

This is especially true if R is large compared to $r_{\pi 1}$. In this case, since $r_{e1} \approx r_{e2}$ as a result of equal emitter currents, then $r_{e2} \ll (\beta + 1) r_{e1}$. For hand analysis of the cascode circuit, the second term in the expression for gain is often neglected.

The gain can then be written as

$$A = A_{MB} \frac{1}{1 + j\frac{f}{f_{in-high}}} \frac{1}{1 + j\frac{f}{f_{out-high}}} \tag{15.34}$$

where

$$f_{out\text{-}high} = \frac{1}{2\pi C_{out} R_3} \tag{15.35}$$

and $f_{in\text{-}high}$ was defined previously.

The capacitance C_{out} is the sum of the current source output capacitance and the output capacitance of $Q2$ giving

$$C_{out} = C_{out2} + C_{outcs}$$

If a current mirror with output stage $Q3$ generates the collector bias current for $Q1$ and $Q2$, the output capacitance is

$$C_{out} = C_{\mu 2} + C_{\mu 3} + C_{cs2} + C_{cs3}$$

15.5.2 Differential Stage

The differential pair or differential stage is very important in the electronic field. Virtually every op-amp chip includes a differential pair as the input stage. Some advantages of the differential stage are its relative immunity to temperature effects and power supply voltage changes, and its ability to amplify dc signals. The differential amplifier uses a pair of identical stages connected in a configuration that allows the temperature drift of one stage to cancel that of the other stage. The basic configuration of a differential stage is shown in Figure 15.15. The two devices can be connected and operated in several different configurations. The mode of operation most often used with the differential amplifier is the differential input-double-ended output mode. Differential input refers to a situation wherein the voltage appearing across the input of one stage is equal in magnitude, but opposite in polarity to the voltage appearing across the input of the other stage. Double-ended output refers to the fact that the output voltage is taken as the difference in voltage between the output voltage of each stage. In single-stage amplifiers, the output voltage appears between the circuit output and ground. This is called a single-ended output. In Figure 15.15, the double-ended output voltage is

$$v_{\text{out}} = v_{\text{o1}} - v_{\text{o2}}$$

or

$$v_{\text{out}} = v_{\text{o2}} - v_{\text{o1}}$$

depending on the choice of output terminals. If this differential pair used a single-ended output, it could be taken between the output of stage 1 and ground or the output of stage 2 and ground.

A second input mode that could be used is the common mode. If the same signal is applied to both inputs, the circuit is said to operate in common mode. If the amplifier stages have exactly equal gains, the signals v_{o1} and v_{o2} will be equal. The double-ended output signal would then be zero in the ideal differential stage operating in common mode. In practice the two amplifier gains will not be identical; thus, the common-mode output signal will have a small value. The common-mode gain A_{CM} is defined as

$$A_{\text{CM}} = \frac{v_{\text{o1}} - v_{\text{o2}}}{v_{\text{CM}}} = A_1 - A_2 \tag{15.36}$$

where v_{CM} is the common-mode voltage applied to both inputs and $A1$ and $A2$ are the voltage gains of the two stages.

The differential gain applies when the input signals v_1 and v_2 are not equal and the gain in this case is

$$A_{\text{D}} = \frac{v_{\text{o1}} - v_{\text{o2}}}{v_1 - v_2} = \frac{A_1 v_1 - A_2 v_2}{v_1 - v_2} = \frac{v_{\text{out}}}{v_1 - v_2} \tag{15.37}$$

where A_{D} is the differential voltage gain of the amplifier with a double-ended output. If the double-ended output had been defined as $v_{\text{o2}} - v_{\text{o1}}$ rather than $v_{\text{o1}} - v_{\text{o2}}$, only the algebraic sign of A_{D} would change.

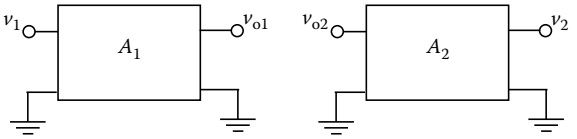


FIGURE 15.15 Differential pair.

In the general case, both common-mode and differential signals will be applied to the amplifier. This situation arises, for example, when the differential inputs are driven by preceding stages that have an output consisting of a dc bias voltage and an incremental signal voltage.

Using superposition, the output voltage for this case can be calculated from

$$v_{\text{out}} = A_{\text{D}}(v_1 - v_2) + A_{\text{CM}}v_{\text{CM}} \tag{15.38}$$

The common-mode input voltage can be found from

$$v_{\text{CM}} = \frac{v_1 + v_2}{2} \tag{15.39}$$

In the ideal situation with perfectly symmetrical stages, the common-mode input would lead to zero output, thus a measure of the asymmetry of the differential pair is the common-mode rejection ratio defined as

$$\text{CMRR} = 20\log\frac{|A_{\text{D}}|}{|A_{\text{CM}}|}\text{dB} \tag{15.40}$$

If $|A_{\text{D}}| = 100$ V/V and $|A_{\text{CM}}| = 0.01$ V/V, this value would be

$$\text{CMRR} = 20\log\frac{100}{0.01} = 80 \text{ dB}$$

The larger the CMRR, the smaller is the effect of A_{CM} on the output voltage compared to A_{D}.

A big advantage of the differential stage is in the cancellation of drift at the double-ended output. Temperature drifts in each stage are often common-mode signals. For example, the change in forward voltage across the base-emitter junction with constant current is about -2 mV/°C. As the temperature changes, each junction voltage changes by the same amount. These changes can be represented as equal voltage signals applied to the two inputs. If the stages are closely matched, very little output drift will be noted. The integrated differential amplifier can perform considerably better than its discrete counterpart since component matching is more accurate and a relatively uniform temperature prevails throughout the chip. Power supply noise is also a common-mode signal and has little effect on the output signal if the common-mode gain is low.

As previously mentioned, the mode of operation most often used with the differential amplifier is the differential input, double-ended output mode. In this configuration, the input voltage applied to one stage should be equal in magnitude, but opposite in polarity to the voltage applied to the other stage. One method of obtaining these equal magnitude, opposite polarity signals using only a single input source is shown in Figure 15.16.

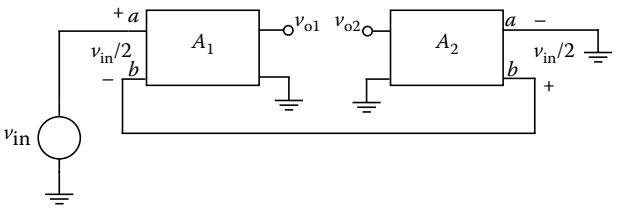


FIGURE 15.16 Differential input obtained from a single input source.

If the input resistances of both stages are equal, half of v_{in} will drop across each stage. While the voltage from terminal *a* to terminal *b* of stage 1 represents a voltage drop, the voltage across the corresponding terminals of stage 2 represents a rise in voltage. We can write

$$v_1 = \frac{v_{in}}{2}$$

and

$$v_2 = -\frac{v_{in}}{2}$$

Each stage has the same magnitude of input voltage, but is opposite in polarity to that of the other stage. From Equation 15.37 we can write the differential gain as

$$\begin{aligned} A_D &= \frac{v_{out}}{v_{in}} = \frac{v_{o2} - v_{o1}}{v_1 - v_2} = \frac{v_{o2} - v_{o1}}{v_{in}/2 - (-v_{in}/2)} \\ &= \frac{A_1 \frac{v_{in}}{2} + A_2 \frac{v_{in}}{2}}{\frac{v_{in}}{2} + \frac{v_{in}}{2}} = \frac{A_1 + A_2}{2} \end{aligned} \tag{15.41}$$

15.5.3 BJT Differential Pair

A BJT differential pair is shown in Figure 15.17.

15.5.3.1 Small-Signal Voltage Gain

Figure 15.18 represents a simple equivalent circuit for the BJT differential pair. The steps in calculating the single-ended or double-ended voltage gain of a differential pair are

1. Calculate the input current as $i_{in} = \frac{v_{in}}{R_{in}}$.
2. Note that the base currents are related to i_{in} by $i_{b1} = i_{in}$ and $i_{b2} = -i_{in}$.
3. Calculate collector currents as $i_{c1} = \beta_1 i_{b1}$ and $i_{c2} = \beta_2 i_{b2}$.
4. Calculate collector voltages from $v_{o1} = -i_c R_{Ceq1}$ and $v_{o2} = -i_2 R_{Ceq2}$, where

$$R_{Ceq} = R_C \parallel r_{out} \tag{15.42}$$

With these values, the single-ended or double-ended voltage gain can be found. In the normal situation, we assume perfectly symmetrical pairs with $\beta_1 = \beta_2 = \beta$ and $R_{C1} = R_{C2} = R_C$. We also assume that the bias current I splits equally between the two stages giving $I_{E1} = I_{E2} = I/2$.

The single-ended gain of the first stage of Figure 15.17 is found to be

$$A_{S1} = \frac{v_{o1}}{v_{in}} = \frac{-\beta R_{Ceq}}{2(\beta + 1) r_e} \tag{15.43}$$

The single-ended gain of the second stage equals this value in magnitude, but shows no phase inversion and therefore, has a positive algebraic sign.

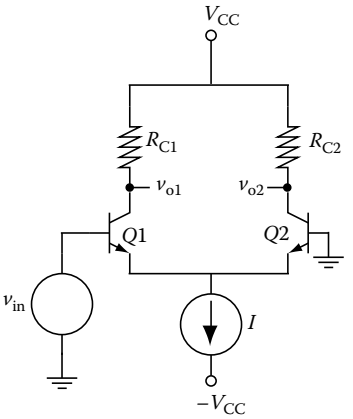


FIGURE 15.17 BJT differential pair.

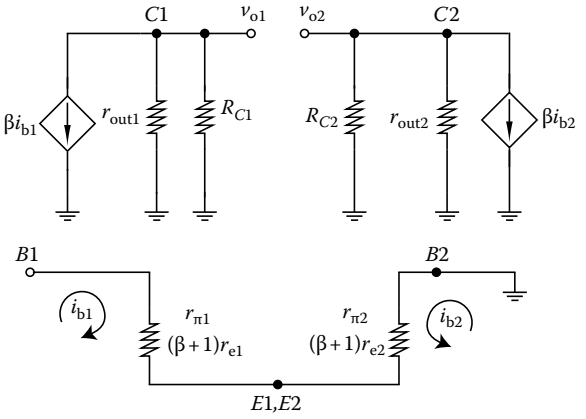


FIGURE 15.18 Equivalent circuit of the differential pair.

The double-ended differential gain is

$$A_D = \frac{v_{o2} - v_{o1}}{v_{in}} = \frac{\frac{\beta v_{in}}{2(\beta+1)r_e} R_{Ceq} - \left(\frac{-\beta v_{in}}{2(\beta+1)r_e} R_{Ceq}\right)}{v_{in}} = \frac{\beta R_{Ceq}}{(\beta+1)r_e} \tag{15.44}$$

This differential double-ended voltage gain is equal in magnitude to the voltage gain of a single transistor amplifier with a load of R_C and no external emitter resistance. The advantages of decreased temperature drift and good power supply rejection in critical applications, and a dc input reference of zero volts are typically far more important than the necessity of using an extra device, especially for IC chips.

The previous calculations for the differential pair assume there is no load across the output terminals. If this circuit drives a succeeding stage, the input impedance to this following stage will load the differential pair. One approach to this problem results from noting that when the collector of one BJT is driven positive by the input signal, the other collector moves an equal voltage in the opposite direction. Each end of the load resistor, R_L, is driven in equal but opposite directions. The midpoint of the resistor is always at 0 V, for incremental signals. To calculate the loaded voltage gain, a resistance of $R_L/2$ can be placed in parallel with each collector resistance to give an equivalent collector resistance of

$$R_{Ceq} = R_C \parallel R_L/2 \tag{15.45}$$

Equation 15.44 can now be used to calculate the loaded differential gain.

15.6 Operational Amplifiers

Very few circuits have had an impact on the electronics field as great as that of the op-amp circuit. This name is a shortened version of operational amplifier, a term that has a rather specific meaning. As mentioned in Section 15.2, before IC chips became available, operational amplifiers were expensive, high-performance amplifiers used primarily in a system called an analog computer. The analog computer was a real-time simulator that could simulate physical systems governed by differential equations. Examples of such systems are circuits, mechanical systems, and chemical systems. One popular use of the analog computer was the simulation of automatic control systems. The control of a space vehicle had to be designed on paper before test flights took place. The test flights could then be used to fine-tune the design. The analog computer was far more efficient than the digital computer of the day in simulating differential equations and was far less costly.

The op-amp was an important component of the analog computer that allowed several mathematical operations to be performed with electronic circuits. The op-amp could be used to create a summing circuit, a difference circuit, a weighting circuit, or an integrating circuit. Using these capabilities, complex differential systems can be created with an output voltage that represents the physical output variable of the simulated system. The time variable may be scaled, but for many simulations, the output represents a real-time solution.

One of the key features of an op-amp is the differential input to the amplifier. This allows differences to be formed, and also allows the creation of a virtual ground or virtual short across the input terminals. This virtual short is used in summing several current signals into a node without affecting the other input current signals. These signals are then summed and easily converted into an output voltage.

The virtual ground also allows the formation of rather accurate integrals the op-amp using an additional resistor and capacitor. This feature is essential in the simulation of differential equations.

Other amplifiers are sometimes mistakenly referred to as op-amps, but unless these amplifiers possess the capability to create a virtual ground and do mathematical operations, this is a misnomer.

The first IC op-amp was introduced by Fairchild Semiconductor Corporation in 1964. This chip, designated the μA702 or 702, was followed shortly by the 709 which was the first analog IC product to receive wide industry acceptance. The National LM101 and Fairchild 741 advanced the field and eliminated the external compensation capacitors required for the earlier models. The ensuing popularity and low price of the 741 allowed op-amps to be treated as a component rather than a subcircuit that must be designed.

Today, op-amps are available in bipolar (BJT), CMOS, and BiCMOS (BJT/CMOS) technologies and designers have the option of including dozens of op-amps and other circuits on a single chip. Modern op-amps generally use the same architecture developed in the LM101/741 circuits with performance improvements resulting from improved processing techniques.

The configuration of an op-amp offers one important advantage for IC amplifiers. This advantage is the op-amp's ability to allow the input signal to be referenced to any dc voltage, including ground, within the allowed input range. This eliminates the need for a large coupling capacitor to isolate a dc input transducer from an amplifier input.

15.6.1 Classical IC Op-Amp Architecture

While the technology used to implement op-amps has changed considerably over the years, the basic architecture has remained remarkably constant [4]. This section first discusses that architecture and some approaches to its implementation. After this topic is considered, the subject will turn to methods of specifying amplifier performance.

The architecture of many op-amps appears as shown in Figure 15.19. The first section is a differential amplifier required by all op-amps to allow a virtual ground and the implementation of mathematical operations. This stage is generally designed to have a very high differential voltage gain, perhaps a few thousand. The bandwidth is generally rather low as a result of the high voltage gain. In some cases, this differential amplifier will use an active load that will also convert the double-ended output of the differential stage to a single-ended output that will drive the second voltage amplifier.

The second stage of Figure 15.19 is a single-ended voltage amplifier with a relatively high voltage gain. This gain may reach values of 500 V/V. Often this stage is used to compensate the op-amp, a topic that will be discussed later in this chapter. For now it is sufficient to say that this stage will probably be an

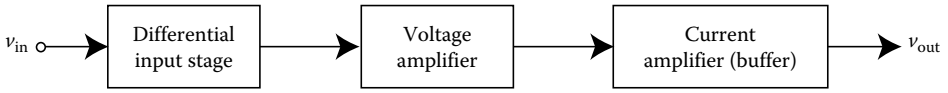


FIGURE 15.19 Classical op-amp architecture.

inverting stage that can multiply the apparent value of some capacitance placed between the input and output of the stage. The large capacitive load presented to the output of the first stage due to the Miller effect will lead to a very low upper corner frequency, perhaps 10–100 Hz, for the first stage.

The last stage is the output stage. It may be nothing more than an emitter follower that has a large current amplification along with a voltage gain that is near unity. This stage will have a very high upper corner frequency.

15.6.2 High-Gain Differential Stage

The normal way to achieve high voltage gains is to use an active load for the amplifying stages. The incremental resistance presented to the amplifying stage is very high while the dc voltage across the active load is small. One popular choice for an active differential stage load is the current mirror. This load provides very high voltage gain and also converts the double-ended output signal to a single-ended signal referenced to ground. Figure 15.20 shows a block diagram of such an arrangement.

With no input signal applied to the differential stage, the tail current splits equally between I_{diff1} and $I_{\text{diff 2}}$. The input current to the mirror equals this value of $I_{\text{tail}}/2$. This value is also mirrored to the output of the mirror giving $I_{\text{out}} = I_{\text{tail}}/2$. We will assume that the voltage between the current mirror output and the second differential stage is approximately zero, although this assumption is unnecessary to achieve the correct result.

When a signal is applied to the differential input, it may increase the current I_{diff1} by a peak value of ΔI. The input current to the mirror now becomes

$$I_{\text{in}} = \frac{I_{\text{tail}}}{2} + \Delta I$$

The output current from the mirror also equals this value. However, the input signal to the differential stage will decrease I_{diff2} by the same amount that I_{diff1} increases. Thus, we can write

$$I_{\text{diff2}} = \frac{I_{\text{tail}}}{2} - \Delta I$$

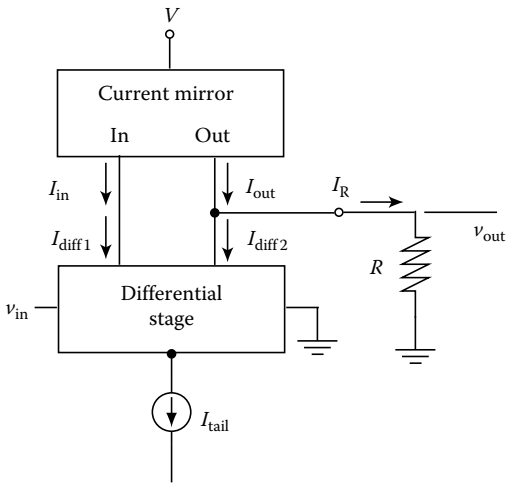


FIGURE 15.20 Differential stage with current mirror load.

The current to the resistance R increases from its quiescent value of zero to

$$I_{\text{R}} = I_{\text{out}} - I_{\text{diff2}} = 2\Delta I$$

The incremental output voltage resulting is then

$$v_{\text{out}} = 2R\Delta I \tag{15.46}$$

When an incremental input signal is applied to the differential pair, half of this voltage will drop across each base–emitter junction of the pair. This results in equal incremental differential stage currents in the two output devices, but they will be in opposite directions. An incremental input signal, v_{in}, will produce incremental currents of

$$i_{\text{diff1}} = \Delta I = \frac{g_{\text{m}} v_{\text{in}}}{2} \tag{15.47}$$

and

$$i_{\text{diff2}} = -\Delta I = -\frac{g_{\text{m}} v_{\text{in}}}{2} \tag{15.48}$$

where g_{m} is the transconductance of devices 1 and 2. Assuming negligibly large output resistances of the current mirror and the differential stage, the incremental output voltage becomes

$$v_{\text{out}} = 2i_{\text{diff1}} R = g_{\text{m}} R v_{\text{in}}$$

with a resulting midband voltage gain of

$$A_{\text{MB}} = g_{\text{m}} R \tag{15.49}$$

If the output resistances of the mirror and differential stage are significant, the voltage gain can be found by combining these resistances in parallel with the load resistance to form $R_{\text{eff}} = R_{\text{out}} \| R$. This resistance then replaces R in Equation 15.49. The load resistance may, in fact, be the incremental input resistance of the following stage. Very large values of voltage gain can result from this configuration. For the BJT, the transconductance is given by $g_{\text{m}} = \alpha / r_{\text{e}} \approx 1/r_{\text{e}}$.

While this expression for voltage gain is the same as that for the differential gain of a resistive load stage, given by Equation 15.44, two significant points should be made. First of all, the impedance R can be much greater than any resistive load that can be used in a differential stage. Large values of R in the differential stage would cause saturation of the stages for reasonable values of tail currents. In addition, large values of R are more difficult to fabricate on an IC chip. The current mirror solves this problem. The second point is that the output voltage of the differential pair with a current mirror load is a single-ended output which can be applied to a following simple amplifier stage. However, the rejection of common-mode variables caused by temperature change or power supply voltage changes is still in effect with the current mirror stage. If a resistive load differential stage must provide a single-ended output, the gain drops by a factor of 2, compared to the double-ended output, and common-mode rejection no longer takes place.

15.6.3 Second Amplifier Stage

Before the purpose of the second amplifier stage can be fully understood, a discussion on circuit stability must take place.

15.6.3.1 Feedback and Stability of the Op-Amp

The op-amp is used in a feedback configuration for essentially all amplifying applications. Because of nonideal or parasitic effects, it is possible for the feedback amplifier to exhibit unstable behavior. Oscillations at the output of the amplifier can exist, having no relationship to the applied input signal. This, of course, negates the desired linear operation of the amplifier. It is necessary to eliminate any undesired oscillation signal from the amplifier.

The open-loop voltage gain of a three-stage amplifier such as that of Figure 15.19 may be represented by a gain function of

$$A = \frac{A_{\text{MB}}}{\left(1 + j\frac{\omega}{P_1}\right)\left(1 + j\frac{\omega}{P_2}\right)\left(1 + j\frac{\omega}{P_3}\right)} \tag{15.50}$$

where P_1, P_2, and P_3 are the dominant poles of gain stages 1, 2, and 3. The quantity A_{MB} is the low-frequency or midband gain of the amplifier.

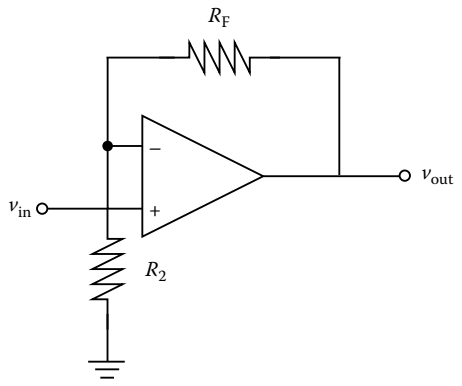


FIGURE 15.21 Noninverting op-amp stage.

When this op-amp is used in a negative feedback configuration such as that shown in Figure 15.21, the loop gain must be analyzed to see if the conditions for oscillation occur. The feedback factor for this circuit is

$$F = \frac{R_2}{R_2 + R_F} \tag{15.51}$$

In order to check the stability of this circuit, a zero volt input signal (short circuit) can be applied to the noninverting input terminal. The loop gain from inverting terminal to output and back to inverting terminal can then be found with the noninverting terminal shorted to ground. When this is done, it is found [4] that the amplifier is unstable for most practical values of R_F and R_2. The worst case condition occurs as R_2 approaches infinity and R_F approaches zero as in the case of a unity-gain voltage buffer.

One measurement of stability is referred to as the phase margin. This quantity is defined in terms of the phase shift of AF that exists when the magnitude of AF has dropped to unity. The number of degrees less than $-180°$ at this frequency is called the phase margin. Most amplifiers target a phase margin of 45°or more.

One possibility to achieve a reasonable phase margin is to intentionally modify one of the three upper corner frequencies of the op-amp. For example, if the pole associated with the first stage is lowered by several factors, the amplifier can become stable, even when used as a unity-gain voltage buffer.

The frequency response of this gain is sketched in Figure 15.22 for a midband gain of 300,000. We note that the magnitude of this gain has fallen below a value of unity (0 dB) before the second upper corner frequency of 2×10^6 is reached. The phase shift of AF at the frequency where the magnitude has dropped to unity may be $-130°$ resulting in a phase margin of 50°. Normally, engineers do not destroy bandwidth of a stage intentionally, however in this case it is necessary to stabilize the operation of the op-amp in the feedback configuration.

Lowering the upper corner frequency of one stage is not a trivial matter. While it may not need to be reduced to 10 rad/s in a practical op-amp, it is often lowered to 10–100 Hz. A capacitor must be added to the appropriate point in the stage to drop this frequency, but a relatively large capacitor is required. In the early days of the IC op-amp, the two terminals between which a capacitor was to be

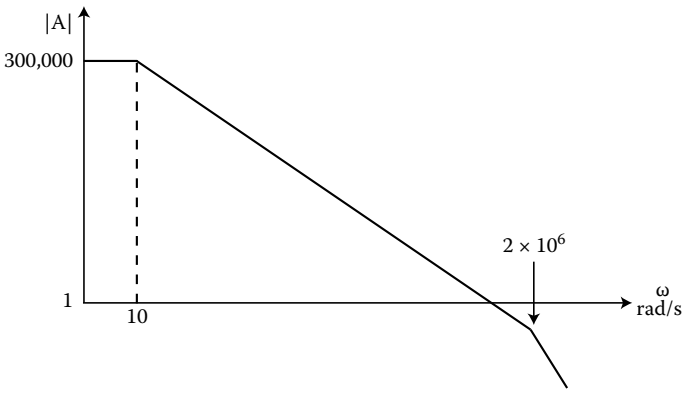


FIGURE 15.22 Op-amp frequency response.

added were connected to two external pins of the chip. A discrete capacitor of sufficient value was then added externally.

In 1967, the capacitor was added to the IC chip using the Miller effect to multiply the capacitor value. Returning to Figure 15.19, it is seen that a capacitor can be added between input and output of the second stage. This is typically a capacitor of value 30 pF that, due to the Miller effect, is multiplied by a factor of $(A_2 + 1)$ where $-A_2$ is the gain of the second stage. The Miller capacitance is reflected to the input of the second stage that loads the output of the first stage. This large effective capacitance, driven by the large output impedance of the first stage produces a very low upper corner frequency for the first stage.

In a 741 op-amp design, the gain A_2 is approximately -400 V/V. With a 30 pF capacitance bridging input to output, the effective capacitance at the input is about 0.012 μF. This creates a bandwidth for the op-amp of 10 Hz or 62.8 rad/s.

This method of solving the instability problem is referred to as dominant pole compensation. The lower value of pole frequency is seen from Figure 15.22 to dominate the amplifier performance up to frequencies above the useful range of gain.

It should be mentioned that op-amps for specific applications may not need to be stabilized for the unity gain configuration. The minimum gain required by the amplifier may be 2 V/V or 3 V/V or some other value that is easier to compensate. In such cases, the dominant pole need not be decreased to the level of the general purpose op-amp. The resulting frequency performance of the op-amp can then be higher than that of the general purpose stage.

15.6.4 Op-Amp Specifications

There are several nonideal effects in the op-amp that detract from its overall performance. Mismatching of devices in the input differential pair, required bias currents, and junction voltage temperature effects can degrade the performance. These effects are described in terms of the following specifications. The diagram of Figure 15.23 is used to define various terms used in these definitions.

Input offset voltage (V_{OS}): Mismatch of the transistors in the differential input stage leads to a finite output dc voltage when both inputs are shorted to ground. This finite output voltage is called the output offset voltage. A slight voltage mismatch in the differential pair is amplified by succeeding stages to create a larger voltage at the output. Inaccurate biasing of later stages also contribute to the output offset. Inaccuracies in later stages are amplified by smaller factors than are early stage inaccuracies.

The input offset voltage is the voltage that must be applied across the differential input terminals to cause the output voltage of the op-amp to equal zero. Theoretically, this voltage could be found by measuring the output voltage when the input terminals are shorted, then dividing this value by the gain of the op-amp. In practice, this may not be possible as the gain may not be known or the output offset may exceed the size of the active region. The value of V_{OS} is typically a few millivolts for monolithic or IC op-amps.

Input offset voltage drift (TCV_{OS}): Temperature changes affect certain parameters of the transistors, leading to a drift in the output dc offset voltage with temperature. In BJT devices, the voltage across the base-to-emitter junction for a constant emitter current drops by approximately 2 mV/°C. Small drifts of voltage in early stages will be amplified by following stages to produce relatively large drifts in output voltage. Because the output dc signal may not exist within the active region of the op-amp, the drift is again referred to the input.

The input offset voltage drift is defined as the change in V_{OS} for a 1°C change in temperature (near room temperature). A value of 10–20 μV/°C is typical for IC op-amps.

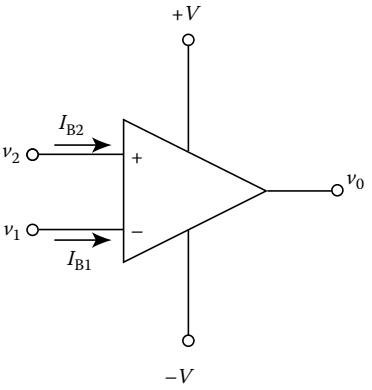


FIGURE 15.23 Op-amp.

Input bias current (I_B): A BJT differential stage will require a finite amount of base current for biasing purposes. This is true even if both inputs are grounded. The input bias current of an op-amp is defined as the average value of bias current into each input with the output driven to zero. The two bias currents are generally slightly different so I_B is

$$I_B = \frac{(I_{B1} + I_{B2})}{2} \tag{15.52}$$

Input offset current (I_{OS}): The input offset current is the difference between the two input bias currents when the output is at zero volts. This parameter is

$$I_{OS} = |I_{B1} - I_{B2}| \tag{15.53}$$

Common-mode input voltage range (CMVR): The voltage range over which the inputs can be simultaneously driven without causing deterioration of op-amp performance is called the common-mode voltage range or CMVR. In most op-amps, the CMVR is a few volts less than the rail-to-rail value of the power supplies. In many applications, both inputs are forced to move together due to the virtual short between the input terminals when negative feedback is used.

Common-mode rejection ratio (CMRR): The ratio of input common-mode voltage to change in input offset voltage is called the common-mode rejection range or CMRR. An equivalent definition is the ratio of differential voltage gain to common-mode voltage gain. IC op-amps range from 80 to 100 dB for the CMRR. This parameter was mentioned earlier in the chapter and is a measure of the mismatch of incremental gain from each of the two inputs to output. If the incremental gains from each input to output were equal, the CMRR would be infinite.

Power supply rejection ratio (PSRR): The power supply rejection ratio or PSRR is the ratio of change in the input offset voltage to a unit change in one of the power supply voltages. An op-amp with two power supplies requires that a PSRR be specified for each power supply.

15.7 Wideband Amplifiers

The development of wideband amplifiers with discrete circuits progressed rapidly during World War II with vacuum tubes and into the 1960s with BJT circuits. Although many important techniques were perfected between 1940 and 1960, these methods are not generally applicable to wideband IC design. Many of these approaches to wideband amplifier design required relatively large inductors for shunt peaking and relatively large coupling capacitors to ac couple individual stages. While small inductors are now available on IC chips, the range of values limit their usage. Likewise, the limitation on IC capacitor sizes precludes the use of coupling capacitors for IC amplifiers. These difficulties are mitigated in IC design by using additional transistors, however, the benefit of past theoretical developments is then largely unused. Furthermore, the absence of coupling capacitors in IC amplifiers leads to a much stronger interaction between dc and ac design of each stage than is present in discrete design. Two of the more popular methods of IC wideband amplifier design that overcome the limitations on IC circuits are based on composite stages or feedback cascades [7,8].

15.7.1 Composite or Compound Stages

The wideband amplifier of Figure 15.24 is a classic architecture on which many present amplifiers are based. This schematic is for the RCA CA3040 [8]. The input is a buffered, differential, cascode pair. The transistors T_1 and T_4 are emitter follower stages while the pair of devices T_2 and T_3 forms a cascode

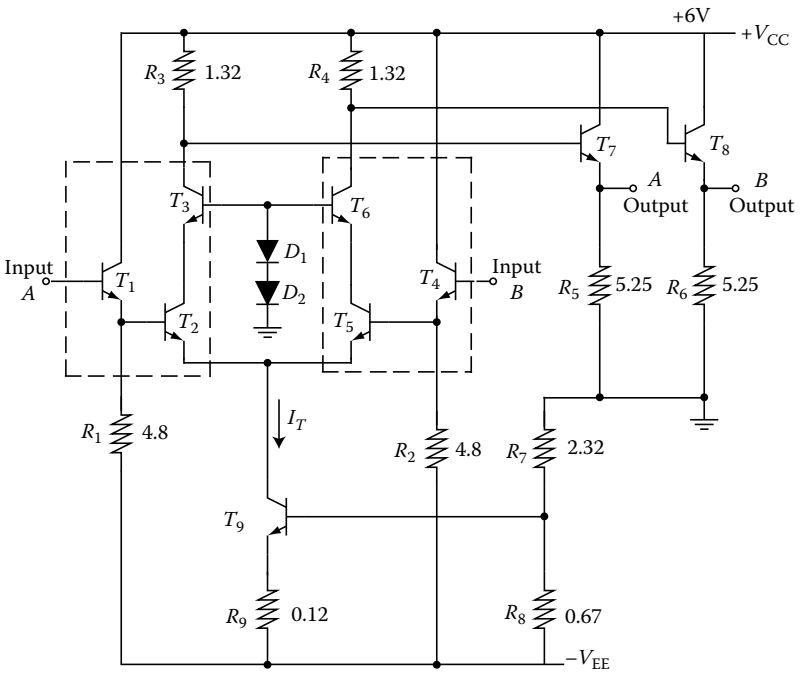


FIGURE 15.24 RCA CA3040 wideband amplifier.

stage as also does the pair of devices T_5 and T_6. Transistor T_9 with its emitter degeneration resistance forms a high output impedance current source. Transistors T_7 and T_8 buffer the output signals to allow relatively high output currents.

Since the input is differential, the input signal can be referenced to ground eliminating the need for a coupling capacitor. The frequency response of the gain extends from dc up to approximately 55 MHz with a constant gain of 30 dB.

The Motorola MC1490 [8] is another typical wideband amplifier chip that can be used for radio frequency or audio applications. It has a power gain of 50 dB at 10 MHz and 35 dB at 100 MHz. It also has a built-in automatic gain control to allow usage as an intermediate frequency amplifier.

15.7.2 Feedback Cascades

The circuit of Figure 15.25 shows the schematic of the MC1553 wideband amplifier using a feedback triple [7]. Transistors T_1, T_2, and T_3 make up the feedback triple using resistor R_F to create a feedback path. This path establishes the incremental voltage gain and also provides dc feedback to keep T_1, T_2, and T_3 in their active regions. There is a second feedback path from the current mirror output transistor T_6, but this is primarily dc feedback. The capacitor C_B is a large, external bypass capacitor that must be added to the circuit to decouple the ac feedback to the input stage. This IC amplifier has a voltage gain of 50 V/V with a bandwidth of 50 MHz.

Although the capacitor C_B is required in some applications, if the amplifier is driven by a low impedance source, it is unnecessary to add this element.

In recent years, both wideband and narrowband amplifiers in the 1–6 GHz range have become more significant. Many companies that provide amplifiers in this frequency range use HBT designs with SiGe, GaAs, GaN, or other material to achieve the necessary frequency performance.

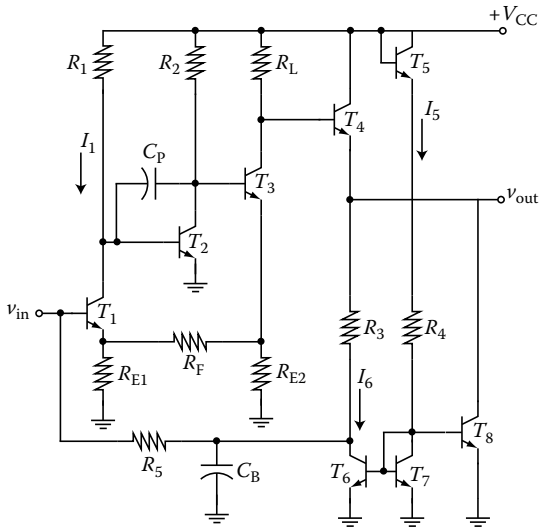


FIGURE 15.25 MC1553 wideband amplifier.

15.8 IC Power Output Stages [3]

If an IC power amplifier is to occupy a relatively small volume, the output power will be limited. This is due to the limitation imposed on the thermal conductivity of a small-area power device. A discrete transistor mounted on a large heat sink will exhibit a much higher thermal conductivity than that of the smaller IC chip. This limits the power that can be dissipated by the output device or devices of the IC as the junction temperature will be higher with the lower thermal conductivity of the IC chip. Typically, this limitation leads to an IC power output stage that is implemented in a high-efficiency configuration. Generally, the output stage for power outputs in the range of a few watts will use a class-B configuration while those with tens or hundreds of watts will use a class-D configuration.

One of the most significant limitations on dissipation is the junction temperature. As the temperature rises, several potentially dangerous effects may occur. First, the solder or alloys used in the transistor can be softened or even melted. Second, the impurity profiles in the doped regions can be affected if elevated temperatures exist for long periods of time. A third result of higher temperatures is the increase in collector leakage current. In power transistors, this current doubles for an incremental increase in temperature of 7°C–10°C. The leakage current, I_{co}, at temperature T_2 can be related to I_{co} at temperature T_1 by

$$I_{co}(T_2) = I_{co}(T_1) \times 2^{(T_2 - T_1)/T_K}$$

where T_K can range from 7°C to 10°C. For a temperature increase of 100°C, the minimum factor of increase in I_{co} is 1024 ($2^{100/10}$), whereas the maximum factor is approximately 20,000 ($2^{100/7}$). This marked increase in I_{co} can lead to increased power dissipation which leads to an increased temperature followed by a further increase in I_{co}. In some cases, this feedback effect is large enough to cause thermal runaway and destroy the transistor. This effect is minimized by placing a resistance in the emitter to decrease the forward bias on the base–emitter junction as current increases.

In other situations, the leakage current can approach the value of the quiescent collector current. Since leakage current is not controlled by the applied input signal, its effects can severely limit the amplifying

properties of the stage. For these reasons, the manufacturer places a limit on the maximum allowable junction temperature of the device.

The maximum junction temperature is therefore an important quantity that limits the power a transistor can deliver. The junction temperature will be determined by the power being dissipated by the transistor, the thermal conductivity of the transistor case, and the heat sink that is being used. The collector junction is the point at which most power is dissipated; hence, it is this junction that concerns us here.

Basically, manufacturers specify the allowable dissipation of a transistor in two ways. One way is to specify the maximum junction temperature along with the thermal resistance between the collector junction and the exterior of the case. Although this method is very straightforward, it incorrectly implies that the allowable power increases indefinitely as the transistor is cooled to lower temperatures. Actually, there is a maximum limit on the allowable dissipation of the transistor which is reflected by the second method of specification. This method shows a plot of allowable power dissipation versus the temperature of the mounting base. Quite often this plot shows power dissipation versus ambient temperature, where an infinite heat sink is used. However, if the transistor could be mounted on an infinite heat sink, the ambient temperature would equal the mounting base temperature; thus both plots convey the same information. This second method indicates the maximum allowable power dissipation, in addition to the maximum junction temperature.

The maximum limit on power dissipation ensures that chip temperature differentials do not become excessive as excess power is dissipated in collector regions. It also minimizes the possibility of excessive collector currents in typical applications.

15.8.1 Thermal Resistance

The thermal resistance of a material is defined as the ratio of the temperature difference across the material to the power flow through the material. This assumes that the temperature gradient is linear throughout. The symbol θ is used for thermal resistance, and

$$\theta = \frac{\Delta T}{P} = \frac{\text{Temperature difference across conductor}}{\text{Power flowing through conductor}} \tag{15.54}$$

The diagram of Figure 15.26 represents the thermal circuit of an IC chip, including the output transistor, surrounded by free air. Here, θ_{JM} is the thermal resistance from collector to mounting base and θ_A is the thermal resistance of that portion of air in contact with the mounting base; T_A is the temperature of air far away from the transistor; T_M is the mounting base temperature; T_J is the collector junction temperature. The power P will be determined by the electrical circuit that includes the output transistor; P, in turn, will determine the temperatures T_J and T_M. The temperature T_J can be written as

$$T_J = T_A + P(\theta_{JM} + \theta_A) \tag{15.55}$$

This equation shows that as more power is dissipated by the output transistor, T_J must rise.

For high-power IC chips mounted in a TO-3 package, θ_A is usually many times greater than θ_{JM}.

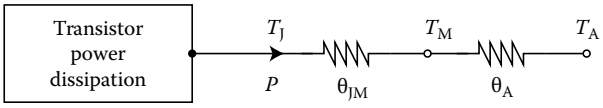


FIGURE 15.26 Transistor dissipating power in free air.

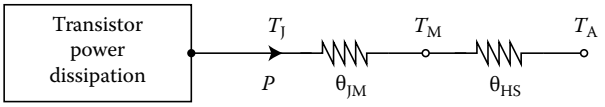


FIGURE 15.27 Transistor mounted on heat sink.

If θ_{JM} were 1°C/W, then θ_A might be 5°C/W to 20°C/W. Of course, θ_A depends on the area of the IC package.

The diagram of Figure 15.27 represents the thermal circuit of the IC chip mounted on a heat sink. The power that can be dissipated without exceeding the maximum junction temperature is found from solving the preceding equation to result in

$$P_{max} = \frac{T_J - T_A}{\theta_{JM} + \theta_{HS}} \tag{15.56}$$

The values of θ_{HS} presented between the chip case and free air might range from 0.4°C/W for air-cooled systems to 2°C/W for flat vertical-finned aluminum heat sinks to 8°C/W for cylindrical heat sinks that slide over the chip package. For each thermal circuit, the amount of allowable power dissipation is fixed.

15.8.2 Circuit or Conversion Efficiency

The efficiency of a power output stage is a measure of its effectiveness in converting dc power to ac output power. It is defined as

$$\eta = \frac{P_{out}}{P_S} \tag{15.57}$$

where

P_{out} is the average output power

P_S is the power supplied by the dc power supply

A useful relationship between the allowable dissipation of the chip and the maximum output power can be found by assuming that the power delivered from the dc source is dissipated by the output transistor and the load. This assumption would be very inaccurate for a class-A, resistive load stage, in which the resistor dissipates significant dc power. For many class-B or class-D stages, the assumption is not unreasonable. In equation form, this assumption is expressed by

$$P_S = P_{out} + P_T \tag{15.58}$$

where P_T is the actual dissipation of the output transistor or transistors.

In terms of the circuit efficiency, the transistor dissipation can be written as

$$P_T = (1 - \eta)P_S = (1 - \eta)\frac{P_{out}}{\eta} \tag{15.59}$$

Solving for P_{out} in terms of P_T leads to

$$P_{out} = \frac{\eta}{1 - \eta}P_T \tag{15.60}$$

The effect of circuit efficiency on output power can be demonstrated by assuming that the maximum allowable output stage dissipation is 5 W. If the circuit efficiency is 50% or $\eta = 0.5$, the maximum output power calculated from Equation 15.60 is also 5 W. If the efficiency is increased to 78.5%, the maximum efficiency of an ideal class-B stage, the maximum output power is found to be 18.26 W. Increasing the circuit efficiency to 98% a figure approached in a near-ideal class-D stage, the maximum output power becomes 245 W. In this case, an output stage that can dissipate 5 W delivers 245 W to the load. Although these calculations are based on some idealizations, it clearly shows the importance of using a circuit configuration that leads to a high value of efficiency. This explains the popularity of the class-B stage and the class-D stage in IC design as opposed to the class-A stage.

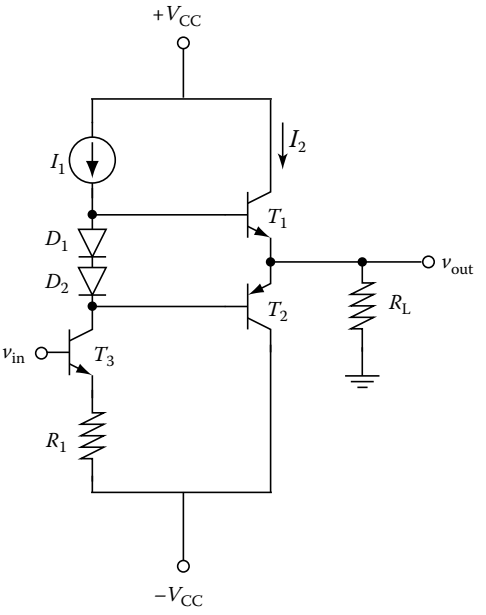


FIGURE 15.28 Class-B IC stage.

15.8.3 Class-B Output Stage

A class-B stage that can be integrated is shown in Figure 15.28. The current source I_1 consists of an IC current-source stage to provide a small bias current through diodes D_1 and D_2. A small quiescent collector current I_2 is necessary to reduce crossover distortion at the output. As v_{in} swings positive the input to T_2 goes negative, turning T_2 on while shutting T_1 completely off. A negative-going waveform at the amplifier input drives the bases of T_1 and T_2 positive to shut T_2 off and pass the signal to the output through T_1.

The amplifier can be made short-circuit proof by limiting the output current that can flow if the output terminal is accidentally shorted to one of the supplies. Figure 15.29 indicates the additional circuitry required for this purpose. The emitter–base junctions of transistors T_4 and T_5 are driven by the voltage drops across the resistances R_{E1} and R_{E2}. Under normal operating conditions, these voltages are too small to turn T_4 and T_5 on; thus, circuit operation is unaffected. If the output is short-circuited to the negative supply voltage, serious damage to device T_1 could result if transistor T_4 were not present. A large voltage would appear across T_1 and the base–emitter junction would be forward biased resulting in a high value of emitter current. The excessive power dissipation could destroy T_1 if the output were to be shorted to the negative supply or even to ground. When this occurs for the circuit of Figure 15.29, T_4 becomes sufficiently forward biased to divert the base-current drive from T_1. The maximum current that can flow in the output circuit is then limited to V_{BE4}/R_{E1}. Typical maximum currents for the short-circuit case range from 10 to 50 mA for modern IC amplifiers.

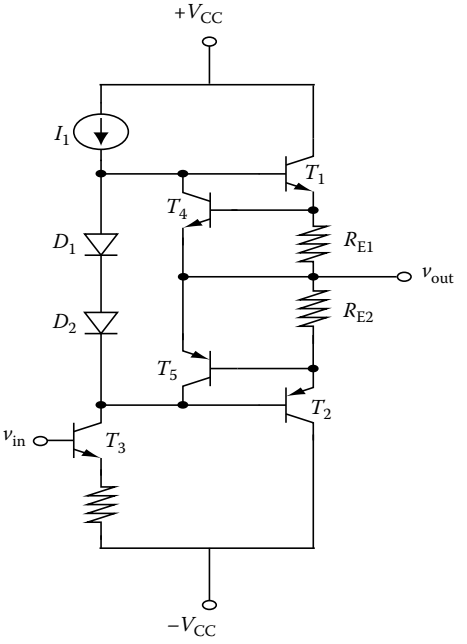


FIGURE 15.29 Class-B stage with short-circuit protection.

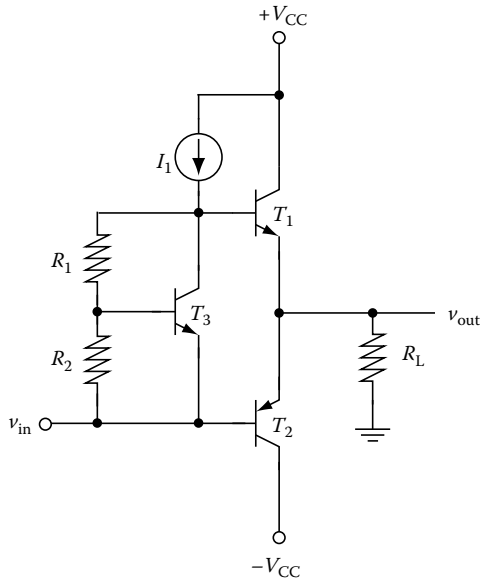


FIGURE 15.30 Bias circuit using V_{BE} multiplier.

The V_{BE} multiplier circuit often replaces the two diodes in Figures 15.28 and 15.29 to get better cancellation of the crossover voltage of the output devices. This circuit appears in Figure 15.30. If negligible base current flows in T_3, the voltage across R_1 and R_2 is

$$V_{CE3} = (1 + R_1/R_2)V_{BE3} \tag{15.61}$$

This voltage is found by noting that the drop across R_2 is constrained to be V_{BE3}, and this value must also equal

$$\frac{R_2}{R_1 + R_2} V_{CE3}$$

The voltage V_{CE3} is used to eliminate crossover distortion and can be adjusted by the ratio of the two resistors. Whereas the absolute values of individual resistors cannot be accurately determined in standard IC fabrication processes, the ratio of two resistors can be determined to the required accuracy.

There are several other output configurations based on the complementary emitter follower. A popular one is the Darlington output stage modified for IC amplifiers as shown in Figure 15.31. The current gain of this stage is very high, approximately equal to β^2. If high-gain *pnp* devices are available, a Darlington pair similar to the upper *npn* pair can replace devices T_2, T_4, and T_5. For the Darlington pairs, a larger difference of input bias voltage must be provided to the inputs of the respective pairs due to the larger voltage drop between each input and output which is now 2 $V_{BE(on)}$ instead of just $V_{BE(on)}$. The V_{BE} multiplier circuit can be designed to generate this increased bias voltage.

15.8.4 Class-D Output Stages

IC class-D amplifiers using PWM have been reported with efficiencies of 90% or more at 10 W output and a frequency response from 20 Hz to 20 kHz. Many IC chips are available that drive power BJTs or MOSFETs, delivering 30–50 W to a load. Larger discrete circuits report audio amplifiers based on the class-D stage that deliver 600 W per stereo channel [9]. This type of amplifier is often used in low-end car radios.

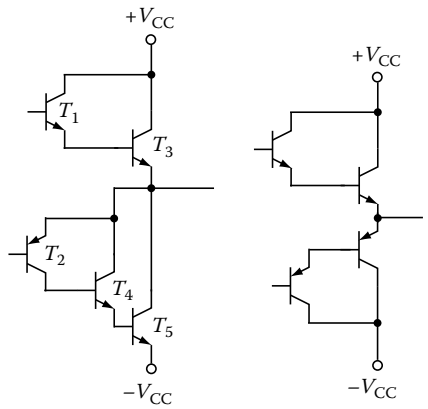


FIGURE 15.31 Modified Darlington and Darlington output stages.

PWM is used to reduce the power dissipated by the transistor while delivering a large power to the load. The varying load signal is applied by means of output devices that switch between on and off states. The resulting load voltage has a rectangular waveform that contains an average or dc value dependent on the duty cycle. In addition, the load voltage would also contain several ac components; however, these unwanted components can be filtered before reaching the load.

Any periodic waveform can be represented by a Fourier series consisting of a dc component (if present), a fundamental frequency component, and higher harmonics of the fundamental frequency. A rectangular wave switching

between $+V$ and $-V$ that remains positive for t^+ seconds is said to have a duty cycle of

$$d = \frac{t^+}{T} \tag{15.62}$$

where T is the repetition period of the waveform. A rectangular waveform with a 50% duty cycle contains no dc or average value, but a change of duty cycle will give the waveform an average value as shown in Figure 15.32. If the period, T, remains constant as the duty cycle is varied, the average value of a rectangular waveform of amplitude V is

$$V_{av} = V(2d - 1) \tag{15.63}$$

The average value varies directly with d. The Fourier coefficients of the ac components also vary as d is changed and new frequencies may be introduced, but in general, these components are of little interest to us as they can be easily eliminated. If, for example, the repetition frequency is 200 kHz, all ac components of the waveform will be greater than this value and far out of the audio range. Now if we vary the duty cycle sinusoidally at some low frequency, the average value will also vary sinusoidally. Mathematically, we can express this by saying that if $d = 0.5 + k \sin \omega t$, then the average value is

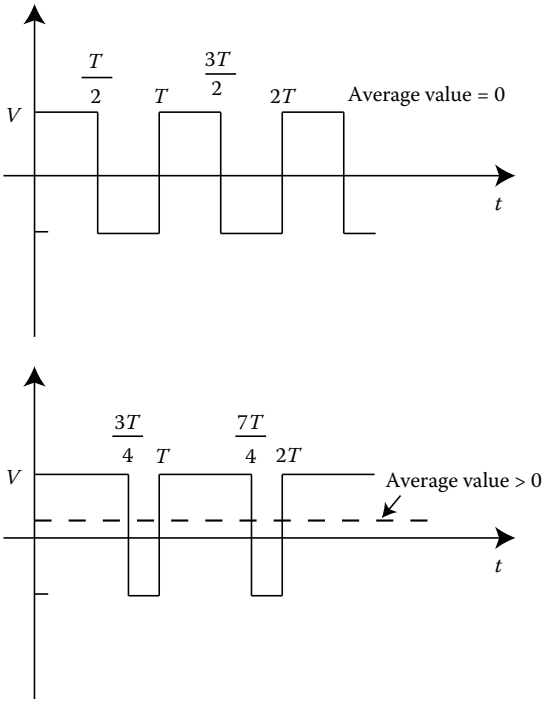


FIGURE 15.32 Rectangular waveforms with different duty cycles.

$$V_{av} = 2kV \sin \omega t \tag{15.64}$$

The waveform with variable duty cycle can be filtered by a low-pass filter to eliminate all frequencies above ω. The result is a low-frequency output sinusoid with an amplitude that varies in the same manner as the duty cycle. A block diagram of PWM amplification is shown in Figure 15.33.

The voltage control circuit must have a 50% duty cycle when the input signal is at zero volts. As v_{in} becomes nonzero, the duty cycle varies proportionally. The high-power switching stage amplifies this rectangular wave and applies the output to a low-pass circuit that allows only the changes in average value to pass to the load. The output signal is then proportional to the input signal, but can be at a much

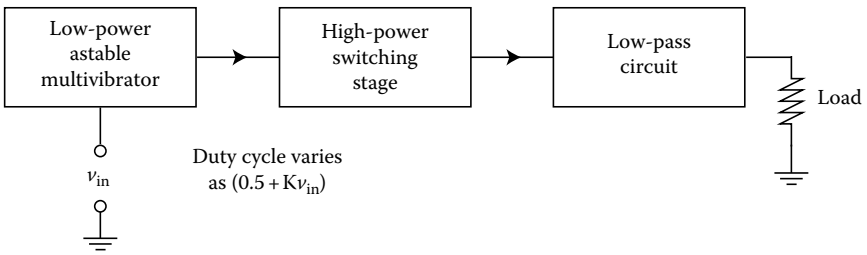


FIGURE 15.33 Architecture of a PWM amplifier.

higher power level than the input signal. The power output stage may dissipate only 5%–10% of the power delivered to the load.

The major advantage of PWM is that the output transistors need operate in only two states to produce the rectangular waveform: either fully on or fully off. In saturation we know that the very small voltage drop across a transistor leads to very low power dissipation. A very small dissipation is also present when the transistor is cut off. If switching times were negligible, no device power loss would occur during the transition between states. Actually, there is a finite switching time and this leads to an increased total dissipation of the output stages. Still, the efficiency figures for the class-D amplifier are very high, as reported earlier. This leads to higher possible power outputs and smaller chip areas for integrated PWM amplifiers. The stages can be direct-coupled to the load, which eliminates the necessity of capacitors. Nonlinear distortion can be less than that of class-B stages, and matching of transistors is unnecessary.

On the other hand, the disadvantages of this amplifier ultimately dictate the limits of usefulness of the PWM scheme. The upper frequency response is limited to a small fraction of the switching frequency. The operating frequency of power transistors generally decreases with higher power ratings. It follows that the upper corner frequency of the amplifier may be lower for higher power transistors. Furthermore, a low-pass filter may be required to eliminate the unwanted frequency components of the waveform. The generation of radio frequencies or electromagnetic interference by the switching circuits can also present problems in certain applications.

In addition to compound emitter followers, the power output stages can be designed in several arrangements. Figure 15.34 shows two possible configurations. The diodes appearing across the output transistors are present to protect the transistors against inductive voltage surges. If the filter is inductive,

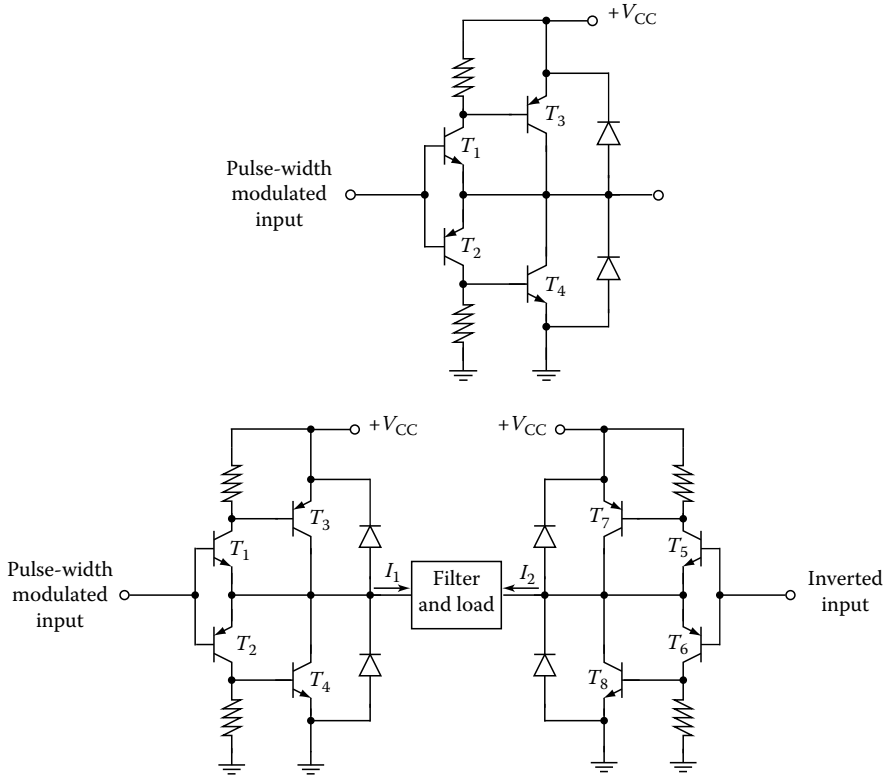


FIGURE 15.34 Class-D output stages.

the current reversals that occur over short switching times generate very large voltage spikes, unless the protective diodes are used.

In the push-pull circuit, the low-power, pulse-width modulated input turns T_1 and T_3 on when the signal is at its maximum value. Transistors T_2 and T_4 are off at this time and current is forced through the load. When the signal switches to the minimum value, T_1 and T_3 go off, while T_2 and T_4 turn on to pull current through the load.

Figure 15.34b shows a bridge circuit that can drive a floating load with a single power supply. When the input signal reaches its maximum value, T_1 and T_3 are on while T_2 and T_4 are held off. The input signal is inverted and applied to the bases of T_5 and T_6. This inverted signal is at its minimum value during the time when the normal input is maximum; thus, T_5 and T_7 will be off while T_6 and T_8 are on. Current will leave the collector of T_3, flow through the load, and enter the collector of T_8. When the input assumes the most negative value, T_1, T_3, T_6, and T_8 turn off while T_2, T_4, T_5, and T_7 turn on. Current now leaves the collector of T_7, flows through the load, and enters the collector of T_4. During this period, the load current flows in the opposite direction to that flowing when the input is maximum. The load current then reverses each time the input signal makes a transition.

In some applications, such as motor control or high-output audio systems, the load serves as a filter of high frequencies since these particular loads cannot respond to the switching frequencies.

References

1. T. Lewis, *Empire of the Air*. Harper Perennial, New York, 1991.
2. J. Bardeen and W. H. Brittain, Physical principles involved in transistor action, *Bell System Technical Journal*, 28, April 1949, 239–247.
3. D. J. Comer and D. T. Comer, *Advanced Electronic Circuit Design*. John Wiley and Sons, Inc., New York, 2002.
4. D. J. Comer and D. T. Comer, *Fundamentals of Electronic Circuit Design*. John Wiley and Sons, Inc., New York, 2002.
5. T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd edn. Cambridge University Press, New York, 2004.
6. A. Vladimirescu, *The SPICE Book*. John Wiley and Sons, Inc., New York, 1994.
7. P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th edn. John Wiley and Sons, Inc., New York, 2001.
8. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*. John Wiley and Sons, Inc., New York, 2003.
9. B. Duncan, *High Performance Audio Power Amplifiers*. Newnes, Jordan Hill, Oxford, UK, 1996.

16

Operational Amplifiers

16.1 Ideal Operational Amplifier **16**-1
Open-Loop Equivalent Circuit • Voltage Op-Amps • Op-Amp Circuit Applications • Comparators • Other Op-Amp Configurations • Summary
References **16**-11
16.2 Nonideal Operational Amplifier **16**-12
Finite Differential Gain • Output Saturation • Offset Voltage • Finite Common-Mode Rejection Ratio and Power-Supply Rejection Ratio • Input Bias Currents • Electrical Noise • Summary
References **16**-24
16.3 Frequency- and Time-Domain Considerations **16**-24
Voltage-Mode Op-Amps • Block Diagram • Current-Mode Op-Amps
Acknowledgment **16**-40
References **16**-40

David G. Nairn
Queen's University

Sergio B. Franco
San Francisco State University

16.1 Ideal Operational Amplifier

The operational amplifier, or op-amp, is a fundamental building block for many electronic circuits. Although the op-amp itself is composed of numerous transistors, it is usually treated as a single circuit element known as the ideal operational amplifier. The ability to treat the op-amp as an ideal circuit element simplifies its use in circuits such as amplifiers, buffers, filters, and data converters. With such varied uses, the op-amp has been implemented in many different forms. Nevertheless, the behavior of each of these different forms can still be characterized as an ideal op-amp.

16.1.1 Open-Loop Equivalent Circuit

The op-amp is primarily a high gain amplifier. Although the op-amp can be used on its own, most op-amps are part of larger circuits in which feedback is used to determine the circuit's overall transfer function. Consequently, the op-amp's precise behavior is only a secondary interest. To simplify preliminary circuit analysis and design, an abstraction of the practical op-amp, known as the ideal op-amp, is often used. The ideal op-amp is characterized by the following four parameters:

- Infinite gain
- Infinite bandwidth
- Draws no signal power at its inputs
- Is unaffected by loading of its output

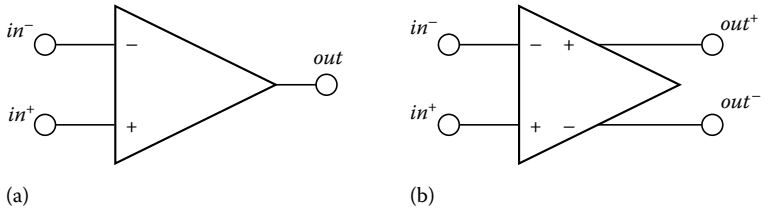


FIGURE 16.1 Circuit symbols for (a) the single-ended op-amp and (b) the fully differential op-amp.

Although such specifications are not achieved in practice, the nonidealities of practical op-amps (see Section 16.2) can be neglected in most applications.

The ideal op-amp is represented schematically as a triangle with two inputs and either one or two outputs, as shown in Figure 16.1. For the single output case, the output is referred to ground and the op-amp is known as a single-ended op-amp. For the two output case, the output is the difference between the out^+ and out^- outputs and the op-amp is known as a fully differential op-amp. Because the op-amp provides gain, it requires an external power source. For the ideal op-amp, the power supply has no effect on the amplifier's performance and is therefore not indicated in the circuit symbol.

Based on the above description, the op-amp's input can be stimulated with either a voltage or current. Also, the controlled output can be either a voltage or a current. Consequently, there are four possible implementations of the ideal op-amp: the voltage op-amp, the transimpedance op-amp, the current op-amp, and the transconductance op-amp, as shown in Table 16.1 and in Figure 16.2.

TABLE 16.1 Ideal Op Amp Types

Input	Output	Gain	Type
V	V	A_v	Voltage
I	V	R_m	Transimpedance
I	I	A_i	Current
V	I	G_m	Transconductance

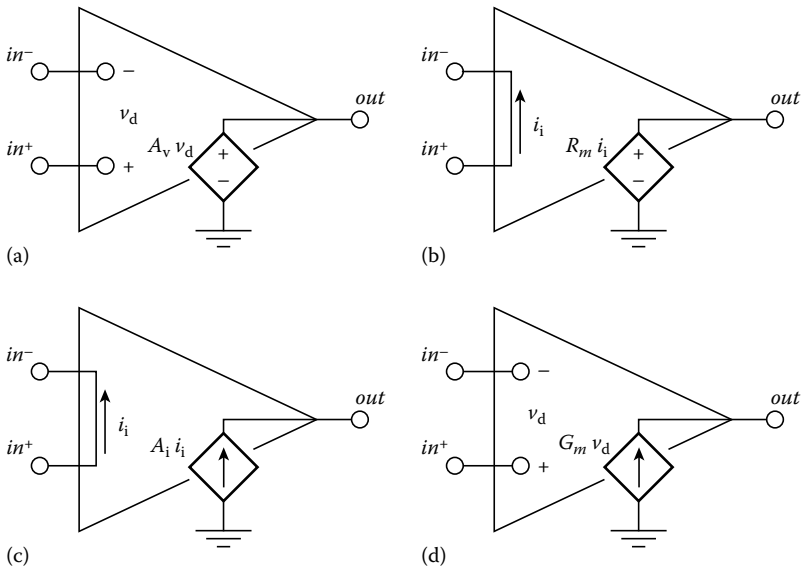


FIGURE 16.2 The four possible op-amp configurations: (a) the voltage op-amp, (b) the transimpedance op-amp, (c) the current op-amp, and (d) the transconductance op-amp.

For most applications, the op-amp is used in a closed-loop configuration with negative feedback. Due to the negative feedback, all four ideal op-amp types perform the same function. When the limitations of practical op-amps are considered, it will be found that different op-amps are preferred for different applications. Of the four types, the voltage op-amp is the most widely known. Therefore, the use of op-amps will be considered from the perspective of the voltage op-amp. Then, the other three types will be considered.

16.1.2 Voltage Op-Amps

The ideal voltage op-amp, illustrated in Figure 16.2a, is a voltage-controlled voltage source with infinite gain. If a potential difference v_d exists between the noninverting terminal in^+ and the inverting terminal in^- the op-amp's output voltage v_{out} will be

$$v_{out} = A_v v_d \tag{16.1}$$

where A_v is the differential gain and is both infinite and frequency independent. Note that only the differential voltage is amplified. As an ideal voltage-controlled voltage source, the op-amp has an infinite resistance and a zero output resistance. The properties of an ideal voltage op-amp may be summarized as follows:

- Infinite differential gain
- Zero common-mode gain
- Infinite bandwidth
- Infinite input resistance
- Zero output resistance

With these ideal properties, the op-amp is relatively easy to use in many circuit applications.

16.1.3 Op-Amp Circuit Applications

The op-amp can be used in both open-loop configurations and closed-loop configurations. If the op-amp is used open-loop, small voltages between the input terminals produced either a positive or negative infinite voltage due to the amplifier's infinite gain. Consequently, the op-amp can be used as a comparator. This application is discussed further in Section 16.1.4. In a closed-loop circuit, feedback allows the op-amp's output voltage to influence its input voltage (see Section 16.1.4). The op-amp can then be made to perform many complex operations.

16.1.3.1 Unity Gain Buffer

The simplest feedback that can be applied to the op-amp is depicted in Figure 16.3. The op-amp's output is connected to the inverting input and an input signal V_{in} is applied to the noninverting input. The feedback forces the voltage at in^- to equal V_{out}. By multiplying the differential voltage at the op-amp's input (i.e., $V_{in} - V_{out}$) by the op-amp's gain, V_{out} is found to be

$$V_{out} = A_v(V_{in} - V_{out}) \tag{16.2}$$

which can be rewritten as

$$\frac{V_{out}}{V_{in}} = \frac{1}{1 + 1/A_v} \tag{16.3}$$

because A_v is infinite, V_{out} equals V_{in}. Therefore, the circuit is a unity gain buffer. It is important to note that the op-amp's high gain and the use of negative feedback forces the voltage at the op-amp's two input

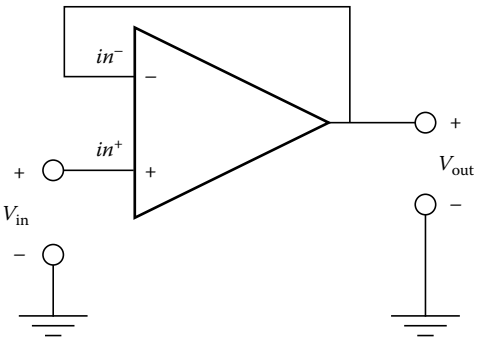


FIGURE 16.3 Unity gain buffer.

terminals to be equal. Hence if V_{in} is varied, V_{out} will follow or track it. The op-amp's two input terminals have the same potential, but no current flows between them. Therefore, a virtual short is said to exist between the inputs. The unity gain buffer draws no current from the signal source due to the op-amp's infinite input resistance and the op-amp's zero output resistance ensures that loading does not affect the voltage at V_{out}.

16.1.3.2 Simple Attenuator

If an output equal to a fraction of V_{in} is required, the circuit is illustrated in Figure 16.4a. The noninverting terminal is now a fraction of V_{in}. Because V_{out} tracks the voltage at the noninverting terminal, the circuit's output voltage will be

$$V_{out} = V_{in}\left[\frac{R_1}{R_1 + R_2}\right] \tag{16.4}$$

Due to the op-amp's infinite input resistance, the voltage divider formed by R_1 and R_2 is not loaded by the op-amp. Therefore, large values of R_1 and R_2 can be used to avoid loading the source voltage V_{in}.

16.1.3.3 The Noninverting Amplifier Configuration

Usually, it is more desirable to amplify a signal than to attenuate it. Therefore, instead of matching an attenuated V_{in} to V_{out}, at the op-amp's input, an attenuated V_{out} can be matched to V_{in}, as illustrated in Figure 16.4b. Due to the negative feedback and the op-amp's finite gain, the op-amp's two inputs have the same potential. Therefore,

$$V_{in} = V_{out}\left[\frac{R_1}{R_1 + R_2}\right] \tag{16.5}$$

which is more commonly written as

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_2}{R_1} \tag{16.6}$$

This circuit configuration is commonly referred to as the noninverting configuration [1]. Gains greater than or equal to unity can be achieved simply by changing the ratio of R_2 to R_1. The actual values of R_2

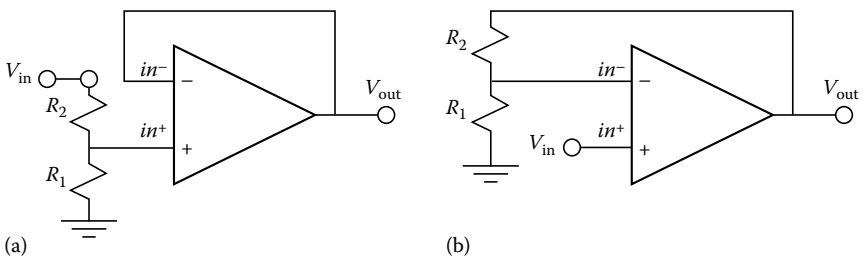


FIGURE 16.4 (a) A simple buffer attenuator and (b) the noninverting amplifier configuration.

and R_1 are unimportant. Only their ratio determines the gain. The op-amp's infinite input resistance ensures that no current is drawn from the source, therefore, V_{in} controls the voltage across R_1 but does not supply the current flowing through it.

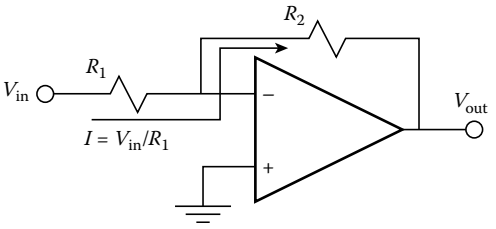


FIGURE 16.5 Inverting amplifier configuration.

16.1.3.4 The Inverting Configuration

If the source V_{in} can supply current, V_{in} can be connected directly to R_1, as illustrated in Figure 16.5. In this case the voltage across R_1 is still equal to V_{in} but of opposite polarity. The voltage at the op-amp's inverting input is now at ground potential, but no current flows to ground. Consequently, a virtual ground exists at the inverting input terminal. Because the current flowing through R_1 cannot flow to ground or into the op-amp, it flows through R_2, causing an output voltage of

$$V_{out} = 0 - R_2\left[\frac{V_{in}}{R_1}\right] \tag{16.7}$$

or

$$\frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1} \tag{16.8}$$

This circuit configuration is commonly referred to as the inverting configuration [1]. Both amplification and attenuation can be achieved by changing the ratio of R_2 to R_1. A very important difference between the inverting and noninverting configurations is that the inverting configuration draws a current equal to V_{in}/R_1 from the source. Consequently, even though the op-amp itself has an infinite input resistance, the inverting configuration only has an input resistance equal to R_1. Fortunately, the gain only depends on the ratio of R_2 to R_1, thereby allowing both resistors to be increased, thus limiting the current drawn from V_{in}.

16.1.3.5 Frequency Dependent and Nonlinear Transfer Functions

Elements other than resistors can be used in both the inverting and noninverting configuration. By using the frequency-dependent elements Z_1 and Z_2 in place of R_1 and R_2, circuits with arbitrary frequency responses can be generated. Two examples of this are illustrated in Figure 16.6a through c. In Figure 16.6a, a capacitor has been added in series with R_1. The inverting amplifier now has the transfer function.

$$\frac{V_{out}}{V_{in}}(s) = \frac{-Z_2}{Z_1} \tag{16.9}$$

where Z_1 equals $(1 + R_1Cs)/(Cs)$ and Z_2 equals R_2. Therefore, the circuit's transfer function is

$$\frac{V_{out}}{V_{in}}(s) = \frac{-R_2Cs}{1 + R_1Cs} \tag{16.10}$$

which is a simple high-pass filter, as illustrated in Figure 16.6b. Alternatively, a capacitor can be added in parallel with R_2 as shown in Figure 16.6c. Now, Z_1 equals R_1 and Z_2 equals $R_2/(1 + R_2Cs)$, which results in the transfer function.

$$\frac{V_{out}}{V_{in}}(s) = \frac{-R_2}{R_1(1 + R_2Cs)} \tag{16.11}$$

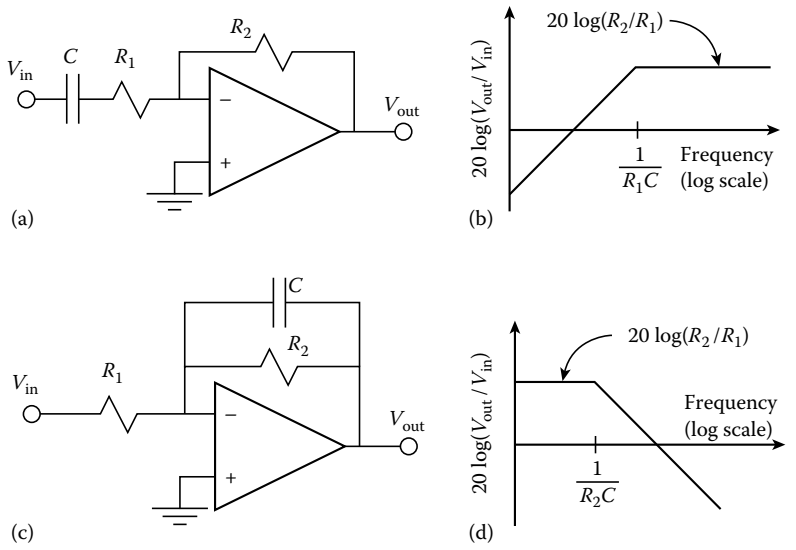


FIGURE 16.6 Frequency-dependent circuits using the inverting configuration and (a) a simple high-pass filter and (b) its frequency response; (c) simple low-pass filter and (d) its frequency response.

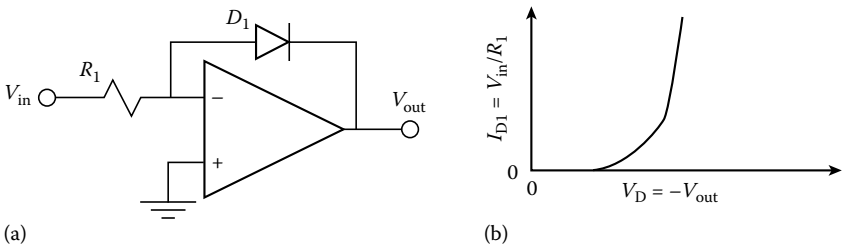


FIGURE 16.7 Obtaining the I–V characteristics of a diode: (a) circuit configuration and (b) oscilloscope display with V_{out} inverted.

This circuit performs as a low-pass filter, as illustrated in Figure 16.6d. By selecting Z_1 and Z_2, arbitrary transfer functions can be generated, thereby making op-amp circuits useful for implementing active filters (see Section II in *Passive, Active, and Digital Filters*). When designing arbitrary transfer functions, the resulting circuits must be stable if they are to perform correctly [2].

If nonlinear elements such as diodes are used in place of R_1 and R_2, nonlinear transfer functions can be obtained [2]. For example, the $I-V$ characteristics of a component can be obtained by replacing R_2 in the inverting configuration with the desired nonlinear element, as depicted in Figure 16.7a. Then, by applying V_{in} and V_{out} to an oscilloscope, the element's $I-V$ characteristic can be displayed directly as depicted in Figure 16.7b. Due to the diode's exponential characteristic, the circuit's output is the logarithm of its input

$$V_{out} = nV_T \ln\left(\frac{V_{in}}{R_1 I_S}\right) \tag{16.12}$$

The constants n and I_S are determined by the diode while V_T is the thermal voltage (see Section 11.5). Circuits of this type are made possible by the presence of the virtual ground at the op-amp's input.

16.1.3.6 Multiple Input Circuits

For the inverting configuration, the presence of the virtual ground at the op-amp's inverting input allow signals from many sources to be combined. As illustrated in Figure 16.8, the currents I_a and I_b are determined independently by V_a and V_b, respectively. These two currents are then summed at the virtual ground and forced through R_2. The resulting output is a weighted sum of V_a and V_b:

$$V_{out} = \frac{R_2}{R_{1a}} V_a - \frac{R_2}{R_{1b}} V_b \tag{16.13}$$

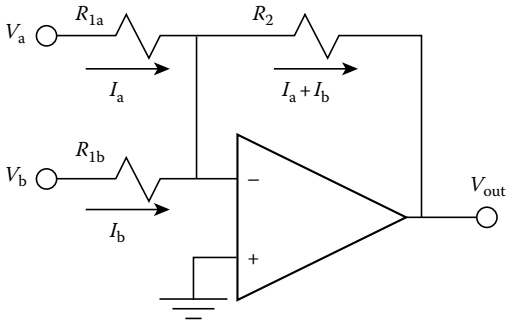


FIGURE 16.8 Weighted summer.

Any number of additional inputs can be added. The virtual ground prevents the different signals from interacting with each other.

The noninverting input can also be used in the multi-input circuit. In this case though, the op-amp's input is no longer at virtual ground and its output ceases to be a weighted sum of the inputs. For example, if V_1 and V_2 are applied to the circuit of Figure 16.9a, the output depends on both V_2 and the difference between V_2 and V_1:

$$V_{out} = V_2 + \frac{R_2}{R_1}(V_2 - V_1) \tag{16.14}$$

This output can also be written as a weighted difference between V_2 and V_1

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_2 - \frac{R_2}{R_1} V_1 \tag{16.15}$$

If, as in Figure 16.9b, V_2 is first attenuated by $R_2/(R_1 + R_2)$, a voltage only proportional to the difference between V_1 and V_2 can be obtained

$$V_{out} = \frac{R_2}{R_1}(V_2 - V_1) \tag{16.16}$$

where it is evident that the circuit amplifies the difference between its inputs and rejects the common-mode component. Consequently, the configuration is referred to as the differential configuration.

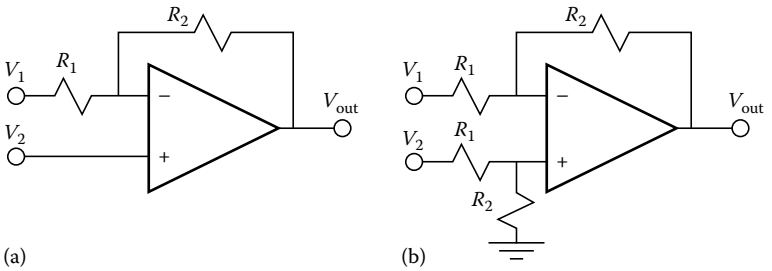


FIGURE 16.9 Circuits for finding weighted differences: (a) a circuit based in the inverting and noninverting configurations and (b) the differential configuration.

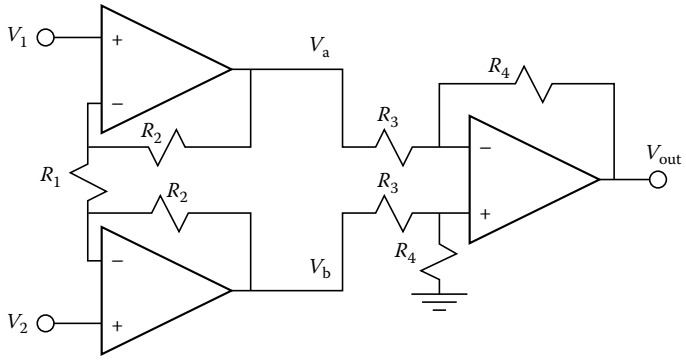


FIGURE 16.10 Instrumentation amplifier.

16.1.3.7 Instrumentation Amplifiers

The differential amplifier depicted in Figure 16.9b is useful for detecting weak signals in a noisy environment. Unfortunately, its input resistance is only $2R_1$. To circumvent this problem, V_1 and V_2 can be buffered using two unity gain buffers. A better solution is to use the instrumentation amplifier illustrated in Figure 16.10. This circuit combines two circuits of the type shown previously in Figure 16.9a and the differential amplifer in Figure 16.9b. Based on Equation 16.15, the voltage at V_a will be

$$V_a = \left(1 + \frac{R_2}{R_1}\right) V_1 - \frac{R_2}{R_1} V_2 \tag{16.17}$$

and the voltage at V_b will be

$$V_b = \left(1 + \frac{R_2}{R_1}\right) V_2 - \frac{R_2}{R_1} V_1 \tag{16.18}$$

With V_a and V_b applied to the differential amplifier, the instrumentation amplifier's output voltage will be

$$V_{out} = \frac{R_2}{R_1}(V_b - V_a) \tag{16.19}$$

Due to the op-amp's ideally zero output resistance, the differential amplifier's low input resistance does not load the other two circuits. Hence, by substituting Equations 16.17 and 16.18 for V_a and V_b, respectively, the output voltage can be expressed as

$$V_{out} = \left[1 + \frac{2R_2}{R_1}\right] \frac{R_4}{R_3}(V_2 - V_1) \tag{16.20}$$

which allows the difference between V_1 and V_2 to be either amplified or attenuated without loading the signal sources. Only one R_1 exists; therefore, it can be made variable to allow for an easily adjustable gain. Due to their usefulness, instrumentation amplifiers are available in a single package from many manufacturers.

This last circuit illustrates the two primary characteristics of op-amps when used in a closed-loop negative feedback configuration. First, a virtual short exists between the op-amp's input terminals

allowing a high impedance source to set the potential of a circuit node. Second, the op-amp's low output resistance allows op-amp circuits to be connected together without altering their individual transfer functions. These two characteristics greatly simplify the analysis and design of circuits containing many op-amps.

16.1.4 Comparators

If the op-amp is not used in a closed-loop, its output will be either high or low, depending on the polarity of the voltage between its inputs. This appears to make op-amps suitable for comparing two closely spaced signal levels. In practice, it is usually better to use a circuit called a comparator for this application. Comparators are similar to op-amps but have been specifically designed to operate in an open-loop circuit.

The basic comparator compares the voltage levels at its two inputs. If the voltage at the positive input, in^+, exceeds that at the negative input, in^-, the comparator will generate a logic high. If the voltage at in^- exceeds that at in^+, a logic low will be produced. Often, complementary outputs are also provided. The logic high and low levels are set either by the manufacturer or by the user. Typically, the logic levels are compatible with common logic families such as TTL, CMOS, or ECL.

The comparator's primary objective is to provide the correct output level as fast as possible while op-amps are usually used in a closed-loop configuration. To ensure closed-loop stability, most op-amps require some form of compensation (see Section 16.2 and [3]), which reduces their speed and bandwidth. On the other hand, comparators are specifically designed for open-loop operation thereby, making them better suited for high-speed comparisons.

The comparator is usually used as a threshold comparator and often has added hysteresis. When used as a threshold detector, a reference level is applied to one input and the signal is applied to the other input. The choice of inputs determines the output's polarity. Circuits of this type are commonly employed as level detectors and in analog-to-digital converters (see Chapter 20).

In many cases, the input signal contains noise that causes the comparator's output to oscillate as the signal passes the threshold level. To avoid this problem, the use of positive feedback, as depicted in Figure 16.11a, is used to generate hysteresis, as depicted in Figure 16.11b. The resistor R_3 is included to reduce the effects of the comparator's bias currents (see Section 16.2.5). For this circuit, the level that causes the output to go low, V_{INL}, is

$$V_{INL} = V_{REF} + (V_{REF} - V_{OH})/K \tag{16.21}$$

where

V_{OH} is the output-high level

K is the ratio of the resistors depicted in Figure 16.11b

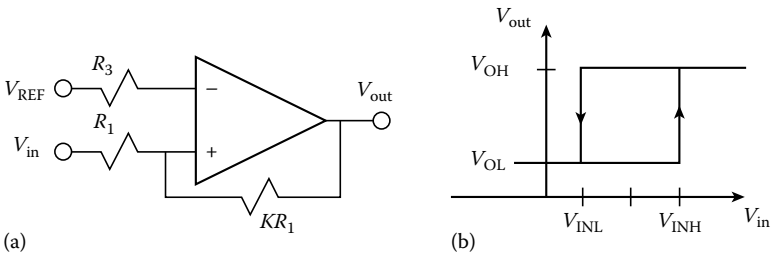


FIGURE 16.11 (a) A comparator with hysteresis and (b) its input–output relationship.

The level that causes the output to go high, V_{INH}, is

$$V_{INH} = V_{REF} + (V_{REF} - V_{OL})/K \tag{16.22}$$

where V_{OL} is the output-low level. By adjusting the ratio K, the amount of noise immunity can be adjusted.

16.1.5 Other Op-Amp Configurations

As mentioned in Section 16.1, the ideal op-amp can be implemented in any of four possible configurations: a voltage amplifier, a current amplifier, a transimpedance amplifier, and a transconductance amplifier. When used in a closed-loop configuration with negative feedback, all four of the ideal op-amps behave the same. In particular, the virtual short circuit between the two inputs remains and the output is unaffected by loading. Consequently, op-amps of all four types exist. Due to practical limitations, some configurations are better suited than others for particular applications.

To illustrate that the ideal op-amp's configuration does not affect the performance of an op-amp circuit, the inverting op-amp circuit in Figure 16.12 has been implemented with a current op-amp. The ideal current op-amp displays zero resistance between its input terminals. The output, which is a current source, has an infinite output resistance. Hence, the current op-amp is the dual of the voltage op-amp. The circuit's output voltage can be found by summing the currents at the op-amp's input. Since there is a physical short between the two input terminals, the current through R_1 will be

$$I_1 = \frac{V_{in}}{R_1} \tag{16.23}$$

while the current through R_2 will be

$$I_2 = \frac{V_{out}}{R_2} \tag{16.24}$$

and due to the op-amp's current gain, A_i, the op-amp's input current will be

$$i_i = \frac{V_{out}}{A_i R_2} \tag{16.25}$$

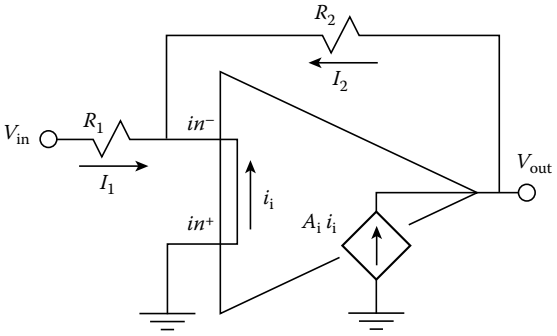


FIGURE 16.12 Inverting amplifier configuration implemented with a current op-amp.

The sum of Equations 16.23 through 16.25 must equal zero. Therefore,

$$\frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1}\left[\frac{1}{1 + 1/A_i}\right] \tag{16.26}$$

which for $A_i = \infty$ results in the same gain as that produced by the inverting configuration implemented with an ideal voltage op-amp. More important, the current through R_1 equals the current through R_2 and the op-amp's input current goes to zero. Hence, even with the physical short between the op-amp's inputs, only a virtual short exists in the closed-loop circuit because no current flows between the

two terminals. At the output, the infinite gain of the ideal current op-amp ensures that the output stays at the value indicated by Equation 16.26 even if a load is added to the output. Consequently, the ideal op-amp's closed-loop behavior is preserved independent of the op amp's configuration.

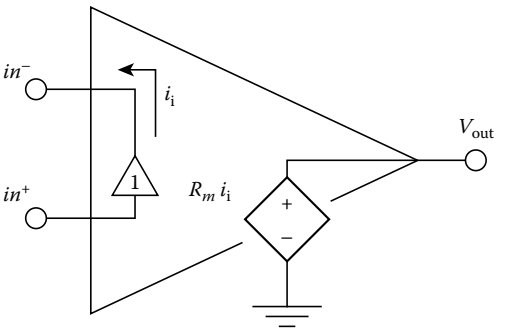


FIGURE 16.13 Practical transimpedance op-amp.

16.1.5.1 Current Op-Amps

Ideally, the current op-amp acts as a current- controlled current source with an infinite current gain. In practical current op-amps, the gain is relatively low. Therefore, the assumption of an infinite gain is unrealistic for most applications. The primary purpose of current op-amps is to boost the output current of a voltage op-amp (see Section 16.2).

16.1.5.2 Transimpedance Op-Amps

The ideal transimpedance op-amp is a current-controlled voltage source with an infinite transimpedance gain. Op amps of this type are commonly referred to as current feedback op-amps. Practical implementations of transimpedance op-amps typically display a much higher speed than most voltage op-amps (see Section 16.3 and [4]). Due to practical considerations, transimpedance op-amps typically have a unity gain buffer between the noninverting and inverting input terminals as illustrated in Figure 16.13.

The added buffer has no effect on the ideal closed-loop performance, but it does increase the input resistance of the noninverting input for practical circuits.

16.1.5.3 Transconductance Op-Amps

The ideal transconductance op-amp is a voltage-controlled current source with an infinite transconductance gain. Practical transconductance op-amps are usually implemented in MOS technologies [5]. MOSFET's themselves are voltage-controlled current sources. Because a practical transimpedance op-amp has a less that infinite output resistance, it is not suited for driving resistive loads. This does not pose a problem because most transimpedance op-amps are used in switched-capacitor circuits, where they are used to drive capacitive loads.

16.1.6 Summary

The ideal op-amp is a high gain circuit element. When used in an open-loop configuration, the op-amp can be used to compare closely spaced signal levels. It is generally much more useful when negative feedback is applied to control its output. With negative feedback, the differential voltage at its input approaches zero and the current between its input approaches zero. This makes it particularly useful for controlling the voltage or current in a circuit without drawing power from the controlling source. The ideal op-amp model, although only approximated in practical op-amps, is very useful for quickly analyzing and understanding the operation of larger circuits. Once the circuit's behavior is understood, the effects of the op-amp's nonidealities can be considered.

References

1. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 3rd edn., New York: Holt, Rinehart & Winston, 1991.
2. D. Sheingold, Op amps and their characteristics, in *Analog Circuit Design*, J. Williams, Ed., New York: Reed, 1991, chap. 30.
3. J. K. Roberge, *Operational Amplifiers*: *Theory and Practice*, New York: John Wiley & Sons, 1975.

4. E. Bruun, Feedback analysis of transimpedance operational amplifier circuits, *IEEE Trans. Circuits Syst.*, 40, 275–278, pt. 1, Apr. 1993.
5. R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits*, New York: John Wiley & Sons, 1986.

16.2 Nonideal Operational Amplifier

Practical op-amps differ significantly from the ideal op-amp. These differences limit the signal levels and range of impedances that can be used in op-amp circuits. Fortunately, the nonidealities are only significant in certain applications. For these applications, circuit design precautions often reduce the effects of the nonidealities to acceptable levels. Alternatively, higher performance op-amps can be used. The op-amp's dominant nonidealities, how they affect various applications, and techniques to compensate for their detrimental effects are discussed in the following sections.

16.2.1 Finite Differential Gain

The op-amp's most critical nonideality is its finite gain. Unlike the infinite gain of the ideal op-amp, the gain of a practical op-amp is typically large at dc and decreases at high frequencies. Most op-amps are internally compensated for a frequency dependent gain of the form

$$\frac{v_o}{v_i} = \frac{A_O}{1 + j\omega(A_O/\omega_t)} \tag{16.27}$$

where
A_O is the dc differential open-loop gain
ω is the frequency
ω_t is the op-amp's unity gain frequency

Unity gain frequencies are typically in the MHz range. At low frequencies, the op-amp's gain simply becomes A_O. At high frequencies, the op-amp's gain can be approximated as

$$\frac{v_o}{v_i} = \frac{\omega_t}{j\omega} \tag{16.28}$$

Further details on the amp's high-frequency behavior will be discussed in Section 16.3. The dc gain, A_O, is typically quite large, hence it is usually expressed in decibels (dB). A_O ranges from 40 dB for high-speed op-amps to 120 dB for high-precision op-amps. General-purpose op-amps usually have differential gains in the 100 dB region. Because A_O is subject to wide variations, manufacturers usually specify a minimum and typical value.

The op-amp's finite A_O reduces the closed-loop gain of most op-amp circuits. To illustrate the problem, the noninverting and inverting amplifiers shown in Figure 16.14 can be analyzed assuming a finite op-amp gain. The finite gain results in a nonzero differential voltage at the op-amp's input:

$$v_d = \frac{v_o}{A_O} \tag{16.29}$$

Consequently, the voltage at the inverting input is not equal to that at the noninverting input. Hence, the voltage across R_1 and the current through it is changed. By equating the current in R_1 and R_2, it is seen that the noninverting amplifier's gain is reduced to

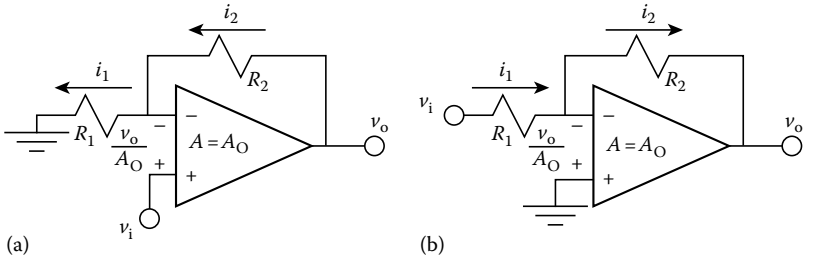


FIGURE 16.14 Analysis of the noninverting (a) and inverting (b) configurations for an op-amp with finite gain A_O.

$$\frac{v_o}{v_i} = \frac{1 + R_2/R_1}{1 + (1 + R_2/R_1)/A_O} \tag{16.30}$$

while the inverting amplifier's gain is reduced to

$$\frac{v_o}{v_i} = \frac{-R_2/R_1}{1 + (1 + R_2/R_1)/A_O} \tag{16.31}$$

For large values of A_O, Equations 16.30 and 16.31 reduce to the gains that would be obtained with an ideal op-amp (i.e., $v_o/v_i = 1 + R_2/R_1$ and $v_o/v_i = -R_2/R_1$, respectively).

It is only when A_O and the desired closed-loop gain become comparable that the op-amp's finite gain leads to a significant reduction in the closed-loop gain. To illustrate this problem, the gain deviation for the noninverting amplifier versus $(1 + R_2/R_1)/A_O$ is plotted in Figure 16.15. Because A_O is subject to wide variations, only gains that are at least 100× lower than A_O should be used to ensure a well-controlled gain. Consequently, the op-amp's finite open-loop differential gain places an upper limit on the closed-loop gain that can be provided accurately.

16.2.2 Output Saturation

Although op-amps can provide high gains, the op-amp's maximum output voltage and current are limited. The maximum output voltage is limited by the op-amp's supply voltages, while the maximum output current is usually limited by the op-amp's allowable power dissipation.

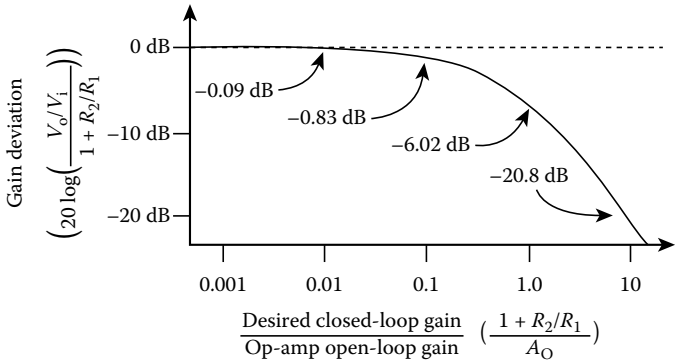


FIGURE 16.15 Gain deviation for the noninverting configuration caused by a finite A_O.

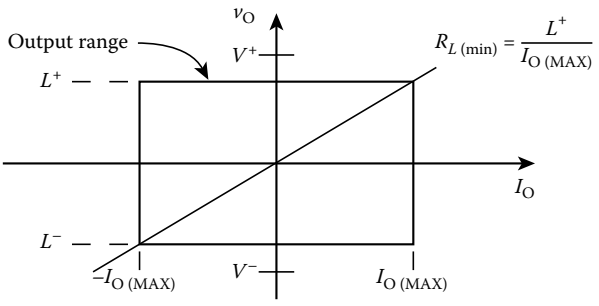


FIGURE 16.16 Voltage and current limitations on the op-amp's output signal swing.

The op-amp, similar to any electronic amplifier, requires a dc power supply. Most op-amps require a positive, V^+, and a negative, V^-, power supply. Because V^+ and V^- are typically the same size and of opposite polarity, they are referred to as dual or split supplies. Usually, the op-amp has no connection to ground. The supply voltage typically ranges from ±5 to ±18 V, with ±15 V being the most common. Special-purpose op-amps include low-voltage/low-power op-amps for use with lower supply voltages, high-voltage op-amps for use with supply voltages beyond ±18 V and single supply op-amps for use with a single supply.

Regardless of the op-amp type, the maximum L^+ and minimum L^- output voltages cannot exceed the supply voltages. Typically, the output saturates within 1–3 V of the supplies as shown in Figure 16.16. Low-voltage op-amps often feature a ''rail-to-rail'' output swing that allows the output signal to extend to both V^+ and V^-. Due to the op-amp's limited output swing, the input signal must be kept small enough to avoid distortion caused by clipping the output signal.

The second limitation on the op-amp's output signal is the op-amp's maximum output current specification. This limitation is determined by the maximum allowed power dissipation of the op-amp. If the power dissipation limit is exceeded, the resulting temperature rise can damage the device. The worst-case power dissipation usually occurs when the op-amp has a load resistance of zero (i.e., the output is shorted to ground). In this situation, the full supply voltage appears across the op-amp's output stage and the power dissipation is

$$P_{\text{Disp}} = V^+ I_O + P_Q \tag{16.32}$$

P_Q is the op-amp's quiescent power dissipation and usually much smaller than $V^+ I_O$. Hence, to avoid an excessive temperature rise, I_O must be limited to a safe value, $I_{O(MAX)}$. Many op-amps are designed with short-circuit protection that limits $I_{O(MAX)}$ to a safe level. For general-purpose op-amps, $I_{O(MAX)}$ is in the 20 mA range.

The limitations imposed by the combination of L^+, L^-, and $I_{O(MAX)}$ are illustrated in Figure 16.16. For loads below $R_{L(min)}$:

$$R_{L(\text{min})} = \frac{L^+}{I_{O(MAX)}} \tag{16.33}$$

the op-amp's output swing will be limited by the op-amp's current-limiting circuitry. For large values of R_L, the signal swing will be limited by the op-amp's maximum and minimum output voltages. Therefore, to ensure that the signal peaks are not clipped, the equivalent load resistance seen by the op-amp must be greater than $R_{L(min)}$ and the amplifier's input signal must be small enough to ensure the output voltage will not exceed either L^+ or L^-.

16.2.3 Offset Voltage

For an ideal op-amp, a zero differential input voltage produces a zero output voltage. For a practical op-amp, as illustrated in Figure 16.17, a zero differential input voltage will, in general, produce a nonzero

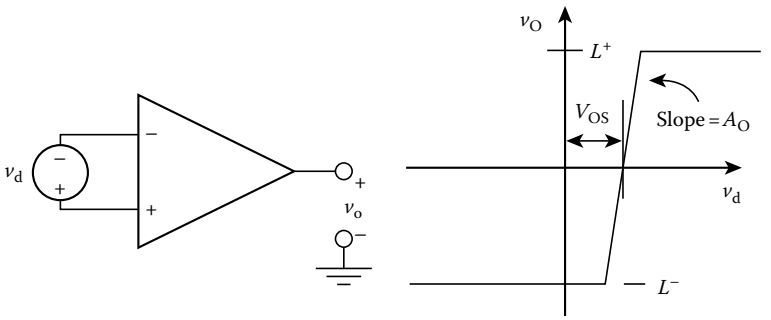


FIGURE 16.17 Transfer function of a practical op-amp illustrating A_O, L^+, L^-, and V_{OS}.

output. Due to the op-amp's high gain, the output will usually saturate at either L^+ or L^- if no feedback is applied. To obtain a zero output voltage, a nonzero input voltage, defined as the input offset voltage, V_{OS}, must be applied between the input terminals.

V_{OS} is generally quite small. It arises from small mismatches in the devices used in the op-amp's input stage and circuit asymmetries. General-purpose op-amps have offset voltages in the 0.1–10 mV range. Typically, op-amps with FET input devices will have higher offset voltages than op-amps with bipolar input devices. If a very low V_{OS} is required, precision and low offset voltage op-amps are available with offset voltages in the μV range. The offset voltage is not constant, it is subject to drift with time and changes in temperature. Consequently, its effects will be evident in most circuits.

To analyze the effect of V_{OS}, a voltage source of unknown polarity, equal to V_{OS}, is connected to one of the inputs of an ideal op-amp. Then, using superposition, the circuit's output voltage due to V_{OS} and the input voltage can be determined. Because superposition applies, the effect of V_{OS} on both the inverting and noninverting configurations is identical. As illustrated in Figure 16.18a, in which a practical op-amp has been replaced with an ideal op-amp in series with V_{OS}, and v_i set to zero, the output v_o, due to V_{OS} alone (i.e., the output offset voltage) is

$$v_o = V_{OS}\left(\frac{1 + R_2}{R_1}\right) \tag{16.34}$$

where it is evident that large dc gains result in a large output offset voltage. If dc gain is not required, a capacitor can be placed in series with R_1 to reduce the dc gain to unity. The resulting output offset voltage then becomes V_{OS}. Another group of circuits affected by V_{OS} are integrating circuits such as the one in Figure 16.18b. If R_f is not included, V_{OS} causes a dc current V_{OS}/R_1 to flow through R_1 and be integrated

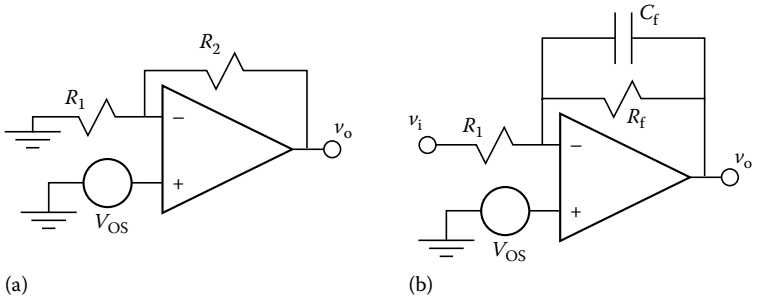


FIGURE 16.18 Some circuits affected by the op-amps offset voltage. (a) The inverting and noninverting configurations. (b) Integrator circuits.

on C_f, thereby causing the output voltage to saturate. Adding R_f limits the dc gain and hence, limits the output offset voltage.

In situations where the smallest possible V_{OS} is required, low V_{OS} op-amps can be used. Alternatively, many op-amps are provided with one or two terminals to which an offset nulling circuit can be attached. The op-amp's V_{OS} can then be trimmed to zero. Because the trimming can only be done for one temperature, V_{OS} will still drift due to temperature and time and hence will limit the circuit's dc accuracy.

16.2.4 Finite Common-Mode Rejection Ratio and Power-Supply Rejection Ratio

If an op-amp's inputs are shorted together, as depicted in Figure 16.19, variations in any one of the three voltages; v_{cm}, the common-mode input voltage, V^+, the positive supply voltage, or V^-, the negative supply voltage, should not affect the output voltage. Nevertheless, if all three voltages are changed by the same amount, it is evident that the output voltage must also change by the same amount [3]. Hence, the op-amp's output voltage will be affected by changes in v_{cm}, V^+ and V^-. The relationship between changes in v_{cm} and v_o is usually characterized by the common-mode rejection ratio (CMRR). The effects of changes in the positive and negative supplies on v_o are usually referred to as the power-supply rejection ratios, $PSRR^+$ and $PSRR^-$, respectively.

16.2.4.1 Common-Mode Rejection Ratio

The op-amp has two input terminals, therefore, two signal types exist: differential signals and common-mode signals. Referring to Figure 16.19, the differential signal v_d is the difference between the two input voltages, v_{in}^+ and v_{in}^-:

$$v_d = v_{in}^+ - v_{in}^- \tag{16.35}$$

while the common-mode signal v_{cm} is their average:

$$v_{cm} = \frac{(v_{in}^+ + v_{in}^-)}{2} \tag{16.36}$$

Ideally, the op-amp rejects (i.e., does not respond to) common-mode signals. For practical op-amps, changes in v_{cm} lead to changes in v_o, resulting in a common-mode gain A_{cm}:

$$A_{cm} = \frac{v_o}{v_{cm}} \tag{16.37}$$

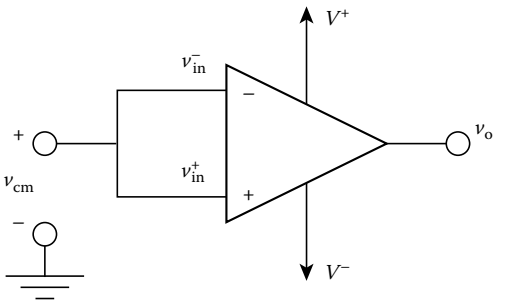


FIGURE 16.19 Op amp with its various input voltages.

Over a specified range, known as the common-mode range, A_{cm} is relatively small. Beyond the common-mode range, A_{cm} rises rapidly and the op-amp ceases to function properly. Typically, the common-mode range does not extend to either the positive or negative supply. Single-supply op-amps, however, are usually designed to have a common-mode range that extends down to and often slightly below the lower supply. Within the common-mode range, A_{cm} is usually specified by the CMRR

$$\text{CMRR} = \frac{A_o}{A_{cm}} \tag{16.38}$$

The CMRR is usually expressed in dB

$$\text{CMRR} = 20\log\left|\frac{A_o}{A_{cm}}\right| \tag{16.39}$$

and ranges from 60 to over 120 dB. An alternate interpretation of the CMRR is the ratio of a change in v_{cm} to the resulting change in the op-amp's V_{OS}.

$$\text{CMRR} = \frac{V_{cm}}{V_{OS}} \tag{16.40}$$

The two interpretations of CMRR are equivalent.

A finite CMRR affects those circuits for which a sizable v_{cm} is applied to the op-amp. Hence, the inverting configuration, with a virtual ground at its input, is unaffected by the common-mode gain. On the other hand, circuits such as the noninverting configuration and the differential configuration in Figure 16.20, have a nonzero v_{cm} and hence display common-mode problems.

The effects of a common-mode signal can be determined as follows. Because a finite CMRR can be interpreted as a change in the op-amp's V_{OS}, due to the presence of a v_{cm}, the V_{OS} due to the nonzero v_{cm} and the finite CMRR can be found as

$$V_{OS} = \frac{v_{cm}}{\text{CMRR}} \tag{16.41}$$

Then, the effect of the resulting V_{OS} can be found by analyzing the op-amp assuming the common-mode gain is zero. For example, the noninverting circuit in Figure 16.20a, has a v_{cm} approximately equal to v_i, which leads to an equivalent V_{OS} of

$$V_{OS} = \frac{v_i}{\text{CMRR}} \tag{16.42}$$

and a total output voltage of

$$v_o = v_i\left(\frac{1+1}{\text{CMRR}}\right)\left(\frac{1+R_2}{R_1}\right) \tag{16.43}$$

Consequently, the CMRR leads to a gain error. For the differential configuration in Figure 16.20b, a finite CMRR leads to an output voltage of the form

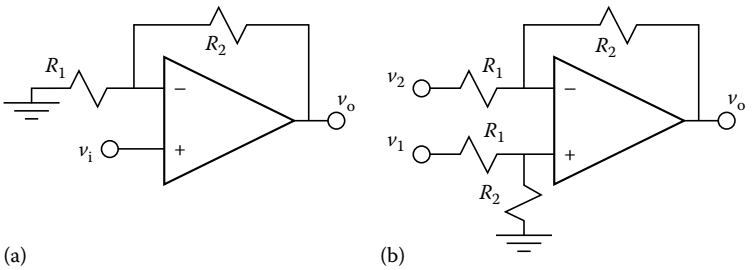


FIGURE 16.20 Some op-amp circuits affected by a finite CMRR. (a) The noninverting configuration. (b) The differential configuration.

$$v_o = \frac{R_2}{R_1}(v_1 - v_2) + \frac{R_2}{R_1}\frac{v_1}{\text{CMRR}} \tag{16.44}$$

where if $(v_1 + v_2)/2 \gg v_1 - v_2$, v_i will be approximately equal to the signal's common-mode voltage and it can be seen that the differential amplifier responds to both the differential and common-mode components of the signal. Consequently, the op-amp's CMRR can lead to problems if both of the op-amp's terminal potentials vary with the input signal.

16.2.4.2 Power-Supply Rejection Ratio

Ideally, changing either or both of an op-amp's power supplies should not affect the op-amp's performance. Practical op-amps though display a power-supply dependent gain and at higher frequencies, power-supply fluctuations are coupled into the op-amp's signal path, leading to variations in the output voltage. These problems can be characterized as an equivalent gain, A_{V+} and A_{V-} from V^+ and V^- terminals, respectively, to the output. Alternatively, the variation can be characterized as a power-supply dependent variation in the op-amp's equivalent input offset voltage. Because the op-amp is only supposed to amplify differential signals applied to its input, and reject signals applied to the power supplies, it is desirable to have $A_O \gg A_{V+}$ and $A_O \gg A_{V-}$. To measure this performance, the PSRRs are used

$$\text{PSRR}^+ = \frac{A_O}{A_{V+}} \tag{16.45}$$

$$\text{PSRR}^- = \frac{A_O}{A_{V-}} \tag{16.46}$$

Usually, the PSRRs are expressed in dB

$$\text{PSRR}^+ = 20\log\left|\frac{A_O}{A_{V+}}\right| \tag{16.47}$$

$$\text{PSRR}^- = 20\log\left|\frac{A_O}{A_{V-}}\right| \tag{16.48}$$

PSRRs of 60–100 dB are common at dc. At higher frequencies, the PSRR decreases.

A noninfinite PSRR may pose a problem if there are variations in the power-supply voltages. Such variations can arise from either the ripple voltage of the supply itself or from large variations in the current being drawn from the supply. To reduce unwanted variations in the op-amp's output voltage, either an op-amp with a better PSRR can be selected or power-supply decoupling capacitors can be used. If decoupling capacitors are used, they should be placed as close as possible to the op-amp's power-supply terminals.

16.2.4.3 Finite Input Impedance and Nonzero Output Impedance

Unlike the ideal op-amp, practical op-amps exhibit a finite input impedance and a nonzero output impedance. The input impedance is composed of a differential and a common-mode component as illustrated in Figure 16.21. R_{id} and C_{id} represent the equivalent resistance and capacitance seen between the op-amp's two input terminals. R_{icm} and C_{icm} represent the total resistance and capacitance to ground that would be seen by a common-mode signal applied to both input terminals. R_{id} ranges from 100 kΩ to over 100 MΩ. The higher differential input resistances are found in op-amps employing FET input stages. R_{icm} is typically two orders of magnitude higher than R_{id}. The input capacitances are generally in the picofarad range. The output resistance R_o is usually in the 50–100 Ω range. Generally, the use of

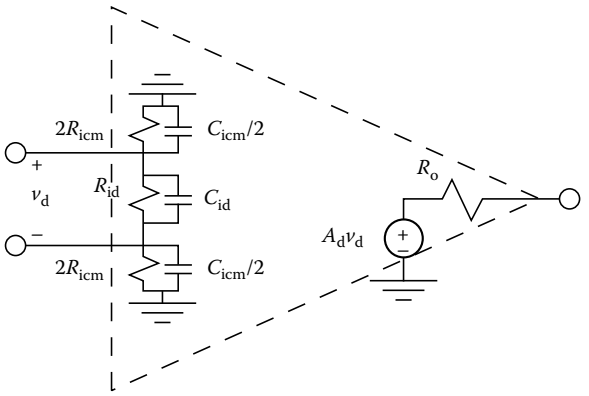


FIGURE 16.21 Input and output impedance of practical op-amp.

negative feedback reduces the effects of these impedances to levels where they can be neglected. Nevertheless, problems can arise in some applications.

The noninverting configuration is often used as a buffering amplifier due to its high input resistance. At low frequencies, the op-amp's input capacitances can be neglected and the negative feedback provided by the op-amp's high gain keeps the voltage across and the current through R_{id} negligible. Hence, the effective input resistance is approximately $2R_{icm}$. At high frequencies, C_{id} shorts the input and the op-amp's decreasing gain causes the voltage across and hence, the current through R_{id} to increase resulting in a significantly decreased input impedance.

Most op-amps employ shunt sampling negative feedback, so the op-amp's effective output resistance is reduced. Hence, even relatively high values of R_o can be tolerated in most circuits. Nevertheless if the circuit is used to drive a capacitive load, problems can arise. Because the op-amp's gain decreases at higher frequencies (see Section 16.2.1), the amount of negative feedback also decreases leading to an output impedance that appears inductive (i.e., increases with frequency). This can be found by analyzing the output impedance of the circuit in Figure 16.22a. Approximating the op-amp's gain as (see Section 16.2.1)

$$A = \frac{w_t}{jw} \tag{16.49}$$

and assuming that $R_o \ll R_1 + R_2$, the output impedance becomes

$$Z_{out} \approx R_o \parallel jwL_{eff} \tag{16.50}$$

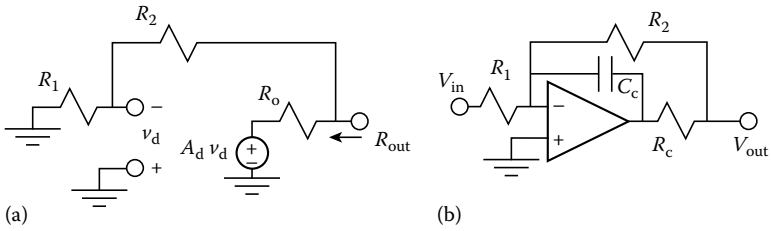


FIGURE 16.22 (a) Determining R_{out} for the inverting and noninverting configurations and (b) compensating for large capacitive loads.

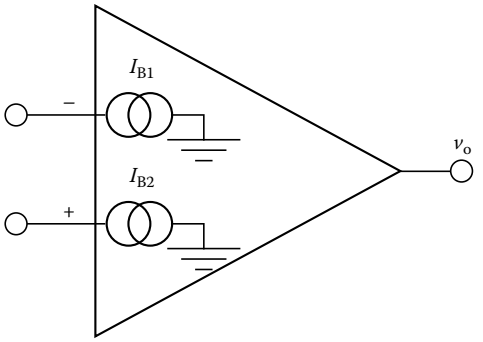


FIGURE 16.23 Op-amp showing its input bias currents.

where

$$L_{\text{eff}} = \frac{R_o}{w_t}\left(\frac{R_1 + R_2}{R_1}\right) \tag{16.51}$$

Consequently, if the circuit drives a large capacitor, it may become unstable due to the presence of L_{eff}. To reduce this problem, a compensation network as illustrated in Figure 16.22b is commonly used.

16.2.5 Input Bias Currents

When operating, current flows in each of the op-amp's input leads as depicted in Figure 16.23. These input currrents, which are due to the internal structure of the op-amp, give rise to errors in many circuits and prevent the practical realization of some circuit configurations. Because these currents cannot be avoided, they should be considered when designing op-amp circuits.

The input currents are determined by the devices used to implement the amplifier's input stage. If BJTs are used, their base currents determine the input currents. If FETs are used, the input currents are due to the gate leakage current. In either case, the average of the two input currents, I_{B1} and I_{B2}, is referred to as the input bias current, I_B:

$$I_B = \frac{(I_{B1} + I_{B2})}{2} \tag{16.52}$$

where I_{B1} and I_{B2} are the input currents that cause the op-amp's output to go to zero with a zero common-mode input voltage. I_B can range from 0.1 pA to 1 μA, which is much higher than would be expected based on the amplifier's finite input resistance alone. Typically, op-amps with FET inputs display a much lower I_B than their bipolar counterparts. Due to mismatches, I_{B1} and I_{B2} are rarely equal. Their difference, referred to as the input offset current, I_{OS}, is defined as

$$I_{OS} = |I_{B1} - I_{B2}| \tag{16.53}$$

I_{OS} is typically an order of magnitude lower than I_B. Therefore, in all but the most critical applications, I_{OS} can be neglected.

The effects of nonzero bias currents on both the inverting and noninverting configurations are illustrated in Figure 16.24a. If I_B is not compensated for, both op-amp configurations will display an output voltage of

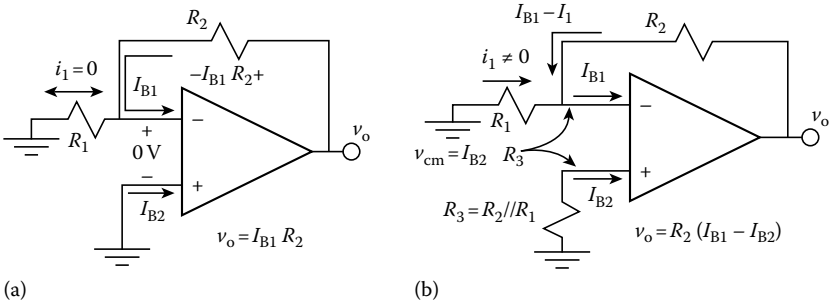


FIGURE 16.24 Analysis of the inverting and noninverting configurations with nonzero I_B. (a) Without bias current compensation and (b) with bias current compensation.

$$v_o = I_{B1}R_2 \tag{16.54}$$

This voltage can be reduced either by reducing R_2 or by selecting an op-amp with lower bias currents. Alternatively, a resistor R_3 equal to the parallel equivalent of R_2 and R_1 can be included in the positive terminal's lead, as illustrated in Figure 16.24b. This added resistor causes the voltage at the op-amp's input to be equal to

$$v_{cm} = -I_{B2}R_1 \parallel R_2 \tag{16.55}$$

which, if two bias currents are equal ($I_{OS} = 0$), causes v_o to be zero. For the practical case of a nonzero offset current, v_o becomes

$$v_o = I_{OS}R_2 \tag{16.56}$$

Because I_{OS} is usually much lower than I_B, the error is greatly reduced. It is important to note that I_B is a dc current. Hence, R_3 should be equal to the equivalent dc resistance seen by the op-amp's negative terminal.

All op-amps require an I_B for proper operation, therefore, a dc path between each input terminal and ground must be provided. For example, the ac coupled buffer of Figure 16.25a requires the addition of R_i to provide a path for I_{B2}. Unfortunately, R_i decreases the buffer's input resistance. In Figure 16.25b, the difference between I_{B1} and $I_{B2}R_3/R_1$ will flow to C_f, quickly leading to saturation of op-amp's output at either L^+ or L^-. By adding R_f, a dc path for the difference current is provided. Unfortunately, R_f makes the integrator nonideal at low frequencies. Consequently, the op-amp's bias currents restrict the dc accuracy and the frequency range of applications for op-amp circuits.

16.2.6 Electrical Noise

Similar to any electronic component, op-amps generate noise that can degrade the system's signal-to-noise ratio (SNR). The amplifier's noise is characterized by the equivalent noise sources in Figure 16.26a. $\overline{e_n}$ is the equivalent input noise voltage density and is expressed in $nV/\sqrt{Hz}$. $\overline{i_{n1}}$ and $\overline{i_{n2}}$ are the equivalent input noise current density and are expressed in $pA/\sqrt{Hz}$. Usually, $\overline{i_{n1}}$ and $\overline{i_{n2}}$ have the same magnitude and are both referred to as $\overline{i_n}$. The typical behavior of $\overline{e_n}$ and $\overline{i_n}$ are illustrated in Figure 16.26b. At higher frequencies, e_n and i_n are independent of frequency (i.e., white noise). In this range, values of $\overline{e_n}$ and $\overline{i_n}$ range from 50 $nV/\sqrt{Hz}$ and $0.6pA/\sqrt{Hz}$, respectively, for general-purpose op-amps to $2nV/\sqrt{Hz}$ and $10fA/\sqrt{Hz}$ for ultra-low-noise op-amps. At low frequencies, op-amps display noise that increases

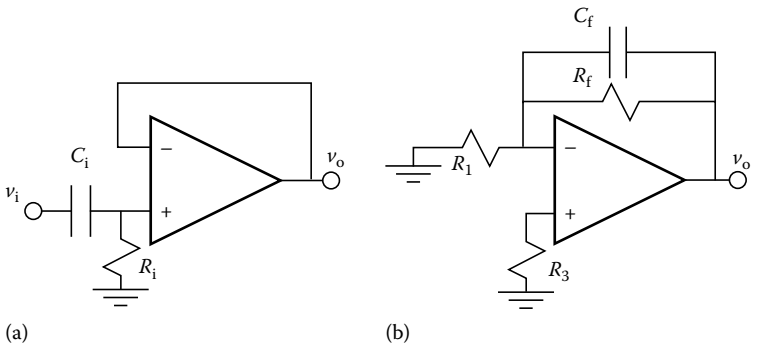


FIGURE 16.25 Some op-amp circuits affected by the op-amp's bias current. (a) The ac coupled buffer. (b) The inverting integrator.

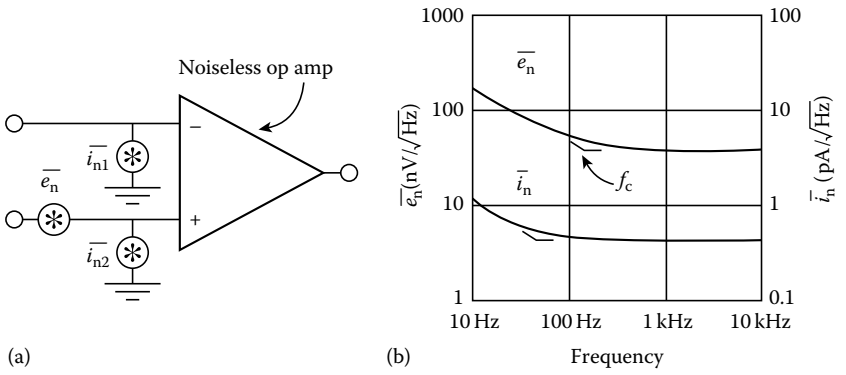
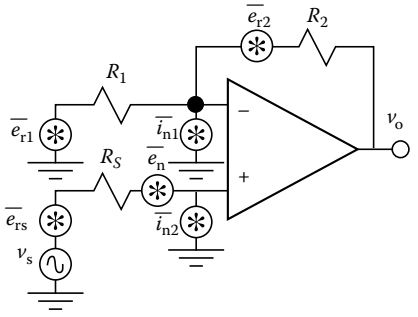


FIGURE 16.26 (a) Noise model of the op-amp. (b) Input noise voltage and current densities versus frequency.



Source	Output × $\sqrt{\text{Hz}}$
$\overline{e_{r1}}$	$\overline{e_{r1}}\,(R_2/R_1)$
$\overline{e_{r2}}$	$\overline{e_{r2}}$
$\overline{i_{n1}}$	$\overline{i_{n1}}\,R_2$
$\overline{i_{n2}}$	$\overline{i_{n2}}\,R_S\,(1+R_2/R_1)$
$\overline{e_n}$	$\overline{e_n}\,(1+R_2/R_1)$
$\overline{e_{rs}}$	$\overline{e_{rs}}\,(1+R_2/R_1)$

FIGURE 16.27 Op-amp circuit showing its noise sources.

with decreasing frequency (i.e., $1/f$ noise). To specify the low frequency noise, a plot such as the one in Figure 16.26b may be provided. In some cases, only the noise corner frequency f_c (see Figure 16.26b) may be specified.

To determine the effects of the op-amp's noise, the circuit in Figure 16.26 can be used. The noise sources are generally uncorrelated, so the total noise is the square root of the sum of the square of each noise source acting independently. Therefore, the first step is to identify all the noise sources, as illustrated in Figure 16.27. Each resistor may be modeled as a noiseless resistor in series with a noise voltage density of

$$\overline{e_r} = \sqrt{4kTR}\Big/\sqrt{\text{Hz}} \tag{16.57}$$

where

k is the Boltzmann constant
T is the absolute temperature

Each noise source then gives rise to the output values shown in the table (see Figure 16.27). The designer is free to choose R_1 and R_2, so the noise sources $\overline{e_n}$, $\overline{e_{rs}}$, and $\overline{i_{n2}}$ typically determine the circuit's total output noise voltage. The total rms white-noise voltage at the output is

$$\overline{v_{\text{output}}} = \sqrt{\frac{\pi}{2}\text{BW}}\sqrt{\overline{e_n^2}\left(1+\frac{R_2}{R_1}\right)^2+\overline{e_{rs}^2}\left(1+\frac{R_2}{R_1}\right)^2+\overline{i_{n2}^2}R_S\left(1+\frac{R_2}{R_1}\right)^2} \tag{16.58}$$

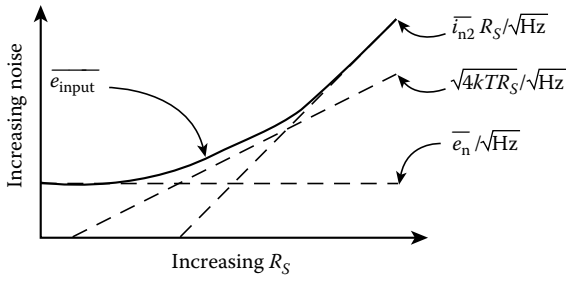


FIGURE 16.28 Effect of R_S on the total noise for the circuit of Figure 16.26a.

where BW is the op-amp's closed-loop bandwidth (see Section 16.3). The factor $\pi/2$ converts the closed-loop bandwidth to the noise equivalent bandwidth, assuming the op-amp is characterized by a single pole. The equivalent voltage white-noise density at the input is

$$\overline{e_{input}} = \sqrt{\overline{e_n^2} + 4kTR_S + \overline{i_{n2}^2}R_S^2}\Big/\sqrt{Hz} \tag{16.59}$$

The relative importance of each of these three factors depends on the value of R_S as shown in Figure 16.28. For low values of R_S, $\overline{e_n}$ dominates. At high values of R_S, $\overline{i_{n2}}$ dominates. In the middle region, R_S dominates if $\sqrt{4kt} > \sqrt{i_n e_n}$.

Two measures are used to specify the circuit's noise performance: the SNR and the noise figure (NF). The SNR is the ratio of the signal power to the total noise power, which assuming only white noise is present, can be expressed as

$$SNR = 20\log\left|\frac{v_s}{\overline{e_{input}}\sqrt{(\pi BW)/2}}\right| \tag{16.60}$$

The second measure, NF, expresses the increase in noise due to the amplifier over that due to the source resistance alone:

$$NF = 10\log\left|\frac{\overline{e_n^2} + \overline{e_{rs}^2} + \overline{i_{n1}^2 R_S^2}}{\overline{e_{rs}^2}}\right| \tag{16.61}$$

Although a low NF is desirable, it is usually more important to minimize the total noise to achieve the highest possible SNR.

16.2.7 Summary

Practical op-amps suffer from a wide range of nonidealities. The dominant effect of these nonidealities is to limit the range of applications for which an op-amp can be used. Problems such as the offset voltage, the CMRR, the bias currents, and the electrical noise will limit the accuracy of op-amp circuits. The op-amp's finite gain and saturation levels will limit the maximum controllable gain of an op-amp circuit. Factors such as the op-amp's saturation limits and its input and output impedances will limit the range of impedances that can be buffered by or driven by an op-amp. In many cases, circuit techniques or special-purpose op-amps can be used to reduce the detrimental effects of the op-amp's nonidealities.

References

1. E. J. Kennedy, *Operational Amplifier Circuits: Theory and Applications*, New York: Holt, Rinehart and Winston, 1988.
2. J. K. Roberge, *Operational Amplifier: Theory and Practice*, New York: Wiley, 1975.
3. E. Säckinger, J. Goette, and W. Guggenbühl, A general relationship between amplifier parameters, and its application to PSRR improvement, *IEEE Trans. Circuits Syst.*, 38, 1173–1181, Oct. 1991.
4. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 3rd edn., New York: Holt, Rinehart and Winston, 1991.

16.3 Frequency- and Time-Domain Considerations

One of the most important limitations of practical op-amps is gain roll-off with frequency. This limitation affects both the frequency-domain and the time-domain behavior of circuits built around op-amps. We have linear effects, such as finite small-signal bandwidth and nonzero rise time, and nonlinear effects, such as slew-rate (SR) limiting and finite full-power bandwidth (FPBW). Additional effects are the settling time and intermodulation distortion. We discuss these limitations both for voltage-mode and current-mode op-amps [1].

16.3.1 Voltage-Mode Op-Amps

Conventional op-amps, the most popular representative of which is without doubt the 741 type, are voltage-mode amplifiers because in order to produce an output they require an input imbalance of the voltage type. Consequently, when a negative feedback loop is created around the op-amp, the signal returned to the input must be in the form of a voltage.

16.3.2 Block Diagram

Shown in Figure 16.29 is a simplified circuit diagram [2] that can be used to describe a wide variety of practical voltage-mode op-amps, including the popular 741. As illustrated in block-diagram form in Figure 16.30, the circuit is made up of three stages.

The *input stage* consists of transistors Q_1 through Q_4, whose function is to sense any imbalance between the inverting and noninverting input voltages V_n and V_p, and convert it to a single-ended output current I_{o1}. This stage is also designed to provide high input impedance and draw negligible input currents.

Q_1 and Q_2 form a differential pair whose task is to split the bias current I_A into two currents I_1 and I_2 in amounts controlled by the imbalance between V_n and V_p. If this imbalance is sufficiently small, we can write $I_1 - I_2 = g_{m1}(V_p - V_n)$, where g_{m1} is the transconductance of Q_1 and Q_2.

Ignoring transistor base currents, we have $I_3 = I_1$. In response to current I_3, Q_3 develops a base–emitter voltage drop that is then applied to Q_4, forcing the latter to draw the same amount of current as the former, or $I_4 = I_3$. For obvious reasons, Q_3 and Q_4 are said to form a current mirror. Summing currents, we obtain $I_{o1} = I_2 - I_4 = I_2 - I_3 = I_2 - I_1$, or

$$I_{o1} = -g_{m1}(V_p - V_n) = -g_{m1}V_d \tag{16.62}$$

where $V_d = V_p - V_n$ is called the differential input voltage.

The *intermediate stage* consists of Darlington pair $Q_5 - Q_6$ and frequency-compensation capacitance C_c. Its function is to provide additional gain as well as to introduce a dominant pole in the open-loop

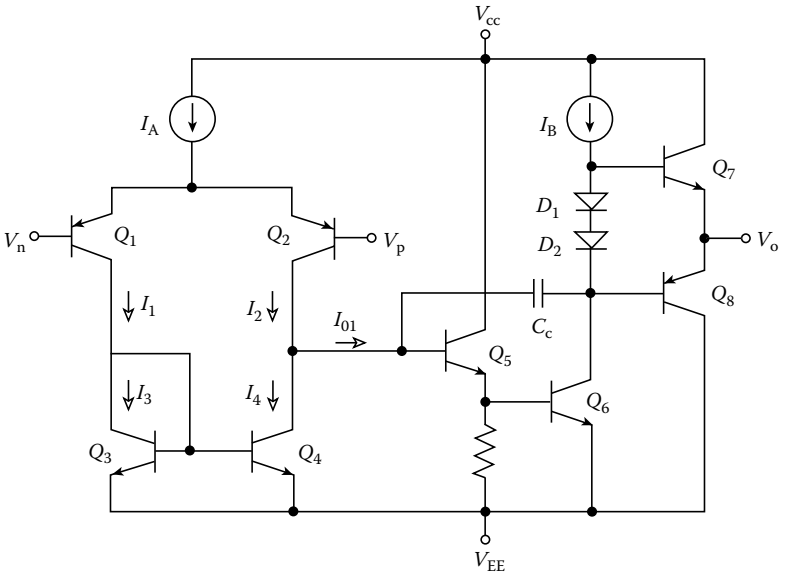


FIGURE 16.29 Simplified circuit diagram of a voltage-mode op-amp.

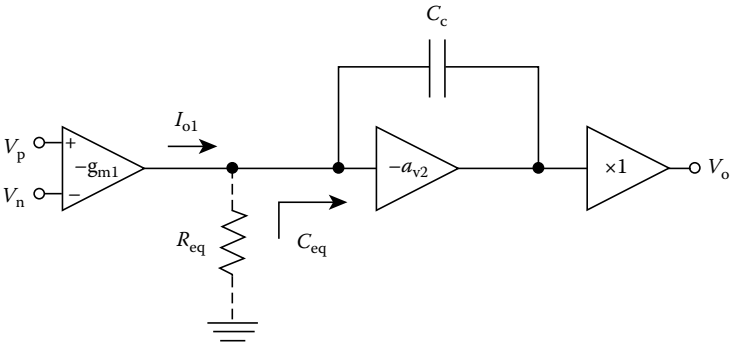


FIGURE 16.30 Voltage-mode op-amp block diagram.

response of the amplifier. Denoting the net equivalent resistance and capacitance between the input node of this stage and ground as R_{eq} and C_{eq}, the pole frequency is

$$f_{OL} = \frac{1}{2\pi R_{eq} C_{eq}} \tag{16.63}$$

By the Miller effect we have $C_{eq} = (1 + a_{v2})C_c$, where $-a_{v2}$ is the voltage gain of the Darlington pair.

The *output stage* consists of emitter–followers Q_7 and Q_8, and biasing diodes D_1–D_2. Though the voltage gain of this stage is only unity, its current gain can be fairly high, indicating that this stage acts as a power amplifier. Its function is also to provide a low output impedance. Q_7 and Q_8 are referred to as a push–pull pair because in the presence of an output load, Q_7 sources (or pushes) current to the load, and Q_8 sinks (or pulls) current from the load.

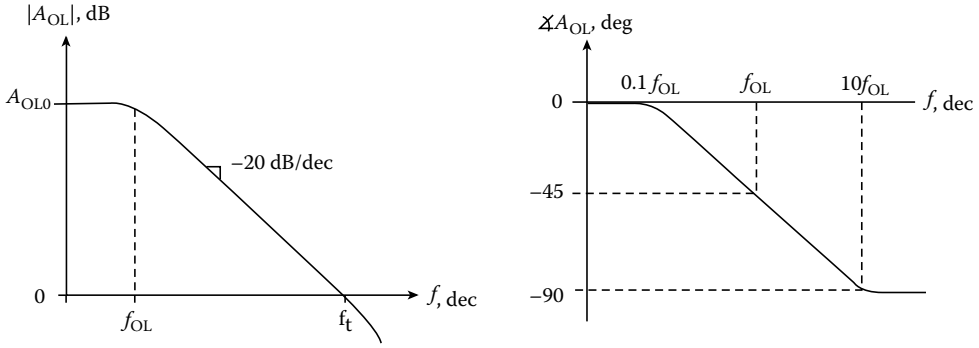


FIGURE 16.31 Frequency plots of the open-loop gain.

The small-signal transfer characteristic of the op-amp is

$$V_o = A_{OL}(jf)V_d \tag{16.64}$$

where $A_{OL}(jf)$, called the *open-loop voltage gain*, is a complex function of frequency f, and $j = \sqrt{-1}$ is the imaginary unit. With dominant pole compensation, this function can be approximated as

$$A_{OL}(jf) = \frac{A_{OL0}}{1 + jf/f_{OL}} \tag{16.65}$$

where A_{OL0} and f_{OL} are, respectively, the *dc value* and *bandwidth* of $A_{OL}(jf)$. For the circuit shown, $A_{OL0} = g_{m1}R_{eq}a_{v2}$. As an example, the popular 741 op-amp has $A_{OL0} \simeq 2 \times 10^5$ V/V and $f_{OL} \simeq 5$ Hz. Figure 16.31 shows the Bode plots of $A_{OL}(jf)$. We make the following observations.

1. For $f \ll f_{OL}$ we have $|A_{OL}(jf)| \simeq A_{OL0}$ and $\measuredangle A_{OL}(jf) \simeq 0°$ indicating an approximately constant gain and negligible delay.
2. For $f = f_{OL}$ we have $|A_{OL}(jf)| = A_{OL0}/\sqrt{2}$ and $\measuredangle A_{OL}(jf) = -45°$. Rewriting as $|A_{OL}(jf)|_{dB} = 20 \log_{10}|A_{OL}(jf)| = |A_{OL0}|_{dB} - 3$ dB explains why f_{OL} is also referred to as the -3 dB frequency or the half-power frequency of the open-loop response.
3. For $f \gg f_{OL}$ gain rolls off with frequency at a constant rate of -20 dB/dec, and it can be approximated as $|A_{OL}(jf)| \simeq A_{OL0}/(f/f_{OL})$ and $\measuredangle A_{OL}(jf) \simeq -90°$ Rewriting as

$$|A_{OL}(jf)| \times f \simeq A_{OL0} \times f_{OL} = f_t \tag{16.66}$$

indicates that in the roll-off region the op-amp exhibits a constant gain–bandwidth product (constant GBP). Increasing frequency by a given amount causes gain to decrease by the same amount. The frequency $f_t = A_{OL0}f_{OL}$ at which gain drops to 0 dB is aptly called the transition frequency. For the 741 op-amp, $f_t = 2 \times 10^5 \times 5 = 1$ MHz.

16.3.2.1 Closed-Loop Frequency Response

Figure 16.32 shows a simplified model of the voltage-mode op-amp, along with external circuitry to create the popular noninverting configuration. The resistors sample V_o and feed the voltage

$$V_n = \frac{R_1}{R_1 + R_2} V_o = \beta V_o \tag{16.67}$$

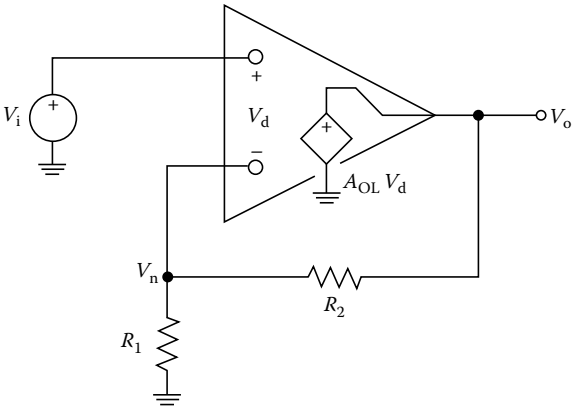


FIGURE 16.32 The noninverting configuration.

back to the inverting input. The parameter

$$\beta = \frac{R_1}{R_1 + R_2} = \frac{1}{1 + R_2/R_1} \tag{16.68}$$

representing the fraction of the output being fed back to the input is called the feedback factor. By inspection,

$$V_o = A_{OL}(jf)V_d = A_{OL}(jf)(V_i - \beta V_o) \tag{16.69}$$

In negative-feedback parlance $V_d = V_i - \beta V_o$ is referred to as the error signal. Collecting and solving for the ratio V_o/V_i yields, after minor algebraic manipulations,

$$A_{CL}(jf) = \frac{V_o}{V_i} = \left(1 + \frac{R_2}{R_1}\right)\frac{1}{1 + 1/T(jf)} \tag{16.70}$$

where $A_{CL}(jf)$ is called the closed-loop gain, and

$$T(jf) = A_{OL}(jf)\beta = \frac{A_{OL}(jf)}{1 + R_2/R_1} \tag{16.71}$$

is called the loop gain. This designation stems from the fact that a voltage propagating clockwise around the loop is first magnified by $A_{OL}(jf)$, and then attenuated by β, thus experiencing an overall gain of $T(jf) = A_{OL}(jf)\beta$.

By Equation 16.70 we have

$$\lim_{T\to\infty} A_{CL} = 1 + \frac{R_2}{R_1} \tag{16.72}$$

a value aptly called ideal closed-loop gain. Clearly, T provides a measure of how close A_{CL} is to ideal: the larger the T, the better. To ensure a substantial loop gain for a range of closed-loop gains, op-amp manufacturers strive to make A_{OL} as large as possible. Consequently, V_d will assume extremely small values since $V_d = V_o/A_{OL}$. In the limit $A_{OL} \to \infty$ we obtain $V_d \to 0$, that is, $V_n \to V_p$. This provides the

basis for the familiar ideal voltage-mode op-amp rule: When operated with negative feedback, an op-amp will provide whatever output is needed to drive its error signal V_d to zero or, equivalently, to force V_n to track V_p.

Substituting Equation 16.65 into Equation 16.71 and then into Equation 16.70, and exploiting the fact that $\beta A_{OL0} \gg 1$, we obtain, after minor algebra,

$$A_{CL}(jf) = \frac{A_{CL0}}{1 + jf/f_{CL}} \tag{16.73}$$

where

$$A_{CL0} = 1 + \frac{R_2}{R_1} = \frac{1}{\beta} \tag{16.74}$$

is the closed-loop dc gain, and

$$f_{CL} = \beta f_t = \frac{f_t}{A_{CL0}} \tag{16.75}$$

is the closed-loop small-signal bandwidth. The quantity $1/\beta$ is also called the dc noise gain because this is the gain with which the amplifier will magnify any dc noise present right at its input pins, such as the input offset voltage V_{OS}.

Equation 16.75 indicates a gain–bandwidth trade-off. As we raise the R_2/R_1 ratio to increase A_{CL0}, we also decrease f_{CL} in the process. Moreover, by Equation 16.71, $T(jf)$ is also decreased, thus leading to a greater departure of $A_{CL}(jf)$ from the ideal.

The above concepts can be visualized graphically as follows. By Equation 16.70 we can write $|T|_{dB} = 20 \log_{10}|T| = 20 \log_{10}|A_{OL}| - 20 \log_{10}(1/\beta)$, or

$$|T|_{dB} = |A_{OL}|_{dB} - |1/\beta|_{dB} \tag{16.76}$$

indicating that the loop gain can be found graphically as the difference between the decibel plot of the open-loop gain and that of the noise gain. This is illustrated in Figure 16.33. The frequency at which the two curves meet is aptly called the crossover frequency. It is readily seen that at this frequency we

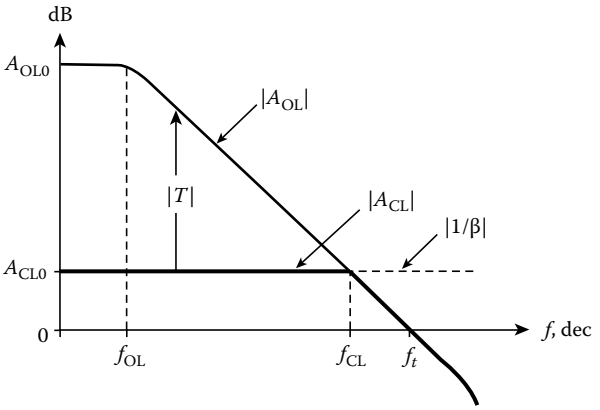


FIGURE 16.33 Graphical interpretation of the loop gain.

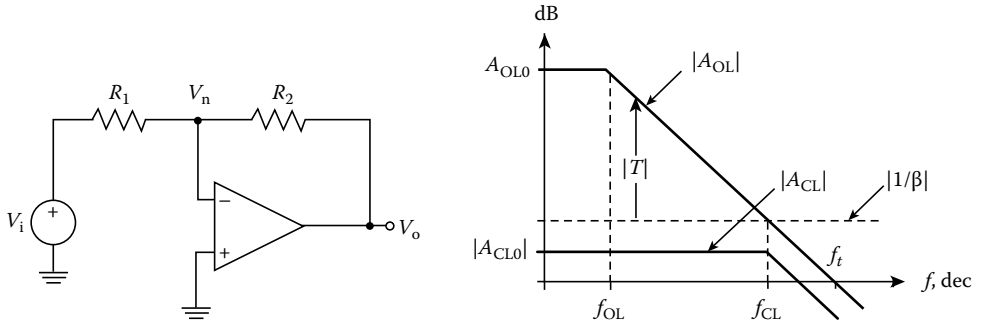


FIGURE 16.34 The inverting configuration.

have $T = 1\angle -90^\circ = -j$, so $|A_{CL}| = A_{CL0}/|1 + j| = A_{CL0}/\sqrt{2}$, by Equations 16.70 and 16.74. Consequently, the crossover frequency represents the -3 dB frequency of $A_{CL}(jf)$, that is, f_{CL}.

We now see that increasing A_{CL0} reduces T and causes the cross-point to move up the $|A_{OL}|$ curve, thus decreasing f_{CL}. The circuit with the widest bandwidth and the highest loop gain is also the one with the lowest closed-loop gain. This is the familiar voltage follower, obtained by letting $R_1 = \infty$ and $R_2 = 0$. Then, by Equations 16.74 and 16.75 we have $A_{CL0} = 1$ and $f_{CL} = f_t$.

Let us now turn to another important configuration, namely, the popular inverting amplifier of Figure 16.34. Since $V_p = 0$, it follows that $V_d = -V_n$. Applying the superposition priniciple we have

$$V_d = -\frac{R_2}{R_1 + R_2} V_i - \frac{R_1}{R_1 + R_2} V_o = \frac{-R_2}{R_1 + R_2} V_i - \beta V_o \tag{16.77}$$

indicating that the feedback factor $\beta = R_1/(R_1+R_2)$ is the same as for the noninverting configuration. Substituting into Equation 16.64, we find the closed-loop gain as

$$A_{CL}(jf) = \frac{V_o}{V_i} = \left(-\frac{R_2}{R_1}\right) \frac{1}{1 + 1/T(jf)} \tag{16.78}$$

Moreover, proceeding as for noninverting configuration, we get

$$A_{CL}(jf) = \frac{A_{CL0}}{1 + if/f_{CL}} \tag{16.79}$$

where

$$A_{CL0} = -\frac{R_2}{R_1} \tag{16.80}$$

is the closed-loop dc gain, and

$$f_{CL} = \beta f_t \tag{16.81}$$

is the closed-loop small-signal bandwidth. We can again find this bandwidth as the intercept of the $|A_{OL}|_{dB}$ and $|1/\beta|_{dB}$ curves. However, since we now have $|A_{CL0}| < |1/\beta|$, it follows that the $|A_{CL}|_{dB}$ curve will be shifted downward, as explicitly shown in Figure 16.34.

Before concluding, we wish to point out that open-loop gain roll-off affects not only the closed-loop gain, but also the closed-loop input and output impedances. Interested readers can find additional information in the literature [3].

16.3.2.2 Closed-Loop Transient Response

To fully characterize the dynamic behavior of an op-amp circuit we also need to know its transient response. This response is usually specified for the case of the op-amp operating as a unity-gain voltage follower. As we know, its small-signal transfer characteristic is $V_o = V_i/(1 + jf/f_t)$. This is formally similar to that of an ordinary *RC* circuit. Subjecting a voltage follower to a step of suitably small amplitude V_m, as shown in Figure 16.35, will cause an exponential output transition with the time constant

$$\tau = \frac{1}{2\pi f_t} \tag{16.82}$$

The rise time t_r, defined as the time it takes for V_o to swing from 10% to 90% of V_m, provides a measure of how rapidly the transition takes place. One can readily see that $t_r = \tau \ln 9 \simeq 2.2\tau$. For the 741 op-amp we have $\tau = 1/(2\pi \times 10^6) \simeq 159$ ns, and $t_r \simeq 350$ ns.

The rate at which V_o changes with time is highest at the beginning of the exponential transition, when its value is V_m/τ. Increasing the step magnitude, V_m increases this initial rate, until a point is reached beyond which the rate saturates at a constant value called the slew rate (SR). The transition is now a ramp, rather than an exponential. Figure 16.36 shows the SR limited response to a pulse.

SR limiting stems from the limited ability of the internal circuitry to charge or discharge the compensation capacitance C_c. To understand this effect, refer to Figure 16.37. As long as the input

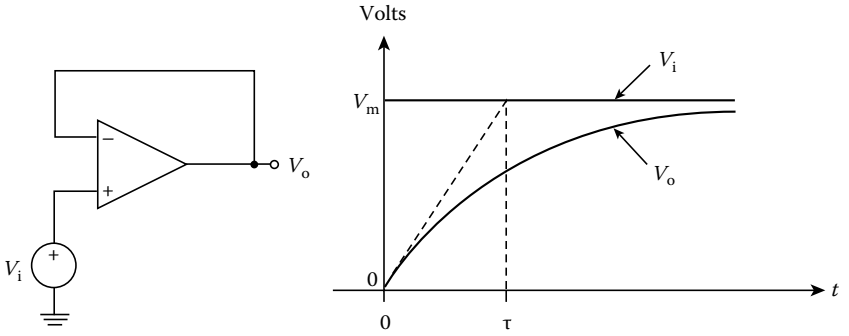


FIGURE 16.35 Voltage follower and its small-signal step response.

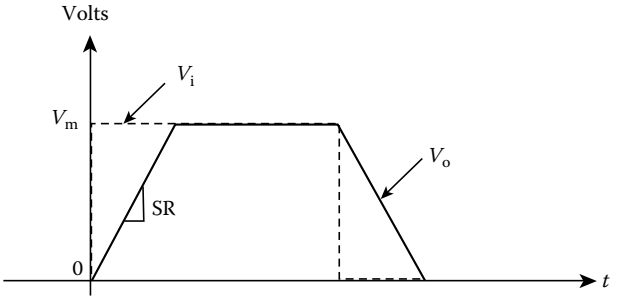


FIGURE 16.36 Large-signal response of the voltage follower.

imbalance V_d is sufficiently small, Equation 16.62 still holds and the step response is exponential. However, for large values of V_d, I_{o1} is no longer linearly proportional to V_d, but saturates at $\pm I_A$, where I_A is the input-stage bias current depicted in Figure 16.29. Turning now to Figure 16.30 and observing that the second stage acts as an integrator, we can state that the maximum rate at which C_c can be charged or discharged is $(dV_o/dt)_{max} = I_A/C_c$. This is precisely the SR,

$$SR = \frac{I_A}{C_c} \tag{16.83}$$

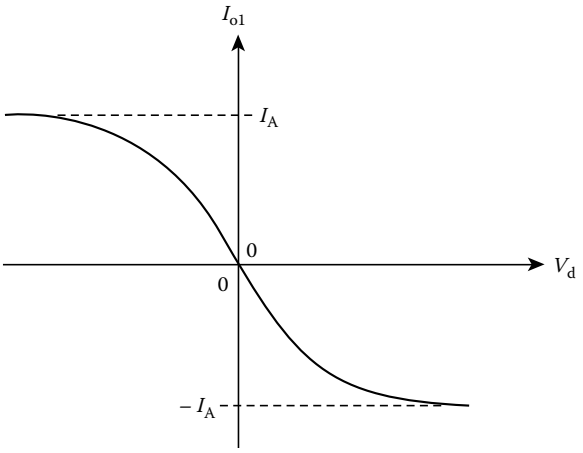


FIGURE 16.37 Actual transfer characteristic of the first stage.

The 741 op-amp has typically $I_A = 20$ μA and $C = 30$ pF, so $SR = 10 \times 10^{-6}/(30 \times 10^{-12}) = 0.67$ V/μs. To respond to a 10 V input step a 741 follower will take approximately $10/0.67 = 15$ μs.

The step magnitude corresponding to the onset of SR limiting in such that $V_m/\tau = SR$, or $V_m = SR \times \tau = SR/(2\pi f_t)$. For the 741 op-amp, $V_m = 0.67 \times 10^6/(2\pi \times 10^6) = 106$ mV. This means that as long as the input step is less than 106 mV, a 741 follower will respond with an exponential transition governed by $\tau \simeq 159$ ns. For a greater input step, however, the output will slew at a constant rate of 0.67 V/μs, and it will do so until it comes within 106 mV of the final value, after which it will complete the transition in exponential fashion.

In certain applications it is important to know the settling time t_s, defined as the time it takes for the output to settle within a specified band around its final value, usually for a full-scale output transition. It is apparent that SR limiting plays an important role in the settling-time characteristic of a circuit.

SR limiting affects also the FPBW, defined as the maximum frequency at which the circuit still yields an undistorted full-power output. Letting $V_o = V_m \sin 2\pi ft$, we have $(dV_o/dt)_{max} = (2\pi f V_m \cos 2\pi ft)_{max} = 2\pi f V_m$. Equating it to the SR and solving for f, whose value is the FPBW, we get

$$FPBW = \frac{SR}{2\pi V_m} \tag{16.84}$$

For instance, for $V_m = 10$ V, the 741 op-amp has $FPBW = SR/(20\pi) = 10.6$ kHz. Figure 16.38 shows the distorted response of a voltage follower to a full-power input with a frequency higher than the FPBW.

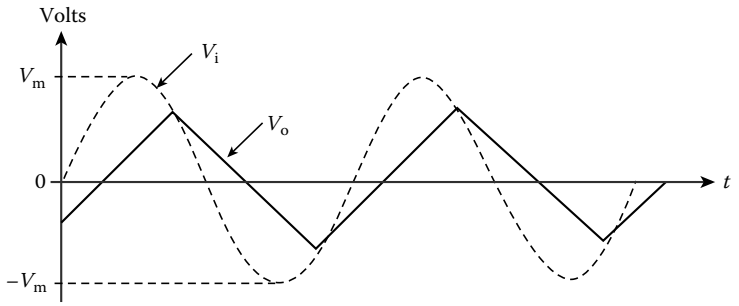


FIGURE 16.38 Distortion when the FPBW is exceeded.

From Equations 16.63 and 16.83, it is apparent that the primary cause of frequency and SR limitations is the capacitance C_c. Why not eliminate C_c altogether? Without C_c the open-loop response would exhibit a much wider bandwidth, but also a much greater phase lag because of the various poles introduced by the transistors making up the op-amp. We are interested in the situation at the crossover frequency, where $|T| = 1$. Should the phase shift at this frequency reach $-180°$, we would have $T = 1\angle -180° = -1$ which, after substitution into Equation 16.70, would yield $|A_{CL}| \rightarrow (1 + R_2/R_1)/(1 - 1) \rightarrow \infty$! The physical meaning of an infinite closed-loop gain is that the circuit would be capable of sustaining a nonzero output with a vanishingly small external input. But, this is the recipe for sustained oscillation! It is precisely to avoid this possibility that the manufacturer incorporates the frequency-compensation capacitance C_c.

As mentioned, C_c causes gain to roll-off, so that by the time the frequency of $-180°$ phase lag is reached, $|A_{OL}|$ has already dropped well below 0 dB, making it impossible for the circuits of Figures 16.32, 16.34, and 16.35 to achieve $T = -1$ at the crossover frequency, regardless of the values of R_1 and R_2. This requires that the dominant-pole frequency be suitably low, and thus, by Equation 16.63, that C_{eq} be suitably large. To avoid the need to manufacture impractically large on-chip capacitances, it is customary to start out with a realistic value, such as 30 pF for the 741 op-amp, and then exploit the multiplicative action of the Miller effect to raise it to the desired equivalent value.

The hardest configuration to compensate is the unity-gain voltage follower because its crossover frequency is the closest to the frequency region of additional phase lag stemming from the higher-order poles of the op-amp. This is why this particular configuration is usually chosen for the specification of the transient response.

16.3.2.3 SPICE Simulation

Op amp dynamics can readily be simulated via SPICE using the op-amp macromodels available from the manufacturers. Shown in Figure 16.39 is the SPICE circuit used to display the frequency and transient responses of the 741 op-amp. The curves of Figure 16.40 are obtained by displaying the dB curves of $|a| = |V_o/(V_p - V_n)| = |V_o/V_n|$ and $|A| = |V_o/V_i|$. The curves of Figures 16.41 and 16.42 display, respectively, the small-signal and the large-signal responses to an input step. The former is dominated by exponential transients, the latter by SR-limited ramps. Note how in either case the inverting input node itself exhibits a transient before becoming a virtual ground once the circuit reaches its steady state.

16.3.3 Current-Mode Op-Amps

Current-mode op-amps exploit a special circuit topology [4], along with high-speed complementary bipolar processes, to achieve much faster dynamics than their voltage-mode amplifier counterparts. The

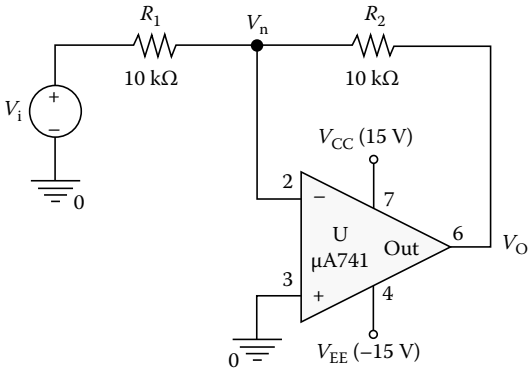


FIGURE 16.39 SPICE circuit to simulate the dynamics of the 741 op-amp.

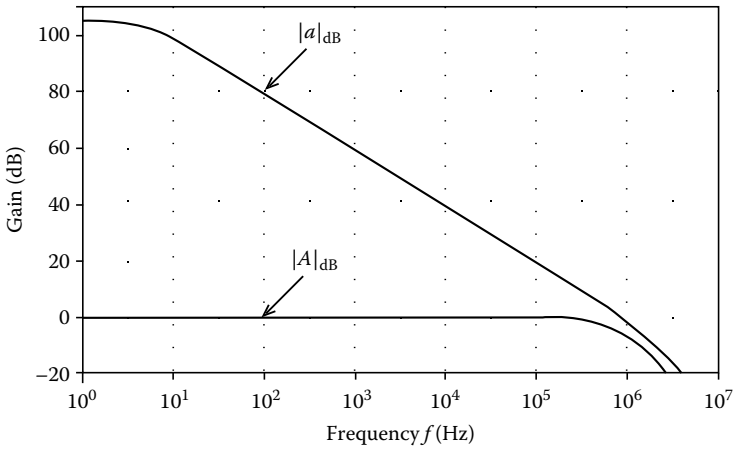


FIGURE 16.40 Open-loop and closed-loop responses of the circuit of Figure 16.39.

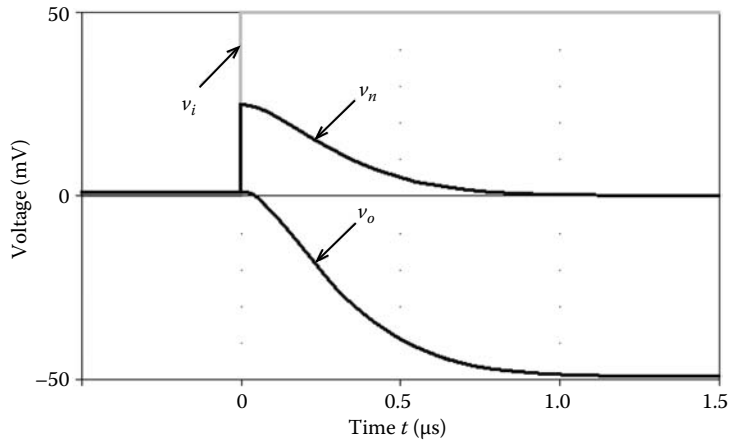


FIGURE 16.41 Small-signal transient response of the 741 op-amp.

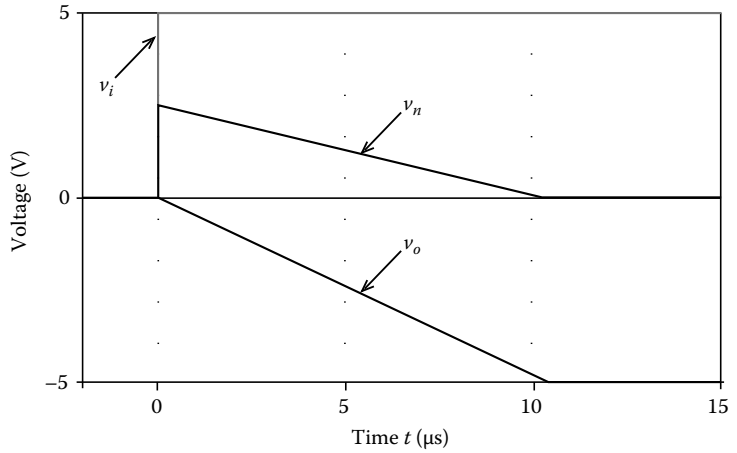


FIGURE 16.42 Large-signal transient response of the 741 op-amp.

name stems from the fact that these amplifiers respond to an input imbalance of the current type, and the signal propagating around the feedback loop is thus in the form of a current rather than a voltage.

16.3.3.1 Block Diagram

Figure 16.43 shows the simplified circuit diagram of a current-mode op-amp. Referring also to Figure 16.44, we identify three functional blocks.

1. The first functional block is a unity-grain input buffer, consisting of transistors Q_1 through Q_4. Q_1 and Q_2 form a low output-impedance push–pull stage, while Q_3 and Q_4 provide V_{BE} compensation as well as a Darlington function to raise the input impedance. This buffer forces V_n to follow V_p, very much like a voltage-mode op-amp does via negative feedback.
   When the op-amp is embedded in a circuit, current can easily flow in or out of its low-impedance inverting input-pin, though we shall see that in the steady-state (nonslewing) condition this current approaches zero. The function of the buffer is to sense this current, denoted as I_n, and produce an imbalance

$$I_1 - I_2 = I_n \tag{16.85}$$

   between the push–pull transistor currents I_1 and I_2.
2. The second block is a pair of current mirrors $Q_5 - Q_6$ and $Q_7 - Q_8$, which reflect currents I_1 and I_2 and sum them at a common junction node. The current into this node thus equals I_n, as shown.
3. Finally, a unity-gain output buffer, consisting of transistors Q_9 through Q_{12}, buffers the summing node voltage to the outside and provides a low output impedance for the overall op-amp.

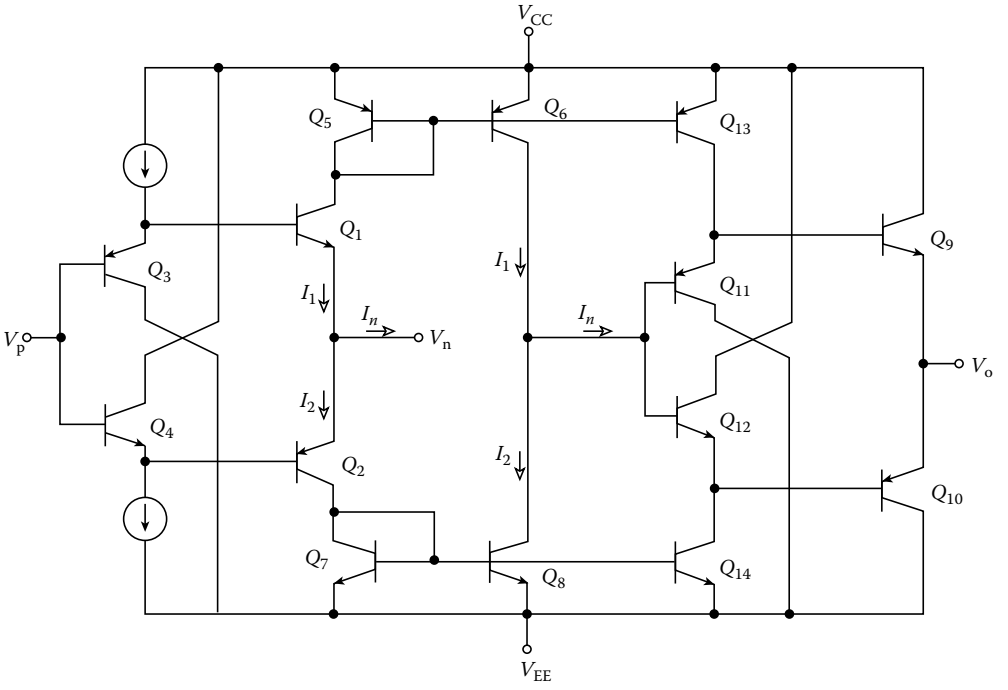


FIGURE 16.43 Simplified circuit diagram of a current-mode op-amp.

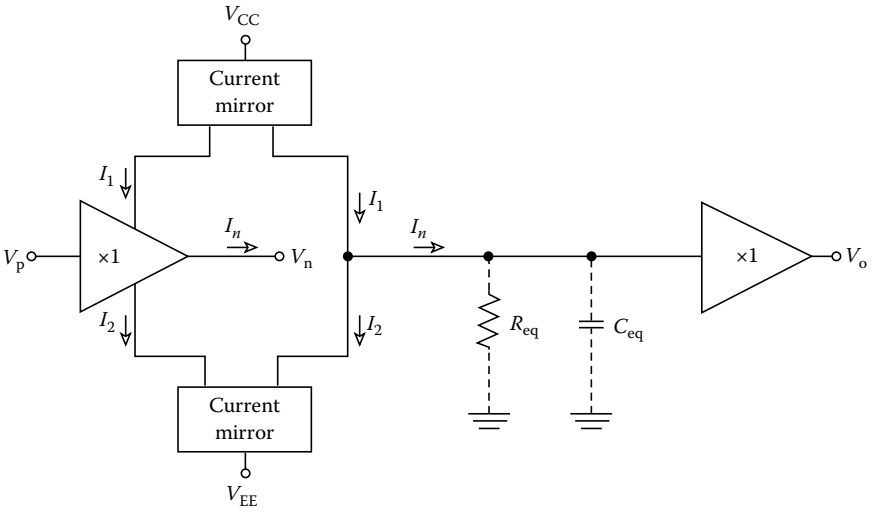


FIGURE 16.44 Current-mode op-amp block diagram.

Denoting the net equivalent resistance and capacitance of the summing node toward ground as R_{eq} and C_{eq}, we can write

$$V_o = Z_{OL}(jf)I_n \tag{16.86}$$

where $Z_{OL}(jf)$, called the *open-loop transimpedance gain*, is the impedance due to the parallel combination of R_{eq} and C_{eq}. This impedance can be expressed as

$$Z_{OL}(jf) = \frac{Z_{OL0}}{1 + jf/f_{OL}} \tag{16.87}$$

where $Z_{OL0} = R_{eq}$, and

$$f_{OL} = \frac{1}{2\pi R_{eq} C_{eq}} \tag{16.88}$$

As an example in Figure 16.44, the CLC401 current-mode op-amp (National Semiconductor Co.) has $Z_{OL0} \simeq 710$ kΩ, $f_{CL} \simeq 350$ kHz, and $C_{eq} = 1(2\pi R_{eq} f_{OL}) \simeq 0.64$ pF.

We observe a formal similarity with voltage-mode op-amps, except that now the error signal I_n is a current rather than a voltage, and the gain $Z_{OL}(jf)$ is in V/A rather than in V/V. For this reason, current-mode op-amps are also referred to as transimpedance op-amps. Than gain $Z_{OL}(jf)$ is approximately constant from dc to f_{OL}, after which it rolls off with frequency at a constant rate of −1 dec/dec.

16.3.3.2 Closed-Loop Characteristics

Figure 16.45 shows a simplified model of the current-mode op-amp, along with an external feedback network to configure it as a noninverting amplifier. Any attempt to unbalance the inputs will cause the input buffer to source (or sink) an imbalance current I_n to the external network. By Equation 16.86, this imbalance causes V_o to swing in the positive (or negative) direction until the original imbalance current is neutralized via the negative-feedback loop.

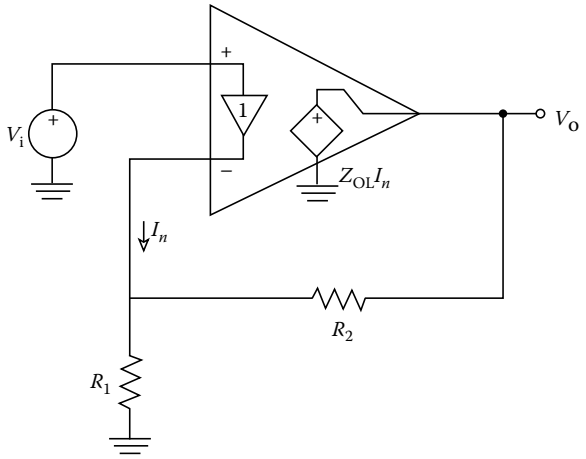


FIGURE 16.45 Noninverting configuration.

Exploiting the fact that the input buffer keeps $V_n = V_p = V_i$, we can apply the superposition principle and write

$$I_n = \frac{V_i}{R_1 \parallel R_2} - \frac{V_o}{R_2} = \frac{V_i}{R_1 \parallel R_2} - \beta V_o \tag{16.89}$$

Clearly, the feedback signal V_o/R_2 is now a current, and the feedback factor

$$\beta = \frac{1}{R_2} \tag{16.90}$$

is now in A/V. Substituting into Equation 16.86 and collecting, we get

$$A_{CL}(jf) = \frac{V_o}{V_i} = \left(1 + \frac{R_2}{R_1}\right) \frac{1}{1 + 1/T(jf)} \tag{16.91}$$

where $A_{CL}(jf)$ is the closed-loop gain of the circuit, and

$$T(jf) = Z_{OL}(jf)\beta = \frac{Z_{OL}(jf)}{R_2} \tag{16.92}$$

is the loop gain. This designation is due again to the fact that a current propagating clockwise around the loop is first multipled by $Z_{OL}(jf)$ to be converted to a voltage, and divided by R_2 to be converted back to a current, thus experiencing an overall gain of $T(jf) = Z_{OL}(jf)/R_2$.

To make $A_{CL}(jf)$ approach the ideal value $1 + R_2/R_1$, it is desirable that $T(jf)$ be as large as possible. To ensure a substantial loop gain over a range of closed-loop gains, manufacturers strive to maximize $Z_{OL}(jf)$ relative to R_2. Consequently, since $I_n = V_o/Z_{OL}$, the inverting input-pin current will be very small, even though this is a low-impedance node because of the input buffer. In the limit $Z_{OL} \to \infty$ we obtain $I_n \to 0$, indicating that a current-mode op-amp will ideally provide whatever output is needed to drive I_n to zero. Thus, the familiar op-amp conditions $V_n \to V_p$, $I_n \to 0$, and $I_p \to 0$ hold also for current-mode op-amps, though for different reasons than their voltage-mode counterparts.

16.3.3.3 Current-Mode Op-Amp Dynamics

Substituting Equation 16.87 into Equation 16.92 and then into Equation 16.91, and exploiting the fact that $Z_{OL0}/R_2 \gg 1$, we obtain

$$A_{CL}(jf) = \frac{A_{CL0}}{1 + jf/f_t} \tag{16.93}$$

where

$$A_{CL0} = 1 + \frac{R_2}{R_1} \tag{16.94}$$

is the closed-loop dc gain, and

$$f_t = \frac{Z_{OL0} f_{OL}}{R_2} = \frac{1}{2\pi R_2 C_{eq}} \tag{16.95}$$

is the closed-loop bandwidth. With R_2 in the kΩ range and C_{eq} in the pF range, f_t is typically in range of 10^8 Hz.

We can again visualize $|T|$ and f_t graphically by noting that if we define $|T|_{dec} = \log_{10}|T|$, then we have, by Equation 16.92, $|T|_{dec} = \log_{10}|Z_{OL}(jf)| - \log_{10}|R_2|$, or

$$|T|_{dec} = |Z_{OL}(jf)|_{dec} - |R_2|_{dec} \tag{16.96}$$

As shown in Figure 16.46, we can visualize the loop gain as the difference between the decade plot of $|Z_{OL}(jf)|$ and that of $|R_2|$, with the latter now acting as the noise gain. Since at the crossover frequency we have $T = 1\ \angle -90° = -j$, Equations 16.91 and 16.94 yield $|A_{CL}| = A_{CL0}/\sqrt{2}$. Consequently, the crossover frequency represents the −3 dB frequency of $A_{CL}(jf)$, that is, f_t.

We are now ready to make two important observations.

1. Equation 16.95 shows that for a given amplifier the closed-loop bandwidth depends only on R_2. We can thus use R_2 to select the bandwidth f_t via Equation 16.95, and R_1 to select the dc gain A_{OL0} via Equation 16.94. The ability to set gain independently of bandwidth, along with the absence of gain-bandwidth trade-off, constitutes the first major advantage of current-mode over voltage-mode op-amps, see Figure 16.47.

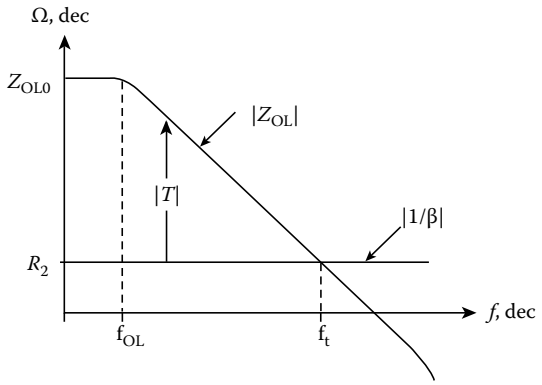


FIGURE 16.46 Graphical interpretation of the loop gain.

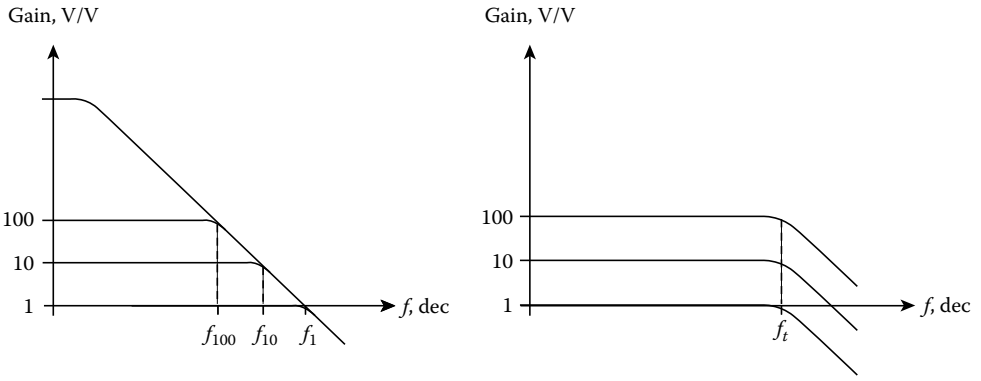


FIGURE 16.47 Comparing the gain–bandwidth characteristics of voltage-mode (left) and current-mode (right) op amps.

2. The other major advantage of current-mode op-amps is the absence of SR limiting. To justify, suppose we apply an input step $V_i = V_m$ to the circuit of Figure 16.45. Referring also to Figure 16.44, we note that the resulting current imbalance I_n yields an output V_o such that $I_n = C_{eq}dV_o/dt + V_o/R_{eq}$. Substituting into Equation 16.89, rearranging, and exploiting the fact that $R_2/Z_{OL0} \ll 1$, we get

$$R_2 C_{eq} \frac{dV_o}{dt} + V_o = A_{OL0} V_m \tag{16.97}$$

indicating an *exponential* output transient regardless of V_m. The time constant governing the transient is

$$T = R_2 C_{eq} \tag{16.98}$$

and is set by R_2, regardless of A_{CL0}. For instance, a CLC401 op-amp with $R_2 = 1.5$ kΩ has $\tau = R_2 C_{eq} = 1.5 \times 10^3 \times 0.64 \times 10^{-12} \simeq 1$ ns. The rise time is $t_r = 2.2\tau \simeq 2.2$ ns, and the settling time within 0.1% of the final value is $t_s \simeq 7\tau \simeq 7$ ns, in reasonable agreement with the data-sheet values $t_r = 2.5$ ns and $t_s = 10$ ns.

16.3.3.4 Higher-Order Effects

The above analysis indicates that once R_2 has been set, the dynamics are unaffected by the closed-loop gain setting. In practice it is found that bandwidth and rise time do vary with gain somewhat, though not as drastically as for voltage-mode op-amps. The main cause is the nonzero output resistance R_n of the input buffer, whose effect is to alter the loop gain and, hence, the closed-loop dynamics. Referring to Figure 16.48, we again use the superposition principle and write

$$I_n = \frac{V_i}{R_n + R_1 \parallel R_2} - \beta V_o \tag{16.99}$$

where the feedback factor is found using the current divider formula and Ohm's law,

$$\beta = \frac{R_1}{R_1 + R_n} \times \frac{1}{R_2 + (R_n \parallel R_1)} = \frac{1}{R_2 + A_{CL0} R_n} \tag{16.100}$$

Comparing with Equation 16.90, we observe that the effect of R_n is to replace R_2 with $R_2 + A_{CL0}R_n$. The $|1/\beta|$ curve of Figure 16.46 will thus be shifted upward, leading to a decrease in the crossover frequency,

which we shall now denote as f_{CL}. This frequency is obtained by letting $R_2 \rightarrow (R_2 + A_{CL0}R_n)$ in Equation 16.95,

$$f_{CL} = \frac{Z_{OL0}f_{OL}}{R_2 + A_{CL0}R_n} = \frac{1}{2\pi(R_2 + A_{CL0}R_n)C_{eq}} \qquad (16.101)$$

As an example, suppose an op-amp has $R_n = 50\ \Omega$, $R_2 = 1.5\ \text{k}\Omega$, and $f_t = 100$ MHz. Then, Equation 16.101 yields $f_{CL} = f_t/(1 + A_{CL0}R_n/R_2) = 10^8/(1 + A_{CL0}/30)$. The bandwidths corresponding to $A_{CL0} = 1$ V/V, 10 V/V, and 100 V/V are, respectively, $f_1 = 96.8$ MHz, $f_{10} = 75.0$ MHz, and $f_{100} = 23.1$ MHz, and are shown in Figure 16.49. The corresponding rise times are, respectively, $t_1 = 3.6$ ns, $t_{10} = 4.7$ ns, and $t_{100} = 15.2$ ns. We note that the above bandwidth reductions still compare favorably with voltage-mode op-amps, where the reduction factors would be, respectively, 1, 10, and 100.

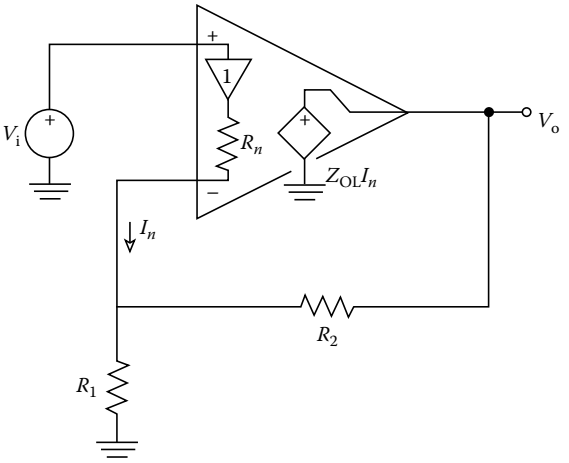


FIGURE 16.48 Investigating the effect of R_n.

The values of R_1 and R_2 can be predistorted to compensate for the bandwidth reduction. Using Equation 16.101 we find R_2 for a given bandwidth f_{CL} and dc gain A_{CL0},

$$R_2 = \frac{Z_{OL0}f_{OL}}{f_{CL}} - A_{CL0}R_n \qquad (16.102)$$

and using Equation 16.94 we find R_1 for the given dc gain A_{CL0},

$$R_1 = \frac{R_2}{A_{CL0} - 1} \qquad (16.103)$$

As an example, suppose an op-amp with $Z_{OL0}f_{OL} = 1.5 \times 10^{11}\ \Omega \times \text{Hz}$ and $R_n = 50\ \Omega$ is to be configured for $A_{CL0} = 10$ V/V and $f_{CL} = 100$ MHz. Then, using Equations 16.102 and 16.103 we find $R_2 = 1.5 \times 10^{11}/10^8 - 10 \times 50 = 1\ \text{k}\Omega$, and $R_1 = 10^3/(10 - 1) = 111\ \Omega$.

Though our analysis has focused on the noninverting configuration, we can readily extend our line of reasoning to other circuits such as the popular inverting configuration. The latter is obtained by

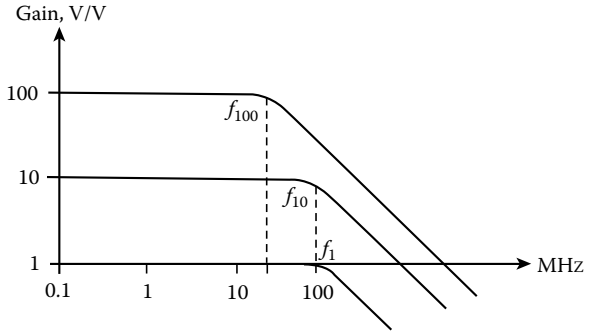


FIGURE 16.49 Bandwidth reduction for different gain settings.

grounding the noninverting input of Figure 16.48, and applying the source V_i via the bottom lead of R_1. The bandwidth is still as in Equation 16.101, but the dc gain is now $-R_2/R_1$. Interested readers can consult the literature [1,5] for additional current-mode op-amp circuits as well as application hints.

We conclude by pointing out that current-mode op-amps, though exhibiting much faster dynamics than their voltage-mode counterparts, in general suffer from poorer input offset voltage and input bias current characteristics. Moreover, having much wider bandwidths, they tend to be noisier. There is no question that the circuit designer must carefully weigh both advantages and disadvantages before deciding which amplifier type is best suited to the application at hand.

Acknowledgment

Parts of this chapter are based on the author's article ''Current feedback amplifiers benefit high-speed designs.'' This article appeared in the Jan. 5, 1989 issue of *EDN* on pages 161–172. Cahners Publishing Company of Newton, MA has granted permission for its appearance.

References

1. S. Franco, Current feedback amplifiers benefit high-speed designs, *EDN*, 161–172, Jan. 5, 1989.
2. J. E. Solomon, The monolithic operational amplifier: A tutorial study, *IEEE Journal of Solid-State Circuits*, SC-9, 314–332, Dec. 1974.
3. S. Franco, *Design with Operational Amplifiers and Analog ICs*, 3rd edn., New York: WCB/McGraw-Hill, 2001.
4. D. Nelson and S. Evans, A new approach to op-amp design, Comlinear Corp. Applicat. Note 3001, Mar. 1985.
5. D. Potson, Current-feedback op-amp applications circuit guide, Comlinear Corp. Applicat. Note OA-07, May 1988.

17

High-Frequency Amplifiers

17.1 Introduction **17**-1
17.2 Current Feedback Op-Amp **17**-2
Current Feedback Op-Amp Basics • CMOS Compound Device • Buffer and CFOA Implementation
17.3 RF Low-Noise Amplifiers **17**-8
Specifications • CMOS Common-Source LNA: Simplified Analysis • CMOS Common-Source LNA: Effect of C_{gd} • Cascode CS LNA
17.4 Optical Low-Noise Preamplifiers **17**-17
Front-End Noise Sources • Receiver Performance Criteria • Transimpedance (TZ) Amplifiers • Layout for HF Operation
17.5 Fundamentals of RF Power Amplifier Design **17**-24
PA Requirements • Power Amplifier Classification • Practical Considerations for RF Power Amplifiers • Conclusions
17.6 Applications of High-Q Resonators in IF-Sampling Receiver Architectures **17**-29
IF Sampling • Linear Region Transconductor Implementation • gm-C Bandpass Biquad
17.7 Log-Domain Processing **17**-34
Instantaneous Companding • Log-Domain Filter Synthesis • Performance Aspects • Basic Log-Domain Integrator • Synthesis of Higher-Order Log-Domain Filters
References **17**-45

Chris Toumazou
Imperial College of Science, Technology and Medicine

Alison Payne
Imperial College of Science, Technology and Medicine

17.1 Introduction

As the operating frequency of communication channels for both video and wireless increases, there is an ever-increasing demand for high-frequency (HF) amplifiers. Furthermore, the quest for single-chip integration has led to a whole new generation of amplifiers predominantly geared toward CMOS VLSI. In this chapter, we will focus on the design of high-frequency amplifiers for potential applications in the front-end of video, optical, and RF systems. Figure 17.1 shows, for example, the architecture of a typical mobile phone transceiver front-end. With channel frequencies approaching the 2 GHz range, coupled with demands for reduced chip size and power consumption, there is an increasing quest for VLSI at microwave frequencies. The shrinking feature size of CMOS has facilitated the design of complex analog circuits and systems in the 1–2 GHz range, where more traditional low-frequency lumped circuit techniques are now becoming feasible. Since the amplifier is the core component in such systems,

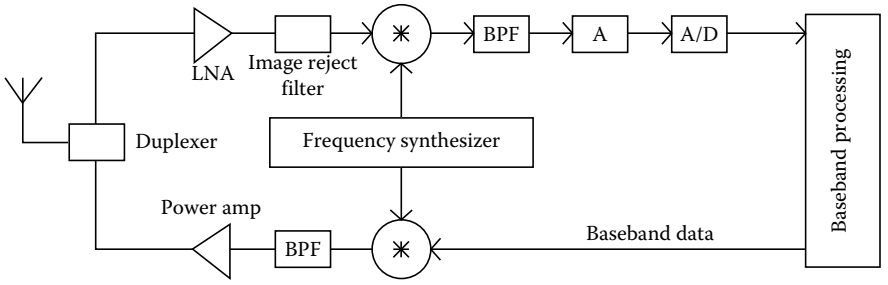


FIGURE 17.1 Generic wireless transceiver architecture.

there has been an abundance of circuit design methodologies for high-speed, low-voltage, low-noise, and low-distortion operation.

This chapter will present various amplifier designs that aim to satisfy these demanding requirements. In particular, we will review, and in some cases present new ideas for power amps, LNAs, and transconductance cells, which form core building blocks for systems such as Figure 17.1. Section 17.2 begins by reviewing the concept of current feedback, and shows how this concept can be employed in the development of low-voltage, high-speed, constant-bandwidth CMOS amplifiers. The next two sections of the chapter focus on amplifiers for wireless receiver applications, investigating performance requirements and design strategies for optical receiver amplifiers (Section 17.3) and HF low-noise amplifiers (LNAs) (Section 17.4). Section 17.5 considers the design of amplifiers for the transmitter side, and in particular the design and feasibility of Class E power amps are discussed. Finally, Section 17.6 reviews a very recent low-distortion amplifier design strategy termed ''log-domain,'' which has shown enormous potential for HF, low-distortion tunable filters.

17.2 Current Feedback Op-Amp

17.2.1 Current Feedback Op-Amp Basics

The operational amplifier (op-amp) is one of the fundamental building blocks of analog circuit design [1,2]. High-performance signal processing functions such as amplifiers, filters, oscillators, etc. can be readily implemented with the availability of high-speed, low-distortion op-amps. In the last decade, the development of complementary bipolar technology has enabled the implementation of single-chip video op-amps [3–7]. The emergence of op-amps with nontraditional topologies, such as the current feedback op-amp (CFOA), has improved the speed of these devices even further [8–11]. CFOA structures are well known for their ability to overcome (to a first-order approximation) the gain-bandwidth trade-off and slew rate limitation that characterizes traditional voltage feedback op-amps (VFOAs) [12].

Figure 17.2 shows a simple macromodel of a CFOA, along with a simplified circuit diagram of the basic architecture. The topology of the CFOA differs from the conventional VFOA in two respects. First, the input stage of a CFOA is a unity-gain voltage buffer connected between the inputs of the op-amp. Its function is to force V_n to follow V_p, very much like a conventional VFOA does via negative feedback. In the case of the CFOA, because of the low output impedance of the buffer, current can flow in or out of the inverting input, although in normal operation (with negative feedback) this current is extremely small. Secondly, a CFOA provides a high open-loop transimpedance gain $Z(j\omega)$, rather than open-loop voltage gain as with a VFOA. This is shown in Figure 17.2, where a current-controlled current source senses the current I_{INV} delivered by the buffer to the external feedback network, and copies this current to a high impedance $Z(j\omega)$. The voltage conveyed to the output is given by Equation 17.1:

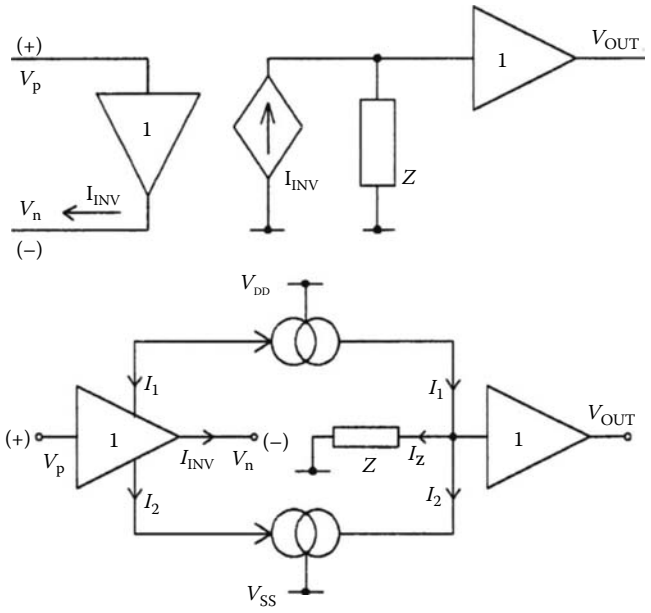


FIGURE 17.2 CFOA macromodel.

$$V_{OUT} = Z(j\omega) \cdot I_{INV} \Rightarrow \frac{V_{OUT}}{I_{INV}}(j\omega) = Z(j\omega) \tag{17.1}$$

When the negative feedback loop is closed, any voltage imbalance between the two inputs due to some external agent will cause the input voltage buffer to deliver an error current I_{INV} to the external network. This error current $I_{INV} = I_1 - I_2 = I_Z$ is then conveyed by the current mirrors to the impedance Z, resulting in an ouput voltage as given by Equation 17.1. The application of negative feedback ensures that V_{OUT} will move in the direction that reduces the error current I_{INV} and equalizes the input voltages.

We can approximate the open-loop dynamics of the CFOA as a single pole response. Assuming that the total impedance $Z(j\omega)$ at the gain node is the combination of the output resistance of the current mirrors R_o in parallel with a compensation capacitor C, we can write:

$$Z(j\omega) = \frac{R_o}{1 + j\omega R_o C} = \frac{R_o}{1 + j\frac{\omega}{\omega_o}} \tag{17.2}$$

where $\omega_o = 1/R_o C$ represents the frequency where the open-loop transimpedance gain is 3 dB down from its low frequency value R_o. In general, R_o is designed to be very high in value.

Referring to the noninverting amplifier configuration shown in Figure 17.3:

$$I_{INV} = \frac{V_{IN}}{R_G} - \frac{V_{OUT} - V_{IN}}{R_F} = \frac{V_{IN}}{R_G // R_F} - \frac{V_{OUT}}{R_F} \tag{17.3}$$

Substituting Equation 17.1 into Equation 17.3 yields the following expression for the closed-loop gain:

$$A_{CL}(j\omega) = \left(1 + \frac{R_F}{R_G}\right) \cdot \frac{Z(j\omega)}{R_F + Z(j\omega)} = \left(1 + \frac{R_F}{R_G}\right) \cdot \frac{1}{1 + \frac{R_F}{Z(j\omega)}} \tag{17.4}$$

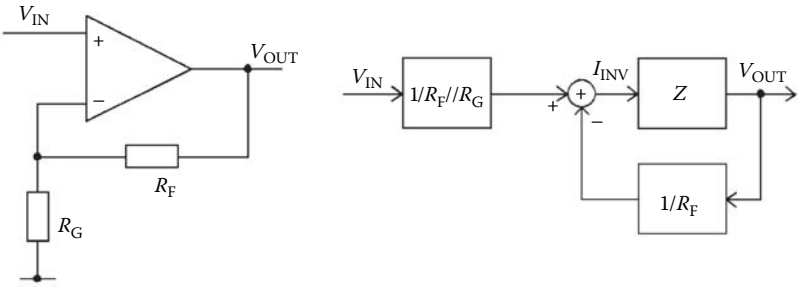


FIGURE 17.3 CFOA noninverting amplifier configuration.

Combining Equations 17.2 and 17.4, and assuming that the low frequency value of the open-loop transimpedance is much higher than the feedback resistor ($R_o \gg R_F$) gives:

$$A_{CL}(j\omega) = \left(1 + \frac{R_F}{R_G}\right) \cdot \frac{1}{1 + j\frac{R_F \cdot \omega}{R_o \cdot \omega_o}} = \frac{A_{Vo}}{1 + j\frac{\omega}{\omega_\alpha}} \tag{17.5}$$

Referring to Equation 17.5, the closed-loop gain $A_{Vo} = 1 + R_F/R_G$, while the closed-loop −3 dB frequency ω_α is given by

$$\omega_\alpha = \frac{R_o}{R_F} \cdot \omega_o \tag{17.6}$$

Equation 17.6 indicates that the closed-loop bandwidth does not depend on the closed-loop gain as in the case of a conventional VFOA, but is determined by the feedback resistor R_F. Explaining this intuitively, the current available to charge the compensation capacitor at the gain node is determined by the value of the feedback resistor R_F and not R_o, provided that $R_o \gg R_F$. So, once the bandwidth of the amplifier is set via R_F, the gain can be independently varied by changing R_G. The ability to control the gain independently of bandwidth constitutes a major advantage of CFOAs over conventional VFOAs.

The other major advantage of the CFOA compared to the VFOA is the inherent absence of slew rate limiting. For the circuit of Figure 17.3, assume that the input buffer is very fast and thus a change in voltage at the noninverting input is instantaneously converted to the inverting input. When a step ΔV_{IN} is applied to the noninverting input, the buffer output current can be derived as

$$I_{INV} = \frac{V_{IN} - V_{OUT}}{R_F} + \frac{V_{IN}}{R_G} \tag{17.7}$$

Equation 17.7 indicates that the current available to charge/discharge the compensation capacitor is proportional to the input step regardless of its size, that is, there is no upper limit. The rate of change of the output voltage is thus

$$\frac{dV_{OUT}}{dt} = \frac{I_{INV}}{C} \Rightarrow V_{OUT}(t) = \Delta V_{IN} \cdot \left(1 + \frac{R_F}{R_G}\right) \cdot (1 - e^{-t/R_f \cdot C}) \tag{17.8}$$

Equation 17.8 indicates an exponential output transition with time constant $\tau = R_F \cdot C$. Similar to the small-signal frequency response, the large-signal transient response is governed by R_F alone, regardless of the magnitude of the closed-loop gain. The absence of slew rate limiting allows for faster settling times and eliminates slew rate-related nonlinearities.

In most practical bipolar realizations, Darlington-pair transistors are used in the input stage to reduce input bias currents, which makes the op-amp somewhat noisier and increases the input offset voltage. This is not necessary in CMOS realizations due to the inherently high MOSFET input impedance. However, in a closed-loop CFOA, R_G should be much larger than the output impedance of the buffer. In bipolar realizations, it is fairly simple to obtain a buffer with low output resistance, but this becomes more of a problem in CMOS due to the inherently lower gain of MOSFET devices. As a result, R_G typically needs to be higher in a CMOS CFOA than in a bipolar realization, and consequently, R_F needs to be increased above the value required for optimum HF performance. Additionally, the fact that the input buffer is not in the feedback loop imposes linearity limitations on the structure, especially if the impedance at the gain node is not very high. Regardless of these problems, CFOAs exhibit excellent HF characteristics and are increasingly popular in video and communications applications [13].

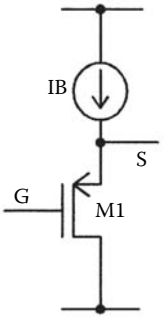


FIGURE 17.4 Simple PMOS source follower.

The following sections outline the development of a novel low output impedance CMOS buffer, which is then employed in a CMOS CFOA to reduce the minimum allowable value of R_G.

17.2.2 CMOS Compound Device

A simple PMOS source follower is shown in Figure 17.4. The output impedance seen looking into the source of M1 is approximately $Z_{out} = 1/g_m$, where g_m is the small-signal transconductance of M1. To increase g_m, the drain current of M1 could be increased, which leads to an increased power dissipation. Alternatively, the dimensions of M1 can be increased, resulting in additional parasitic capacitance and hence an inferior frequency response. Figure 17.5 shows a configuration that achieves a higher transconductance than the simple follower of Figure 17.3 for the same bias current [11]. The current of M2 is fed back to M1 through the a:1 current mirror. This configuration can be viewed as a compound transistor whose gate is the gate of M1 and whose source is the source of M2. The impedance looking into the compound source can be approximated as $Z_{out} = (g_{m1} - a \cdot g_{m2})/(g_{m1} \cdot g_{m2})$, where g_{m1} and g_{m2} represent the small-signal transconductance of M1 and M2, respectively. The output impedance can be made small by setting the current mirror transfer ratio $a = g_{m1}/g_{m2}$.

The p-compound device is practically implemented as in Figure 17.6. In order to obtain a linear voltage transfer function from node 1 to 2, the gate-source voltages of M1 and M3 must cancel. The current mirror (M4–M2) acts as an NMOS–PMOS gate-source voltage matching circuit [14] and compensates for the difference in the gate-source voltages of M1 and M3, which would normally appear as an output offset. DC analysis, assuming a square law model for the MOSFETs, shows that the output voltage exactly follows the input voltage. However, in practice, channel length modulation and body effects preclude exact cancellation [15].

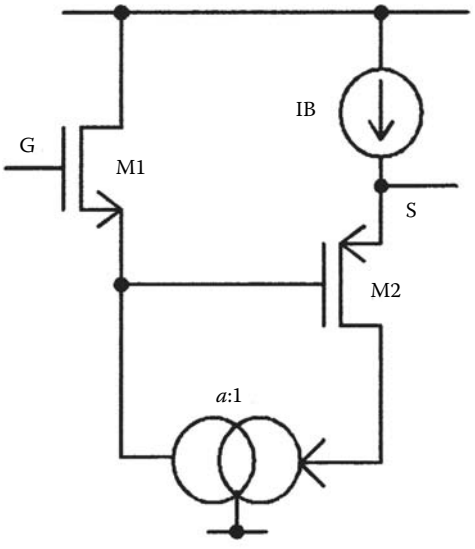


FIGURE 17.5 Compound MOS device.

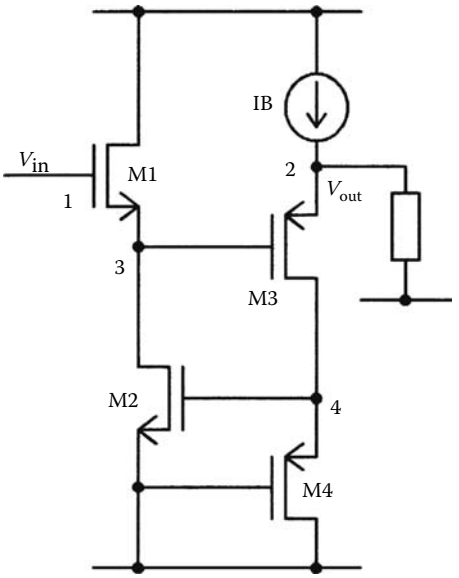


FIGURE 17.6 Actual p-compound device implementation.

17.2.3 Buffer and CFOA Implementation

The CFOA op-amp shown in Figure 17.7 has been implemented in a single-well 0.6 μm digital CMOS process [11]; the corresponding layout plot is shown in Figure 17.8. The chip has an area of 280 μm by 330 μm and a power dissipation of 12 mW. The amplifier comprises two voltage followers (input and output) connected by cascoded current mirrors to enhance the gain node impedance. A compensation capacitor ($C_c = 0.5$ pF) at the gain node ensures adequate phase margin and thus closed-loop stability. The voltage followers have been implemented with two compound transistors, p-type and n-type, in a push–pull arrangement. Two such compound transistors in the output stage are shown shaded in Figure 17.7. The input voltage follower of the CFOA was initially tested open-loop, and measured results are summarized in Table 17.1. The load is set to 10 kΩ/10 pF, except where mentioned otherwise, 10 kΩ being a limit imposed by overall power dissipation of the chip. Intermodulation distortion was

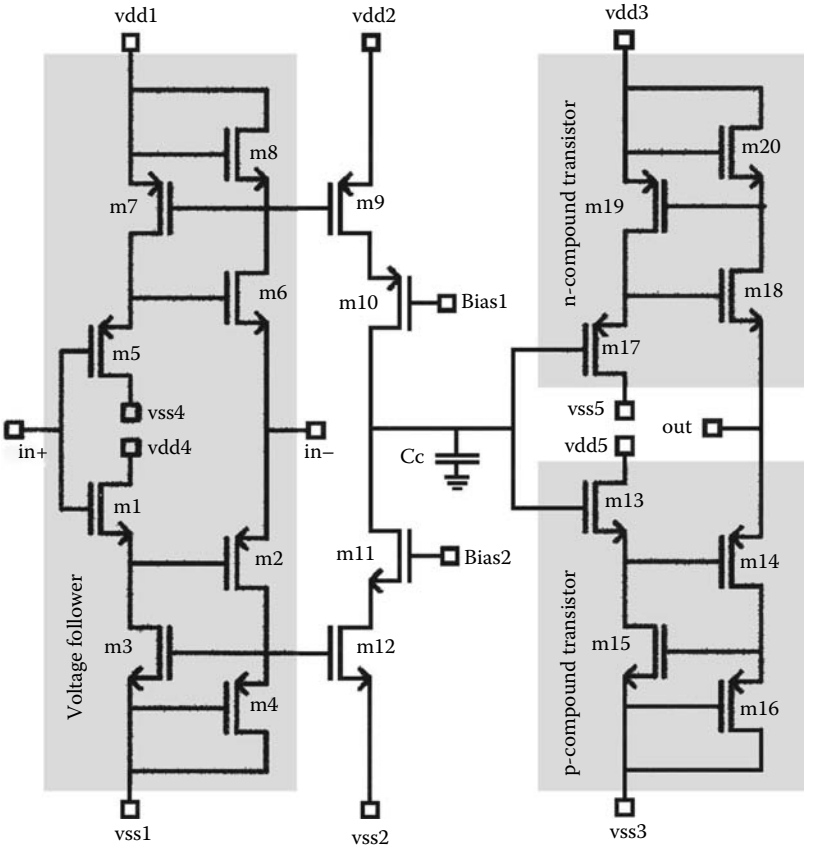


FIGURE 17.7 CFOA schematic.

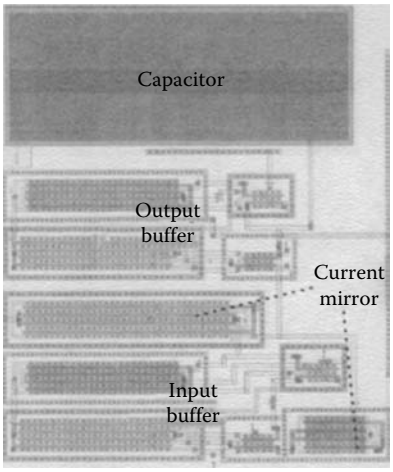


FIGURE 17.8 CFOA layout plot.

measured with 2 tons separated by 200 kHz. The measured output impedance of the buffer is given in Figure 17.9. It remains below 80 Ω up to a frequency of about 60 MHz, when it enters an inductive region. A maximum impedance of 140 Ω is reached around 160 MHz. Beyond this frequency, the output impedance is dominated by parasitic capacitances. The inductive behavior is characteristic of the use of feedback to reduce output impedance, and can cause stability problems when driving capacitive loads. Small-signal analysis (summarized in Table 17.2) predicts a double zero in the output impedance [15].

Making factor G in Table 17.2 small will reduce the output impedance, but also moves the double zero to lower frequencies and intensifies the inductive behavior. The principal trade-off in this configuration is between output impedance magnitude and inductive behavior. In practice, the output impedance can be reduced by a factor of 3 while still maintaining good stability when driving capacitive loads. Figure 17.10 shows the measured frequency response of the buffer. Given the low power dissipation, excellent slew rates have been achieved (Table 17.2).

After the characterization of the input buffer stage, the entire CFOA was tested to confirm the suitability of the compound transistors for the implementation of more complex building blocks. Open-loop transimpedance measurements are shown in Figure 17.11. The bandwidth of the amplifier was measured at gain settings of 1, 2, 5, and 10 in a noninverting configuration, and the feedback resistor was trimmed to achieve maximum bandwidth at each gain setting separately. CFOA measurements are summarized in Table 17.3, loading conditions are again 10 kΩ/10 pF.

Figure 17.12 shows the measured frequency response for various gain settings. The bandwidth remains constant at 110 MHz for gains of 1, 2, and 5, consistent with the expected behavior of a CFOA. The bandwidth falls to 42 MHz for a gain of 10 due to the finite output impedance of the input buffer stage which series as the CFOA inverting input. Figure 17.13 illustrates the step response of the CFOA driving a 10 kΩ/10 pF load at a voltage gain of 2. It can be seen that the inductive behavior of the buffers has little effect on the step response. Finally, distortion measurements were carried out for the entire CFOA for gain settings 2, 5, and 10 and are summarized in Table 17.3. HD2 levels can be further improved by employing a double-balanced topology. A distortion spectrum is shown in Figure 17.14; the onset of HD3 is due to clipping at the test conditions.

TABLE 17.1 Voltage Buffer Performance

Power Supply	5 V	Dissipation	5 mW
DC gain (no load)	−3.3 dB	Bandwidth	140 MHz
Output impedance	75 Ω	Min. load resistance	10 kΩ
HD2 ($V_{in} = 200$ mV$_{rms}$)	1 MHz	−50 dB	
	10 MHz	−49 dB	
	20 MHz	−45 dB	
IM3 ($V_{in} = 200$ mV$_{rms}$)	20 MHz, $\Delta f = 200$ kHz	−53 dB	
Slew rate	(Load = 10 pF)	+130 V/μs	−72 V/μs
Input-referred noise	10 nV$\sqrt{\text{Hz}}$		

Note: Load = 10 kΩ/10 pF, except for slew rate measurement.

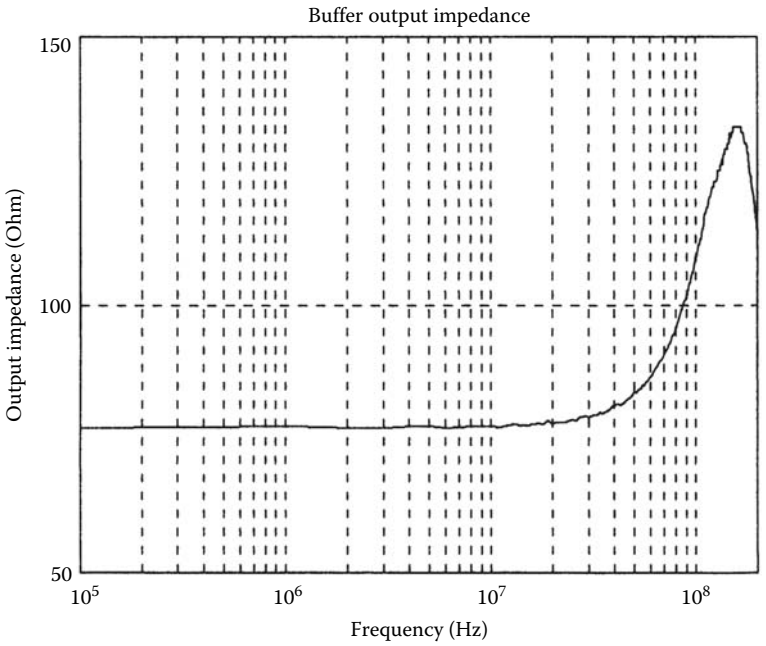


FIGURE 17.9 Measured buffer output impedance characteristics.

TABLE 17.2 Voltage Transfer Function and Output Impedance of Compound Device

$Z_{out} = \dfrac{G}{(g_{m1} + g_{ds1} + g_{ds2}) \cdot (g_{m3} + g_{ds3}) \cdot (g_{m4} + g_{ds4})}$
$\dfrac{V_{out}}{V_{in}} = \dfrac{g_{m1} \cdot g_{m3} \cdot (g_{m4} + g_{ds4})}{(g_{m1} + g_{ds1} + g_{ds2}) \cdot (g_{m3} + g_{ds3}) \cdot (g_{m4} + g_{ds4}) + g_L \cdot G}$
$G = (g_{m1} + g_{ds1} + g_{ds2}) \cdot (g_{m4} + g_{ds4} + g_{ds3}) - g_{m2} \cdot g_{m3}$

17.3 RF Low-Noise Amplifiers

This section reviews the important performance criteria demanded of the front-end amplifier in a wireless communication receiver. The design of CMOS LNAs for front-end wireless communication receiver applications is then addressed. Section 17.4 considers the related topic of LNAs for optical receiver front-ends.

17.3.1 Specifications

The front-end amplifier in a wireless receiver must satisfy demanding requirements in terms of noise, gain, impedance matching, and linearity.

17.3.1.1 Noise

Since the incoming signal is usually weak, the front-end circuits of the receiver must possess very low noise characteristics so that the original signal can be recovered. Provided that the gain of the front-end amplifier is sufficient so as to suppress noise from the subsequent stages, the receiver noise performance is determined predominantly by the front-end amplifier. Hence, the front-end amplifier should be an LNA.

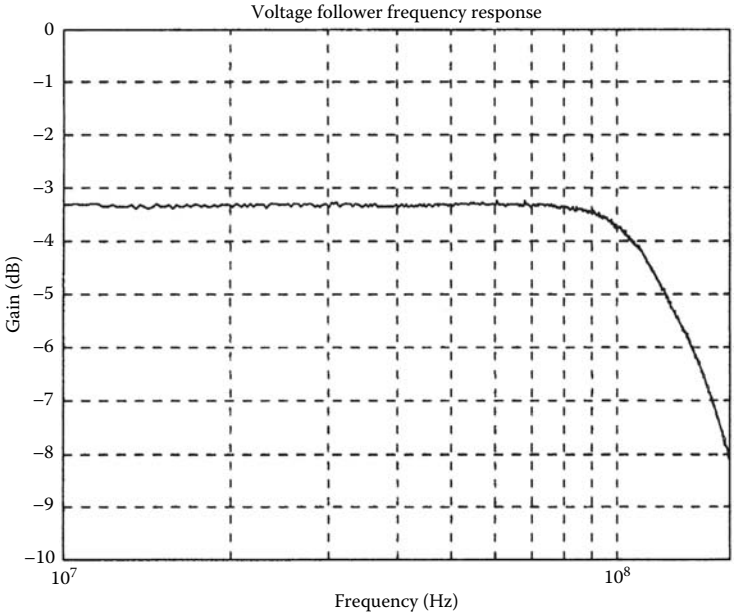


FIGURE 17.10 Measured buffer frequency response.

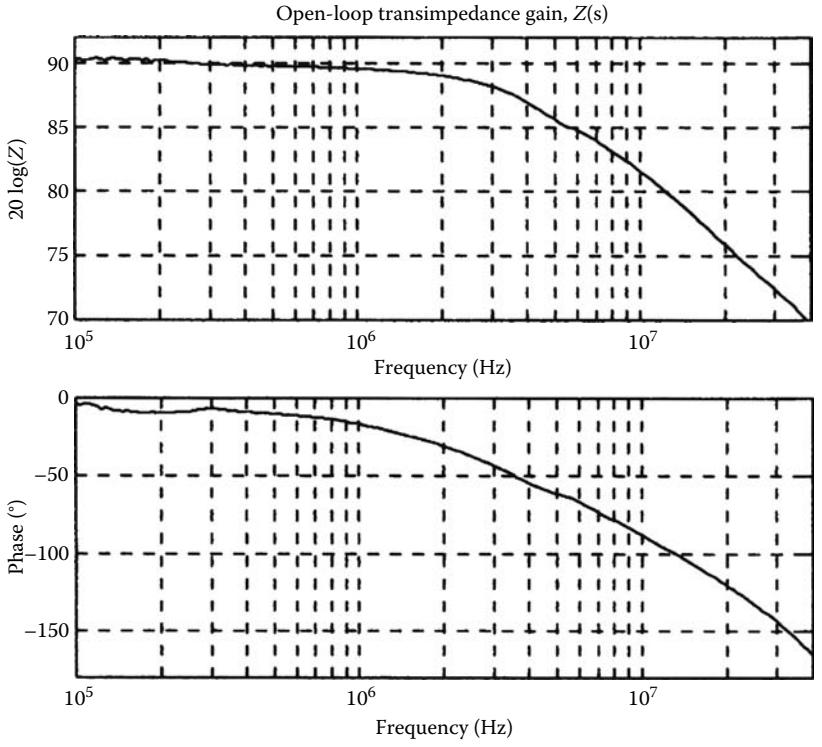


FIGURE 17.11 Measured CFOA open-loop transimpedance gain.

TABLE 17.3 CFOA Measurement Summary

Power Supply	5 V	Power Dissipation	12 mW
Gain	Bandwidth (MHz)		
1	117		
2	118		
5	113		
10	42		
Frequency	**Input (mV rms)**	**Gain**	**HD2 (dB)**
1 MHz	140	2	−51
	40	5	−50
	10	10	−49
10 MHz	80	2	−42
	40	5	−42
	13	10	−43

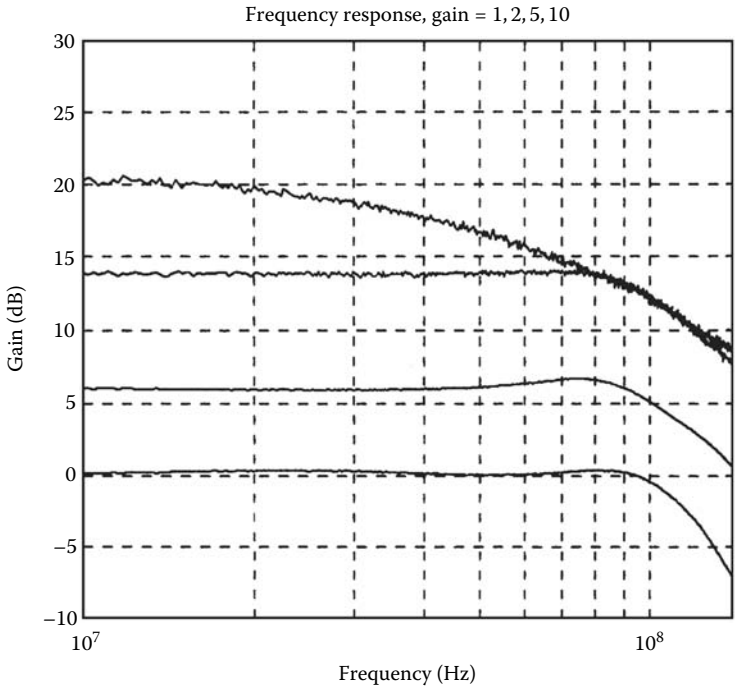


FIGURE 17.12 Measured CFOA closed-loop frequency response.

17.3.1.2 Gain

The voltage gain of the LNA must be high enough to ensure that noise contributions from the following stages can be safely neglected. As an example, Figure 17.15 shows the first three stages in a generic front-end receiver, where the gain and output-referred noise of each stage are represented by G_i and N_i ($i = 1, 2, 3$), respectively. The total noise at the third stage output is given by

$$N_{\text{out}} = N_{\text{in}}G_1G_2G_3 + N_1G_2G_3 + N_2G_3 + N_3 \tag{17.9}$$

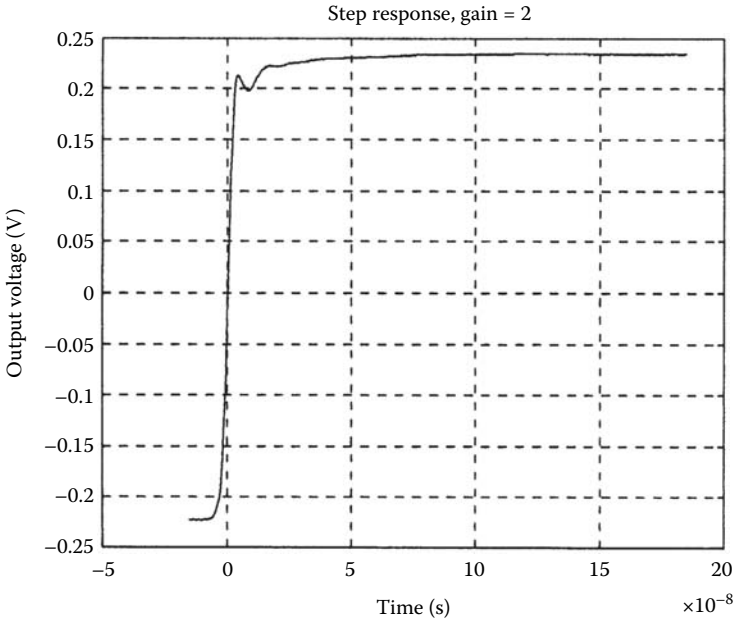


FIGURE 17.13 Measured CFOA step response.

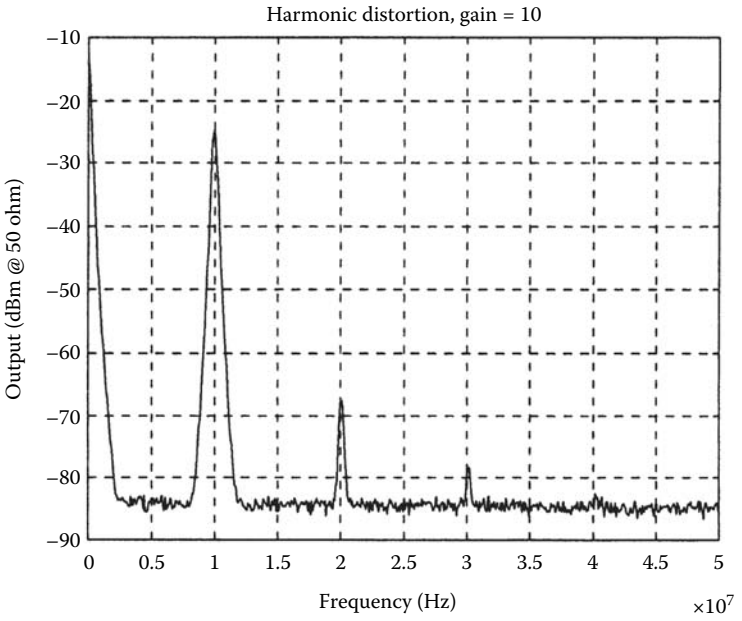


FIGURE 17.14 CFOA harmonic distortion measurements.

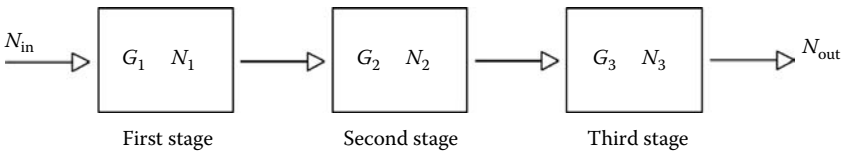


FIGURE 17.15 Three-stage building block with gain G_i and noise N_i per stage.

This output noise (N_{out}) can be referred to the input to derive an equivalent input noise (N_{eq}):

$$N_{eq} = \frac{N_{out}}{\text{Gain}} = \frac{N_{out}}{G_1 G_2 G_3} = N_{in} + \frac{N_1}{G_1} + \frac{N_2}{G_1 G_2} + \frac{N_3}{G_1 G_2 G_3} \tag{17.10}$$

According to Equation 23.10, the gain of the first stage should be high in order to reduce noise contributions from subsequent stages. However, if the gain is too high, a large input signal may saturate the subsequent stages, yielding intermodulation products which corrupt the desired signal. Thus, optimization is inevitable.

17.3.1.3 Input Impedance Matching

The input impedance of the LNA must be matched to the antenna impedance over the frequency range of interest, in order to transfer the maximum available power to the receiver.

17.3.1.4 Linearity

Unwanted signals at frequencies fairly near the frequency band of interest may reach the LNA with signal strengths many times higher than that of the wanted signal. The LNA must be sufficiently linear to prevent these out-of-band signals from generating intermodulation products within the wanted frequency band, and thus degrading the reception of the desired signal. Since third-order mixing products are usually dominant, the linearity of the LNA is related to the "third-order intercept point" (IP3), which is defined as the input power level that results in equal power levels for the output fundamental frequency component and the third-order intermodulation components. The dynamic range (DR) of a wireless receiver is limited at the lower bound by noise and at the upper band by nonlinearity.

17.3.2 CMOS Common-Source LNA: Simplified Analysis

17.3.2.1 Input Impedance Matching by Source Degeneration

For maximum power transfer, the input impedance of the LNA must be matched to the source resistance, which is normally 50 Ω. Impedance-matching circuits consist of reactive components and therefore are (ideally) lossless and noiseless. Figure 17.16 shows the small-signal equivalent circuit of a common-source (CS) LNA input stage with impedance-matching circuit, where the gate-drain capacitance C_{gd} is

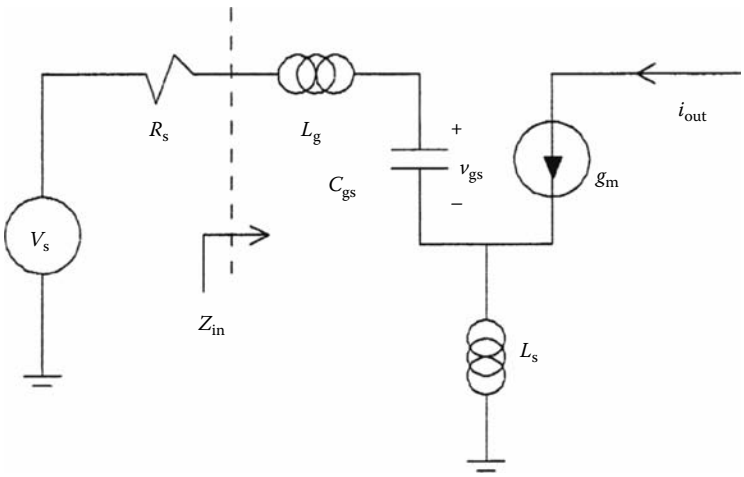


FIGURE 17.16 Simplified small-signal equivalent circuit of the CS stage.

assumed to have negligible effect and is thus neglected [16,17]. The input impedance of this CS input stage is given by

$$Z_{in} = j\omega(L_g + L_s) + \frac{1}{j\omega C_{gs}} + \frac{g_m}{C_{gs}} L_s \tag{17.11}$$

Thus, for matching, the two conditions below must be satisfied:

$$\text{(i) } \omega_o^2 = \frac{1}{(L_g + L_s)C_{gs}} \quad \text{and} \quad \text{(ii) } \frac{g_m}{C_{gs}} L_s = R_s \tag{17.12}$$

17.3.2.2 Noise Figure of CS Input Stage

Two main noise sources exist in a CS input stage as shown in Figure 17.17; thermal noise from the source resistor R_s (denoted $\overline{v_{Rs}^2}$) and channel thermal noise from the input transistor (denoted $\overline{i_d^2}$). The output noise current due to $\overline{v_{Rs}^2}$ can be determined from Figure 17.17 as

$$\overline{i_{nout1}^2} = \frac{g_m^2 \overline{v_{Rs}^2}}{\omega^2 (g_m L_s + R_s C_{gs})^2} = \frac{g_m^2}{4\omega^2 R_s^2 C_{gs}^2} \overline{v_{Rs}^2} \tag{17.13}$$

while the output noise current due to $\overline{i_d^2}$ can be evaluated as

$$\overline{i_{nout2}} = \frac{\overline{i_d}}{\left(1 + \frac{g_m L_s}{R_s C_{gs}}\right)} = \frac{1}{2}\overline{i_d} \quad \therefore \ \overline{i_{nout2}^2} = \frac{1}{4}\overline{i_d^2} \tag{17.14}$$

From Equations 17.13 and 17.14, the noise figure (NF) of the CS input stage is determined as

$$NF = 1 + \frac{\overline{i_{nout2}^2}}{i_{nout1}^2} = 1 + \Gamma\left(\frac{\omega_o^2 R_s C_{gs}^2}{g_m}\right) = 1 + \Gamma\left(\frac{L_s}{L_s + L_g}\right) \tag{17.15}$$

In practice, any inductor (especially a fully integrated inductor) has an associated resistance that will contribute thermal noise, degrading the NF in Equation 17.15.

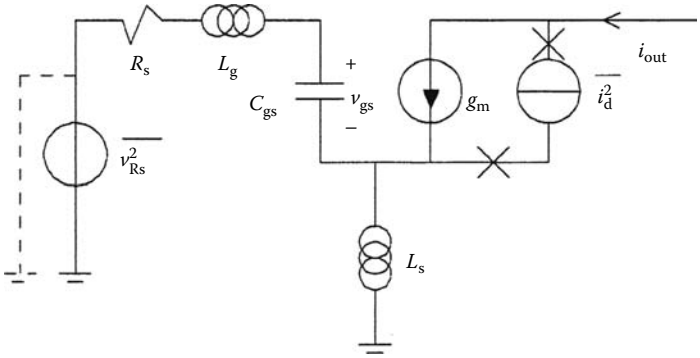


FIGURE 17.17 Simplified noise equivalent circuit of the CS stage. $\overline{V_{Rs}^2} = 4kT_{Rs}$; $\overline{id^2} = KT\Gamma g_{dc}$.

17.3.2.3 Voltage Amplifier with Inductive Load

Referring to Figure 17.15, the small-signal current output is given by

$$i_{\text{out}} = \frac{g_{\text{m}} v_{\text{s}}}{[1 - \omega^2 C_{\text{gs}}(L_{\text{g}} + L_{\text{s}})] + j\omega(g_{\text{m}} L_{\text{s}} + R_{\text{s}} C_{\text{gs}})} \tag{17.16}$$

For an inductive load (L_1) with a series internal resistance r_{L1}, the output voltage is thus

$$v_{\text{out}} = -i_{\text{out}}(r_{\text{L1}} + j\omega L_1) = \frac{-(r_{\text{L1}} + j\omega L_1) g_{\text{m}} v_{\text{s}}}{[1 - \omega^2 C_{\text{gs}}(L_{\text{g}} + L_{\text{s}})] + j\omega(g_{\text{m}} L_{\text{s}} + R_{\text{s}} C_{\text{gs}})} \tag{17.17}$$

Assuming that the input is impedance matched, the voltage gain at the output is given by

$$\left|\frac{v_{\text{out}}}{v_{\text{s}}}\right| = \frac{\sqrt{(r_{\text{L1}})^2 + (\omega_{\text{o}})^2 (L_1)^2}}{2\omega_{\text{o}} L_{\text{s}}} = \frac{r_{\text{L1}}}{2\omega_{\text{o}} L_{\text{s}}} \sqrt{1 + \left(\frac{\omega_{\text{o}} L_1}{r_{\text{L1}}}\right)^2} \cong \frac{1}{2}\omega_{\text{o}} \left(\frac{L_1}{L_{\text{s}}}\right)\left(\frac{L_1}{r_{\text{L1}}}\right) \tag{17.18}$$

17.3.3 CMOS Common-Source LNA: Effect of C_{gd}

In the analysis so far, the gate-drain capacitance (C_{gd}) has been assumed to be negligible. However, at very high frequencies, this component cannot be neglected. Figure 17.18 shows the modified input stage of a CS LNA including C_{gd} and an input ac-coupling capacitance C_{in}. Small-signal analysis shows that the input impedance is now given by

$$Z_{\text{in}} = \frac{g_{\text{m}} L_{\text{s}}}{C_{\text{gs}} + C_{\text{gd}}\left[j\omega L_{\text{s}} g_{\text{m}} + \frac{g_{\text{m}}(1 - \omega^2 L_{\text{s}} C_{\text{gd}})}{\frac{1}{Z_{\text{L}}} + j\omega C_{\text{gd}}}\right]} \tag{17.19}$$

Equation 17.19 exhibits resonance frequencies that occur when

$$1 - \omega^2 L_{\text{s}} C_{\text{gs}} = 0 \quad \text{and} \quad 1 - \omega^2 L_{\text{g}} C_{\text{in}} = 0 \tag{17.20}$$

Equation 17.19 indicates that the input impedance matching is degraded by the load Z_L when C_{gd} is included in the analysis.

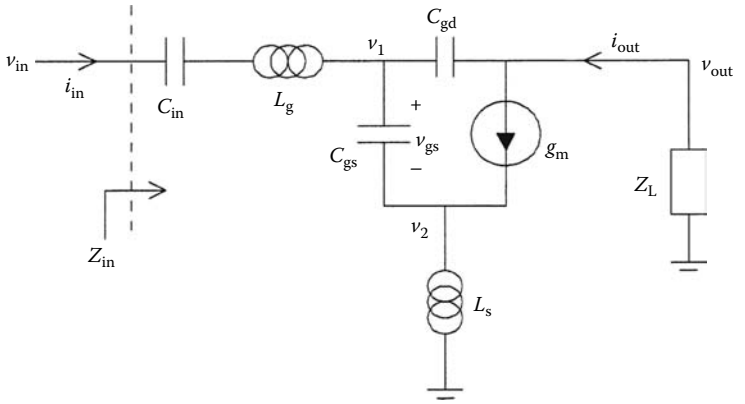


FIGURE 17.18 Noise equivalent circuit of the CS stage, including effects of C_{gd}.

17.3.3.1 Input Impedance with Capacitive Load

If the load Z_L is purely capacitive, that is,

$$Z_L = \frac{1}{j\omega C_L} \tag{17.21}$$

then the input impedance can be easily matched to the source resistor R_s. Substituting Equation 17.21 for Z_L, the bracketed term in the denominator of Equation 17.19 becomes

$$d_1 = j\omega L_s g_m + \frac{g_m(1-\omega^2 L_s C_{gd})}{j\omega(C_{gd}+C_L)} = 0 \tag{17.22}$$

under the condition that

$$1 - \omega^2 L_s(2C_{gd} + C_L) = 0 \tag{17.23}$$

The three conditions in Equations 17.20 and 17.23 should be met to ensure input impedance matching. However, in practice, we are unlikely to be in the situation of using a load capacitor.

17.3.3.2 Input Impedance with Inductive Load

If $Z_L = j\omega L_L$, the CS LNA input impedance is given by

$$Z_{in} = \frac{g_m L_s}{C_{gs} + j\omega C_{gd} g_m \left[L_s + L_L\left(\frac{1-\omega^2 L_s C_{gd}}{1-\omega^2 L_L C_{gd}}\right)\right]} \tag{17.24}$$

In order to match to a purely resistive input, the value of the reactive term in Equation 17.24 must be negligible, which is difficult to achieve.

17.3.4 Cascode CS LNA

17.3.4.1 Input Matching

As outlined in the paragraph above, the gate-drain capacitance (C_{gd}) degrades the input impedance matching and therefore reduces the power transfer efficiency. In order to reduce the effect of C_{gd}, a cascoded structure can be used [18–20]. Figure 17.19 shows a cascode CS LNA. Since the voltage gain from the gate to the drain of M1 is unity, the gate-drain capacitance (C_{gd1}) no longer sees the full input–output voltage swing which greatly improves the input–output isolation. The input impedance can be approximated by Equation 17.11, thus allowing a simple matching circuit to be employed [18].

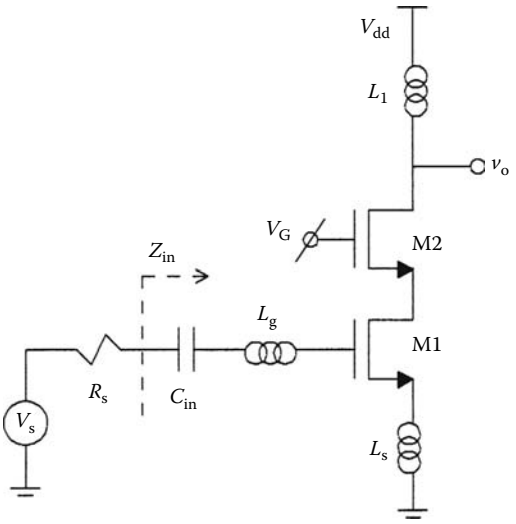


FIGURE 17.19 Cascode CS LNA.

17.3.4.2 Voltage Gain

Figure 17.20 shows the small-signal equivalent circuit of the cascode CS LNA. Assuming that input is fully matched to the source, the voltage gain of the amplifier is given by

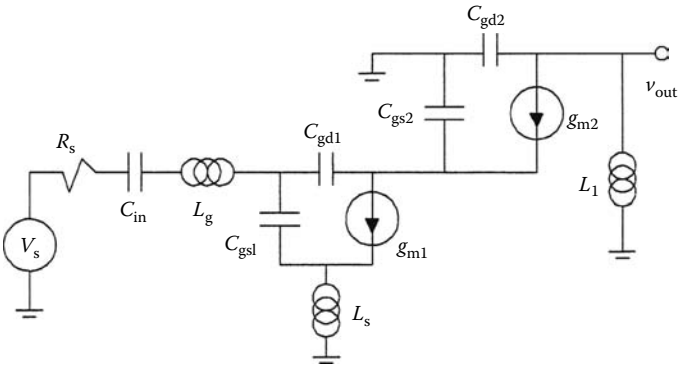


FIGURE 17.20 Equivalent circuit of cascode CS LNA.

$$\frac{v_{out}}{v_s} = -\frac{1}{2}\left(\frac{j\omega L_1}{1-\omega^2 L_1 C_{gd2}}\right)\left(\frac{g_{m2}}{g_{m2}+j\omega C_{gs2}}\right)\left[\frac{g_{m1}}{(1-\omega^2 L_s C_{gs1})+j\omega L_s g_{m1}}\right] \tag{17.25}$$

At the resonant frequency, the voltage gain is given by

$$\frac{v_{out}}{v_s}(\omega_o) = -\frac{1}{2}\left(\frac{L_1}{L_s}\right)\left(\frac{1}{1-\omega_o^2 L_1 C_{gd2}}\right)\times\frac{1}{1+j\omega_o\left(\frac{C_{gs1}}{g_{m2}}\right)} \approx -\frac{1}{2}\left(\frac{L_1}{L_s}\right)\times\frac{1}{1+j\left(\frac{\omega_o}{\omega_T}\right)} \tag{17.26}$$

From Equation 17.26, the voltage gain is dependent on the ratio of the load and source inductance values. Therefore, high gain accuracy can be achieved since this ratio is largely process independent.

17.3.4.3 Noise Figure

Figure 17.21 shows an equivalent circuit of the cascode CS LNA for noise calculations. Three main noise sources can be identified: the thermal noise voltage from R_s, and the channel thermal noise currents from M1 and M2. Assuming that the input impedance is matched to the sources, the output noise current due to $\overline{v_{Rs}^2}$ can be derived as

$$\overline{i_{out1}} = \frac{1}{2j\omega_o L_s(1-\omega_o^2 L_1 C_{gd2})}\left(\frac{g_{m2}}{g_{m2}+j\omega_o C_{gs2}}\right)\overline{v_{Rs}} \tag{17.27}$$

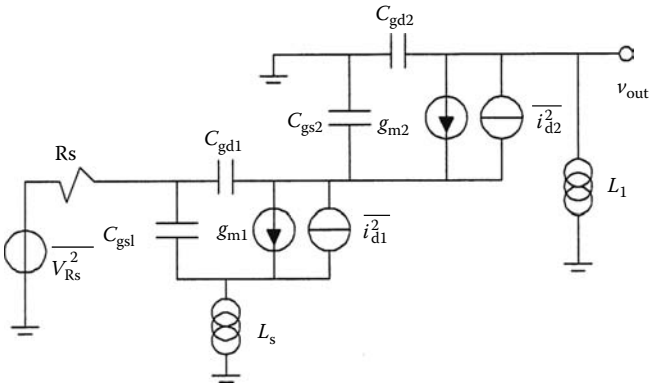


FIGURE 17.21 Noise equivalent circuit of cascode CS LNA.

The output noise current contribution due to $\overline{i_{d1}^2}$ of M1 is given by

$$\overline{i_{out2}} = \frac{1}{2(1-\omega_o^2 L_1 C_{gd2})}\left(\frac{g_{m2}}{g_{m2}+j\omega_o C_{ds2}}\right)\overline{i_{d1}} \tag{17.28}$$

The output noise current due to $\overline{i_{d2}^2}$ of M2 is given by

$$\overline{i_{out3}} = \frac{j\omega_o C_{gs2}}{(1-\omega_o^2 L_1 C_{gd2})(g_{m2}+j\omega_o C_{gs2})}\overline{i_{d2}} \tag{17.29}$$

The NF of the cascode CS LNA can thus be derived as

$$\mathrm{NF} = 1 + \frac{\overline{i_{out2}^2}}{\overline{i_{out1}^2}} + \frac{\overline{i_{out3}^2}}{\overline{i_{out1}^2}} = 1 + \Gamma\left(1 + \frac{4\omega_o^2 C_{gs2}^2}{g_{m1}g_{m2}}\right) \tag{17.30}$$

In order to improve the NF, the transconductance values (g_m) of M1 and M2 should be increased. Since the gate-source capacitance (C_{gs2}) of M2 is directly proportional to the gate width, the gate width of M2 cannot be enlarged to increase the transconductance. Instead, this increase should be realized by increasing the gate bias voltage.

17.4 Optical Low-Noise Preamplifiers

Figure 17.22 shows a simple schematic diagram of an optical receiver, consisting of a photodetector, a preamplifier, a wide-band voltage amplifier, and a predetection filter. Since the front-end transimpedance preamplifier is critical in determining the overall receiver performance, it should possess a wide bandwidth so as not to distort the received signal, high gain to reject noise from subsequent stages, low noise to achieve high sensitivity, wide DR, and low intersymbol interference (ISI).

17.4.1 Front-End Noise Sources

Receiver noise is dominated by two main noise sources: the detector (PIN photodiode) noise and the amplifier noise. Figure 17.23 illustrates the noise equivalent circuit of the optical receiver.

17.4.1.1 PIN Photodiode Noise

The noise generated by a PIN photodiode arises mainly from three shot noise contributions: quantum noise $S_q(f)$, thermally generated dark-current shot noise $S_D(f)$, and surface leakage-current shot noise $S_L(f)$. Other noise sources in a PIN photodiode, such as series resistor noise, are negligible in comparison. The quantum noise $S_q(f)$, also called signal-dependent shot noise, is produced by the light-generating

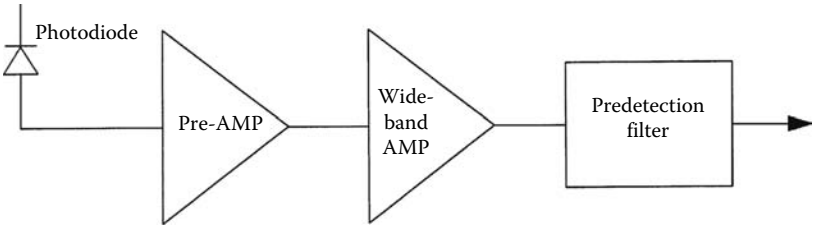


FIGURE 17.22 Front-end optical receiver.

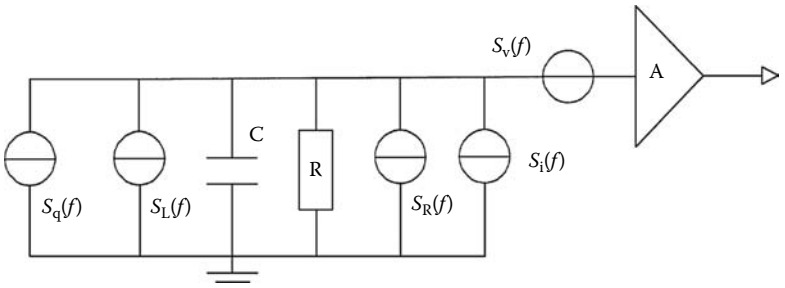


FIGURE 17.23 Noise equivalent circuit of the front-end optical receiver.

nature of photonic detection and has a spectral density $S_q(f) = 2qI_{pd}\Delta f$, where I_{pd} is the mean signal current arising from the Poisson statistics. The dark-current shot noise $S_D(f)$ arises in the photodiode bulk material. Even when there is no incident optical power, a small reverse leakage current still flows, resulting in shot noise with a spectral density $S_D(f) = 2qI_{DB}\Delta f$, where I_{DB} is the mean thermally generated dark current. The leakage shot noise $S_L(f)$ occurs because of surface effects around the active region, and is described by $S_L(f) = 2qI_{SL}\Delta f$, where I_{SL} is the mean surface leakage current.

17.4.1.2 Amplifier Noise

For a simple noise analysis, the pre- and postamplifiers in Figure 17.22 are merged to a single amplifier with a transfer function of $A_v(\omega)$. The input impedance of the amplifier is modeled as a parallel combination of R_{in} and C_{in}.

If the photodiode noise is negligibly small, the amplifier noise will dominate the whole receiver noise performance, as can be inferred from Figure 17.23. The equivalent noise current and voltage spectral densities of the amplifier are represented as S_i (A^2/Hz) and S_v (V^2/Hz), respectively.

17.4.1.3 Resistor Noise

The thermal noise generated by a resistor is directly proportional to the absolute temperature T and is represented by a series noise voltage generator or by a shunt noise current generator [21] of value:

$$\overline{v_R^2} = 4kTR\Delta f \quad \text{or} \quad \overline{i_R^2} = 4kT\frac{1}{R}\Delta f \tag{17.31}$$

where

k is Boltzmann's constant

R is the resistance

17.4.2 Receiver Performance Criteria

17.4.2.1 Equivalent Input Noise Current $\overline{\langle i_{eq}^2 \rangle}$

The transfer function from the current input to the amplifier output voltage is given by

$$Z_T(\omega) = \frac{V_{out}}{I_{pd}} = Z_{in}A_v(\omega) = \frac{R_{in}}{1j\omega R_{in}(C_{pd} + C_{in})}A_v(\omega) \tag{17.32}$$

where

C_{pd} is the photodiode capacitance

R_{in} and C_{in} are the input resistance and capacitance of the amplifier, respectively

Assuming that the photodiode noise contributions are negligible and that the amplifier noise sources are uncorrelated, the equivalent input noise current spectral density can be derived from Figure 17.23 as

$$S_{\text{eq}}(f) = S_i + \frac{S_v}{[Z_{\text{in}}]^2} = S_i + S_v\left[\frac{1}{R_{\text{in}}^2} + (2\pi f)^2(C_{\text{pd}} + C_{\text{in}})^2\right] \tag{17.33}$$

The total mean-square noise output voltage $\langle\overline{v_{\text{no}}^2}\rangle$ is calculated by combining Equations 17.32 and 17.33 as follows:

$$\langle\overline{v_{\text{no}}^2}\rangle = \int_0^\infty S_{\text{eq}}(f)|Z_{\text{T}}(f)|^2\,\text{d}f \tag{17.34}$$

This total noise voltage can be referred to the input of the amplifier by dividing it by the squared dc gain $|Z_{\text{T}}(0)|^2$ of the receiver, to give an equivalent input mean-square noise current:

$$\begin{aligned}\left\langle\overline{i_{\text{eq}}^2}\right\rangle &= \frac{\langle\overline{v_{\text{no}}^2}\rangle}{|Z_{\text{T}}(0)|^2} = \left(S_i + \frac{S_v}{R_{\text{in}}^2}\right)\int_0^\infty \frac{|Z_{\text{T}}(f)|^2}{|Z_{\text{T}}(0)|^2}\,\text{d}f + S_v[2\pi(C_{\text{pd}} + C_{\text{in}})]^2\int_0^\infty f^2\frac{|Z_{\text{T}}(f)|^2}{|Z_{\text{T}}(0)|^2}\,\text{d}f \\ &= \left(S_i + \frac{S_v}{R_{\text{in}}^2}\right)I_2B + [2\pi(C_{\text{pd}} + C_{\text{in}})]^2 I_3B^3S_v\end{aligned} \tag{17.35}$$

where

B is the operating bit-rate

$I_2(=0.56)$ and $I_3(=0.083)$ are the Personick second and third integrals, respectively, as given in Ref. [22]

According to Morikuni et al. [23], the Personick integral in Equation 17.35 is correct only if a receiver produces a raised-cosine output response from a rectangular input signal at the cut-off bit-rate above which the frequency response of the receiver is zero. However, the Personick integration method is generally preferred when comparing the noise (or sensitivity) performance of different amplifiers.

17.4.2.2 Optical Sensitivity

Optical sensitivity is defined as the minimum received optical power incident on a perfectly efficient photodiode connected to the amplifier, such that the presence of the amplifier noise corrupts on average only 1 bit per 10^9 bits of incoming data. Therefore, a detected power greater than the sensitivity level guarantees system operation at the desired performance. The optical sensitivity is predicted theoretically by calculating the equivalent input noise spectral density of the receiver, and is calculated [24] via Equation 17.36:

$$S = 10\log_{10}\left(Q\frac{hc}{q\lambda}\sqrt{\left\langle i_{\text{eq}}^2\right\rangle}\cdot\frac{1}{1\text{mW}}\right)(\text{dBm}) \tag{17.36}$$

where

h is Planck's constant

c is the speed of light

q is the electronic charge

λ (μm) is the wavelength of light in an optical fiber

$Q = \sqrt{\text{SNR}}$, where SNR represents the required signal-to-noise ratio (SNR)

The value of Q should be 6 for a bit error rate (BER) of 10^{-9}, and 7.04 for a BER of 10^{-12}. The relation between Q and BER is given by

$$\text{BER} = \frac{\exp(-Q^2/2)}{\sqrt{2\pi}Q} \tag{17.37}$$

Since the number of photogenerated electrons in a single bit is very large (more than 10^4) for optoelectronic integrated receivers [25], Gaussian statistics of the above BER equation can be used to describe the detection probability in PIN photodiodes.

17.4.2.3 SNR at the Photodiode Terminal [22]

Among the photodiode noise sources, quantum noise is generally dominant and can be estimated as

$$\left\langle \overline{i_n^2} \right\rangle_q = 2qI_{\text{pd}}B_{\text{eq}} \tag{17.38}$$

where

I_{pd} is the mean signal current
B_{eq} is the equivalent noise bandwidth

The SNR referred to the photodiode terminal is thus given by

$$\text{SNR} = \frac{I_{\text{pd}}^2}{\left\langle \overline{i_{\text{n}}^2} \right\rangle_{\text{pd}} + \frac{4kTB_{\text{eq}}}{R_{\text{B}}} + \left\langle \overline{i_{\text{eq}}^2} \right\rangle_{\text{amp}}} \tag{17.39}$$

where all noise contributions due to the amplifier are represented by the equivalent noise current $\left\langle \overline{i_{\text{eq}}^2} \right\rangle_{\text{amp}}$. It is often convenient to combine the noise contributions from the amplifier and the photodiode with the thermal noise from the bias resistor, by defining an NF:

$$\left\langle \overline{i_{\text{n}}^2} \right\rangle_{\text{pd}} + \frac{4kTB_{\text{eq}}}{R_{\text{B}}} + \left\langle \overline{i_{\text{eq}}^2} \right\rangle_{\text{amp}} = \frac{4kTB_{\text{eq}}\text{NF}}{R_{\text{B}}} \tag{17.40}$$

The SNR at the photodiode input is thus given by

$$\text{SNR} \cong \frac{I_{\text{pd}}^2 R_{\text{B}}}{4kTB_{\text{eq}}\text{NF}} \tag{17.41}$$

17.4.2.4 Intersymbol Interference

When a pulse passes through a band-limited channel, it gradually disperses. When the channel bandwidth is close to the signal bandwidth, the expanded rise and fall times of the pulse signal will cause successive pulses to overlap, deteriorating the system performance and giving higher error rates. This pulse overlapping is known as ISI. Even with raised signal power levels, the error performance cannot be improved [26].

In digital optical communication systems, sampling at the output must occur at the point of maximum signal in order to achieve the minimum error rate. The output pulse shape should therefore be chosen to maximize the pulse amplitude at the sampling instant and give a zero at other sampling points; that is, at multiples of $1/B$, where B is the data-rate. Although the best choice for this purpose is the sinc-function

pulse, in practice a raised-cosine spectrum pulse is used instead. This is because the sinc-function pulse is very sensitive to changes in the input pulse shape and variations in component values, and because it is impossible to generate an ideal sinc-function.

17.4.2.5 Dynamic Range

The DR of an optical receiver quantifies the range of detected power levels within which correct system operation is guaranteed. DR is conventionally defined as the difference between the minimum input power (which determines sensitivity) and the maximum input power (limited by overload level). Above the overload level, the bit-error-rate (BER) rises due to the distortion of the received signal.

17.4.3 Transimpedance (TZ) Amplifiers

High-impedance (HZ) amplifiers are effectively open-loop architectures, and exhibit a high gain but a relatively low bandwidth. The frequency response is similar to that of an integrator, and thus HZ amplifiers require an output equalizer to extend their frequency capabilities. In contrast, the transimpedance (TZ) configuration exploits resistive negative feedback, providing an inherently wider bandwidth and eliminating the need for an output equalizer. In addition, the use of negative feedback provides a relatively low input resistance and thus the architecture is less sensitive to the photodiode parameters. In a TZ amplifier, the photodiode bias resistor R_B can be omitted, since bias current is now supplied through the feedback resistor.

In addition to wider bandwidth, TZ amplifiers offer a larger DR because the transimpedance gain is determined by a linear feedback resistor, and not by a non-linear open-loop amplifier as is the case for HZ amplifiers. The DR of TZ amplifiers is set by the maximum voltage swing available at the amplifier output, provided no integration of the received signal occurs at the front end. Since the TZ output stage is a voltage buffer, the voltage swing at the output can be increased with high current operation. The improvement in DR in comparison to the HZ architecture is approximately equal to the ratio of open-loop to closed-loop gain [27]. Conclusively, the TZ configuration offers the better performance compromise compared to the HZ topology, and hence this architecture is preferred in optical receiver applications.

A schematic diagram of a TZ amplifier with PIN photodiode is shown in Figure 17.24. With an open-loop, high-gain amplifier and a feedback resistor, the closed-loop transfer function of the TZ amplifier is given by

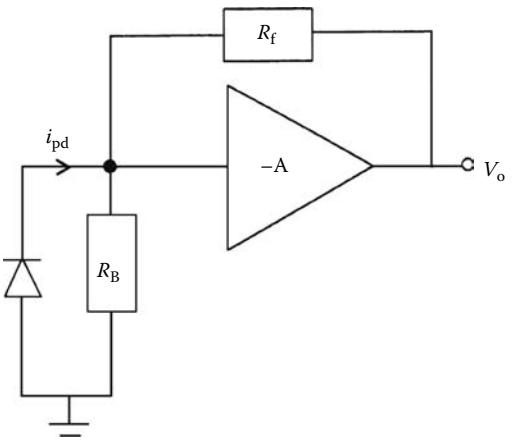


FIGURE 17.24 Schematic diagram of a transimpedance amplifier with photodiode.

$$Z_T(s) = \frac{-R_f}{\left(\frac{1+A}{A}\right) + sR_f\left[\frac{C_{in}+(1+A)C_f}{A}\right]} \cong \frac{R_f}{1 + sR_f\left(\frac{C_{in}}{A} + C_f\right)} \tag{17.42}$$

where

A is the open-loop mid-band gain of the amplifier which is assumed to be greater than unity
R_f is the feedback resistance
C_{in} is the total input capacitance of the amplifier including the photodiode and the parasitic capacitance
C_f represents the stray feedback capacitance

The −3 dB bandwidth of the TZ amplifier is approximately given by

$$f_{-3d} = \frac{(1+A)}{2\pi R_f C_T} \tag{17.43}$$

where C_T is the total input capacitance including the photodiode capacitance. The TZ amplifier can thus have wider bandwidth by increasing the open-loop gain, although the open-loop gain cannot be increased indefinitely without stability problems.

However, a trade-off between low noise and wide bandwidth exists, since the equivalent input noise current spectral density of TZ amplifier is given by

$$S_{eq}(f) = \frac{4kT}{R_f} + \frac{4kT}{R_B} + S_i(f) + S_v(f)\left[\left(\frac{1}{R_f} + \frac{1}{R_B}\right)^2 + (2\pi f)^2(C_{pd} + C_{in})^2\right] \tag{17.44}$$

where C_{in} is the input capacitance of the input transistor. Increasing the value of R_f reduces the noise current in Equation 17.44 but also shrinks the bandwidth in Equation 17.43. This conflict can be mitigated by making A in Equation 17.43 as large as the closed-loop stability allows [28]. However, the feedback resistance R_f cannot be increased indefinitely due to the DR requirements of the amplifier, since too large a feedback resistance causes the amplifier to be overloaded at high signal levels. This overloading can be avoided by using automatic gain control (AGC) circuitry, which automatically reduces the transimpedance gain in discrete steps to keep the peak output signal constant [27].

The upper limit of R_f is set by the peak amplitude of the input signal. Since the dc transimpedance gain is approximately equal to the feedback resistance R_f, the output voltage is given by $I_{pd} \times R_f$, where I_{pd} is the signal photocurrent. If this output voltage exceeds the maximum voltage swing at the output, the amplifier will be saturated and the output will be distorted, yielding bit errors. The minimum value of R_f is determined by the output signal level at which the performance of the receiver is degraded due to noise and offsets. For typical fiber-optic communication systems, the input signal power is unknown, and may vary from just above the noise floor to a large value enough to generate 0.5 mA at the detector diode [29].

The TZ configuration has some disadvantages over HZ amplifiers. The power consumption is fairly high, partly due to the broadband operation provided by negative feedback. A propagation delay exists in the closed-loop of the feedback amplifier that may reduce the phase margin of the amplifier and cause peaking in the frequency response. Additionally, any stray feedback capacitance C_f will further deteriorate the ac performance.

Among three types of TZ configuration in CMOS technology (common-source, common-drain, and common-gate TZ amplifiers), the common-gate configuration has potentially the highest bandwidth due to its inherently lower input resistance. Using a common-gate input configuration, the resulting amplifier bandwidth can be made independent of the photodiode capacitance (which is usually the limiting factor in achieving GHz preamplifier designs). Recently, a novel common-gate TZ amplifier has been demonstrated, which shows superior performance compared to various other configurations [30,31].

17.4.4 Layout for HF Operation

Wideband high-gain amplifiers have isolation problems irrespective of the choice of technology. Couplings from output to input, from the power supply rails, and from the substrate are all possible. Therefore, careful layout is necessary, and special attention must be given to stray capacitance, both on the integrated circuit and associated with the package [32].

17.4.4.1 Input/Output Isolation

For stable operation, a high level of isolation between I/O is necessary. Three main factors degrade the I/O isolation [33,34]: (1) capacitive coupling between I/O signal paths through the air and through the substrate; (2) feedback through the dc power supply rails and ground-line inductance; and (3) the package cavity resonance since at the cavity resonant frequency, the coupling between I/O can become very large.

In order to reduce the unwanted coupling (or to provide good isolation, typically more than 60 dB) between I/O, the I/O pads should be laid out to be diagonally opposite to each other on the chip with a thin ''left-to-right'' geometry between I/O. The small input signal enters on the left-hand side of the chip, while the large output signal exits on the far right-hand side. This helps to isolate the sensitive input stages from the larger signal output stages [35,36].

The use of fine line-widths and shielding are effective techniques to reduce coupling through the air. Substrate coupling can be reduced by shielding and by using a thin and low-dielectric substrate. Akazawa et al. [33] suggest a structure for effective isolation: a coaxial-like signal-line for high shielding, and a very thin dielectric dc feed-line structure for low characteristic impedance.

17.4.4.2 Reduction of Feedback through the Power Supply Rails

Careful attention should be given to the layout of power supply rails for stable operation and gain flatness. Power lines are generally inductive; thus, on-chip capacitive decoupling is necessary to reduce the HF power line impedance. However, a resonance between these inductive and capacitive components may occur at frequencies as low as several hundred megahertz, causing a serious dip in the gain–frequency response and an upward peaking in the isolation-frequency characteristics. One way to reduce this resonance is to add a series damping resistor to the power supply line, making the Q factor of the LC resonance small. Additionally, the power supply line should be widened to reduce the characteristic impedance/inductance. In practice, if the characteristic impedance is as small as several ohms, the dip and peaking do not occur, even without resistive termination [33].

Resonance also occurs between the IC pad capacitance (C_{pd}) and the bond-wire inductance (L_{bond}). This resonance frequency is typically above 2 GHz in miniature RF packages. Also in layout, the power supply rails of each IC chip stage should be split from the other stages in order to reduce the parasitic feedback (or coupling effect through wire-bonding inductance), which causes oscillation [34]. This helps to minimize crosstalk through power supply rail. The IC is powered through several pads and each pad is individually bonded to the power supply line.

17.4.4.3 I/O Pads

The bond pads on the critical signal path (e.g., input pad and output pads) should be made as small as possible to minimize the pad-to-substrate capacitance [35]. A floating n-well placed underneath the pad will further reduce the pad capacitance since the well capacitance will appear in series with the pad capacitance. This floating well also prevents the pad metal from spiking into the substrate.

17.4.4.4 High-Frequency (HF) Ground

The best possible HF grounds to the sources of the driver devices (and hence the minimization of interstage crosstalk) can be obtained by separate bonding of each source pad of the driver MOSFETs to the ground plane that is very close to the chip [36]. A typical bond-wire has a self-inductance of a few nH,

which can cause serious peaking within the bandwidth of amplifiers or even instability. By using multiple bond-wires in parallel, the ground-line inductance can be reduced to less than 1 nH.

17.4.4.5 Flip-Chip Connection

In noisy environments, the noise-insensitive benefits of optical fibers may be lost at the receiver connection between the photodiode and the preamplifier. Therefore, proper shielding, or the integration of both components onto the same substrate, is necessary to prevent this problem. However, proper shielding is costly, while integration restricts the design to GaAs technologies.

As an alternative, the flip-chip interconnection technique using solder bumps has been used [37,38]. Small solder bumps minimize the parasitics due to the short interconnection lengths and avoid damages by mechanical stress. Also, it needs relatively low-temperature bonding and hence further reduces damage to the devices. Easy alignment and precise positioning of the bonding can be obtained by a self-alignment effect. Loose chip alignment is sufficient because the surface tension of the molten solder during reflow produces precise self-alignment of the pads [34]. Solder bumps are fabricated onto the photodiode junction area to reduce parasitic inductance between the photodiode and the preamplifier.

17.5 Fundamentals of RF Power Amplifier Design

17.5.1 PA Requirements

An important functional block in wireless communication transceivers is the power amplifier (PA). The transceiver PA takes as input the modulated signal to be transmitted, and amplifies this to the power level required to drive the antenna. Because the levels of power required to transmit the signal reliably are often fairly high, the PA is one of the major sources of power consumption in the transceiver. In many systems, power consumption may not be a major concern, as long as the signal can be transmitted with adequate power. For battery-powered systems, however, the limited amount of available energy means that the power consumed by all devices must be minimized so as to extend the transmit time. Therefore, power efficiency is one of the most important factors when evaluating the performance of a wireless system.

The basic requirement for a PA is the ability to work at low supply voltages as well as high operating frequencies, and the design becomes especially difficult due to the trade-offs between supply voltage, output power, distortion, and power efficiency that can be made. Moreover, since the PA deals with large signals, small-signal analysis methods cannot be applied directly. As a result, both the analysis and the design of PAs are challenging tasks.

This section will first present a study of various configurations employed in the design of state-of-the-art nonlinear RF PAs. Practical considerations toward achieving full integration of PAs in CMOS technology will also be highlighted.

17.5.2 Power Amplifier Classification

PAs currently employed for wireless communication applications can be classified into two categories: linear PAs and nonlinear PAs. For linear PAs, the output signal is controlled by the amplitude, frequency, and phase of the input signal. Conversely, for nonlinear PAs, the output signal is only controlled by the frequency of input signal.

Conventionally, linear PAs can be classified as Class A, Class B, or Class AB. These PAs produce a magnified replica of the input signal voltage or current waveform, and are typically used where accurate reproduction of both the envelope and the phase of the signal is required. However, either poor power efficiency or large distortion prevents them from being extensively employed in wireless communications.

Many applications do not require linear RF amplification. Gaussian minimum shift keying (GMSK) [39], the modulation scheme used in the European standard for mobile communications (GSM), is an

example of constant envelope modulation. In this case, the system can make use of the greater efficiency and simplicity offered by nonlinear PAs. The increased efficiency of nonlinear PAs, such as Class C, Class D, and Class E, results from techniques that reduce the average collector voltage–current product (i.e., power dissipation) in the switching device. Theoretically, these switching-mode PAs have 100% power efficiency since, ideally, there is no power loss in the switching device.

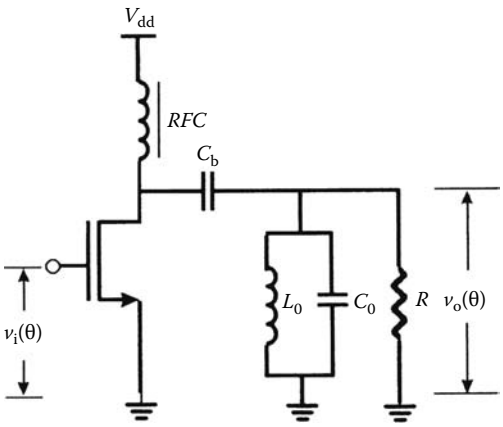


FIGURE 17.25 Single-ended Class A PA.

17.5.2.1 Linear Power Amplifiers

17.5.2.1.1 Class A

The basic structure of the Class A PA is shown in Figure 17.25 [40]. For Class A amplification, the conduction angle of the device is 360°, that is, the transistor is in its active region for the entire input cycle. The serious shortcoming with Class A PAs is their inherently poor power efficiency, since the transistor is always dissipating power. The efficiency of a single-ended Class A PA is ideally limited to 50%. However, in practice, few designs can reach this ideal efficiency due to additional power loss in the passive components. In an inductorless configuration, the efficiency is only about 25% [41].

17.5.2.1.2 Class B

A PA is defined as Class B when the conduction angle for each transistor of a push–pull pair is 180° during any one cycle. Figure 17.26 shows an inductorless Class B PA. Since each transistor only conducts for half of the cycle, the output suffers crossover distortion due to the finite threshold voltage of each transistor. When no signal is applied, there is no current flowing; as a result, any current through either device flows directly to the load, thereby maximizing the efficiency. The ideal efficiency can reach 78% [41], allowing this architecture to be of use in applications where linearity is not the main concern.

17.5.2.1.3 Class AB

The basic idea of Class AB amplification is to preserve the Class B push–pull configuration while improving the linearity by biasing each device slightly above threshold. The implementation of Class AB PAs is similar to Class B configurations. By allowing the two devices to conduct current for a short period, the output voltage waveform during the crossover period can be smoothed, which thus reduces the crossover distortion of the output signal.

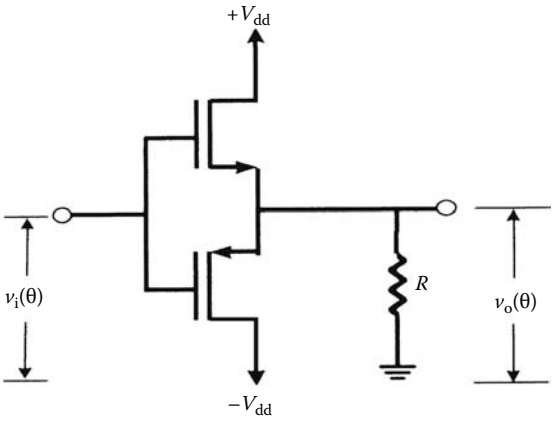


FIGURE 17.26 Inductorless Class B PA.

17.5.2.2 Nonlinear Power Amplifiers

17.5.2.2.1 Class C

A Class C PA is the most popular nonlinear PA used in the RF band. The conduction angle is less than 180° since the switching transistor is biased on the verge of conduction. A portion of the input signal will make the transistor operate in the amplifying region, and thus the drain current of the transistor is a pulsed signal.

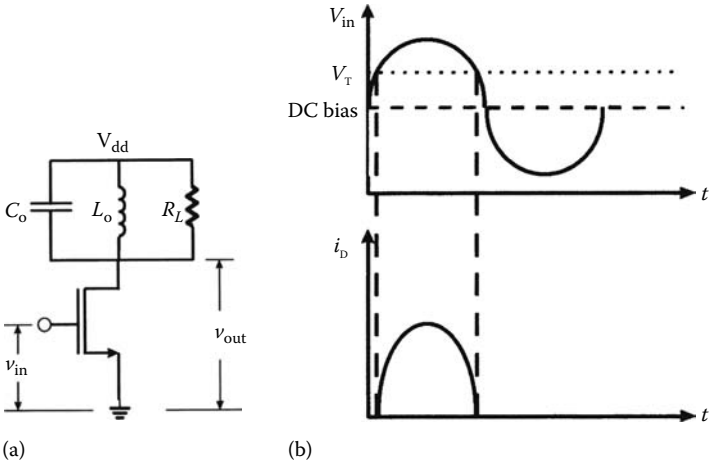


FIGURE 17.27 (a) Class C PA and (b) Class C waveforms.

Figure 17.27a and b shows the basic configuration of a Class C PA and its corresponding waveforms; clearly, the input and output voltages are not linearly related.

The efficiency of an ideal Class C amplifier is 100% since at any point in time, either the voltage or the current waveforms are zero. In practice, this ideal situation cannot be achieved, and the power efficiency should be maximized by reducing the power loss in the transistor. That is, minimize the current through the transistor when the voltage across the output is high, and minimize the voltage across the output when the current flows through the device.

17.5.2.2.2 Class D

A Class D amplifier employs a pair of transistors and a tuned output circuit, where the transistors are driven to act as a two-pole switch and the output circuit is tuned to the switching frequency. The theoretical power efficiency is 100%. Figure 17.28 shows the voltage-switching configuration of a Class D amplifier. The input signals of transistors Q_1 and Q_2 are out of phase, and consequently when Q_1 is on, Q_2 is off, and vice versa. Since the load network is a tuned circuit, we can assume that it provides little

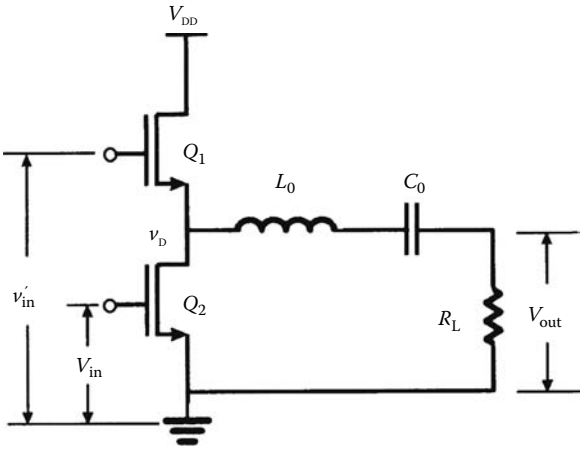


FIGURE 17.28 Class D PA.

impedance to the operating frequency of the voltage v_d and high impedance to other harmonics. Since v_d is a square wave, its Fourier expansion is given by

$$v_d(\omega t) = V_{dc}\left[\frac{1}{2} + \frac{2}{\pi}\sin(\omega t) + \frac{2}{3\pi}\sin(3\omega t)\ldots\right] \tag{17.45}$$

The impedance of the RLC series load at resonance is equal to R_L, and thus the current is given by

$$i_L(\omega t) = \frac{2V_{dc}}{\pi R_L}\sin(\omega t) \tag{17.46}$$

Each of the devices carries the current during one half of the switching cycle. Therefore, the output power is given by

$$P_o = \frac{2}{\pi^2}\frac{V_{dc}^2}{R_L} \tag{17.47}$$

Design efforts should focus on reducing the switching loss of both transistors as well as generating the input driving signals.

17.5.2.2.3 Class E

The idea behind the Class E PA is to employ nonoverlapping output voltage and output current waveforms. Several criteria for optimizing the performance can be found in Ref. [42]. Following these guidelines, Class E PAs have high power efficiency, simplicity, and relatively high tolerance to circuit variations [43]. Since there is no power loss in the transistor as well as in the other passive components, the ideal power efficiency is 100%. Figure 17.29 shows a Class E PA, and the corresponding waveforms are given in Figure 17.30.

The Class E waveforms indicate that the transistor should be completely off before the voltage across it changes, and that the device should be completely on before it starts to allow current to flow through it. Refs. [44,45] demonstrate practical Class E operation at RF frequencies using a GaAs process.

17.5.3 Practical Considerations for RF Power Amplifiers

More recently, single-chip solutions for RF transceivers have become a goal for modern wireless communications due to potential savings in power, size, and cost. CMOS must clearly be the technology

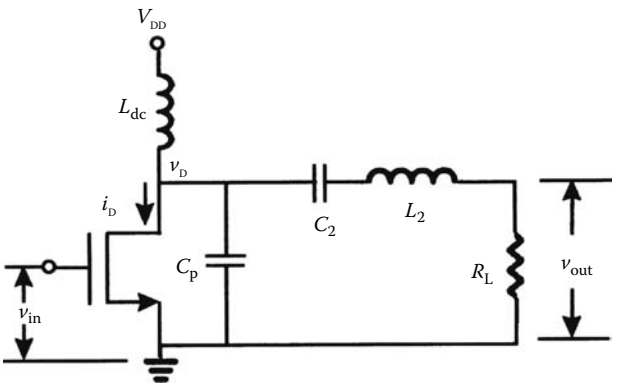


FIGURE 17.29 Class E PA.

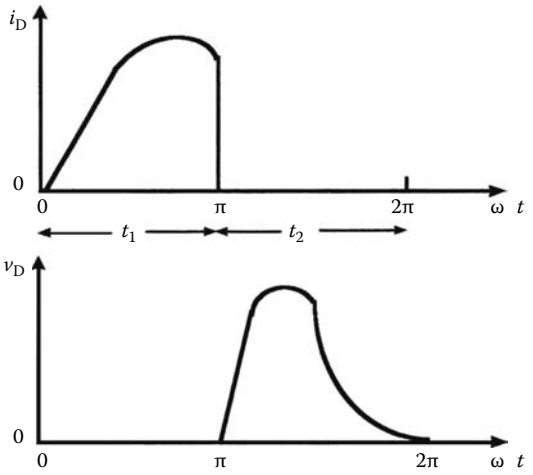


FIGURE 17.30 Waveforms of Class E operation.

of choice for a single-chip transceiver due to the large amount of digital baseband processing required. However, the PA design presents a bottleneck toward full integration, since CMOS PAs are still not available. The requirements of low supply voltage, gigahertz-band operation, and high output power make the implementation of CMOS PAs very demanding. The proposal of ''microcell'' communications may lead to a relaxed demand for output power levels that can be met by designs such as that described in Ref. [46], where a CMOS Class C PA has demonstrated up to 50% power efficiency with 20 mW output power.

Nonlinear PAs seem to be popular for modern wireless communications due to their inherent high power efficiency. Since significant power losses occur in the passive inductors as well as the switching devices, the availability of on-chip, low-loss passive inductors is important. The implementation of CMOS on-chip spiral inductors has therefore become an active research topic [47].

Due to the poor spectral efficiency of a constant envelope modulation scheme, the high power efficiency benefit of nonlinear PAs is eliminated. A recently proposed linear transmitter using a nonlinear PA may prove to be an alternative solution [48]. The development of high mobility devices such as SiGe HBTs has led to the design of PAs demonstrating output power levels up to 23 dBm at 1.9 GHz with power-added efficiency of 37% [49]. Practical PA designs require that much attention be paid to issues of package and harmonic terminations. Power losses in the matching networks must be absolutely minimized, and trade-offs between power-added efficiency and linearity are usually achieved through impedance matching. Although GaAs processes provide low-loss impedance matching structures on the semi-insulating substrate, good shielding techniques for CMOS may prove to be another alternative.

17.5.4 Conclusions

Although linear PAs provide conventional ''easy-design'' characteristics and linearity for modulation schemes such as $\pi/4$-DQPSK, modern wireless transceivers are more likely to employ nonlinear PAs due to their much higher power efficiency. As the development of high-quality on-chip passive components makes progress, the trend toward full integration of the PA is becoming increasingly plausible.

The rapid development of CMOS technology seems to be the most promising choice for PA integration, and vast improvements in frequency performance have been gained through device scaling. These improvements are expected to continue as silicon CMOS technologies scale further, driven by the

demand for high-performance microprocessors. The further development of high mobility devices such as SiGe HBTs may finally see GaAs MOSFETs being replaced by wireless communication applications, since SiGe technology is compatible with CMOS.

17.6 Applications of High-Q Resonators in IF-Sampling Receiver Architectures

Transconductance-C (gm-C) filters are currently the most popular design approach for realizing continuous-time filters in the intermediate frequency range in telecommunications systems. This section will consider the special application area of high-Q resonators for receiver architectures employing IF sampling.

17.6.1 IF Sampling

A design approach for contemporary receiver architectures that is currently gaining popularity is IF digitization, whereby low-frequency operations such as second mixing and filtering can be performed more efficiently in the digital domain. A typical architecture is shown in Figure 17.31. The IF signal is digitized, multiplied with the quadrature phases of a digital sinusoid, and lowpass filtered to yield the quadrature baseband signals. Since processing takes place in the digital domain, I/Q mismatch problems are eliminated. The principal issue in this approach, however, is the performance required from the A/D converter (ADC). Noise referred to the input of the ADC must be very low so that selectivity remains high. At the same time, the linearity of the ADC must be high to minimize corruption of the wanted signal through intermodulation effects. Both the above requirements should be achieved at an input bandwidth commensurate with the value of the IF frequency, and at an acceptable power budget.

Oversampling has become popular in recent years because it avoids many of the difficulties encountered with conventional methods for A/D and D/A conversion. Conventional converters are often difficult to implement in fine-line, very large-scale integration (VLSI) technology, because they require precise analog components and are very sensitive to noise and interference. In contrast, oversampling converters trade off resolution in time for resolution in amplitude, in such a way that the imprecise nature of the analog circuits can be tolerated. At the same time, they make extensive use of digital signal processing power, taking advantage of the fact that fine-line VLSI is better suited for providing fast digital circuits than for providing precise analog circuits. Therefore, IF-digitization techniques utilizing oversampling Sigma-Delta modulators are very well suited to modern submicron CMOS technologies, and their potential has made them the subject of active research.

Most Delta-Sigma modulators are implemented with discrete-time circuits, switched-capacitor (SC) implementations being by far the most common. This is mainly due to the ease with which monolithic SC filters can be designed, as well as the high linearity which they offer. The demand for high-speed $\Sigma\Delta$ oversampling ADCs, especially for converting bandpass signals, makes it necessary to look for a

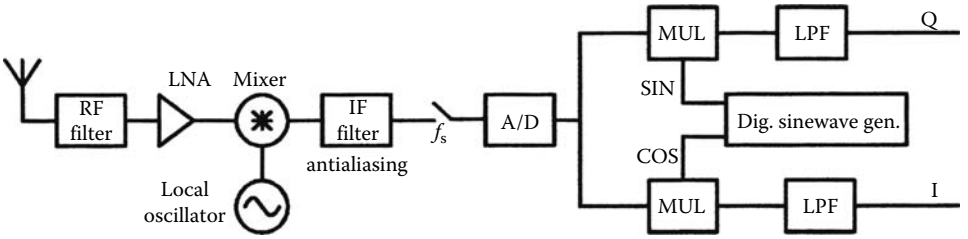


FIGURE 17.31 IF-sampling receiver.

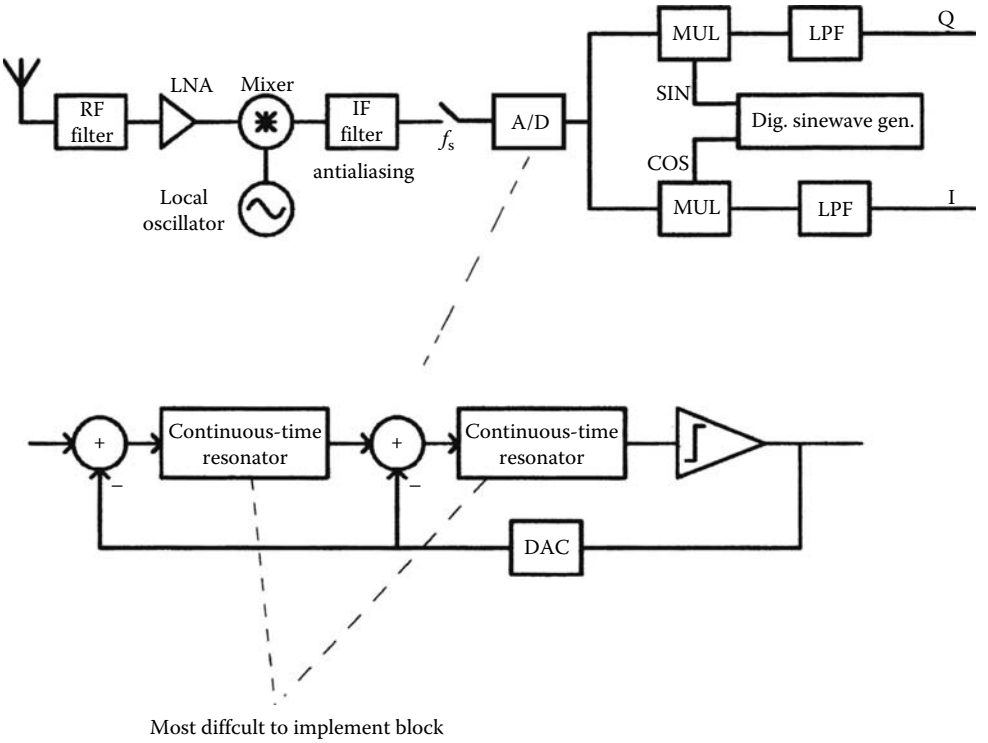


FIGURE 17.32 Continuous-time ΣΔ A/D in IF-sampling receiver.

technique that is faster than SC. This demand has stimulated researchers to develop a method for designing continuous-time ΣΔ ADCs (not checked with MS to identify that there is an missing symbol). Although continuous-time modulators are not easy to integrate, they possess a key advantage over their discrete-time counterparts. The sampling operation takes place inside the modulator loop, making it is possible to "noise-shape" the errors introduced by sampling, and provide a certain amount of anti-aliasing filtering at no cost. On the other hand, they are sensitive to memory effects in the DACs and are very sensitive to jitter. They must also process continuous-time signals with high linearity. In communications applications, meeting the latter requirement is complicated by the fact that the signals are located at very high frequencies.

As shown in Figure 17.32, integrated bandpass implementations of continuous-time modulators require integrated continuous-time resonators to provide the noise shaping function. The gm-C approach of realizing continuous-time resonators offers advantages of complete system integration and total design freedom. However, the design of CMOS high-Q high-linearity resonators at the tens of megahertz is very challenging. Since the linearity of the modulator is limited by the linearity of the resonators utilized, the continuous-time resonator is considered to be the most demanding analog subblock of a bandpass continuous-time Sigma-Delta modulator. Typical specifications for a gm-C resonator used to provide the noise-shaping function in a ΣΔ modulator in a mobile receiver (see Figure 17.32) are summarized in Table 17.4.

17.6.2 Linear Region Transconductor Implementation

The implementation of fully integrated, high-selectivity filters operating at tens to hundreds of megahertz provides benefits for wireless transceiver design, including chip area economy and cost reduction. The main disadvantages of on-chip active filter implementations when compared to off-chip passives include

TABLE 17.4 Fully Integrated Continuous-Time Resonator Specifications

Resonator Specifications	
Center frequency	50 MHz
Quality factor	50
SFDR	>30 dB
Power dissipation	Minimal

increased power dissipation, deterioration in the available DR with increasing Q, and Q and resonant frequency integrity (because of process variations, temperature drifts, and aging, automatic tuning is often unavoidable, especially in high-Q applications). The transconductor-capacitor (gm-C) technique is a popular technique for implementing high-speed continuous-time filters and is widely used in many industrial applications [52]. Because gm-C filters are based on integrators built from an open-loop transconductance amplifier driving a capacitor, they are typically very fast but have limited linear DR. Linearization techniques that reduce distortion levels can be used, but often lead to a compromise between speed, DR, and power consumption.

As an example of the trade-offs in design, consider the transconductor shown in Figure 17.33. This design consists of a main transconductor cell (M1, M2, M3, M4, M10, M11, and M14) with a negative resistance load (M5, M6, M7, M8, M9, M12, and M13). Transistors M1 and M2 are biased in the triode region of operation using cascode devices M3 and M4 and determine the transconductance gain of the cell. In the triode region of operation, the drain current versus terminal voltage relation can be approximated (for simple hand calculations) as $I_D = K[2(V_{GS} - V_T)V_{DS} - V_{DS}^2]$, where K and V_T are the transconductance parameter and the threshold voltage, respectively. Assuming that V_{DS} is constant for both M1 and M2, both the differential mode and the common-mode transconductance gains can be derived as $G_{DM} = G_{CM} = 2KV_{DS}$, which can thus be tuned by varying V_{DS}.

The high value of common-mode transconductance is undesirable since it may result in regenerative feedback loops in high-order filters. To improve the CMRR transistor and avoid the formation of such loops, M10 is used to bias the transconductor, thus transforming it from a pseudo differential to a fully differential transconductor [53]. Transistors M11 and M14 constitute a floating voltage source, thus maintaining a constant drain-source voltage for M1 and M2.

The nonlinearities in the voltage-to-current transfer of this stage are mainly due to three effects. The first is the finite impedance levels at the sources of the cascode devices, which cause a signal-dependent variation of the corresponding drain-source voltages of M1 and M2. A fast floating voltage source and large

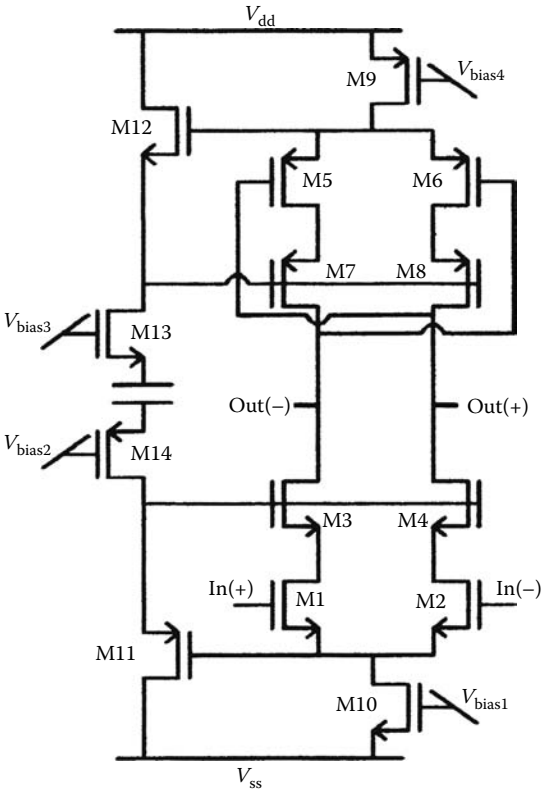


FIGURE 17.33 Triode region transconductor.

cascode transistors therefore need to be used to minimize this nonlinearity. The second cause of nonlinearity is the variation of carrier mobility μ of the input devices M1 and M2 with $V_{GS} - V_T$, which becomes more apparent when short-channel devices are used ($K = \mu \cdot C_{ox} \cdot W/2 \cdot L$) . A simple first-order model for transverse-field mobility degradation is given by $\mu = \mu_0/(1 + \theta \cdot (V_{GS} - V_T))$, where μ_0 and θ are the zero-field mobility and the mobility reduction parameter, respectively. Using this model, the third-order distortion can be determined by a Maclaurin series expansion as $\theta^2/4(1 + \theta(V_{CM} - V_T))$ [54]. This expression cannot be regarded as exact, although it is useful to obtain insight. Furthermore, it is valid only at low frequencies, where reactive effects can be ignored and the coefficients of the Maclaurin series expansion are frequency independent. At high frequencies or when very low values of distortion are predicted by the Maclaurin series method, a generalized power series method (Volterra series) must be employed [55,56]. Finally, a further cause of nonlinearity is a mismatch between M1 and M2, which can be minimized by good layout. A detailed linearity analysis of this transconductance stage is presented in Ref. [15].

To provide a load for the main transconductor cell, a similar cell implemented by p-devices is used. The gates of the linear devices M5 and M6 are now cross-coupled with the drains of the cascode devices M7 and M8. In this way, weak positive feedback is introduced. The differential-mode output resistance can now become negative and is tuned by the V_{DS} of M5 and M6 (M12 and M13 form a floating voltage source), while the common-mode output resistance attains a small value.

When connected to the output of the main transconductor cell as shown in Figure 17.33, the cross-coupled p-cell forms a high-ohmic load for differential signals and a low-ohmic load for common-mode signals, resulting in a controlled common-mode voltage at the output [54,57]. CMRR can be increased even further using M10, as described previously. Transistor M9 is biased in the triode region of operation and is used to compensate the offset common-mode voltage at the output.

The key performance parameter of an integrator is the phase shift at its unity-gain frequency. Deviations from the ideal $-90°$ phase include phase lead due to finite dc gain and phase lag due to HF parasitic poles. In the transconductor design of Figure 17.33, dc gain is traded for phase accuracy, thus compensating the phase lag introduced by the parasitic poles. The reduction in dc gain for increased phase accuracy is not a major problem for bandpass filter applications, since phase accuracy at the center frequency is extremely important while dc gain has to be adequate to ensure that attenuation specifications are met at frequencies below the passband.

From simulation results using parameters from a 0.8 μm CMOS process, with the transconductor unity-gain frequency set at 50 MHz, third-order intermodulation components were observed at -78 dB with respect to the fundamental signals (two input signals at 49.9 and 50.1 MHz were applied, each at 50 mVpp).

17.6.3 gm-C Bandpass Biquad

17.6.3.1 Filter Implementation

The implementation of on-chip high-Q resonant circuits presents a difficult challenge. Integrated passive inductors have generally poor quality factors, which limits the Q of any resonant network in which they are employed. For applications in the hundreds of megahertz to a few gigahertz, one approach is to implement the resonant circuit using low-Q passive on-chip inductors with additional Q-enhancing circuitry. However, for lower frequencies (tens of megahertz), on-chip inductors occupy a huge area and this approach is not attractive.

As disscussed above, an alternative method is to use active circuitry to eliminate the need for inductors. gm-C-based implementations are attractive due to their high-speed potential and good tunability. A bandpass biquadratic section based upon the transconductor of Figure 17.33 is shown in Figure 17.34. The transfer function of Figure 17.34 is given by

$$\frac{V_o}{V_i} = \frac{g_{mi} \cdot R_o}{(R_o \cdot C)^2} \cdot \frac{(1 + s \cdot R_o \cdot C)}{\left\{ s^2 + s\frac{2 \cdot R_o + g_m^2 \cdot R_o^2 \cdot R}{R_o^2 \cdot C} + \frac{1 + g_m^2 \cdot R_o^2}{R_o^2 \cdot C^2} \right\}} \tag{17.48}$$

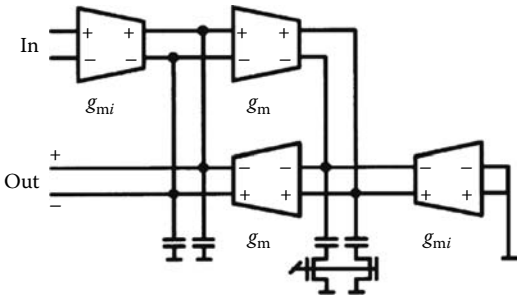


FIGURE 17.34 Biquad bandpass.

R_o represents the total resistance at the nodes due to the finite output resistance of the transconductors. R represents the effective resistance of the linear region transistors in the transconductor (see Figure 17.33), and is used here to introduce damping and control the Q. From Equation 17.48, it can be shown that $\omega_o \approx g_m/C$, $Q \approx g_m \cdot R_o/(2 + R_o \cdot R \cdot g_m^2)$, $Q_{max} = Q|_{r=0} = (g_m \cdot R_o)/2$ and $A_o = g_{mi} \supseteq Q$. Thus, g_m is used to set the central frequency, R is used to control the Q, and g_{mi} controls the bandpass gain A_o. A dummy g_{mi} is used to provide symmetry and thus better stability due to process variations, temperature, and aging.

One of the main problems when implementing high-Q HF resonators is maintaining the stability of the center frequency ω_o and the quality factor Q. This problem calls for very careful layout and the implementation of an automatic tuning system. Another fundamental limitation regarding available DR occurs: namely, that the DR of high-Q gm-C filters has been found to be inversely proportional to the filter Q [57]. The maximum DR is given by

$$\mathrm{DR} = \frac{V_{max}^2}{V_{noise}^2} = \frac{V_{max}^2 \cdot C}{4 \cdot k \cdot T \cdot \xi \cdot Q} \tag{17.49}$$

where

V_{max} is the maximum rms voltage across the filter capacitors
C is the total capacitance
k is Boltzman's constant
T is the absolute temperature
ξ is the noise factor of the active circuitry ($\xi = 1$ corresponds to output noise equal to the thermal noise of a resistor of value $R = 1/g_m$, where g_m is the transconductor value used in the filter)

In practice, the DR achieved will be less than this maximum value due to the amplification of both noise and intermodulation components around the resonant frequency. This is a fundamental limitation, and the only solution is to design the transconductors for low noise and high linearity. The linearity performance in narrowband systems is characterized by the spurious-free dynamic range (SFDR). SFDR is defined as the SNR when the power of the third-order intermodulation products equals the noise power. As shown in Ref. [15], the SFDR of the resonator in Figure 17.34 is given by

$$\mathrm{SFDR} = \frac{1}{2(k \cdot T)^{2/3}} \left(\frac{3 \cdot V_{o,peak}^2 \cdot C}{4 \cdot \xi \cdot IM_{3,int}} \right)^{2/3} \frac{1}{Q^2} \tag{17.50}$$

where $IM_{3,int}$ is the third-order intermodulation point of the integrator used to implement the resonator. The SFDR of the resonator thus deteriorates by 6 dB if the quality factor is doubled, assuming that the output swing remains the same. In contrast, implementing a resonant circuit using low-Q passive on-chip

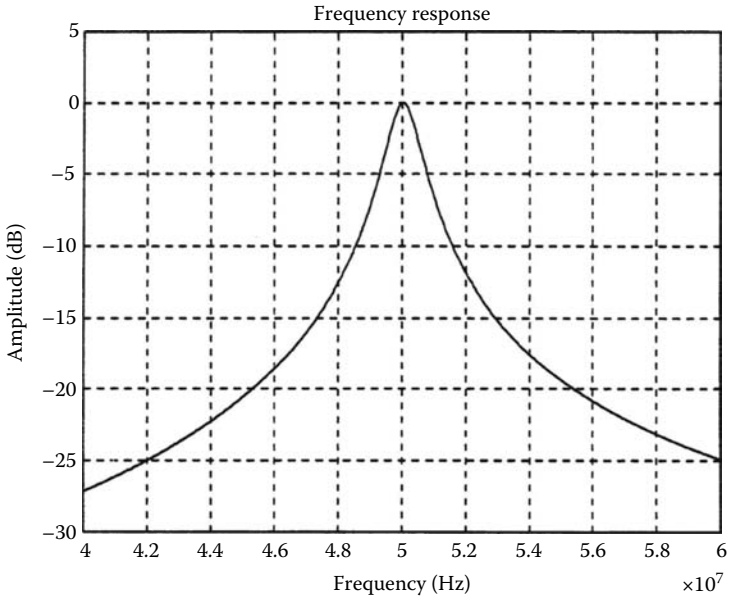


FIGURE 17.35 Simulated bandpass frequency response.

inductors with additional Q-enhancing circuitry leads to a DR amplified by a factor Q_o, where Q_o is the quality factor of the on-chip inductor itself [59]. However, as stated above, for frequencies in the tens of megahertz, on-chip inductors occupy a huge area and thus the Q_o improvement in DR is not high enough to justify the area increase.

17.6.3.2 Simulation Results

To confirm operation, the filter shown in Figure 17.34 has been simulated in HSPICE using process parameters from a commercial 0.8 μm CMOS process. Figure 17.35 shows the simulated frequency and phase response of the filter for a center frequency of 50 MHz and a quality factor of 50. Figure 17.36 shows the simulated output of the filter when the input consists of 2 tons at 49.9 and 50.1 MHz, respectively, each at 40 mVpp. At this level of input signal, the third-order intermodulation components were found to be at the same level as the noise. Thus, the predicted SFDR is about 34 dB with $Q = 50$. Table 17.5 summarizes the simulation results.

17.7 Log-Domain Processing

17.7.1 Instantaneous Companding

The concept of *instantaneous companding* is an emerging area of interest within the field of analog integrated circuit design. Currently, the main area of application for this technique is the implementation of continuous-time, fully integrated filters with wide DR, HF potential, and wide tunability.

With the drive toward lower supply voltages and higher operating frequencies, traditional analog integrated circuit design methodologies are proving inadequate. Conventional techniques to linearize inherently nonlinear devices require an overhead in terms of increased power consumption or reduced operating speed. Recently, the use of companding, originally developed for audio transmission, has been proposed as an elegant solution to the problem of maintaining DR and HF operation under low supply

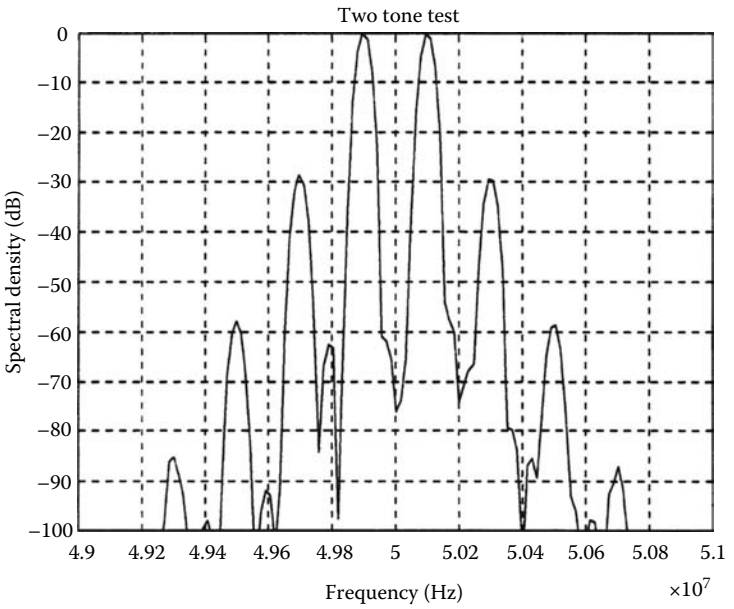


FIGURE 17.36 Simulated two-tone intermodulation test.

TABLE 17.5 Simulation Results

Power dissipation (supply voltage = 5 V)	12.5 mW
Common-mode output offset	<1 mV
Center frequency	50 MHz
Quality factor	50
Output noise voltage (integrated over the band from 40 to 60 MHz with $Q = 50$)	500 μV_{rms}
Output signal voltage (so that intermodulation components are at the same level as the noise, $Q = 50$)	25.2 mV_{rms}
SFDR ($Q = 50$)	34 dB

voltage [50]. In this approach, nonlinear amplifiers are used to compress the DR of the input signal to ensure, for example, that signal levels always remain above a certain noise threshold but below levels that may cause overload. The overall system must thus adapt its operation to ensure that input–output linearity is maintained as the instantaneous signal level alters. Although the system is input–output linear, signals internal to the system are inherently nonlinear. Companding has traditionally been realized in two ways—syllabic and instantaneous—depending on how the system adapts in response to the changing input signal level. In *syllabic* companding, the level of compression (and expansion) is a nonlinear function of a slowly varying property of the signal (e.g., envelope or power). In contrast, *instantaneous* companding adapts the compression and expansion ratios instantaneously with the changing input signal amplitude.

Perhaps the best-known recent example of this approach is the log-domain technique, where the exponential I–V characteristics of bipolar junction transistors (BJTs) are directly exploited to implement

input–output linear filters. Since the large signal device equations are utilized, there is no need for small-signal operation or local linearization techniques, and the resulting circuits have the potential for wide DR and HF operation under low power supply voltages. Log-domain circuits are generally implemented using BJTs and capacitors only and thus are inherently suitable for integration. The filter parameters are easily tunable, which makes them robust. In addition, the design procedure is systematic, which suggests that these desirable features can be reproduced for different system implementations. Section 17.7.2 provides an introduction to the synthesis of log-domain filters and highlights various performance issues. Interested readers are advised to consult Refs. [61–76] for a more detailed treatment of the subject.

17.7.2 Log-Domain Filter Synthesis

The synthesis of log-domain filters using state-space transformations was originally proposed by Frey [62]. As an example of this methodology, consider a biquad filter with the following transfer function:

$$Y(s) = \frac{s\omega_o U_1(s) + \omega_o^2 U_2(s)}{s^2 + (\omega_o/Q)s + \omega_o^2} \tag{17.51}$$

Thus, using u_1 as input gives a bandpass response at the output y, while using u_2 as input gives a lowpass response. This system can also be described by the following state space equations:

$$\begin{aligned} \dot{x}_1 &= -(\omega_o/Q)x_1 - \omega_o x_2 + \omega_o u_1 \\ \dot{x}_2 &= -\omega_o x_1 - \omega_o u_2 \\ y &= x_1 \end{aligned} \tag{17.52}$$

where a ''dot'' denotes differentiation in time. To transform these linear state equations into nonlinear nodal equations that can be directly implemented using bipolar transistors, the following exponential transformations are defined:

$$\begin{aligned} x_1 &= I_S \exp\left(\frac{V_1}{V_t}\right) \quad & u_1 &= \left(\frac{I_S^2}{I_o}\right) \exp\left(\frac{V_{in1}}{V_t}\right) \\ x_2 &= I_o \exp\left(\frac{V_2}{V_t}\right) \quad & u_2 &= I_S \exp\left(\frac{V_{in2}}{V_t}\right) \end{aligned} \tag{17.53}$$

These transformations map the linear state variables to currents flowing through bipolar transistors (BJTs) biased in the active region. I_S represents the BJT reverse saturation current, while V_t is the thermal voltage. Substituting these transformations into the state (Equation 17.52) gives

$$\begin{aligned} C\dot{V}_1 &= -\frac{I_o}{Q} - \frac{I_o^2}{I_S} \exp\left(\frac{V_2 - V_1}{V_t}\right) + I_S \exp\left(\frac{V_{o1} - V_1}{V_t}\right) \\ C\dot{V}_2 &= I_S \exp\left(\frac{V_1 - V_2}{V_t}\right) - I_S \exp\left(\frac{V_{o2} - V_2}{V_t}\right) \\ y &= I_S \exp\left(\frac{V_1}{V_t}\right) \end{aligned} \tag{17.54}$$

In Equation 17.54, a tuning current $I_o = C\omega_o V_t$ is defined, where C is a scaling factor which represents a capacitance. The linear state space equations have thus been transformed into a set of nonlinear nodal equations, and the task for the designer is now to realize a circuit architecture that will implement these

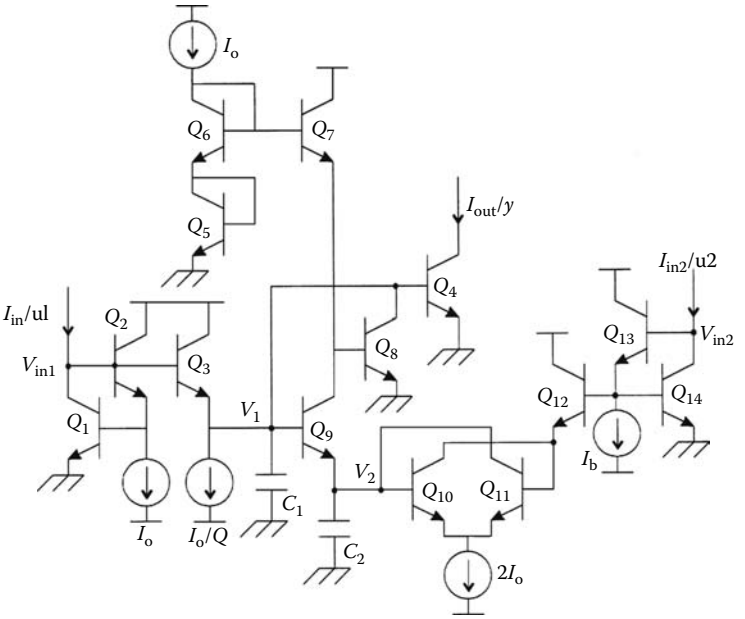


FIGURE 17.37 A log-domain biquadratic filter. A lowpass response is obtained by applying a signal at I_{u2} and keeping I_{u1} constant, while a bandpass response is obtained by applying a signal at I_{u1} and keeping I_{u2} constant.

nonlinear equations. Considering the first two equations in Equation 17.54, the terms on the LHS can be implemented as currents flowing through grounded capacitors of value C connected at nodes V_1 and V_2, respectively. The expressions on the RHS can be implemented by constant current sources in conjunction with appropriately biased bipolar transistors to realize the exponential terms. The third equation in Equation 17.54 indicates that the output y can be obtained as the collector current of a bipolar transistor biased with a base–emitter voltage V_1.

Figure 17.37 shows a possible circuit implementation, which is derived using Gilbert's translinear circuit principle [63]. The detailed circuit implementation is described in more detail in Ref. [62]. The center frequency of this filter is given by $\omega_o = (I_o/CV_t)$ and can thus be tuned by varying the value of the bias current I_o.

17.7.3 Performance Aspects

17.7.3.1 Tuning Range

Both the quiescent bias current I_o and capacitance value C can be varied to alter the filter response. However, the allowable capacitor values are generally limited to within a certain range. On the lower side, C must not become smaller than the parasitic device capacitance. The base–emitter diffusion capacitance of the transistors is particularly important, since this is generally the largest device capacitance (up to the picofarad range) and is also nonlinear. In addition, as the value of C decreases, it becomes more difficult to match the capacitance values. The silicon area available limits the maximum value of C; for example, in a typical technology, a 50 pF poly–poly capacitor consumes 100×1000 μm of silicon area.

The range of allowable currents in a modern BJT is, in principle, fairly large (several decades); however, at very low and very high current levels, the current gain (β) is degraded. For HF operation, particular attention must be paid to the actual f_t of the transistors, which is given by [6]

$$f_t = \frac{g_m}{2\pi(C_\pi + C_\mu)} = \frac{g_m}{2\pi(C_d + C_{je} + C_{jc})} \tag{17.55}$$

where

C_{je} and C_{jc} represent the emitter and collector junction capacitance, respectively, and are only a weak function of the collector bias current

C_d represents the base–emitter diffusion capacitance and is proportional to the instantaneous collector current; $C_d = (\tau_f I_c/V_t)$, where τ_f is the effective base transit time

g_m represents the device transconductance, which is again dependent on the collector current; $g_m = (I_c/V_t)$

At high current levels, the diffusion capacitance is much larger than the junction capacitance, and thus:

$$f_t = \frac{g_m}{2\pi C_d} = \frac{1}{2\pi\tau_f} \tag{17.56}$$

At lower current levels, C_d and g_m decrease, whereas C_{je} and C_{jc} remain constant and f_t is reduced:

$$f_t = \frac{g_m}{2\pi(C_{je} + C_{jc})} \tag{17.57}$$

At very high current levels, f_t again reduces due to the effects of high-level injection.

17.7.3.2 Finite Current Gain

To verify the performance of the biquad filter, the circuit of Figure 17.37 was simulated using HSPICE with transistor parameters from a typical bipolar process. Transient (large signal) simulations were carried out to confirm the circuit operation, and the lowpass characteristic (input at I_{u2}) is found to be very close to the ideal response. However, with an input signal applied at I_{u1} to obtain a bandpass response, the filter performance clearly deviates from the ideal characteristic as shown in Figure 17.38. At low frequencies, the stop-band attenuation is only 25 dB. Resimulating the circuit with ideal transistor models gives the required ideal bandpass characteristic as shown in Figure 17.38; and by reintroducing the transistor parameters one at a time, the cause of the problem is found to be the finite current gain (β) of the bipolar transistors.

To increase the effective β, each transistor can be replaced by a Darlington pair, as shown in Figure 17.39. This combination acts as a bipolar transistor with $\beta = \beta_a\beta_b + \beta_a + \beta_b \approx \beta^2$. Simulating the bandpass response of the circuit with Darlington pairs results in improved stop-band attenuation as shown in Figure 17.40, where the dc attenuation $H(0)$ is now approximately -50 dB. A disadvantage, however, is that a higher supply voltage is now required, since the effective base–emitter voltage V_{be} of the Darlington pair is double that of a single device. In addition, the current in device Q_a of Figure 17.39 is now β times smaller than the design current I_o, resulting in a much lower f_t for this device.

An alternative method to improve the bandpass characteristic is described below. The (nonideal) transfer function from the bandpass input I_{u1} is described by

$$H_1(s) = \frac{I_{out}(s)}{I_{u1}(s)} = \frac{\omega_o(s + \omega_z)}{s^2 + (\omega_o/Q)s + \omega_o^2} \tag{17.58}$$

ω_z is a parasitic zero, which describes the low-frequency "flattening-out" of the bandpass characteristic. The transfer function from the second input I_{u2} is a lowpass response:

$$H_2(s) = \frac{I_{out}(s)}{I_{u2}(s)} = \frac{\omega_o^2}{s^2 + (\omega_o/Q)s + \omega_o^2} \tag{17.59}$$

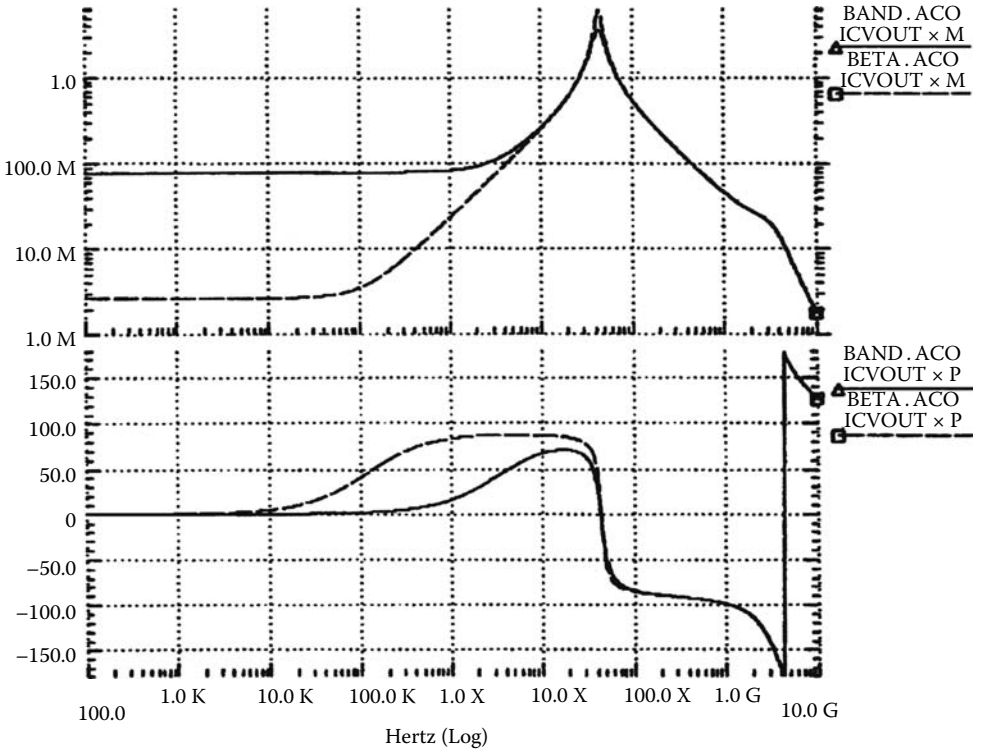


FIGURE 17.38 Bandpass response of the biquad of Figure 17.37. Notice the poor low-frequency stop-band attenuation (approx. 25 dB) if real transistor models are used (solid line), versus the much better stop-band attenuation with β set to 1000 (dashed line).

By applying a scaled version of the input signal I_{u1} to the lowpass input I_{u2}, the unwanted zero ω_z can thus be compensated. Setting $I_{u2} = -(\omega_z/\omega_o)\, I_{u1}$

$$I_{\text{out}}(s) = \frac{\omega_o(s+\omega_z)I_{u1}(s) + \omega_o^2\left(-\frac{\omega_z}{\omega_o}I_{u1}(s)\right)}{s^2 + (\omega_o/Q)s + \omega_o^2} = \frac{\omega_o s}{s^2 + (\omega_o/Q)s + \omega_o^2} I_{u1}(s) \tag{17.60}$$

This idea is confirmed by simulation as shown in Figure 17.40. However, in practice, since this technique is very sensitive to the exact factor by which the input signal is scaled, active on-chip tuning would be required.

17.7.3.3 Frequency Performance

The log-domain filter operates in current mode, and all nodes within the circuit have an impedance of the order of $1/g_m$. Thus, any parasitic poles or zeros within the circuit are of the order of g_m/C_p, where C_p represents the parasitic capacitance at the given node. Typically, the parasitic capacitors are dominated by base–emitter diffusion capacitance Cπ, and the parasitic poles are situated close to the f_t of the technology. This underscores the potential of the log-domain technique for HF

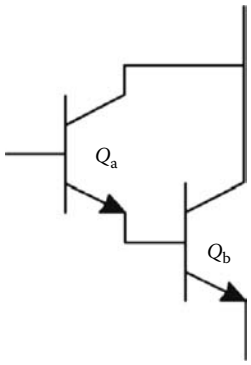


FIGURE 17.39 Darlington pair.

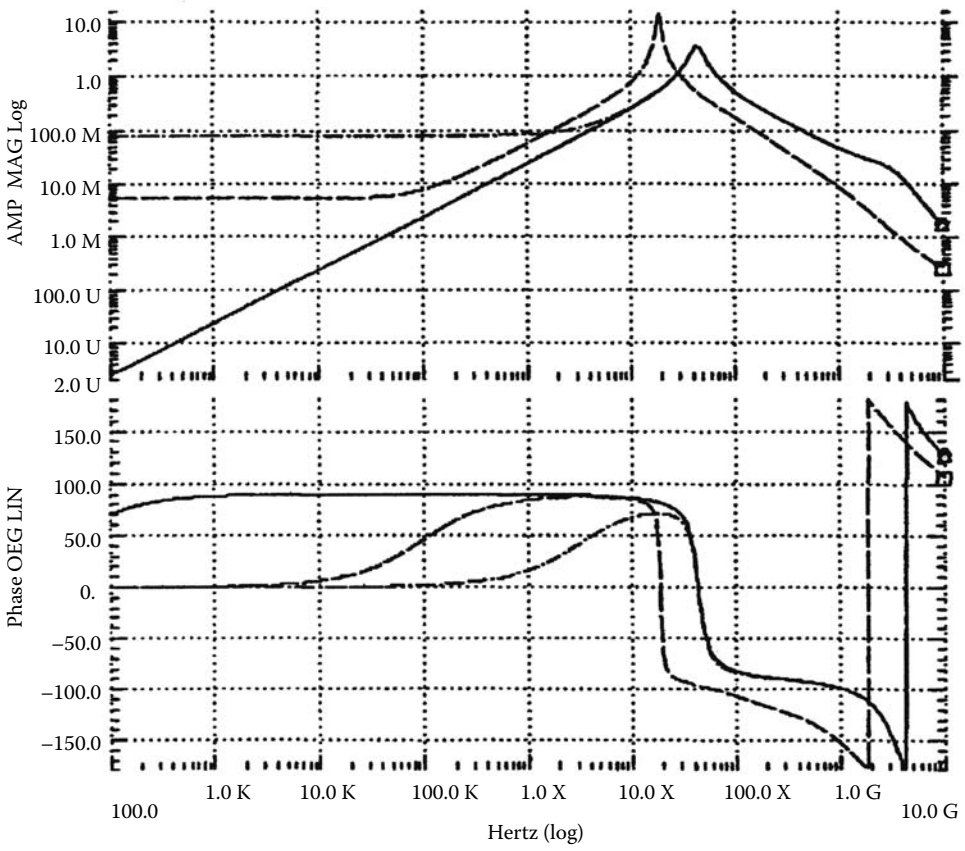


FIGURE 17.40 Possible solutions for the effect of finite β. The "dashed-dotted" line is the original response with a stop-band attenuation of only 25 dB. The dashed line (−50 dB) is the result of replacing all BJTs by Darlington pairs. The solid line is the result of feeding a fraction of the input signal to the second (i.e., lowpass) input.

operation. In practice, as with all HF circuits, careful design and layout are required to achieve the maximum frequency potential [75,76].

17.7.4 Basic Log-Domain Integrator

Section 17.7.3 has outlined some of the limitations of practical log-domain circuits that result from intrinsic device parasitics. This section analyzes these nonidealities in more detail by considering the simplest log-domain circuit, a first-order lowpass filter (lossy integrator).

A log-domain first-order lowpass filter is shown in Figure 17.41. Referring to this circuit, and assuming an ideal exponential characteristic for each BJT, the following set of equations can be written (assuming matched components and neglecting the effect of base currents):

$$\begin{aligned} I_{\text{in}} &= I_{\text{S}} \exp\left(\frac{V_{\text{in}}}{V_{\text{t}}}\right) \quad I_{\text{out}} = I_{\text{S}} \exp\left(\frac{V_{\text{out}}}{V_{\text{t}}}\right) \\ V_{\text{out}} &= V_{\text{c}} + V_{\text{t}} \ln\left(\frac{I_{\text{o}}}{I_{\text{S}}}\right) \quad I_{\text{o}} + C\frac{\text{d}V_{\text{c}}}{\text{d}t} = I_{\text{S}} \exp\left(\frac{V_{\text{in}} - V_{\text{c}}}{V_{\text{t}}}\right) \end{aligned} \tag{17.61}$$

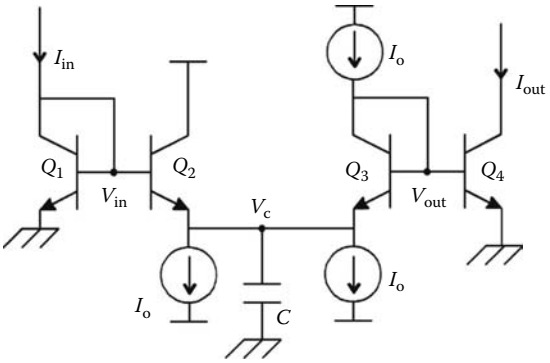


FIGURE 17.41 A first-order lowpass filter (lossy integrator).

Combining these equations results in the following linear first-order differential equation:

$$\left(\frac{CV_t}{I_o}\right)\frac{dI_{out}}{dt} + I_{out} = I_{in} \tag{17.62}$$

Taking the Laplace transform produces the following linear transfer function:

$$\frac{I_{out}(s)}{I_{in}(s)} = \frac{I_o}{I_o + sCV_t} \tag{17.63}$$

Equation 17.63 describes a first-order lowpass circuit with −3 dB frequency $f_c = I_o/2\pi CV_t$. This transfer function has been derived assuming an ideal exponential BJT characteristic; however, in practice, device nonidealities will introduce deviations from this ideal exponential characteristic, resulting in transfer function errors and distortion. This is similar to the case of ''conventional'' linear system design, where any deviation from the ideal linear building block response will contribute to output distortion. A brief discussion of the performance limitations of log-domain filters due to device nonidealities is given below; further discussion of distortion and performance limitations can be found in Refs. [65,66].

17.7.4.1 Effects of Finite Current Gain

Assuming all transistors have equal β, Equation 17.62 is modified to the following nonlinear differential equation (neglecting terms with β^2 or higher in the denominator):

$$\left(\frac{CV_t}{I_o}\right)\frac{dI_{out}}{dt}\left(1 + \frac{I_o}{I_{out}(1+\beta)}\right) + I_{out}\left(1 + \frac{I_{out}}{\beta I_o}\right) + \frac{2I_o}{1+\beta} = I_{in}\left(1 + \frac{I_{in}}{\beta}\right) \tag{17.64}$$

An analytical solution to Equation 17.64 is difficult to obtain; thus, a qualitative discussion is given. At low frequencies, neglecting the differential terms, finite β causes a dc gain error and quadratic (even-order) distortion. At high frequencies, the differential term is modified, depending on the values of I_{out} and β; therefore, scalar error and modulation of the 3 dB frequency are expected. In practice, the effects of finite β are further complicated due to its dependence on frequency and device collector current.

17.7.4.2 Component Mismatch

Emitter area mismatches cause variations in the saturation current I_S between transistors. Taking the emitter area into account, Equation 17.62 can be rewritten as

$$\left(\frac{CV_t}{I_o}\right)\frac{dI_{out}}{dt} + I_{out} = \lambda I_{in} \tag{17.65}$$

where $\lambda = (I_{S3}I_{S4}/I_{S1}I_{S2}) = (A_3A_4/A_1A_2)$. It is clear from Equation 17.65 that area mismatches introduce only a change in the proportionality constant or dc gain of the integrator, and do not have any effect on the linearity or time constant of the integrator. The gain error can be compensated by easily adjusting one of the dc bias currents; thus, I_s mismatches do not seem to be a significant problem.

17.7.4.3 Ohmic Resistance

Ohmic resistances include the base and emitter diffusion resistance, and the resistance of interconnects and contact interfaces. For simplicity, all these resistances can be referred to the base as an equivalent ohmic base resistance r_b. The device collector current can thus be defined as

$$I_c = I_S \exp\left(\frac{V_{be} - I_b r_b}{V_t}\right) = I_S \exp\left(\frac{V_{be}}{V_t}\right) \exp\left(-\alpha I_c\right) \tag{17.66}$$

where

V_{be} is the applied (extrinsic) base–emitter voltage
$\alpha = r_b/\beta V_t$

When the first-order filter of Figure 17.41 is analyzed using the expression given in Equation 17.66, a modified differential equation is obtained (assuming all base resistances are equal):

$$\left(\frac{CV_t}{I_o}\right)\frac{d[I_{out}e^{\alpha I_{out}}]}{dt} + I_{out}e^{\alpha I_{out}} = (e^{\partial\alpha I_o})I_{in}e^{\alpha I_{in}}e^{\left(-\alpha\frac{CV_t}{I_o}\frac{dI_{out}}{dt}(1+\alpha I_{out})\right)} \tag{17.67}$$

The term $\partial\alpha$ represents the difference between two α values and will be close to zero if all base resistances are assumed equal, and thus can be neglected. At frequencies well below ω_o, the time derivative in the exponent can also be neglected to give

$$\left(\frac{CV_t}{I_o}\right)\frac{d[I_{out}\exp(\alpha I_{out})]}{dt} + I_{out}\exp(\alpha I_{out}) = I_{in}\exp(\alpha I_{in}) \tag{17.68}$$

Expanding the differential term in Equation 17.68:

$$I_{out}\exp(\alpha I_{out}) + \tau\frac{dI_{out}}{dt}\exp(\alpha I_{out}) + \tau\alpha I_{out}\frac{dI_{out}}{dt}\exp(\alpha I_{out}) = I_{in}\exp(\alpha I_{in}) \tag{17.69}$$

Equation 17.69 is clearly no longer linear; and thus, distortion products will be present at the output. To quantify this distortion, we apply as an input signal $I_{in} = A\exp(j\omega t)$, and expect as output:

$$I_{out} = B\exp j(\omega w + \theta) + c\exp j(2\omega t + \phi) + D\exp j(3\omega t + \psi) \tag{17.70}$$

Expanding the exponential terms in Equation 17.69 by the first few terms of their series expansion:

$$\exp(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} \tag{17.71}$$

and identifying the terms in $\exp(j\omega t)$, $\exp(2j\omega t)$, etc. results in

$$B\exp j\theta = \frac{A}{1+s\tau} \quad C\exp j\phi = \alpha A^2 p(\tau) \quad D\exp j\psi = \alpha^2 A^3 q(s\tau) \tag{17.72}$$

The expression for B is (as expected) the first-order transfer function of the system. The expressions for C and D give the second and third harmonic terms:

$$\mathrm{HD}_2 = \frac{|C \exp j\phi|}{|\mathrm{A}|} \quad \mathrm{HD}_3 = \frac{|D \exp j\psi|}{|\mathrm{A}|} \tag{17.73}$$

$p(s\tau)$ and $q(s\tau)$ are rational functions in $s\tau$; thus, the output distortion is frequency dependent. Distortion is low at low frequencies, and peaks around the cut-off frequency ($s\tau = 1$), where $p = 0.25$ and $q = 2$. The maximum distortion levels can be approximated as

$$\begin{aligned} \mathrm{HD}_{2(\max)} &= 0.25\alpha A = 0.25 \frac{r_\mathrm{b}}{\beta V_\mathrm{t}} I_\mathrm{in} \\ \mathrm{HD}_{3(\max)} &= 2(\alpha A)^2 = 2\left(\frac{r_\mathrm{b}}{\beta V_\mathrm{t}}\right)^2 I_\mathrm{in}^2 \end{aligned} \tag{17.74}$$

These expressions demonstrate that when the voltage drop across the base resistance becomes comparable to V_t, there is significant distortion. This analysis also predicts that the distortion is larger at frequencies closer to the cut-off frequency, which is confirmed by circuit simulation.

In practice, the base resistance can be minimized by good layout (multiple base contacts), or by connecting several transistors in parallel. The latter, however, decreases the current through each transistor, which may lead to a decrease in f_t. It should be noted that, from a technological point of view, a high f_t and a low r_b are opposing goals. To achieve a high f_t, the base should be as shallow as possible, reducing base transit time. At the same time, however, a shallow base increases the silicon (intrinsic) base resistance.

17.7.4.4 Early Effect

The early effect (base-width modulation) causes the collector current to vary with the collector–emitter and base–collector voltages. Considering the variation of collector–emitter voltage, the collector current can be written as $I_\mathrm{c} = (1 + V_\mathrm{ce}/V_\mathrm{A}) \exp(V_\mathrm{be}/V_\mathrm{t})$, where V_A is the forward-biased early voltage. An analysis of the circuit of Figure 17.41 shows that the early effect introduces a scalar error to the dc gain as in the case of emitter area (I_S) mismatch. In practice, the base-width modulation of the devices also introduces distortion because V_ce and V_bc are signal dependent. However, since voltage swings in current-mode circuits are minimized, this is not believed to be a major source of distortion.

17.7.4.5 Frequency Limitations

Each bipolar transistor has various intrinsic capacitors, the most important being C_μ (base–collector junction capacitance), C_cs (collector–substrate junction capacitance), and C_π (the sum of the base–emitter junction capacitance and the base–emitter diffusion capacitance). The junction capacitors depend only slightly on the operating point, while the base–emitter diffusion capacitance is proportional to the bias current, as given by $C_\mathrm{d} = \tau_\mathrm{f} I_\mathrm{c}/V_\mathrm{t}$. To determine the position of the parasitic poles and zeros, Figure 17.41 should be analyzed using large-signal device models. Unfortunately, the complexity of the resulting expressions renders this approach impractical even for the simplest log-domain circuits. To gain an intuitive understanding of the HF limitations of the circuit, a small-signal analysis can be performed. Although log-domain circuits are capable of large-signal operation, this small-signal approach is justified to some extent since small signals can be considered ''special case'' of large signals. If the circuit fails to operate correctly for small signals, then it will almost certainly fail in large-signal operation (unfortunately, the opposite does not hold; if a circuit operates correctly for small signals, it does not necessarily work well for large signals). Analyzing the circuit of Figure 17.41, replacing each transistor by a

small-signal (hybrid-π) equivalent model (comprising g_m, r_b, r_π, C_π, C_μ) [64], results in the following expression:

$$\frac{I_{out}}{I_{in}} = \left(\frac{g_{m4}}{g_{m1}}\right)\frac{1+\tau_z s}{(1+\tau_{p1}s)(1+\tau_{p2}s)} \tag{17.75}$$

τ_{p1} is the time constant of the first (dominant) pole, given by

$$\tau_{p1} = \frac{C}{g_{m2}} + (C_{\mu4}+C_{\pi4})\left(r_{b4}+\frac{1}{g_{m2}}+\frac{1}{g_{m3}}\right)+\frac{C_{\pi1}}{g_{m1}}+\frac{C_{\pi2}}{g_{m2}} + \frac{C_{\pi3}}{g_{m3}} + C_{\mu1}r_{b1} + C_{\mu2}r_{b2} + C_{\mu3}r_{b3} \tag{17.76}$$

Ideally, this pole location should depend only on the design capacitance C, and not on the device parasitics. Therefore, $C_\mu\ r_b$, $C_\pi/g_m \ll C/g_{m2}$. Since $1/C_\mu r_b$ is typically much greater than f_t, this first constraint does not form a limit. The value of C_π/g_m depends on the operating point. For large currents, C_π is dominated by diffusion capacitance C_d so that $C_\pi/g_m = 1/f_t$. For smaller currents, g_m decreases while C_π becomes dominated by junction capacitance C_{je}, so that $C_\pi/g_m \gg 1/f_t$. Thus, it would seem that the usable cut-off frequency of the basic log-domain first-order filter is limited by the actual f_t of the transistors. The second pole time constant τ_{p2} (assuming that $\tau_{p1} = C/g_{m2}$) is

$$\tau_{p2} = (C_{\mu4}+C_{\pi4})\left(r_{b4}+\frac{1}{g_{m3}}\right)+(C_{\mu2}+C_{\pi2})\left(r_{b2}+\frac{1}{g_{m1}}\right)+\frac{C_{\pi1}+C_{cs1}}{g_{m1}} + \frac{C_{\pi13}+C_{cs3}}{g_{m3}} + C_{\mu1}r_{b1} + C_{\mu4}r_{b4} \tag{17.77}$$

This corresponds approximately to the f_t of the transistors, although Equation 17.77 shows that the collector–substrate capacitance also contributes toward limiting the maximum operating frequency. The zero time constant τ_z is given by

$$\tau_z = (C_{\pi1}+C_{\mu1})r_{b1} + \frac{C_{\pi2}}{g_{m2}} + \frac{C_{\pi4}}{g_{m4}} + C_{\mu4}r_{b4} \tag{17.78}$$

This is of the same order of magnitude as the second pole. This means that the first zero and the second pole will be close together, and will compensate to a certain degree. However, in reality, there are more poles and zeros than Equation 17.75 would suggest, and it is likely that others will also occur around the actual f_t of the transistors.

17.7.4.6 Noise

Noise in companding and log-domain circuits is discussed in some detail in Refs. [68–70], and a complete treatment is beyond the scope of this discussion. For linear (noncompanding) circuits, noise is generally assumed to be independent of signal level, and the SNR will increase with increasing input signal level. This is not true for log-domain systems. At small input signal levels, the noise value can be assumed approximately constant, and an increase in signal level will give an increase in SNR. At high signal levels, the instantaneous value of noise will increase, and thus the SNR levels out at a constant value. This can be considered as an intermodulation of signal and noise power. For the Class A circuits discussed above, the peak signal level is limited by the dc bias current. In this case, the large-signal noise is found to be of the same order of magnitude as the quiescent noise level, and thus a linear approximation is generally acceptable (this is not the case for Class AB circuits).

17.7.5 Synthesis of Higher-Order Log-Domain Filters

The state-space synthesis technique outlined above proves difficult if implementation of high-order filters is required, since it becomes difficult to define and manipulate a large set of state equations. One solution is to use the signal flow graph (SFG) synthesis method proposed by Perry and Roberts [71] to simulate LC ladder filters using log-domain building blocks. The interested reader is also referred to Refs. [72–74], which present modular and transistor-level synthesis techniques that can be easily extended to higher-order filters.

References

1. A. Sedra and K. Smith, *Microelectronic Circuits*, Oxford University Press, Oxford, 1998.
2. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, Wiley, New York, 1993.
3. W. H. Gross, New high speed amplifier designs, design techniques and layout problems, in *Analog Circuit Design*, Eds. J. S. Huijsing, R. J. van der Plassche, and W. Sansen, Kluwer Academic, Norwell, MA, 1993.
4. D. F. Bowers, The impact of new architectures on the ubiquitous operational amplifier, in *Analog Circuit Design*, Eds. J. S. Huijsing, R. J. van der Plassche, and W. Sansen, Kluwer Academic, Norwell, MA, 1993.
5. J. Fonderie and J. H. Huijsing, Design of low-voltage bipolar opamps, in *Analog Circuit Design*, Eds. J. S. Huijsing, R. J. van der Plassche, and W. Sansen, Kluwer Academic, Norwell, MA, 1993.
6. M. Steyaert and W. Sansen, Opamp design towards maximum gain-bandwidth, in *Analog Circuit Design*, Eds. J. S. Huijsing, R. J. van der Plassche, and W. Sansen, Kluwer Academic, Norwell, MA, 1993.
7. K. Bult and G. Geelen, The CMOS gain-boosting technique, in *Analog Circuit Design*, Eds. J. S. Huijsing, R. J. van der Plassche, and W. Sansen, Kluwer Academic, Norwell, MA, 1993.
8. J. Bales, A low power, high-speed, current feedback opamp with a novel class AB high current output stage, *IEEE Journal Solid-State Circuits*, 32(9), 1470, Sep. 1997.
9. C. Toumazou, Analogue signal processing: The 'current way' of thinking, *International Journal of High-Speed Electronics*, 32(3–4), 297, 1992.
10. K. Manetakis and C. Toumazou, A new CMOS CFOA suitable for VLSI technology, *Electronics Letters*, 32(12), June 1996.
11. K. Manetakis, C. Toumazou, and C. Papavassiliou, A 120 MHz, 12 mW CMOS current feedback opamp, *Proceedings of IEEE Custom Integrated Circuits Conference*, Santa Clara, CA, p. 365, 1998.
12. D. A. Johns and K. Martin, *Analog Integrated Circuit Design*, Wiley, Hoboken, NJ, 1997.
13. C. Toumazou, J. Lidgey, and A. Payne, Emerging techniques for high-frequency BJT amplifier design: A current-mode perspective, *Parchment Press for International Conference on Electronics, Circuits and Systems*, Cairo, 1994.
14. M. C. H. Cheng and C. Toumazou, 3V MOS current conveyor cell for VLSI technology, *Electronics Letters*, 29, 317, 1993.
15. K. Manetakis, Intermediate frequency CMOS analogue cells for wireless communications, PhD thesis, Imperial College, London, 1998.
16. R. A. Johnson et al., A 2.4 GHz silicon-on-sapphire CMOS low-noise amplifier, *IEEE Microwave and Guided Wave Letters*, 7(10), 350–352, Oct. 1997.
17. A. N. Karanicolas, A 2.7 V 900 MHz CMOS LNA and mixer, *IEEE Digest of International Solid-State Circuits Conference*, San Francisco, pp. 50–51, 1996.
18. D. K. Shaffer and T. H. Lee, A 1.5-V, 1.5-GHz CMOS low noise amplifier, *IEEE Journal of the Solid-State Circuits*, 32 (5), 745–759, May 1997.
19. J. C. Rudell et al., A 1.9 GHz wide-band if double conversion CMOS integrated receiver for cordless telephone applications, *Digest of IEEE International Solid-State Circuits Conference*., pp. 304–305, 1997.

20. E. Abou-Allam et al., CMOS front end RF amplifier with on-chip tuning, *Proceedings of IEEE ISCAS96*, Atlanta, pp. 148–151, 1996.
21. P. R. Gray and R. G. Meyer, *Analysis and Design of Analogue Integrated Circuits and Systems*, Chap. 11, 3rd edn., John Wiley & Sons, New York, 1993.
22. M. J. N. Sibley, *Optical Communications*, Chaps. 4–6, Macmillan, UK, 1995.
23. J. J. Morikuni et al., Improvements to the standard theory for photoreceiver noise, *Journal of Lightwave Technology*, 12(4), 1174–1184, July 1994.
24. A. A. Abidi, Gigahertz transresistance amplifiers in fine line NMOS, *IEEE Journal of the Solid-State Circuits*, SC-19(6), 986–994, Dec. 1984.
25. M. B. Das, J. Chen, and E. John, Designing optoelectronic integrated circuit (OEIC) receivers for high sensitivity and maximally flat frequency response, *Journal of Lightwave Technology*, 13(9), 1876–1884, Sep. 1995.
26. B. Sklar, *Digital Communication: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
27. S. D. Personick, Receiver design for optical fiber systems, *Proceedings of the IEEE*, 65(12), 1670–1678, Dec. 1977.
28. J. M. Senior, *Optical Fiber Communications: Principles and Practice*, Chaps. 8–10, Prentice Hall International, UK, 1985.
29. N. Scheinberg et al., Monolithic GaAs transimpedance amplifiers for fiber-optic receivers, *IEEE Journal of the Solid-State Circuits*, 26(12), 1834–1839, Dec. 1991.
30. C. Toumazou and S. M. Park, Wide-band low noise CMOS transimpedance amplifier for gigahertz operation, *Electronics Letters*, 32(13), 1194–1196, June 1996.
31. S. M. Park and C. Toumazou, Giga-hertz low noise CMOS transimpedance amplifier, *Proceedings of the IEEE ISCAS*, 1, 209–212, June 1997.
32. D. M. Pietruszynski et al., A 50-Mbit/s CMOS monolithic optical receiver, *IEEE Journal of the Solid-State Circuits*, 23, 6, 1426–1432, Dec. 1988.
33. Y. Akazawa et al., A design and packaging technique for a high-gain, gigahertz-band single-chip amplifier, *IEEE Journal of the Solid-State Circuits*, SC-21(3), 417–423, June 1986.
34. N. Ishihara et al., A design technique for a high-gain, 10-GHz class-bandwidth GaAs MESFET amplifier IC module, *IEEE Journal of the Solid-State Circuits*, 27(4), 554–561, Apr. 1992.
35. M. Lee and M. A. Brooke, Design, fabrication, and test of a 125Mb/s transimpedance amplifier using MOSIS 1.2 μm standard digital CMOS process, *Proceedings of the 37th Midwest Symposium, Circuits and Systems*, 1, 155–157, Aug. 1994.
36. R. P. Jindal, Gigahertz-band high-gain low-noise AGC amplifiers in fine-line NMOS, *IEEE Journal of the Solid-State Circuits*, SC-22(4), 512–520, Aug. 1987.
37. N. Takachio et al., A 10Gb/s optical heterodyne detection experiment using a 23 GHz bandwidth balanced receiver, *IEEE Transactions on Microwave Theory and Techniques*, 38(12), 1900–1904, Dec. 1990.
38. K. Katsura et al., A novel flip-chip interconnection technique using solder bumps for high-speed photoreceivers, *Journal of Lightwave Technology*, 8(9), 1323–1326, Sep. 1990.
39. K. Murota and K. Hirade, GMSK modulation for digital mobile radio telephony, *IEEE Transactions on Communications*, 29, 1044–1050, 1981.
40. H. Krauss, C. W. Bostian, and F. H. Raab, *Solid State Radio Engineering*, Wiley, New York, 1980.
41. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 4th edn., Oxford University Press, UK, 1998.
42. N. O. Sokal and A. D. Sokal, Class E, a new class of high efficiency tuned single-ended switching power amplifiers, *IEEE Journal of Solid-State Circuits*, SC-10, 168–176, June 1975.
43. F. H. Raab, Effects of circuit variations on the class E tuned power amplifier, *IEEE Journal of Solid-State Circuits*, SC-13, 239–247, 1978.
44. T. Sowlati, C. A. T. Salama, J. Sitch, G. Robjohn, and D. Smith, Low voltage, high efficiency class E GaAs power amplifiers for mobile communications, 1994 *GaAs IC Symposium Digest*, pp. 171–174, Philadelphia, PA, October 1994.

45. T. Sowlati, C. A. T. Salama, J. Sitch, G. Robjohn, and D. Smith, Low voltage, high efficiency GaAs class E power amplifiers for wireless transmitters, *IEEE Journal of Solid-State Circuits*, SC-13(10), 1074–1080, 1995.
46. A. Rofougaran et al., A single-chip 900 MHz spread-spectrum wireless transceiver in 1-μm CMOS. Part I: Architecture and transmitter design, *IEEE Journal of Solid-State Circuits*, SC-33(4), 515–534, 1998.
47. J. Chang, A. A. Abidi, and M. Gaitan, Large suspended inductors on silicon and their use in a 2μm CMOS RF amplifier, *IEEE Electron Device Letters*, 14(5), 246–248, May 1993.
48. T. Sowlati et al., Linearized high efficiency class E power amplifier for wireless communications, *Proceedings of the IEEE 1996 Custom Integrated Circuits Conference*, San Diego, California, May 5–8, pp. 201–204, 1996.
49. G. N. Henderson, M. F. OKeefe, T.E. Boless, P. Noonan, et al., SiGe bipolar junction transistors for microwave power applications, *IEEE MTT-S International Microwave Symposium Digest*, pp. 1299–1302, 1997.
50. O. Shoaei and W. M. Snelgrove, A wide-range tunable 25 MHz–110 MHz BiCMOS continuous-time filter, *Proceedings of the IEEE ISCAS*, Atlanta, 1996.
51. P.-H. Lu, C.-Y. Wu, and M.-K. Tsai, Design techniques for VHF/UHF high-Q tunable bandpass filters using simple CMOS inverter-based transresistance amplifiers, *IEEE Journal of Solid-State Circuits*, 31(5), May 1996.
52. Y. Tsividis, Integrated continuous-time filter design—An overview, *IEEE Journal of Solid-State Circuits*, 29(3), 166–176, Mar. 1994.
53. F. Rezzi, A. Baschirotto, and R. Castello, A 3V 12–55MHz BiCMOS pseudo-differential continuous-time filter, *IEEE Transactions on Circuits and Systems-I*, 42(11), Nov. 1995.
54. B. Nauta, *Analog CMOS Filters for Very High Frequencies*, Kluwer Academic Publishers, Norwell, MA, 1992.
55. C. Toumazou, F. Lidgey, and D. Haigh, *Analogue IC Design: The Current-Mode Approach*, Peter Peregrinus Ltd., for IEEE Press, Herts, UK, 1990.
56. S. Szczepanski and R. Schauman, Nonlinearity-induced distortion of the transfer function shape in high-order filters, *Kluwer Journal of Analog Integrated Circuits and Signal Processing*, 3, 143–151, 1993.
57. S. Szczepanski, VHF fully-differential linearized CMOS transconductance element and its applications, *Proceedings of the IEEE International Symposium on Circuits Systems* (*ISCAS*), London 1994.
58. A. A. Abidi, Noise in active resonators and the available dynamic range, *IEEE Transactions on Circuits and Systems-I*, 39(4), 296–299, Apr. 1992.
59. S. Pipilos and Y. Tsividis, RLC active filters with electronically tunable center frequency and quality factor, *Electronics Letters*, 30(6), 472–474, Mar. 1994.
60. K. Manetakis and C. Toumazou, A 50 MHz high-Q bandpass CMOS filter, *Proceedings of the IEEE International Symposium on Circuits Systems* (*ISCAS*), Hong Kong, 1997.
61. Y. Tsividis, Externally linear time invariant systems and their application to companding signal processors, *IEEE Transactions on CAS-II*, 44(2), 65–85, Feb. 1997.
62. D. Frey, Log-domain filtering: An approach to current-mode filtering, *IEE Proceedings-G*, 140, 406–416, 1993.
63. B. Gilbert, Translinear circuits: A proposed classification, 1975, *Electronics Letters*, 11(1), 1416, 1975.
64. P. Grey and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd edn., John Wiley & Sons Inc., New York, 1993.
65. E. M. Drakakis, A. Payne, and C. Toumazou, Log-domain state-space: A systematic transistor-level approach for log-domain filtering, accepted for publication in *IEEE Transactions on CAS-II*, 1998.
66. V. Leung, M. El-Gamal, and G. Roberts, Effects of transistor non-idealities on log-domain filters, *Proceedings of the IEEE International Symposium on Circuits Systems*, Hong Kong, pp. 109–112, 1997.

67. D. Perry and G. Roberts, Log domain filters based on LC-ladder synthesis, *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems* (*ISCAS*), pp. 311–314, Seattle, 1995.
68. J. Mulder, M. Kouwenhoven, and A. van Roermund, Signal × noise intermodulation in translinear filters, *Electronics Letters*, 33(14), 1205–1207.
69. M. Punzenberger and C. Enz, Noise in instantaneous companding filters, *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems*, Hong Kong, pp. 337–340, June 1997.
70. M. Punzenberger and C. Enz, A 1.2V low-power BiCMOS class-AB log-domain filter, *IEEE Journal of Solid-State Circuits*, SC-32(12), 1968–1978, Dec. 1997.
71. D. Perry and G. Roberts, Log-domain filters based on LC ladder synthesis, *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems*, Seattle, pp. 311–314, 1995.
72. E. Drakakis, A. Payne, and C. Toumazou, Bernoulli operator: A low-level approach to log-domain processing, *Electronics Letters*, 33(12), 1008–1009, 1997.
73. F. Yang, C. Enz, and G. Ruymbeke, Design of low-power and low-voltage log-domain filters, *Proceedings of the 1996 IEEE International Symposium on Circuits and Systems*, Atlanta, pp. 125–128, 1996.
74. J. Mahattanakul and C. Toumazou, Modular log-domain filters, *Electronics Letters*, 33(12), 1130–1131, 1997.
75. D. Frey, A 3.3 V electronically tuneable active filter useable to beyond 1 GHz, *Proceedings of the 1994 IEEE International Symposium on Circuits and Systems*, London, pp. 493–496, 1994.
76. M. El-Gamal, V. Leung, and G. Roberts, Balanced log-domain filters for VHF applications, *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems*, Monterey, pp. 493–496, 1997.

III

Linear Circuit Analysis

Wai-Kai Chen
University of Illinois at Chicago

18 **Fundamental Circuit Concepts** *John Choma, Jr.* **18**-1
Electrical Circuit • Circuit Classifications

19 **Network Laws and Theorems** *Ray R. Chen, Artice M. Davis, and Marwan A. Simaan* **19**-1
Kirchhoff's Voltage and Current Laws • References • Network Theorems

20 **Terminal and Port Representations** *James A. Svoboda* **20**-1
Introduction • Terminal Representations • Port Representations • Port Representations and Feedback Systems • Conclusions • References

21 **Signal Flow Graphs in Filter Analysis and Synthesis** *Pen-Min Lin* **21**-1
Formulation of Signal Flow Graphs for Linear Networks • Synthesis of Active Filters Based on Signal Flow Graph Associated with a Passive Filter Circuit • Synthesis of Active Filters Based on Signal Flow Graph Associated with a Filter Transfer Function • References

22 **Analysis in the Frequency Domain** *Jiri Vlach and John Choma, Jr.* **22**-1
Network Functions • Advanced Network Analysis Concepts • References

23 **Tableau and Modified Nodal Formulations** *Jiri Vlach* **23**-1
Introduction • Nodal and Mesh Formulations • Graphs and Tableau Formulation • Modified Nodal Formulation • Nonlinear Elements • Nodal Analysis of Active Networks • Conclusion • References

24 **Frequency-Domain Methods** *Peter B. Aronhime* **24**-1
Network Functions • Network Theorems • Sinusoidal Steady-State Analysis and Phasor Transforms • Acknowledgments • References

25 **Symbolic Analysis** *Benedykt S. Rodanski and Marwan M. Hassoun* **25**-1
Introduction and Definition • Frequency Domain Analysis • Traditional Methods (Single Expressions) • Hierarchical Methods (Sequence of Expressions) • Approximate Symbolic Analysis • Methods Based on Determinant Decision Diagrams • Time Domain Analysis • References

26 Analysis in the Time Domain *Robert W. Newcomb* **26**-1
Signal Types • References • First-Order Circuits • References • Second-Order Circuits • References

27 State-Variable Techniques *Kwong S. Chao* **27**-1
Concept of States • State-Variable Formulation via Network Topology • Natural Response and State Transition Matrix • Complete Response • References

18

Fundamental Circuit Concepts

18.1 Electrical Circuit **18**-1
Current and Current Polarity • Energy and Voltage • Power
18.2 Circuit Classifications **18**-10
Linear vs. Nonlinear • Active vs. Passive • Time-Varying vs. Time-Invariant • Lumped vs. Distributed

John Choma, Jr.
University of Southern California

18.1 Electrical Circuit

An electrical circuit or electrical network is an array of interconnected elements wired so as to be capable of conducting current. As discussed earlier, the fundamental two-terminal elements of an electrical circuit are the resistor, capacitor, inductor, voltage source, and current source. The circuit schematic symbols of these elements, together with the algebraic symbols used to denote their respective general values, shown in Figure 18.1.

As suggested in Figure 18.1, the value of a resistor is known as its resistance, R, and its dimensional units are ohms. The case of a wire used to interconnect the terminals of two electrical elements corresponds to the special case of a resistor whose resistance is ideally zero ohms; that is, $R = 0$. For the capacitor in Figure 18.1b, the capacitance, C, has units of farads, and from Figure 18.1c, the value of an inductor is its inductance, L, the dimensions of which are henries. In the case of the voltage sources depicted in Figure 18.1d, a constant, time-invariant source of voltage, or battery, is distinguished from a voltage source that varies with time. The latter type of voltage source is often referred to as a time-varying signal or simply, a signal. In either case, the value of the battery voltage, E, and the time-varying signal, $v(t)$, is in units of volts. Finally, the current source of Figure 18.1e has a value, I, in units of amperes, which is typically abbreviated as amps.

Elements having three, four, or more than four terminals can also appear in practical electrical networks. The discrete component bipolar junction transistor (BJT), which is schematically portrayed in Figure 18.2a, is an example of a three-terminal element, in which the three terminals are the collector, base, and emitter. On the other hand, the monolithic metal-oxide-semiconductor field-effect transistor (MOSFET) depicted in Figure 18.2b has four terminals: drain, gate, source, and bulk substrate.

Multiterminal elements appearing in circuits identified for systematic mathematical analyses are routinely represented, or modeled, by equivalent subcircuits formed of only interconnected two-terminal elements. Such a representation is always possible, provided that the list of two-terminal elements itemized in Figure 18.1 is appended by an additional type of two-terminal element known as the controlled source, or dependent generator. Two of the four types of controlled sources are voltage sources and two are current sources. In Figure 18.3a, the dependent generator is a voltage-controlled

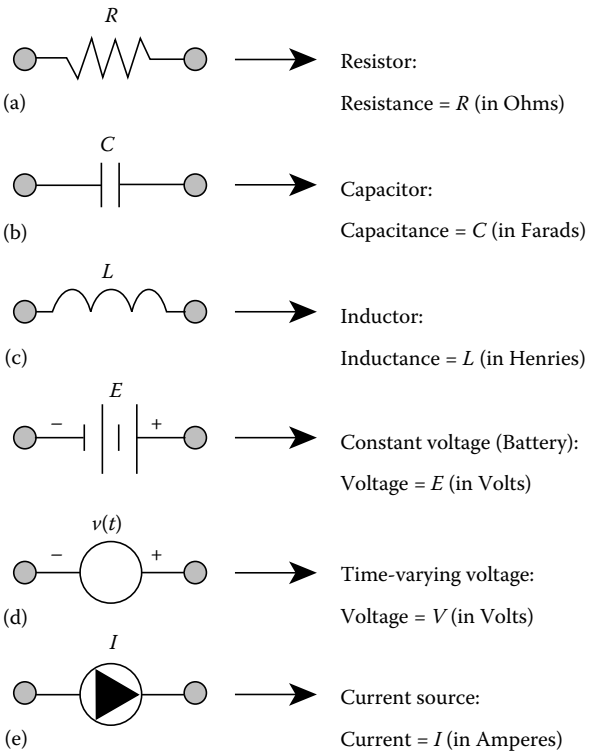


FIGURE 18.1 Circuit schematic symbol and corresponding value notation for (a) resistor, (b) capacitor, (c) inductor, (d) voltage source, and (e) current source. Note that a constant voltage source, or battery, is distinguished from a voltage source that varies with time.

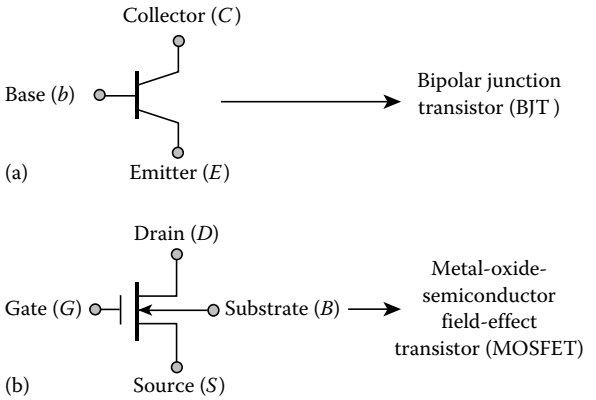


FIGURE 18.2 Circuit schematic symbol for (a) discrete component BJT and (b) MOSFET.

voltage source (VCVS) in that the voltage, $v_o(t)$, developed from terminal 3 to terminal 4 is a function of, and is therefore dependent on, the voltage, $v_i(t)$, established elsewhere in the considered network from terminal 1 to terminal 2. The controlled voltage, $v_o(t)$, as well as the controlling voltage, $v_i(t)$, can be constant or time-varying. Regardless of the time-domain nature of these two voltage, the value of $v_o(t)$ is

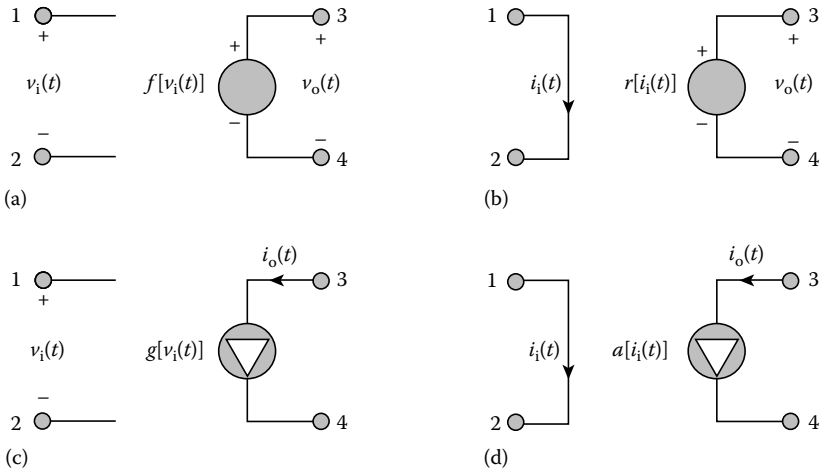


FIGURE 18.3 Circuit schematic symbol for (a) VCVS, (b) CCVS, (c) VCCS, and (d) CCCS.

not an independent number. Instead, its value is determined by $v_i(t)$ in accordance with a prescribed functional relationship, for example,

$$v_o(t) = f[v_i(t)] \tag{18.1}$$

If the function, $f(\cdot)$, is linearly related to its argument, Equation 18.1 collapses to the form

$$v_o(t) = f_\mu v_i(t) \tag{18.2}$$

where f_μ is a constant, independent of either $v_o(t)$ or $v_i(t)$. When the function on the right-hand side of Equation 18.1 is linear, the subject VCVS becomes known as a linear voltage-controlled voltage source.

The second type of controlled voltage source is the current-controlled voltage source (CCVS) depicted in Figure 18.3b. In this dependent generator, the controlled voltage, $v_o(t)$, developed from terminal 3 to terminal 4 is a function of the controlling current, $i_i(t)$, flowing elsewhere in the network between terminals 1 and 2, as indicated. In this case, the generalized functional dependence of $v_o(t)$ on $i_i(t)$ is expressible as

$$v_o(t) = r[i_i(t)] \tag{18.3}$$

which reduces to

$$v_o(t) = r_m i_i(t) \tag{18.4}$$

when $r(\cdot)$ is a linear function of its argument.

The two types of dependent current sources are diagrammed symbolically in Figure 18.3c and d. Figure 18.3c depicts a voltage-controlled current source (VCCS), for which the controlled current $i_o(t)$, flowing in the electrical path from terminal 3 to terminal 4, is determined by the controlling voltage, $v_i(t)$, established across terminals 1 and 2. Therefore, the controlled current can be written as

$$i_o(t) = g[v_i(t)] \tag{18.5}$$

In the current-controlled current source (CCCS) of Figure 18.3d

$$i_o(t) = a[i_i(t)] \tag{18.6}$$

where the controlled current, $i_o(t)$, flowing from terminal 3 to terminal 4 is a function of the controlling current, $i_i(t)$, flowing elsewhere in the circuit from terminal 1 to terminal 2. As is the case with the two controlled voltage sources studied earlier, the preceding two equations collapse to the linear relationships

$$i_o(t) = g_m v_i(t) \tag{18.7}$$

and

$$i_o(t) = a_\alpha i_i(t) \tag{18.8}$$

when $g(\cdot)$ and $a(\cdot)$, respectively, are linear functions of their arguments.

The immediate implication of the controlled source concept is that the definition for an electrical circuit given at the beginning of this section can be revised to read "an electrical circuit or electrical network is an array of interconnected two-terminal elements wired in such a way as to be capable of conducting current." Implicit in this revised definition is the understanding that the two-terminal elements allowed in an electrical circuit are the resistor, capacitor, inductor, voltage source, current source, and any of the four possible types of dependent generators.

In, an attempt to reinforce the engineering utility of the foregoing definition, consider the voltage-mode operational amplifier, or op-amp, whose circuit schematic symbol is submitted in Figure 18.4a. Observe that the op-amp is a five-terminal element. Two terminals, labeled 1 and 2, are provided to receive input signals that derive either from external signal sources or from the output terminals of subcircuits that feedback a designable fraction of the output signal established between terminal 3 and the system ground. Battery voltages, identified as E_{CC} and E_{BB} in the figure, are applied to the remaining two op-amp terminals (terminals 4 and 5) with respect to ground to bias or activate the op-amp for its intended application. When E_{CC} and E_{BB} are selected to ensure that the subject op-amp behaves as a linear circuit element, the voltages, E_{CC} and E_{BB}, along with the corresponding terminals at which they are incident, are inconsequential. In this event the op-amp of Figure 18.4a can be modeled by the electrical circuit appearing in Figure 18.4b, which exploits a linear VCVS. Thus, the voltage amplifier of Figure 18.4c, which interconnects two batteries, a signal source voltage, three resistors, a capacitor, and an op-amp, can be represented by the network given in Figure 18.4d. Note that the latter configuration uses only two terminal elements, one of which is a VCVS.

18.1.1 Current and Current Polarity

The concept of an electrical current is implicit to the definition of an electrical circuit in that a circuit is said to be an array of two-terminal elements that are connected in such a way as to permit the condition of current. Current flow through an element that is capable of current conduction requires that the net charge observed at any elemental cross-section change with time. Equivalently, a net nonzero charge, $q(t)$, must be transferred over finite time across any cross-sectional area of the element. The current, $i(t)$, that actually flows is the time rate of change of this transferred charge:

$$i(t) = \frac{dq(t)}{dt} \tag{18.9}$$

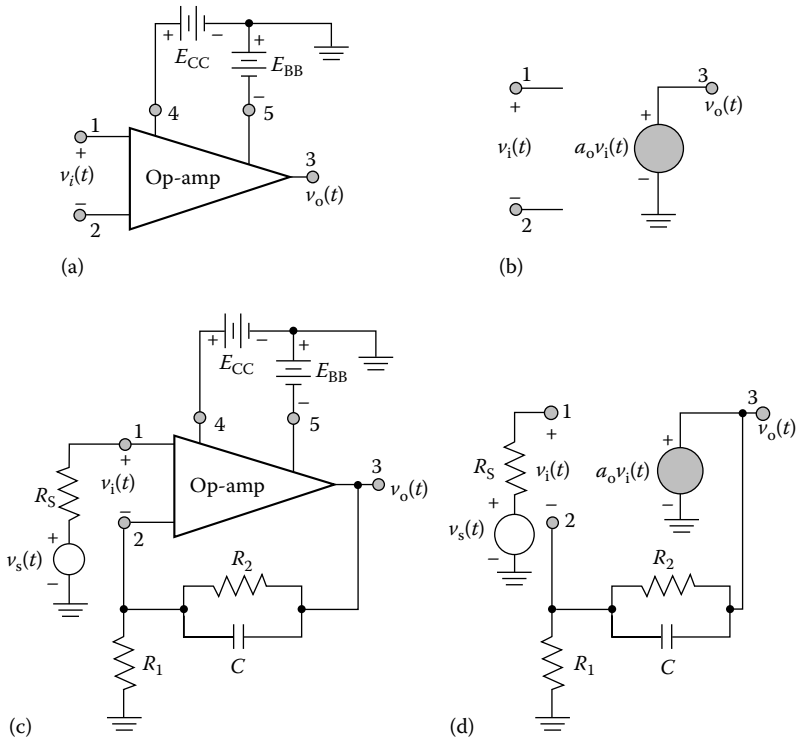


FIGURE 18.4 (a) Circuit schematic symbol for a voltage-mode operational amplifier. (b) First-order linear model of the op-amp. (c) A voltage amplifier realized with the op-amp functioning as the gain element. (d) Equivalent circuit of the voltage amplifier in (c).

where

MKS unit of charge is the coulomb
time t is measured in seconds
resultant current is measured in units of amperes

Note that zero current does not necessarily imply a lack of charge at a given cross-section of a conductive element. Instead, zero current implies only that the subject charge is not changing with time, that is, the charge is not moving through the elemental cross-section.

Electrical charge can be negative, as in the case of electrons transported through a cross-section of a conductive element such as aluminum or copper. A single electron has a charge of $-(1.6021 \times 10^{-19})$ C. Thus, Equation 18.9 implies a need to transport an average of (6.242×10^{18}) electrons in 1 s through a cross-section of aluminum if the aluminum element is to conduct a constant current of 1 A. Charge can also be positive, as in the case of holes transported through a cross-section of a semiconductor such as germanium or silicon. Hole transport in a semiconductor is actually electron transport at an energy level that is smaller than the energy required to effect electron transport in that semiconductor. To first order, therefore, the electrical charge of a hole is the negative of the charge of an electron, which implies that the charge of a hole is $+(1.602 \times 10^{-19})$ C.

A positive charge, $q(t)$, transported from the left of the cross-section to the right of the cross-section in the element abstracted in Figure 18.5a gives rise to a positive current, $i(t)$, which also flows from left to right across the indicated cross-section. Assume that, prior to the transport of such charge, the volumes to the left and to the right of the cross-section are electrically neutral, that is, these volumes have zero

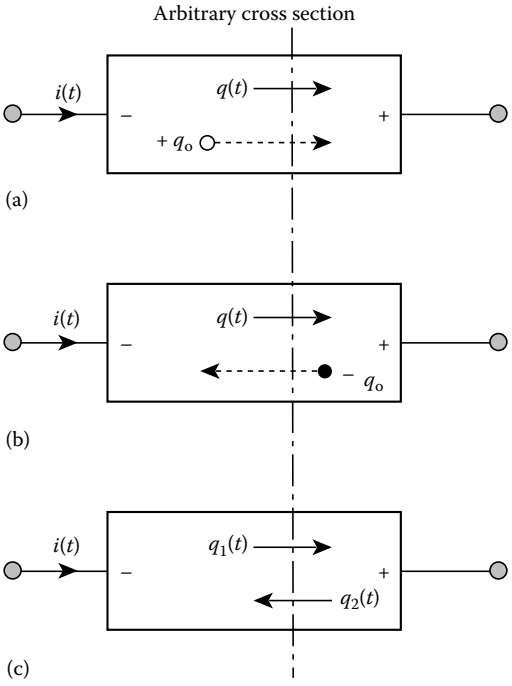


FIGURE 18.5 (a) Transport of a positive charge from the left-hand side to the right-hand side of an arbitrary cross-section of a conductive element. (b) Transport of a negative charge from the right-hand side to the left-hand side of an arbitrary cross-section of a conductive element. (c) Transport of positive or negative charges from either side to the other side of an arbitrary cross-section of a conductive element.

initial net charge. Then, the transport of a positive charge, q_0, from the left side to the right side of the element charges the right side to $+1q_0$ and the left side to $-1q_0$.

Alternatively, suppose a negative charge in the amount of $-q_0$ is transported from the right side of the element to its left side, as suggested in Figure 18.5b. Then, the left side charges to $-q_0$, and the right side charges to $+q_0$, which is identical to the electrostatic condition incurred by the transport of a positive charge in the amount of q_0 from left- to right-hand sides. As a result, the transport of a net negative charge from right to left produces a positive current, $i(t)$, flowing from left to right, just as positive charge transported from left- to right-hand sides induces a current flow from left to right.

Assume, as portrayed in Figure 18.5c, that a positive or a negative charge, say, $q_1(t)$, is transported from the left side of the indicated cross-section to the right side. Simultaneously, a positive or a negative charge in the amount of $q_2(t)$ is directed through the cross-section from right to left. If $i_1(t)$ is the current arising from the transport of the charge $q_1(t)$, and if $i_2(t)$ denotes the current corresponding to the transport of the charge, $q_2(t)$, the net effective current $i_e(t)$, flowing from the left side of the cross-section to the right side of the cross-section is

$$i_e(t) = \frac{d}{dt}[q_1(t) - q_2(t)] = i_1(t) - i_2(t) \tag{18.10}$$

where the charge difference, $[q_1(t) - q_2(t)]$, represents the net charge transported from left to right. Observe that if $q_1(t) + q_2(t)$, the net effective current is zero, even though conceivably large numbers of charges are transported back and forth across the junction.

18.1.2 Energy and Voltage

The preceding section highlights the fundamental physical fact that the flow of current through a conductive electrical element mandates that a net charge be transported over finite time across any arbitrary cross-section of that element. The electrical effect of this charge transport is a net positive charge induced on one side of the element in question and a net negative charge (equal in magnitude to the aforementioned positive charge) mirrored on the other side of the element. This ramification conflicts with the observable electrical properties of an element in equilibrium. In particular, an element sitting in free space, without any electrical connection to a source of energy, is necessarily in equilibrium in the sense that the net positive charge in any volume of the element is precisely counteracted by an equal amount of charge of opposite sign in said volume. Thus, if none of the elements abstracted in Figure 18.5 is connected to an external source of energy, it is physically impossible to achieve the indicated electrical charge differential that materializes across an arbitrary cross-section of the element when charge is transferred from one side of the cross-section to the other.

The energy commensurate with sustaining current flow through an electrical element derives from the application of a voltage, $v(t)$, across the element in question. Equivalently, the application of electrical energy to an element manifests itself as a voltage developed across the terminals of an element to which energy is supplied. The amount of applied voltage, $v(t)$, required to sustain the flow of current, $i(t)$, as diagrammed in Figure 18.6a, is precisely the voltage required to offset the electrostatic implications of the differential charge induced across the element through which $i(t)$ flows. This is to say that without the connection of the voltage, $v(t)$, to the element in Figure 18.6a, the element cannot be in equilibrium. With $v(t)$ connected, equilibrium for the entire system comprised of element and voltage source is reestablished by allowing for the conduction of the current, $i(t)$.

Instead of viewing the delivery of energy to an electrical element as the ramification of a voltage source applied to the element, the energy delivery may be interpreted as the upshot of a current source used to excite the element, as depicted in Figure 18.6b. This interpretation follows from the fact that energy must be applied in an amount that effects charge transport at a desired time rate of change. It follows that the application of a current source in the amount of the desired current is necessarily in one-to-one

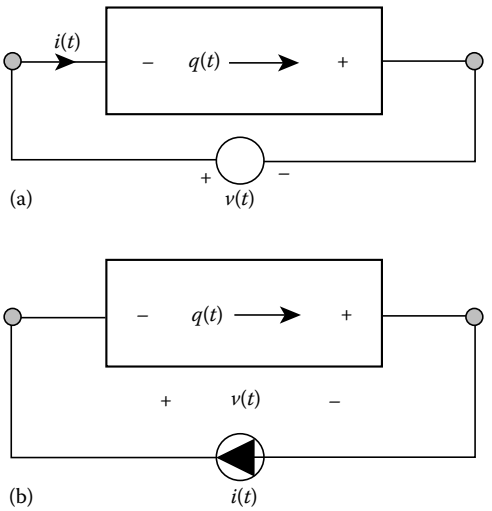


FIGURE 18.6 (a) The application of energy in the form of a voltage applied to an element that is made to conduct a specified current. The applied voltage, $v(t)$, causes the current, $i(t)$, to flow. (b) The application of energy in the form of a current applied to an element that is made to establish a specified terminal voltage. The applied current, $i(t)$, causes the voltage, $v(t)$, to be developed across the terminals of the electrical element.

correspondence with the voltage required to offset the charge differential manifested by the charge transport that yields the subject current. To be sure, a voltage source is a physical entity, while current source is not, but the mathematical modeling of energy delivery to an electrical element can nonetheless be accomplished through either a voltage source or a current source.

In Figure 18.6, the terminal voltage, $v(t)$, corresponding to the energy, $w(t)$, required to transfer an amount of charge, $q(t)$, across an arbitrary cross-section of the element is

$$v(t) = \frac{dw(t)}{dq(t)} \tag{18.11}$$

where

$v(t)$ is in units of volts when $q(t)$ is expressed in coulombs
$w(t)$ is specified in joules

Thus, if 1 J of applied energy results in the transport of 1 C of charge through an element, the elemental terminal voltage manifested by the 1 J of applied energy is 1 V.

It should be understood that the derivative on the right-hand side of Equation 18.11, and thus the terminal voltage demanded of an element that is transporting a certain amount of charge through its cross-section, is a function of the properties of the type of material from which the element undergoing study is fabricated. For example, an insulator such as paper, air, or silicon dioxide is ideally incapable of current conduction and hence, intrinsic charge transport. Thus, $q(t)$ is essentially zero in an insulator and by Equation 18.11, an infinitely large terminal voltage is required for even the smallest possible current. In a conductor such as aluminum, iron, or copper, large amounts of charge can be transported for very small applied energies. Accordingly, the requisite terminal voltage for even very large currents approaches zero in ideal conductors. The electrical properties of semiconductors such as germanium, silicon, and gallium arsenide lie between the extremes of those for an insulator and a conductor. In particular, semiconductor elements behave as insulators when their terminals are subjected to small voltages, while progressively larger terminal voltages render the electrical behavior of semiconductors akin to conductors. This conditional conductive property of a semiconductor explains why semiconductor devices and circuits generally must be biased to appropriate voltage levels before these devices and circuits can function in accordance with their requirements.

18.1.3 Power

The foregoing material underscores the fact that the flow of current through a two-terminal element, or more generally, through any two terminals of an electrical network, requires that charge be transported over time across any cross-section of that element or network. In turn, such charge transport requires that energy be supplied to the network, usually through the application of an external voltage source. The time rate of change of this applied energy is the power delivered by the external voltage or current source to the network in question. If $p(t)$ denotes this power in units of watts

$$p(t) = \frac{dw(t)}{dq} \tag{18.12}$$

where, of course, $w(t)$ is the energy supplied to the network in joules. By rewriting Equation 18.12 in the form

$$p(t) = \frac{dw(t)}{dq(t)} \frac{dq(t)}{dt} \tag{18.13}$$

and applying Equations 18.9 and 18.11, the power supplied to the two terminals of an element or a network becomes the more expedient relationship

$$p(t) = v(t)i(t) \tag{18.14}$$

Equation 18.14 expresses the power delivered to an element as a simple product of the voltage applied across the terminals of the element and the resultant current conducted by that element. However, care must be exercised with respect to relative voltage and current polarity, when applying Equation 18.14 to practical circuits.

To the foregoing end, it is useful to revisit the simple abstraction of Figure 18.6a, which is redrawn as the slightly modified form in Figure 18.7. In this circuit, a signal source voltage, $v_s(t)$, is applied across the two terminals, 1 and 2, of an element, which responds by conducting a current $i(t)$, from terminal 1 to terminal 2 and developing a corresponding terminal voltage $v(t)$, as illustrated. If the wires (zero resistance conductors, as might be approximated by either aluminum or copper interconnects) that connect the signal source to the element are ideal, the voltage, $v(t)$, is identical to $v_s(t)$. Moreover, because the current is manifested by the application of the signal source, which thereby establishes a closed electrical path for current conduction, the element current, $i(t)$, is necessarily the same as the current, $i_s(t)$, that flows through $v_s(t)$.

If attention is focused on only the element in Figure 18.7, it is natural to presume that the current conducted by the element actually flows from terminal 1 to terminal 2 when (as shown) the voltage developed across the element is positive at terminal 1 with respect to terminal 2. This assertion may be rationalized qualitatively by noting that the positive voltage nature at terminal 1 acts to repel positive charges from terminal 1 to terminal 2, where the negative nature of the developed voltage, $v(t)$, tends to attract the repulsed positive charges. Similarly, the positive nature of the voltage at terminal 1 serves to attract negative charges from terminal 2, where the negative nature of $v(t)$ tends to repel such negative charges. Because current flows in the direction of transported positive charge and opposite to the direction of transported negative charge, either interpretation gives rise to an elemental current, $i(t)$, which flows from terminal 1 to terminal 2. In general, if current is indicated as flowing from the "high" (+) voltage terminal to the "low" (−) voltage terminal of an element, the current conducted by the element and the voltage developed across the element to cause this flow of current are said to be in associated reference polarity. When the element current, $i(t)$, and the corresponding element voltage, $v(t)$, as exploited in the defining power relationship of Equation 18.14, are in associated reference polarity, the resulting computed power is a positive number and is said to represent the power delivered to the element. In contrast, $v(t)$ and $i(t)$ are said to be in disassociated reference polarity when $i(t)$ flows from the "low" voltage terminal of the element to its "high" voltage terminal. In this case the voltage-current product in Equation 18.14 is a negative number. Instead of stating that the resulting negative power is delivered to the element, it is more meaningful to assert that the computed negative power is a positive power that is generated by the element in question.

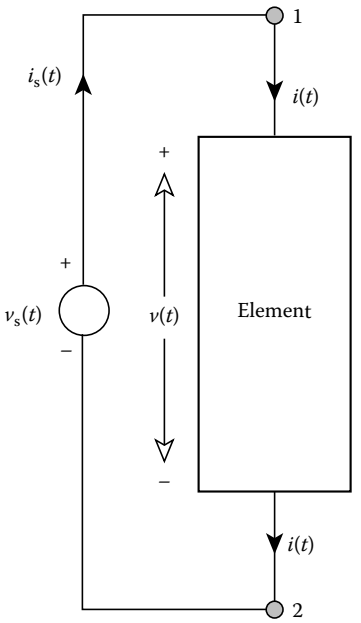


FIGURE 18.7 Circuit used to illustrate power calculations and the associated reference polarity convention.

At first glance, it may appear as though the latter polarity disassociation between element voltage and current variables is an impossible circumstance. Not only is polarity disassociation possible, it is absolutely necessary if electrical circuits are to subscribe to the fundamental principle of conversation of power. This

principle states that the net power dissipated by a circuit must be identical to the net power supplied to that circuit. A confirmation of this basic principle derives from a further consideration of the topology in Figure 18.7. The electrical variables, $v(t)$ and $i(t)$, pertinent to the element delineated in this circuit, are in associated reference polarity. Accordingly, the power, $p_e(t)$, dissipated by this element is positive and given by Equation 18.14:

$$p_e(t) = v(t)i(t) \tag{18.15}$$

However, the voltage and current variables, $v_s(t)$ and $i_s(t)$, relative to the signal source voltage are in disassociated polarity. It follows that the power, $p_s(t)$, delivered to the signal source is

$$p_s(t) = -v_s(t)i_s(t) \tag{18.16}$$

Because, as stated previously, $v_s(t) = v(t)$ and $i_s(t) = i(t)$, for the circuit at hand, Equation 18.16 can be written as

$$p_s(t) = -v(t)i(t) \tag{18.17}$$

The last result implies that the

$$\text{power delivered by the signal source} = +v(t)i(t) \equiv p_e(t) \tag{18.18}$$

that is, the power delivered to the element by the signal source is equal to the power dissipated by the element.

An alternative statement to conservation of power, as applied to the circuit in Figure 18.7 derives from combining Equations 18.15 and 18.17 to arrive at

$$p_s(t) + p_e(t) = 0 \tag{18.19}$$

The foregoing result may be generalized to the case of a more complex circuit comprised of an electrical interconnection of N elements, some of which may be voltage and current sources. Let the voltage across the kth element by $v_k(t)$, and let the current flowing through this kth element, in associated reference polarity with $v_k(t)$, be $i_k(t)$. Then, the power, $p_k(t)$, delivered to the kth electrical element is $v_k(t)\ i_k(t)$. By conservation of power,

$$\sum_{k=1}^{N} p_k(t) = \sum_{k=1}^{N} v_k(t) = i_k(t) = 0 \tag{18.20}$$

The satisfaction of the expression requires that at least one of the $p_k(t)$ be negative, or equivalently, at least one of the N elements embedded in the circuit at hand must be a source of energy.

18.2 Circuit Classifications

It was pointed out earlier that the relationship between the current that is made to flow through an electrical element and the applied energy, and thus voltage, that is required to sustain such current flow is dictated by the material from which the subject element is fabricated. The element material and the associated manufacturing methods exploited to realize a particular type of circuit element determine the mathematical nature between the voltage applied across the terminals of the element and the resultant

current flowing through the element. To this end, electrical elements and circuits in which they are embedded are generally codified as linear or nonlinear, active or passive, time-varying or time-invariant, and lumped or distributed.

18.2.1 Linear vs. Nonlinear

A linear two-terminal circuit element is one for which the voltage developed across, and the current flowing through, are related to one another by a linear algebraic or a linear integro-differential equation. If the relationship between terminal voltage and corresponding current is nonlinear, the element is said to be nonlinear. A linear circuit contains only linear circuit elements, while a circuit is said to be nonlinear if a least one of its embedded electrical elements is nonlinear.

All practical circuit elements, and thus all practical electrical networks, are inherently nonlinear. However, over suitably restricted ranges of applied voltages and corresponding currents, the volt–ampere characteristics of these elements and networks emulate idealized linear relationships. In the design of an electronic linear signal processor, such as an amplifier, an implicit engineering task is the implementation of biasing subcircuits that constrain the voltages and currents of internal semiconductor elements to ranges that ensure linear elemental behavior over all possible operating conditions.

The voltage–current relationship for the linear resistor offered in Figure 18.8a is

$$v(t) = Ri(t) \tag{18.21}$$

where the voltage, $v(t)$, appearing across the terminals of the resistor and the resultant current, $i(t)$, conducted by the resistor are in associated reference polarity. The resistance, R, is independent of either $v(t)$ or $i(t)$. From Equation 18.14, the dissipated resistor power, which is mainfested in the form of heat is

$$p_r(t) = i^2(t)R = \frac{v^2(t)}{R} \tag{18.22}$$

The linear capacitor and the linear inductor, with schematic symbols that appear, respectively, in Figure 18.8b and c, store energy as opposed to dissipating power. Their volt–ampere equations are the linear relationships

$$i(t) = C\frac{dv(t)}{dt} \tag{18.23}$$

for the capacitor, whereas for the inductor in Figure 18.8c,

$$v(t) = L\frac{di(t)}{dt} \tag{18.24}$$

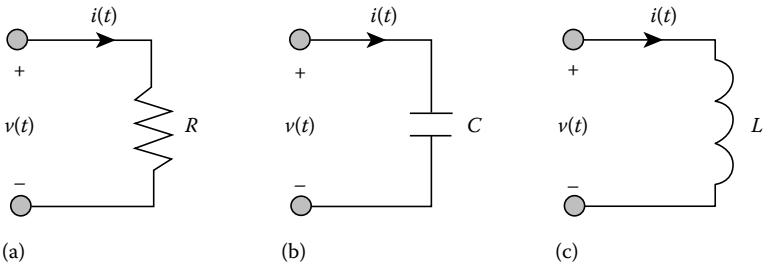


FIGURE 18.8 Circuit schematic symbol and corresponding voltage and current notation for (a) linear resistor, (b) linear capacitor, and (c) linear inductor.

Observe from Equations 18.23 and 18.14 that the power, $p_c(t)$, delivered to the linear capacitor is

$$p_c(t) = v(t)i(t) = Cv(t)\frac{dv(t)}{dt} \tag{18.25}$$

From Equation 18.12, this power is related to the energy, for example, $w_c(t)$, stored in the form of charge deposited on the plates of the capacitor by

$$Cv(t)dv(t) = dw_c(t) \tag{18.26}$$

It follows that the energy delivered to, and hence stored in, the capacitor from time $t = 0$ to time t is

$$w_c(t) = \frac{1}{2}Cv^2(t) \tag{18.27}$$

It should be noted that this stored energy, like the energy associated with a signal source or a battery voltage, is available to supply power to other elements in the network in which the capacitor is embedded. For example, if very little current is conducted by the capacitor in question, Equation 18.23 implies that the voltage across the capacitor is essentially constant. However, an element whose terminal voltage is a constant and in which energy is stored and therefore available for use behaves as a battery.

If the preceding analysis is repeated for the inductor of Figure 18.8c, it can be shown that the energy, $w_l(t)$, stored in the inductive element form time $t = 0$ to time t is

$$w_l(t) = \frac{1}{2}Li^2(t) \tag{18.28}$$

Although an energized capacitor conducting almost zero current functions as a voltage source, an energized inductor supporting almost zero terminal voltage emulates a constant current source.

18.2.2 Active vs. Passive

An electrical element or network is said to be passive if the power delivered to it, defined in accordance with Equation 18.14, is positive. This definition exploits the requirement that the terminal voltage, $v(t)$, and the element current $i(t)$, appearing in Equation 18.14 be in associated reference polarity. In contrast, an element or network to which the delivered power is negative is said to be active, that is, an active element or network generates power instead of dissipating it.

Conventional two-terminal resistors, capacitors, and inductors are passive elements. It follows that networks formed of interconnected two-terminal resistors, capacitors, and inductors are passive networks. Two-terminal voltage and current sources generally behave as active elements. However, when more than one source of externally applied energy is present in an electrical network, it is possible for one more of these sources to behave as passive elements. Comments similar to those made in conjunction with two-terminal voltage and current sources apply equally well to each of the four possible dependent generators. Accordingly, multiterminal configurations, whose models exploit dependent sources, can behave as either passive or active networks.

18.2.3 Time-Varying vs. Time-Invariant

The elements of a circuit are defined electrically by an identifying parameter, such as resistance, capacitance, inductance, and the gain factors associated with dependent voltage or current sources. An

element whose identifying parameter changes as a function of time is said to be a time-varying element. If said parameter is a constant over time, the element in question is time-invariant. A network containing at least one time-varying electrical element it is said to be a time-varying network. Otherwise, the network is time-invariant. Excluded from the list of elements whose electrical character establishes the time variance or time invariance of a considered network are externally applied voltage and current sources. Thus, for example, a network with internal elements that are exclusively time-invariant resistors, capacitors, inductors, and dependent sources, but which is excited by a sinusoidal signal source, is nonetheless a time-invariant network.

Although some circuits, and particularly electromechanical networks, are purposely designed to exhibit time-varying volt–ampere characteristics, parametric time variance is generally viewed as a parasitic phenomena in the majority of practical circuits. Unfortunately, a degree of parametric time variance is unavoidable in even those circuits that are specifically designed to achieve input–output response properties that closely approximate time-invariant characteristics. For example, the best of network elements exhibit a slow aging phenomenon that shifts the values of its intrinsic physical parameters. The upshot of these shifts is electrical circuits where overall performance deteriorates with time.

18.2.4 Lumped vs. Distributed

Electrons in conventional conductive elements are not transported instantaneously across elemental cross sections, but their transport velocities are very high. In fact, these velocities approach the speed of light, say c, which is (3×10^8) m/s or about 982 ft./μs. Electrons and holes in semiconductors are transported at somewhat slower speeds, but generally no less than an order of magnitude or so smaller than the speed of light. The time required to transport charge from one terminal of a two-terminal electrical element to its other terminal, compared with the time required to propagate energy uniformly through the element, determines whether an element is lumped or distributed. In particular, if the time required to transport charge through an element is significantly smaller than the time required to propagate the energy through the element that is required to incur such charge transport, the element in question is said to be lumped. On the other hand, if the charge transport time is comparable to the energy propagation time, the element is said to be distributed.

The concept of a lumped, as opposed to a distributed, circuit element can be qualitatively understood through a reconsideration of the circuit provided in Figure 18.7. As argued, the indicated element current, $i(t)$, is identical to the indicated source current, $i_s(t)$. This equality implies that $i(t)$, is effectively circulating around the loop that is electrically formed by the interconnection of the signal source voltage, $v_s(t)$, to the element. Equivalently, the subject equality implies that $i(t)$ is entering terminal 1 of the element and simultaneously is exiting at terminal 2, as illustrated. Assuming that the element at hand is not a semiconductor, the current, $i(t)$, arises from the transport of electrons through the element in a direction opposite to that of the indicated polarity of $i(t)$. Specifically, electrons must be transported from terminal 2, at the bottom of the element, to terminal 1, at the top of the element, and in turn the requisite amount of energy must be applied in the immediate neighborhoods of both terminals. The implication of presuming that the element at hand is lumped is that $i(t)$ is entering terminal 1 at precisely the same time that it is leaving terminal 2. Such a situation is clearly impossible, for it mandates that electrons be transported through the entire length of the element in zero time. However, given that electrons are transported at a nominal velocity of 982 ft./μs, a very small physical elemental length renders the approximation of zero electron transport time reasonable. For example, if the element is $\frac{1}{2}$ in. long (a typical size for an off-the-shelf resistor), the average transport time for electrons in this unit is only about 42.4 ps. As long as the period of the applied excitation, $v_s(t)$, is significantly larger than 42.4 ps, the electron transport time is significantly smaller than the time commensurate with the propagation of this energy through the entire element. A period of 42.4 ps corresponds to a signal whose frequency of approximately 23.6 GHz. Thus, a $\frac{1}{2}$ in. resistive element excited by a signal whose frequency is significantly smaller than 23.6 GHz can be viewed as a lumped circuit element.

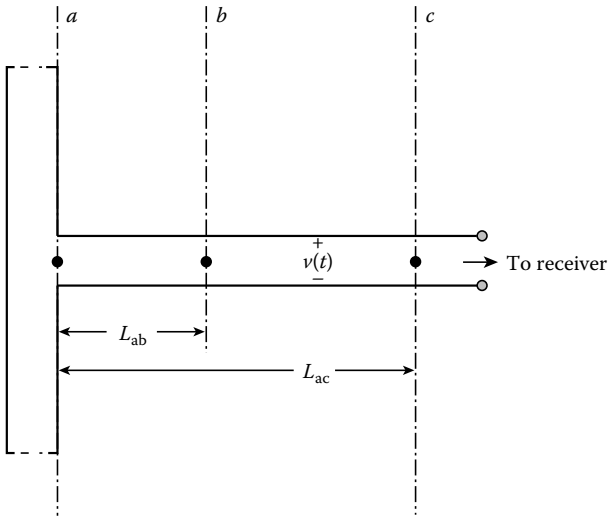


FIGURE 18.9 Schematic abstraction of a dipole antenna for an FM receiver application.

In the vast majority of electrical and electronic networks it is difficult not to satisfy the lumped circuit approximation. Nevertheless, several practical electrical systems cannot be viewed as lumped entities. For example, consider the lead-in wire that connects the antenna input terminals of a frequency-modulated (FM) radio receiver to an antenna, as diagrammed in Figure 18.9. Let the signal voltage, $v_a(t)$, across the lead-in wires at point "a" be the sinusoid

$$v_a(t) = V_M \cos(\omega t) \tag{18.29}$$

where

V_M represents the amplitude of the signal
ω is its frequency in units of radians per second

Consider the case in which $\omega = 2\pi(103.5 \times 10^6)$ rad/s, which is a carrier frequency lying within the commercial FM broadcast band. This high signal frequency makes the length of antenna lead-in wire critically important for proper signal reception.

In an attempt to verify the preceding contention, let the voltage developed across the lead-in lines at point "b" in Figure 18.9 be denoted as $v_b(t)$, and let point "b" be 1 ft. displaced from point "a"; that is, $L_{ab} = 1$ ft. The time, τ_{ab} required to transport electrons over the indicated length, L_{ab}, is

$$\tau_{ab} = \frac{L_{ab}}{c} = 1.018 \text{ ns} \tag{18.30}$$

Thus, assuming an idealized line in the sense of zero effective resistance, capacitance, and inductance, the signal, $v_b(t)$, at point "b" is seen as the signal appearing at "a," delayed by approximately 1.02 ns. It follows that

$$v_b(t) = V_M \cos[\omega(t - \tau_{ab})] = V_M \cos(\omega t - 0.662) \tag{18.31}$$

where the phase angle associated with $v_b(t)$ is 0.662 rad, or almost 38°. Obviously, the signal established at point "b" is a significantly phase-shifted version of the signal presumed at point "a."

An FM receiver can effectively retrieve the signal voltage, $v_a(t)$, by detecting a phase-inverted version of $v_a(t)$ at its input terminals. To this end, it is of interest to determine the length, L_{ac}, such that the signal, $v_c(t)$, established at point "c" in Figure 18.9 is

$$v_c(t) = V_M \cos(\omega t - \pi) \tag{18.32}$$

The required phase shift of 180°, or π radians, corresponds to a time delay, τ_{ac}, of

$$\tau_{ac} = \frac{\pi}{\omega} = 4.831 \text{ ns} \tag{18.33}$$

In turn, a time delay of τ_{ac} implies a required line length, L_{ac} of

$$L_{ac} = c\tau_{ac} = 4.744 \text{ ft.} \tag{18.34}$$

A parenthetically important point is the observation that the carrier frequency of 103.5 MHz corresponds to a wavelength, λ, of

$$\lambda = \frac{2\pi c}{\omega} = 9.489 \text{ ft.} \tag{18.35}$$

Accordingly, the lead-in length computed in Equation 18.34 is $\lambda/2$; that is, a half-wavelength.

19

Network Laws and Theorems

Ray R. Chen
San Jose State University

Artice M. Davis
San Jose State University

Marwan A. Simaan
University of Pittsburgh

19.1 Kirchhoff's Voltage and Current Laws **19**-1
Nodal Analysis • Mesh Analysis • Fundamental Cutset-Loop Circuit Analysis
References **19**-41
19.2 Network Theorems **19**-42
Superposition Theorem • Thévenin Theorem • Norton Theorem • Maximum Power Transfer Theorem • Reciprocity Theorem

19.1 Kirchhoff's Voltage and Current Laws

Circuit analysis, like Euclidean geometry, can be treated as a mathematical system, that is, the entire theory can be constructed upon a foundation consisting of a few fundamental concepts and several axioms relating these concepts. As it happens, important advantages accrue from this approach—it is not simply a desire for mathematical rigor, but a pragmatic need for simplification that prompts us to adopt such a mathematical attitude.

The basic concepts are conductor, element, time, voltage, and current. Conductor and element are axiomatic, thus, they cannot be defined, only explained. A conductor is the idealization of a piece of copper wire; an element is a region of space penetrated by two conductors of finite length termed leads and pronounced ''leeds.'' The ends of these leads are called terminals and are often drawn with small circles as in Figure 19.1.

Conductors and elements are the basic objects of circuit theory; we will take time, voltage, and current as the basic variables. The time variable is measured with a clock (or, in more picturesque language, a chronometer). Its unit is the second, s. Thus, we will say that time, like voltage and current, is defined operationally, that is, by means of a measuring instrument and a procedure for measurement. Our view of reality in this context is consonant with that branch of philosophy termed operationalism [1].

Voltage is measured with an instrument called a voltmeter, as illustrated in Figure 19.2. In Figure 19.2, a voltmeter consists of a readout device and two long, flexible conductors terminated in points called probes that can be held against other conductors, thereby making electrical contact with them. These conductors are usually covered with an insulating material. One is often colored red and the other black. The one colored red defines the positive polarity of voltage, and the other the negative polarity. Thus, voltage is always measured between two conductors. If these two conductors are element leads, the voltage is that across the corresponding element. Figure 19.3 is the symbolic description of such a measurement;

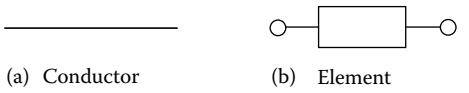


FIGURE 19.1 Conductors and elements.

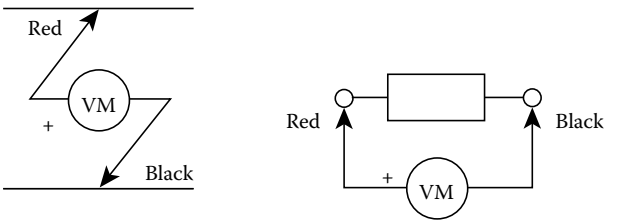


FIGURE 19.2 Operational definition of voltage.

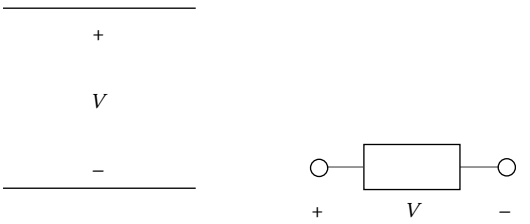


FIGURE 19.3 Symbolic description of the voltage measurement.

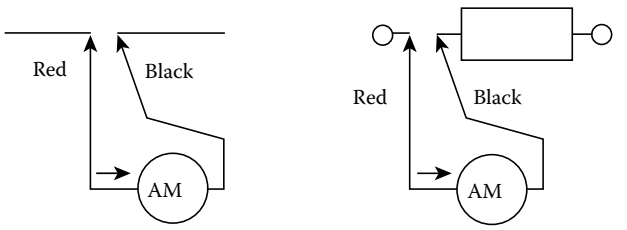


FIGURE 19.4 Operational definition of current.

the variable v, along with the corresponding plus and minus signs, means exactly the experimental procedure depicted in Figure 19.2, neither more nor less. The outcome of the measurement, incidentally, can be either positive or negative. Thus, a reading of $v = -12$ V, for example, has meaning only when viewed within the context of the measurement. If the meter leads are simply reversed after the measurement just described, a reading of $v' = +12$ V will result. The latter, however, is a different variable; hence, we have changed the symbol to v'. The V after the numerical value is the unit of voltage, the volt, V.

Although voltage is measured across an element (or between conductors), current is measured through a conductor or element. Figure 19.4 provides an operational definition of current. One cuts the conductor or element lead and touches one meter lead against one terminal thus formed and the other against the second. A shorthand symbol for the meter connection is an arrow close to one lead of the ammeter. This arrow, along with the meter reading, defines the current. We show the shorthand symbol for a current in Figure 19.5. The reference arrow and the symbol i are shorthand for the complete measurement in Figure 19.4—merely this and nothing more. The variable i can be either positive or negative; for example, one possible

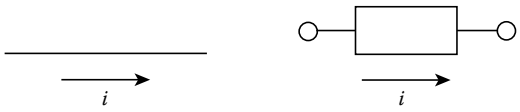


FIGURE 19.5 Symbolic representation of a current measurement.

TABLE 19.1 Summary of the Basic Concepts of Circuit Theory

Objects	Variables			
Conductor	Element	Time	Voltage	Current
—	—	Seconds, s	Volt, V	Ampere, A

outcome of the measurement might be $i = -5$ A. The A signifies the unit of current, the ampere. If the red and black leads in Figure 19.4 were reversed, the reading sign would change.

Table 19.1 provides a summary of the basic concepts of circuit theory: the two basic objects and the three fundamental variables. Notice that we are a bit at variance with the SI system here because although time and current are considered fundamental in that system, voltage is not. Our approach simplifies things, however, for one does not require any of the other SI units or dimensions. All other quantities are derived. For instance, charge is the integral of current and its unit is the ampere-second, or the coulomb, C. Power is the product of voltage and current. Its unit is the watt, W. Energy is the integral power, and has the unit of the watt-second, or joule, J. In this manner one avoids the necessity of introducing mechanical concepts, such as mechanical work, as being the product of force and distance.

In the applications of circuit theory, of course, one has need of the other concepts of physics. If one is to use circuit analysis to determine the efficiency of an electric motor, for example, the concept of mechanical work is necessary. However—and this is the main point of our approach—the introduction of such concepts is not essential in the analysis of a circuit itself. This idea is tied in to the concept of modeling. The basic catalog of elements used here does not include such things as temperature effects or radiation of electromagnetic energy. Furthermore, a ''real'' element such as resistor is not ''pure.'' A real resistor is more accurately modeled, for many purposes, as a resistor plus series inductance and shunt capacitance. The point is this: In order to adequately model the ''real world'' one must often use complicated combinations of the basic elements. Additionally, to incorporate the influence of variables such as temperature, one must assume that certain parameters (such as resistance or capacitance) are functions of that variable. It is the determination of the more complicated model or the functional relationship of a given parameter to, for example, temperatures that fall within the realm of the practitioner. Such ideas were discussed more fully in Chapter 18. Circuit analysis merely provides the tools for analyzing the end result.

The radiation of electromagnetic energy is, on the other hand, a quite different aspect of circuit theory. As will be seen, circuit analysis falls within a regime in which such behavior can be neglected. Thus, the theory of circuit analysis we will expound has a limited range of application: to low frequencies or, what is the same in the light of Fourier analysis, to waveforms that do not vary too rapidly.

We are now in a position to state two basic axioms, which we will assume all circuits obey:

Axiom 1: The behavior of an element is completely determined by its v–i characteristic, which can be determined by tests made on the element in isolation from the other elements in the circuit in which it is connected.

Axiom 2: The behavior of a circuit is independent of the size or the shape or the orientation of its elements, the conductors that interconnect them, and the element leads.

At this point, we loosely consider a circuit to be any collection of elements and conductors, although we will sharpen our definition a bit later. Axiom 1 means that we can run tests on an element in the laboratory, then wire it into a circuit and have the assurance that it will not exhibit any new and different behavior. Axiom 2 means that it is only the topology of a circuit that matters, not the way the circuit is stretched or bent or rearranged, so long as we do not change the listing of which element leads are connected to which others or to which conductors. Another way of saying this is that the circuit graph, described in Chapter 7, give all the necessary information about the way a circuit is wired.

The remaining two axioms are somewhat more involved and require some discussion of circuit topology. Consider, for a moment, the collection of elements in Figure 19.6. We labeled each element with a letter to distinguish it from the others. First, notice the two solid dots. We refer to them as joints. The idea is that they represent ''solder joints,'' where the ends of two or more leads or conductors were connected. If only two ends are connected we do not show the joints explicitly; where three or more are connected, however, they are drawn. We temporarily (as a test) erase all of the element bodies and replace them with open space. The result is given in Figure 19.7. We refer to each of the interconnected ''islands'' of a conductor as a node. This example circuit has six nodes, and we labeled them with the numbers one through six for identification purposes.

Axiom 3 (Kirchhoff's current law): The charge on a node or in an element is identically zero at all instants of time.

Kirchhoff's current law (KCL) is not usually phrased in quite this manner. Thus, let us consider the closed (or ''Gaussian'') surface S in Figure 19.8. We assume that it is penetrated only by conductors. The elements, of course, are there; we simply do not show them so that we can concentrate on the conductors. We have arbitrarily defined the currents in the conductors penetrating S. Now, recalling that charge is the time integral of the current and thus has the same direction as the current from which it is derived, one can phrase Axiom 3 as follows:

$$\sum_{S} q_{\text{in}} = q_{\text{enclosed}} = 0 \tag{19.1}$$

at each instant of time. This equation is simply one form of conservation of charge. Because current is the time derivative of voltage, one can also state that

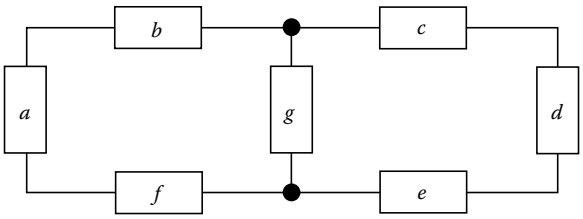


FIGURE 19.6 Example circuit.

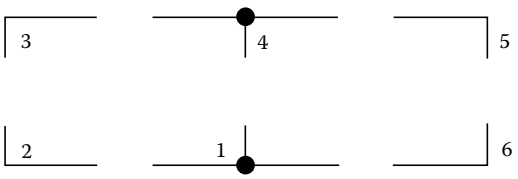


FIGURE 19.7 Nodes of the example circuit.

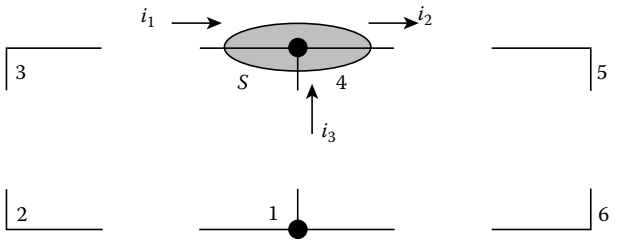


FIGURE 19.8 Illustration of KCL.

$$\sum_S i_{\text{in}} = 0 \tag{19.2}$$

at each and every time instant. This last equation is the usual phrasing of KCL. The subscript "in" means that a current reference pointed inward is to be considered positive; by default, therefore, a current with its reference pointed outward is to have a negative sign affixed. This sign is in addition to any negative sign that might be present in the value of each variable. For node 4 in Figure 19.8, KCL in its current form, therefore, reads

$$i_1 - i_2 + i_3 = 0 \tag{19.3}$$

Two other ways of expressing KCL (in current form) are

$$\sum_S i_{\text{out}} = 0 \tag{19.4}$$

and

$$\sum_S i_{\text{in}} = \sum_S i_{\text{out}} \tag{19.5}$$

The equivalent charge forms are also clearly valid. We emphasize the latter to a greater extent than is usual in the classical treatment because of the current interest in charge distribution and transfer circuits.

The Gaussian surface used to express KCL is not constrained to enclose only conductors. It can enclose elements as well, although it still can be penetrated by only conductors (which can be element leads). Thus, consider Figure 19.9, which illustrates the same circuit with which we have been working. Now, however, the elements are given and the Gaussian surface encloses three elements, as well as conductors carrying the currents previously defined. Because these currents are not carried in the conductors

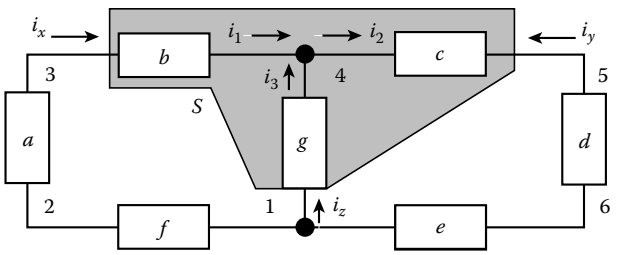


FIGURE 19.9 KCL for a more general surface.

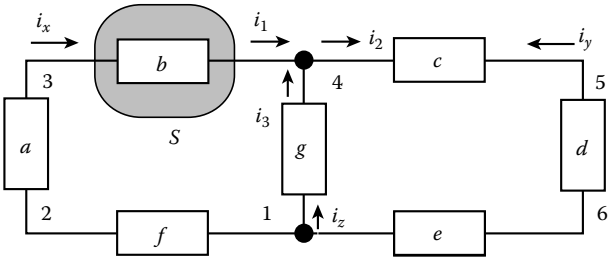


FIGURE 19.10 KCL for a single element.

penetrating the surface under consideration, they do not enter into KCL for that surface. Instead, KCL becomes

$$i_x + i_y + i_z = 0 \tag{19.6}$$

As a special case let us look once more at the preceding figure, but use a different surface, one enclosing only the element *b*. This is depicted in Figure 19.10. If we refer to Axiom 3, which notes that charge cannot accumulate inside an element, and apply charge conservation, we find that

$$i_x = i_1 \tag{19.7}$$

This states that the current into any element in one of its leads is the same as the current leaving in the other lead. In addition, we see that KCL for nodes and KCL for elements (both of which are implied by Axiom 3) imply that KCL holds for any general closed surface penetrated only by conductors such as the one used in connection with Figure 19.9.

In order to phrase our last axiom, we must discuss circuit topology a bit more, and we will continue to use the circuit just considered previously. We define a path to be an ordered sequence of elements having the property that any two consecutive elements in the sequence share a common node. Thus, referring for convenience back to Figure 19.10, we see that {*f*, *a*, *b*} is a path. The elements *f* and *a* share node 2 and *a* and *b* share node 3. One lead of the last element in a path is connected to a node that is not shared with the preceding element. Such a node is called the terminal node of the path. Similarly, one lead of the first element in the sequence is connected to a node that is not shared with the preceding element.* It is called the initial node of the path. Thus, in the example just cited, node 1 is the initial node and node 4 is the final node. Thus, a direction is associated with a path, and we can indicate it diagrammatically by means of an arrow on the circuit. This is illustrated in Figure 19.11 for the path $P_1 = \{f, a, b\}$ and $P_2 = \{g, c, d, e\}$.

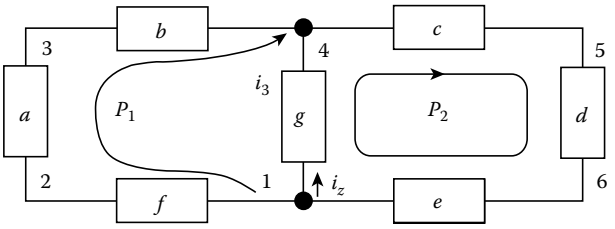


FIGURE 19.11 Circuit paths.

* We assume that no element has its two leads connected together and that more than two elements are in the path in this definition.

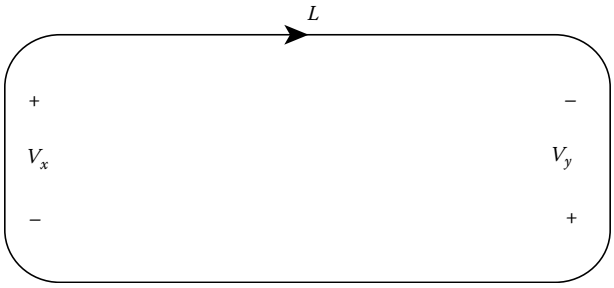


FIGURE 19.12 Voltage rise and drop.

If the initial node is identical to the terminal node, then the corresponding path is called a loop. An example is $\{f, a, b, g\}$. The patch P_2 is a loop. An alternate definition of a loop is as a collection of branches having the property that each node connected to a patch branch is connected to precisely two path branches, that is, it has degree two relative to the path branches.

We can define the voltage across each element in our circuit in exactly two ways, corresponding to the choices of which lead is designated plus and which is designated minus. Figure 19.12 presents two voltages and a loop L in a highly stylized manner. We have purposely not drawn the circuit itself so that we can concentrate on the essentials in our discussion. If the path enters the given element on the lead carrying the minus and exits on the one carrying the positive, its voltage will be called a voltage rise; however, if it enters on the positive and exits on the minus, the voltage will be called a voltage drop. If the signs of a voltage are reversed and a negative sign is affixed to the voltage variable, the value of that variable remains unchanged; thus, note that a negative rise is a drop, and vice versa.

We are now in a position to state our fourth and final axiom:

Axiom 4 (Kirchhoff's voltage law): The sum of the voltage rises around any loop is identically zero at all instants of time.

We refer to this law as Kirchhoff's voltage law (KVL) for the sake of economy of space. Just as KCL was phrased in terms of charge, KVL could just as well be phrased in terms of flux linkage. Flux linkage is the time integral of voltage, so it can be said that the sum of the flux linkages around a loop is zero. In voltage form, we write

$$\sum_{\text{loop}} v_{\text{rises}} = 0 \tag{19.8}$$

We observed that a negative rise is a drop, so

$$\sum_{\text{loop}} v_{\text{drops}} = 0 \tag{19.9}$$

or

$$\sum_{\text{loop}} v_{\text{rises}} = \sum_{\text{loop}} v_{\text{drops}} \tag{19.10}$$

Thus, in Figure 19.13, we could write (should we choose to use the form of Equation 19.8)

$$v_x - v_y - v_a + v_b = 0 \tag{19.11}$$

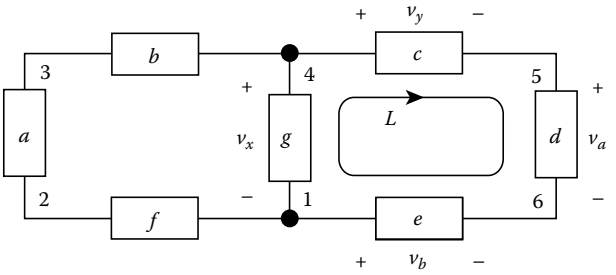


FIGURE 19.13 Illustration of KVL.

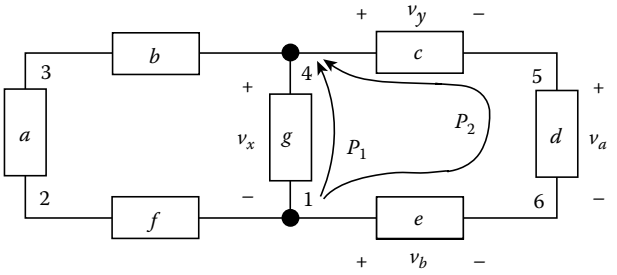


FIGURE 19.14 Path form of KVL.

Clearly, one can rearrange KVL into many different algebraic forms that are equivalent to those just stated; one form, however, is more useful in circuit computations than many others. It is known as the path form of KVL. To better appreciate this form, review Figure 19.13. This time, however, the paths are defined a bit differently. As illustrated in Figure 19.14, we consider two paths, P_1 and P_2, having the same initial and terminal nodes, 1 and 4, respectively.* We can rearrange Equation 19.11 into the form

$$v_x = -v_b + v_a + v_y \tag{19.12}$$

This form is often useful for finding one unknown voltage in terms of known voltages along some given path. In general, if P_1 and P_2 are two paths having the same initial and final nodes,

$$\sum_{P_1} v_{\text{rises}} = \sum_{P_2} v_{\text{rises}} = 0 \tag{19.13}$$

Be careful to distinguish this equation from Equation 19.10. In the present case two paths are involved; in the former we find only a single loop, and drops are located on one side of the equation and rises on the other. One might call the path form the ''all roads lead to Rome'' form.

We covered four basic axioms, and these are all that are needed to construct a mathematical theory of circuit analysis. The first axiom is often referred to by means of the phrase ''lumped circuit analysis,'' for we assume that all the physics of a given element are internal to that element and are of no concern to us; we are only interested in the v–i characteristic. That is, we are treating all the elements as lumps of matter that interact with the other elements in a circuit by means of the voltage and current at their leads.

* If one defines the negative of a path as a listing of the same elements as the original path in the reverse order and summation of two paths as a concatenation of the two listings, one sees that $P_1 - P_2 = L$, the loop in Figure 19.13.

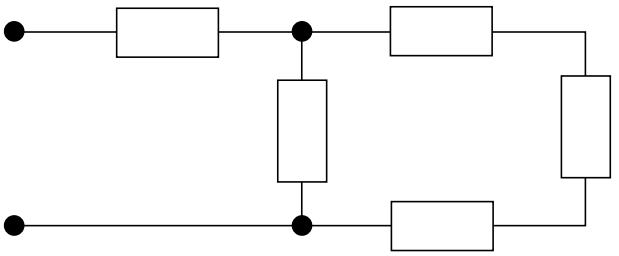


FIGURE 19.15 Noncircuit.

The second axiom says that the physical construction is irrelevant and that the interconnections are completely described by means of the circuit graph. KCL is an expression of conservation of charge, plus the assumption that neither conductors nor elements can maintain a net charge. In this connection, observe that a capacitor maintains a charge separation internally, but it is a separation of two charges of opposite sign; thus, the total algebraic charge within it is zero. Finally, KVL is an expression of conservation of flux linkage. If $\mathrm{l}(t) = \int_{-\infty}^{t} v(\alpha)\mathrm{d}\alpha$ is the flux linkage, then one can write* (using one form of KVL)

$$\sum_{\text{loop}} \lambda_{\text{rises}}^{(t)} = 0 \tag{19.14}$$

In the theory of electromagnetics, one finds that this equation does not hold exactly; in fact, the right-hand side is equal to the negative of the derivative of the magnetic flux contained within the loop (this is the Faraday–Lenz law). If, however, the time variation of all signals in the circuit are slow, then the right-hand side is approximately zero and KVL can be assumed to hold. A similar result holds also for KCL. For extremely short instants of time, a conductor can support an unbalanced charge. One finds, however, that the ''relaxation'' time of such unbalanced charge is quite short in comparison with the time variations of interest in the circuits considered in this text.

Finally, we tie up a loose end left hanging at the beginning of this section. We consider a circuit to be, not just any collection of elements that are interconnected, but a collection having the property that each element is contained in at least one loop. Thus, the circuit in Figure 19.15 is not a circuit; instead, it must be treated as a subcircuit, that is, as part of a larger circuit in which it is to be imbedded.

The remainder of this section develops the application of the axioms presented here to the analysis of circuits. The reader is referred to Refs. [2–4] for a more detailed treatment.

19.1.1 Nodal Analysis

Nodal analysis of electric circuits, although using all four of the fundamental axioms presented in the introduction, concentrates upon KCL explicitly. KVL is also satisfied automatically in view of the way the basic equations are formulated. This effective method uses the concept of a node voltage. Figure 19.16 illustrates the concept. Observe a voltmeter, with its black probe attached to a single node, called the reference node, which remains fixed during the course of the investigation. In the case shown node 1 is the reference node. The red probe is shown being touched to node 4; therefore, we call the resulting voltage v_4. The subscript denotes the node and the result is always assumed to have its positive reference

* One might anticipate a constant on the right side of Equation 19.14; however, a closer investigation reveals that it is more realistic and pragmatic to assume that all signals are one-sided and that all elements are causal. This implies that the constant is zero. Two-sided signals only arise legitimately within the context of steady-state behavior of stable circuits and systems.

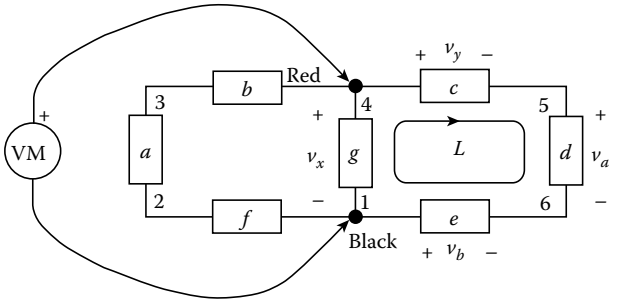


FIGURE 19.16 Node voltages.

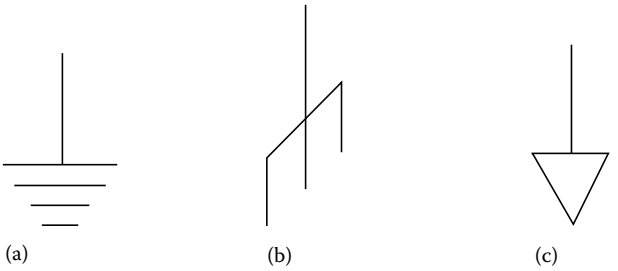


FIGURE 19.17 Reference node symbols.

on the given node. In the present instance v_4 is identical to the element voltage because element g (across which v_g is defined) is connected between node 4 and the reference node. Note that the voltage of such an element is always either the node voltage or its negative, depending upon the reference polarities of its associated element voltage. If we were to touch the red probe to node 5, however, no element voltage would have this relationship to the resulting node voltage v_5 because no elements are connected directly between nodes 5 and 1.

The concept of reference node is used so often that a special symbol is used for it (see Figure 19.17a); alternate symbols often seen on circuit diagrams are shown in the figure as well. Often one hears the terms ''ground'' or ''ground reference'' used. This is commonly accepted argot for the reference node; however, one should be aware that a safety issue is involved in the process of grounding a circuit or appliance. In such cases, sometimes one symbol specifically means ''earth ground'' and one or more other symbols are used for such things as ''signal ground'' or ''floating ground,'' although the last term is something of an oxymoron. Here, we use the terms ''reference'' or ''reference node.'' The reference symbol is quite often used to simplify the drawing of a circuit. The circuit in Figure 19.16, for instance, can be redrawn as in Figure 19.18; circuit operation will be unaffected. Note that all four of the reference symbols refer to a single node, node 1, although they are shown separated from one another. In fact, the circuit is not changed electrically if one bends the elements around and thereby separates the ground symbols even more, as we have done in Figure 19.19. Notice that the loop L shown in the original figure, Figure 19.16, remains a loop, as in Figures 19.18 and 19.19. Redrawing a circuit using ground reference symbols does not alter the circuit topology, the circuit graph.

Suppose the red probe were moved to node 5. As described previously, no element is directly connected between nodes 5 and 1; hence, node voltage v_5 is not an element voltage. However, the element voltages and the node voltages are directly related in a one-to-one fashion. To see how, look at Figure 19.20. This figure shows a ''floating element,'' e, which is connected between two nodes, k and j,

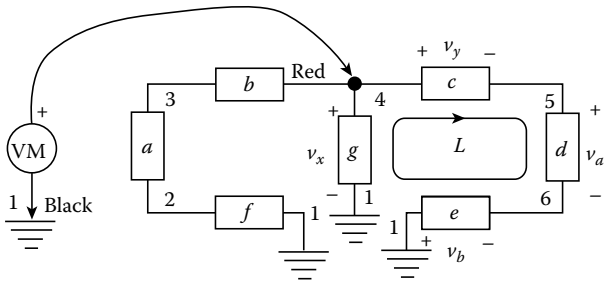


FIGURE 19.18 Alternate drawing.

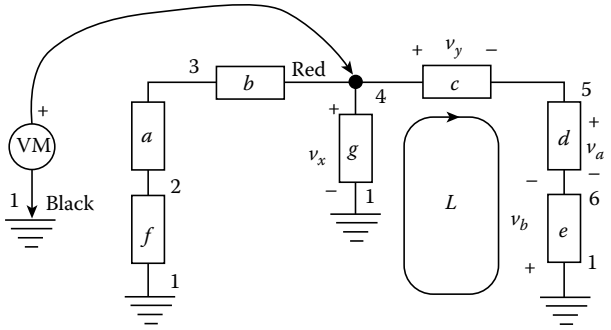


FIGURE 19.19 Equivalent drawing.

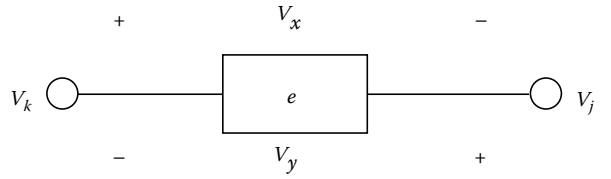


FIGURE 19.20 Floating element and its voltage.

neither of which is the reference node. It is vital here to remember that all node voltages are assumed to have their positive reference polarities on the nodes themselves and their negative reference on the reference node. Now, we can define the element voltage in either of two possible ways, as illustrated in the figure. KVL (the simplest form perhaps being the path form) shows at once that

$$v_x = v_k - v_j \tag{19.15}$$

and

$$v_y = v_j - v_k \tag{19.16}$$

An easy mnemonic for this result is the following:

$$v_{\text{floating element}} = v_+ - v_- \tag{19.17}$$

where v_+ is the node voltage of the node to which the element lead associated with the positive reference for the element voltage is connected, and v_- is the node voltage of the node to which the lead carrying the negative reference for the element voltage is connected. We refer to an element that is not floating, by the way, as being "grounded."

It is easy to see that a circuit having N nodes has $N-1$ node voltages; further, if one uses Equation 19.17, any element voltage can be expressed in terms of these $N-1$ node voltages. Then, for any invertible element,* one can determine the element current. The nodal analysis method uses this fact and considers the node voltages to be the unknown variables.

To illustrate the method, first consider a resistive circuit that contains only resistors and/or independent sources. Furthermore, we initially limit our investigation to circuits whose only independent sources (if any) are current sources. Such a circuit is depicted in Figure 19.21. Because nodal analysis relies upon the node voltages as unknowns, one must first select an arbitrary node for the reference. For circuits that contain voltage sources, one can achieve some simplification for hand analysis by choosing the reference wisely; however, if current sources are the only type of independent source present, one can choose it arbitrarily. As it happens, physical intuition is almost always better served if one chooses the bottom node. Such is done here and the circuit is redrawn using reference symbols as in Figure 19.22. Here, we have arbitrarily assigned node voltages to the $N-1=2$ nonreference nodes. In performing these two steps, we have "prepared the circuit for nodal analysis." The next step involves writing one KCL equation at each of the nonreference nodes. As it happens, the resulting equations are nice and compact if the form

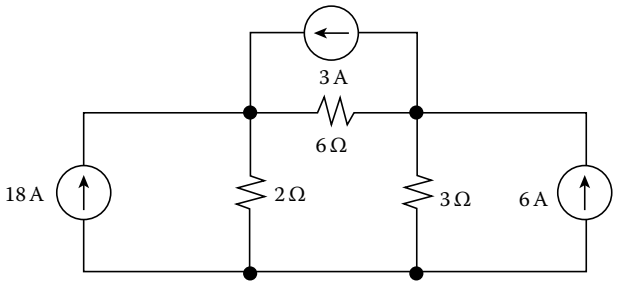


FIGURE 19.21 Example circuit.

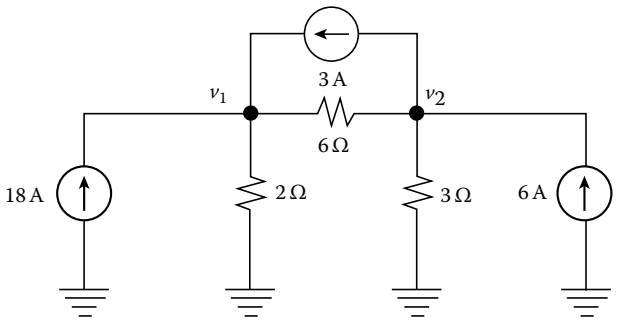


FIGURE 19.22 Example circuit prepared for nodal analysis.

* For instance, resistors, capacitors, and inductors are invertible in the sense that one can determine their element currents if their element voltages are known.

$$\sum_{\text{node}} i_{\text{out}}(R's) = \sum_{\text{node}} i_{\text{in}}(I - \text{Sources}) \tag{19.18}$$

is used. Here, we mean that the currents leaving a node through the resistors must sum up to be equal to the current being supplied to that node from current sources. Because these two types of elements are exhaustive for the circuits we are considering, this form is exactly equivalent to the other forms presented in the introduction. Furthermore, for a current leaving a node through a resistor, the floating element KVL result in Equation 19.17 is used along with Ohm's law:

$$\sum_{j=1}^{N-1} \frac{v_k - v_j}{R_{kj}} = \sum_{q=1}^{M_k} i_{\text{sq}}(\text{node } k). \tag{19.19}$$

In this equation for node k, R_{kj} is the resistance between nodes k and j (or the equivalent resistance of the parallel combination if more than one are found), i_{sq} is the value of the qth current source connected to node k (positive if its reference is toward node k), and M_k is the number of such sources. Clearly, one can simply omit the $j = k$ term on the left side because $v_k - v_k = 0$.

The nodal equations for our example circuit are

$$\frac{v_1}{2} + \frac{v_1 - v_2}{6} = 18 + 3 \tag{19.20}$$

and

$$\frac{v_2 - v_1}{6} + \frac{v_2}{3} = 6 - 3 \tag{19.21}$$

Notice, by the way, that we are using units of A, Ω, and V. It is a simple matter to show that KVL, KCL, and Ohm's law remain invariant if we use the consistent units of mA, kΩ, and V. The latter is often a more practical system of units for filter design work. In the present case the matrix form of these equations is

$$\begin{bmatrix} \frac{2}{3} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 21 \\ 3 \end{bmatrix} \tag{19.22}$$

It can be verified easily that the solution is $v_1 = 36$ V and $v_2 = 18$ V. To see that one can compute the value of any desired variable from these two voltages, consider the problem of determining the current i_6 (let us call it) through the horizontal 6 Ω resistor from right to left. One can simply use the equation

$$i_6 = \frac{v_2 - v_1}{6} = \frac{18 - 36}{6} = -3 \text{ A} \tag{19.23}$$

The previous procedure works for essentially all circuits encountered in practice. If the coefficient matrix on the left in Equation 19.22 (which will always be symmetric for circuits of the type we are considering) is nonsingular, a solution is always possible. It is surprisingly difficult, however, to determine conditions on the circuit under which solutions do not exist, although this is discussed at greater length in a later section.

Suppose, now, that our circuit to be solved contains one or more independent voltage sources in addition to resistors and/or current sources. This constrains the node voltages because a given voltage source value must be equal to the difference between two node voltages if it is floating and to a node

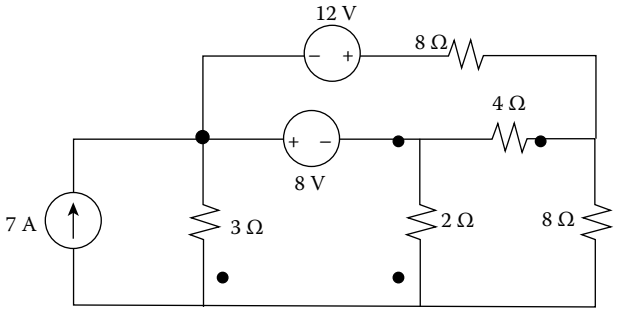


FIGURE 19.23 Example circuit.

voltage or its negative if it is grounded. One might expect that this complicates matters, but fortunately the converse is true.

To explore this more fully, examine the example circuit in Figure 19.23. The algorithm just presented will not work as is because it relies upon balancing the current between resistors and current sources. Thus, it seems that we must account in some fashion for the currents in the voltage sources. In fact, we do not, as the following analysis shows. The key step in our reasoning is this: the analysis procedure should not depend upon the values of the independent circuit variables, that is, on the values of the currents in the current sources and voltages across the voltage sources. This is almost inherent in the definition of an independent source, for it can be adjusted to any value whatsoever. What we are assuming in addition to this is simply that we would not write one given set of equations for a specific set of source values, then change to another set of equations when these values are altered. Thus, let us test the circuit by temporarily deactivating all the independent sources (i.e., by making their values zero). Recalling that a deactivated voltage source is equivalent to a short circuit and a deactivated current source to an open circuit, we have the resulting configuration of Figure 19.24. The resulting nodes are shaded for convenience. Note carefully, however, that the nodes in the circuit under test are not the same as those in the original circuit, although they are related. Notice that, for the circuit under test, all the resistor voltages would be determined by the node voltages as expected; however, the number of nodes has been reduced by one for each voltage source. Hence, we suspect that the required number of KCL equations N_{ne} (and the number of independent node voltages) is

$$N_{ne} = N - 1 - N_v \tag{19.24}$$

where N_v is the number of voltage sources. In the example circuit one can easily compute this required number to be $5 - 1 - 2 = 2$. This is compatible with the fact that clearly three nodes ($3 - 1 = 2$ nonreference nodes) are clearly found in Figure 19.24.

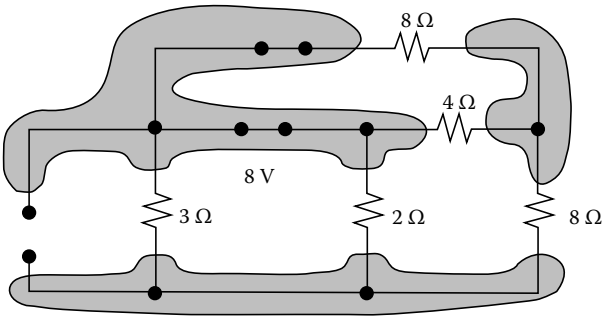


FIGURE 19.24 Example circuit deactivated.

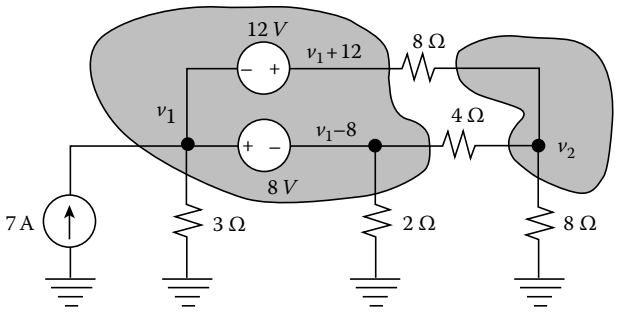


FIGURE 19.25 Example circuit prepared for nodal analysis.

It should also be rather clear that there is only one independent voltage within each of the shaded regions shown in Figure 19.24. We can use KVL to express any other in terms of that one. For example, in Figure 19.25 we have redrawn our example circuit with the bottom node arbitrarily chosen as the reference. We have also arbitrarily chosen a node voltage within the top left surface as the unknown v_1. Note how we have used KVL (the path form, again, is perhaps the most effective) to determine the node voltages of all the other nodes within the top left surface. Any set of connected conductors, leads, and voltage sources to which only one independent voltage can be assigned is called a generalized node. If that generalized node does not include the reference node, it is termed a supernode. The node within the shaded surface at the top left in Figure 19.25, however, has no voltage sources; hence, it is called an essential node.

As pointed out earlier, the equations that one writes should not depend upon the values of the independent sources. If one were to reduce all the independent sources to zero, each generalized node would reduce to a single node; hence, only one equation should be written for each supernode. One equation should be written for essential node also; it is unaffected by deactivation of the independent sources. Observe that deactivation of the current sources does not reduce the number of nodes in a circuit.

Writing one KCL equation for the supernode and one for the essential node in Figure 19.25 results in

$$\frac{v_1}{3}+\frac{v_1-8}{2}+\frac{v_1-8-v_2}{4}+\frac{v_1+12-v_2}{8}=7 \tag{19.25}$$

and

$$\frac{v_2}{8}+\frac{v_2-(v_1-8)}{4}+\frac{v_2-(v_1+12)}{8}=0 \tag{19.26}$$

In matrix form, one has

$$\begin{bmatrix} \frac{29}{24} & -\frac{3}{8} \\ -\frac{3}{8} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}=\begin{bmatrix} \frac{23}{2} \\ -\frac{1}{2} \end{bmatrix} \tag{19.27}$$

The solution is $v_1=12$ V and $v_2=8$ V. Notice once again that the coefficient matrix on the left-hand side is symmetric. This actually follows from our earlier observation about this property for circuits containing only current sources and resistors because the voltage sources only introduce knowns into the nodal equations, thus modifying the right-hand side of Equation 19.27.

The general form for nodal equations in any circuit containing only independent sources and resistors, based upon our foregoing development, is

$$A\bar{v}_n=F_v\bar{v}_s+F_I\bar{i}_s \tag{19.28}$$

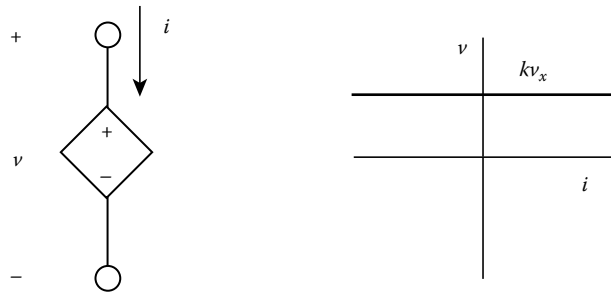


FIGURE 19.26 Dependent source.

where

A is a symmetric square matrix of constant coefficients
F_v and F_I are rectangular matrices of constants
$\bar{v}_n$ is the column matrix of independent mode voltages

The vectors $\bar{v}_s$ and $\bar{i}_s$ are column matrices of independent source values. Clearly, if A is a nonsingular matrix, Equation 19.28 can be solved for the node voltages. Then, using KVL and/or Ohm's law, one can solve for any element current or voltage desired. Equally clearly, if a solution exists, it is a multilinear function of the independent source values.*

Now suppose that the circuit under consideration contains one or more dependent sources. Recall that the two-terminal characteristics of such elements are indistinguishable from those of the corresponding independent sources except for the fact that their value depends upon some other circuit variable. For instance, in Figure 19.26 a voltage-controlled voltage source (VCVS) is shown. Its v–i characteristic is identical to that of an independent source except for the fact that its voltage (the controlled variable) is a constant multiple† of another circuit variable (the controlling variable), in this case another voltage. This fact will be relied upon to develop a modification to the nodal analysis procedure.

We will adopt the following attitude: We will imagine that the dependent relationship, kv_x in Figure 19.26, is a label pasted to the surface of the source in much the same way that a battery is labeled with its voltage. We will imagine ourselves to take a small piece of opaque masking tape and apply it over this label; we will call this process taping the dependent source. This means that we are—temporarily—treating it as an independent source. The usual nodal analysis procedure is then followed, which results in Equation 19.28. Then, we imagine ourselves to remove the tape from the dependent source(s) and note that the relationship is linear, with the controlling variables as the independent ones and the controlled variables the dependent ones. We next express the controlling variables—and thereby the controlled ones as well—in terms of the node voltages using KVL, KCL, and Ohm's law. The resulting relationships have the forms

$$\bar{v}_c = B'\bar{v}_n + C'\bar{y}_{si} + D'\bar{i}_{si} \tag{19.29}$$

and

$$\bar{i}_c = B''\bar{v}_n + C''\bar{v}_{si} + D''\bar{i}_{si} \tag{19.30}$$

* That is, it is a linear function of the vector consisting of all of the independent source values.
† Thus, one should actually refer to such a device as a linear dependent source.

Here, the subscript i refers to the fact that the corresponding sources are the independent ones. Noting that $\bar{v}_c$ and $\bar{i}_c$ appear on the right-hand side of Equation 19.28 because they are source values, one can use the last two results to express the vectors of all source voltages and all source currents in that equation in the form

$$\bar{v}_c = B'''\bar{v}_{si} + C'''\bar{i}_{si} + D'''\bar{v}_n \tag{19.31}$$

and

$$\bar{i}_c = B''''\bar{v}_{si} + C''''\bar{i}_{si} + D''''\bar{v}_n \tag{19.32}$$

Finally, using the last two equations in Equation 19.28, one has

$$A\bar{v}_n = F'_v\bar{v}_s + F'_I\bar{i}_s + B\bar{v}_n \tag{19.33}$$

Now,

$$(A - B)\bar{v}_n = F'_v\bar{v}_s + F'_I\bar{i}_s \tag{19.34}$$

This equation can be solved for the node voltages, provided that $A - B$ is nonsingular. This is even more problematic than for the case without dependent sources because the matrix B is a function of the gain coefficients of the dependent sources; for some set of such values the solution might exist and for others it might not. In any event if $A - B$ is nonsingular, one obtains once more a response that is linear with respect to the vector of independent source values.

Figure 19.27 shows a rather complex example circuit with dependent sources. As pointed out earlier, there are often reasons for preferring one reference node to another. Here, notice that if we choose one of the nodes to which a voltage source is attached it is not necessary to write a nodal equation for the nonreference node because, when the circuit is tested by deactivation of all the sources, the node disappears into the ground reference; thus, it is part of a generalized node including the reference called a nonessential node. For this circuit, choose the node at the bottom of the 2 V independent source. The resulting circuit, prepared for nodal analysis, is shown in Figure 19.28. Surfaces have been drawn around both generalized nodes and the one essential node and they have been shaded for emphasis. Note that we have chosen one node voltage within the one supernode arbitrarily and have expressed the other node

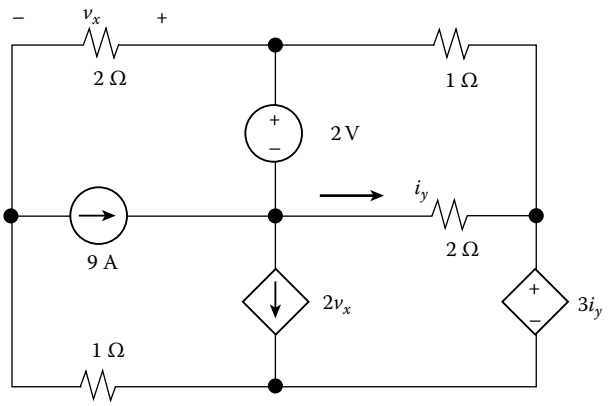


FIGURE 19.27 Example circuit.

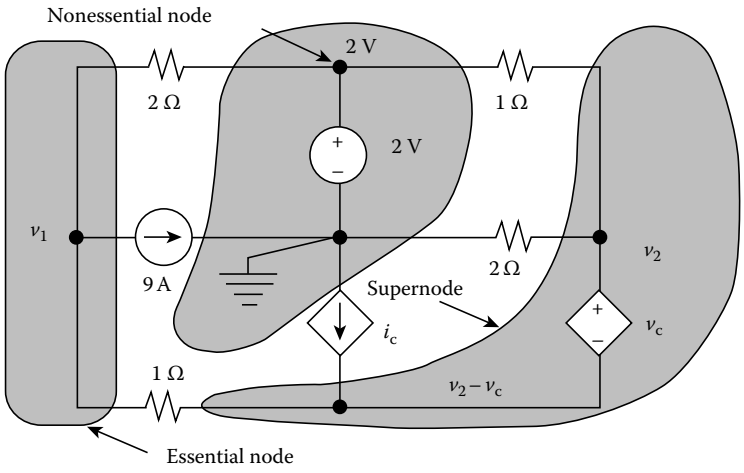


FIGURE 19.28 Example circuit prepared for nodal analysis.

voltage within that supernode in terms of the first and the voltage source value; furthermore, we have taped both dependent sources and written in the known value at the one nonessential node.

The nodal equations for the supernode and for the essential node are

$$\frac{v_1 - 2}{2} + \frac{v_1 - (v_2 - v_c)}{1} = -9 \quad \text{(essential node)} \tag{19.35}$$

and

$$\frac{v_2}{2} + \frac{v_2 - 2}{1} + \frac{v_2 - v_c - v_1}{1} = i_c \quad \text{(supernode)} \tag{19.36}$$

Now, the two dependent sources are untaped and their values expressed in terms of the unknown node voltages and known values using KVL, KCL, and Ohm's law. This results in (referring to the original circuit for the definitions)

$$v_c = -\frac{3}{2} v_2 \tag{19.37}$$

and

$$i_c = 4 - 2v_2 \tag{19.38}$$

Solving these four equations simultaneously gives $v_1 = -2$ V and $v_2 = 2$ V.

If the circuit under consideration contains op-amps, one can first replace each op amp by a VCVS, using the above procedure, and then allow the voltage gain to go to infinity. This is a bit unwieldy, so one often models the op-amp in a different way as a circuit element called a nullor. This is explored in more detail elsewhere in the book and is not discussed here.

Thus far, this chapter has considered only nondynamic circuits whose independent sources were all constants (DC). If these independent sources are assumed to possess time-varying waveforms, no essential modification ensues. The only difference is that each node voltage, and hence each circuit variable, becomes a time-varying function. If the circuit considered contains capacitors and/or inductors,

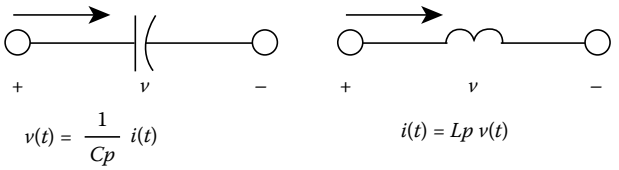


FIGURE 19.29 Dynamic element relationships.

however, the nodal equations are no longer algebraic; they become differential equations. The method developed above remains applicable, however. We will now show why.

Capacitors and inductors have the *v–i* relationships given in Figure 19.29. The symbols p and $1/p$ are referred to as operators, differential operators, or Heaviside operators. The last term is in honor of Oliver Heaviside, who first used them in circuit analysis. They are defined by

$$p = \frac{d}{\mathrm{d}t} \tag{19.39}$$

$$\frac{1}{p} = \int_{-\infty}^{t} (\,)\mathrm{d}a \tag{19.40}$$

The notation suggests that they are inverses of each other, and this is true; however, one must suitably restrict the signal space in order for this to hold. The most realistic assumption is that the signal space consists of all piecewise continuous functions whose derivatives of all orders exist except on a countable set of points that does not have any finite points of accumulation—plus all generalized derivatives of such functions. In fact, Laurent Schwartz, on the first page of the preface of his important work on the theory of distributions, acknowledges that this work was motivated by that of Heaviside. Thus, we will simply assume that all derivatives of all orders of any waveform under consideration exists in a generalized function sense. Higher order differentiation and integration operators are defined in power notation, as expected:

$$p^n = p \cdot p \cdots p = \frac{\mathrm{d}^n}{\mathrm{d}t^n} \tag{19.41}$$

and

$$\frac{1}{p^n} = \frac{1}{p} \cdot \frac{1}{p} \cdots \frac{1}{p} = \int_{-\infty}^{t} \int_{-\infty}^{b} \cdots \int_{-\infty}^{g} (\,)\mathrm{d}a \tag{19.42}$$

Another fact of the preceding issue often escapes notice, however. Look at any arbitrary function in the above-mentioned signal set, compute its running integral, and differentiate it. This action results in

$$p\left[\frac{1}{p}x(t)\right] = \frac{d}{\mathrm{d}t} \int_{-\infty}^{t} x(\alpha)\mathrm{d}\alpha = x(t) \tag{19.43}$$

In fact, it is precisely this property that characterizes the set of all generalized functions. It is closed under differentiation. However, suppose the computation is done in the reverse order:

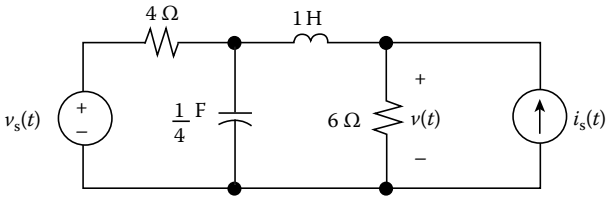


FIGURE 19.30 Example circuit.

$$\frac{1}{p}[px(t)] = \int_{-\infty}^{t} x'(a)\mathrm{d}a = x(t) = x(t) - x(\infty) \tag{19.44}$$

We have assumed here that the fundamental theorem of calculus holds. This is permissible within the framework of generalized functions, provided that the waveform $x(t)$ has a value in the conventional sense at time t. The problem with the previous result is that one does not regain $x(t)$. If it is assumed, however, that $x(t)$ is one sided, that is, $x(t)$ is identically zero for sufficiently large negative values of (t), $x(t)$ will be regained and p and $1/p$ will be inverses of one another. Thus, in the following, we will assume that all independent waveforms are one-sided. We will, in fact, interpret this as meaning that they are all zero for $t < 0$. We will also assume that all circuit elements possess one property in addition to their defining v–i relationship, namely, that they are causal. Thus, all waveforms in any circuit under consideration will be zero for $t \leq 0$ and the previous two operators are inverses of one another. The only physically reasonable situation in which two-sided waveforms can occur is that of a stable circuit operating in the steady state, which we recognize as being an approximate mode of behavior derived from the previous considerations in the limit as time becomes large.

Referring to Figure 19.29 once more, we define

$$Z_c(p) = \frac{1}{Cp} \tag{19.45}$$

$$Z_L(p) = Lp \tag{19.46}$$

to be the impedance operators (or operator impedances) for the capacitor and the inductor, respectively. With our one-sidedness causality assumptions, we can manipulate these qualities just as we would manipulate algebraic functions of a real or complex variable.

The analysis of a dynamic circuit is illustrated by Figure 19.30. The circuit is shown prepared for nodal analysis, with the reference node at the bottom of the circuit and the dynamic elements expressed in terms of their impedance operators, in Figure 19.31. Note that if the circuit were to contain dependent sources, we would have taped them at this step. The nodal equations at the two essential nodes are

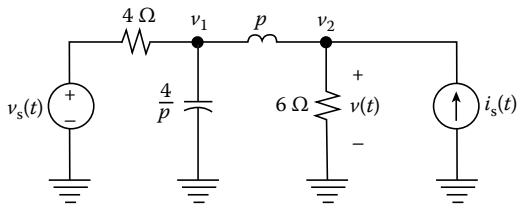


FIGURE 19.31 Example circuit prepared for nodal analysis.

$$\frac{v_1 - v_2}{4} + \frac{v_1}{4/p} + \frac{v_1 - v_2}{p} = 0 \tag{19.47}$$

and

$$\frac{v_2}{6} + \frac{v_2 - v_1}{p} = i_s \tag{19.48}$$

In matrix form, merely rationalizing and collecting terms,

$$\begin{bmatrix} \frac{1}{4} + \frac{p}{4} + \frac{1}{p} & -\frac{1}{p} \\ -\frac{1}{p} & \frac{1}{6} + \frac{1}{p} \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{4} v_s(t) \\ i_s(t) \end{bmatrix} \tag{19.49}$$

Notice that the coefficient matrix is once again symmetric because no dependent sources exist. Multiplying the first row of each side by $4p$ and the second by $6p$, thus clearing fractions, one obtains

$$\begin{bmatrix} p^2 + p + 4 & -4 \\ -6 & p + 6 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \begin{bmatrix} p v_s(t) \\ 6 p i_s(t) \end{bmatrix} \tag{19.50}$$

Now, multiply both sides by the inverse of the 2×2 coefficient matrix to get

$$\begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} = \frac{1}{p(p^2 + 7p + 10)} \begin{bmatrix} p + 6 & 4 \\ 6 & p^2 + p + 4 \end{bmatrix} \begin{bmatrix} p v_s(t) \\ 6 p i_s(t) \end{bmatrix} \tag{19.51}$$

Multiplying the two matrices on the right and canceling the common p factor (legitimate under our assumptions), we finally have

$$v(t) = v_2(t) = \frac{6 v_s(t) + 6(p^2 + p + 4) i_s(t)}{p^2 + 7p + 10} \tag{19.52}$$

We can, on the one hand, consider the result of our nodal analysis process to be a differential equation which we obtain by cross-multiplication:

$$\left[p^2 + 7p + 10\right] v(t) = 6 v_s(t) + 6\left(p^2 + p + 4\right) i_s(t) \tag{19.53}$$

In conventional notation, using the distributive properties of the p operators, one has

$$\frac{d^2 v(t)}{dt^2} + 7\frac{dv(t)}{dt} + 10 v(t) = 6 v_s(t) + 6\frac{d^2 i_s(t)}{dt^2} + 6\frac{di_s(t)}{dt} + 24 i_s(t) \tag{19.54}$$

On the other hand, it is possible to interpret Equation 19.52 directly as a solution operator equation. We simply note that the denominator factors, then do a partial fraction expansion to get

$$\begin{aligned} v(t) &= \frac{6}{(p+2)(p+5)} v_s(t) + \frac{6(p^2 + p + 4)}{(p+2)(p+5)} i_s(t) \\ &= \frac{2}{p+2} v_s(t) - \frac{2}{p+2} v_s(t) + 6 i_s(t) + \frac{2}{p+2} i_s(t) - \frac{8}{p+2} i_s(t) \end{aligned} \tag{19.55}$$

Thus, we have expressed the two second-order operators in terms of operators of order one. It is quite easy to show that the first-order operator has the following simple form:

$$\frac{1}{p+a}x(t) = e^{-at}\frac{1}{p}[e^{at}x(t)] \tag{19.56}$$

Using this result, one can quickly show that the impulse and step responses of the first-order operator are

$$h(t) = \frac{1}{p+a}\delta(t) = e^{-at}u(t) \tag{19.57}$$

and

$$s(t) = \frac{1}{p+a}u(t) = [1 - e^{-at}]u(t) \tag{19.58}$$

respectively. Thus, if $i_s(t) = \delta(t)$ and $v_s(t) = u(t)$, one has

$$v(t) = 6\delta(t) + \frac{1}{5}\left[3 + 5e^{-2t} - 38e^{-5t}\right]u(t) \tag{19.59}$$

References [5,6] demonstrate that all the usual algebraic results valid for the Laplace transform are also valid for Heaviside operators.

19.1.2 Mesh Analysis

The central concept in nodal analysis is, of course, the node. The central idea in the method we will discuss here is the loop. Just as KCL formed the primary set of equations for nodal analysis, KVL will serve a similar function here. We will begin with the idea of a mesh. A mesh is a special kind of loop in a planar circuit (one that can be drawn on a plane) a loop that does not contain any other loop inside it. If one reflects on this definition a bit, one will see that it depends upon how the circuit is drawn. Figure 19.32 illustrates the idea of a mesh. The nodes have been numbered and the elements labeled with letters for clarity. The circuit graph in Figure 19.32 abstracts all of the information about how the elements are connected, but does not show them explicitly. The lines represent the elements and the solid dots represent the nodes. If we apply the definition given in the introduction to this section, we can

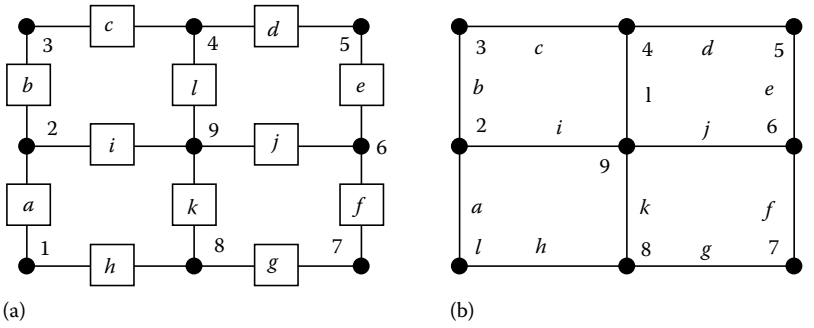


FIGURE 19.32 Circuit (a) and its graph (b).

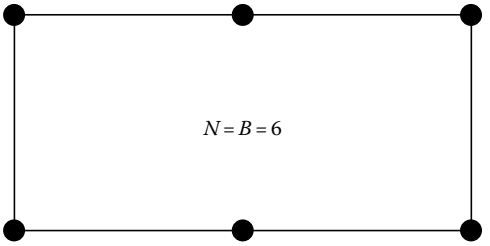


FIGURE 19.33 One-mesh (series) circuit.

quickly verify that {*h*, *a*, *i*, *k*} is a loop. It is a simple loop because each of its elements share only one node with any of the other path elements. It is a mesh because no other loops are inside it.

It is an important fact that the number of meshes in a circuit is given by

$$N_{\text{me}} = B - N + 1 \tag{19.60}$$

where

B is the number of branches (elements)
N is the number of nodes

To see this, just look at the simple one-mesh graph in Figure 19.33. The number of branches is the same as the number of nodes for such a graph (or circuit). Imagine constructing the graph by placing an element on a planar surface, thereby forming two nodes with the one element. $B - N + 1 = 1 - 2 + 1 = 0$ in this case, and no meshes exist. Now, add another element by connecting one of its leads to one of the leads of the first element. Now, $B - N + 1 = 2 - 3 + 1 = 0$. This can be done indefinitely (or until you tire). At this point, connect one lead of the last element to the free lead of the one immediately preceding and the other lead of the last element to a node already placed. N nodes and $N - 1$ branches will have been put down, and exactly one mesh will have been formed. Thus, it is true that $B - N + 1 = N - (N - 1) + 1 = 1$ mesh and the formula is verified. Now connect a new element to one of the old nodes; the result is that one new element and one new node have been added. A glance at the formula verifies that it remains valid. Again, continue indefinitely, and then connect one new element and no new nodes by connecting the free lead of the last element with one of the nodes in the original one-loop circuit. Clearly, the number of added branches exceeds the number of added nodes by one; once again, the formula is verified. Figure 19.34 shows the new circuit. For the graph shown in the figure, $B = 13$ and $N = 12$, so $B - N + 1 = 2$, as expected. Induction generalizes the result, and Equation 19.60 has been proved.

We now define a fictitious set of currents circulating around the meshes of a circuit. Figure 19.35 illustrates this idea with a circuit graph. All mesh currents are assumed to be circulating in a clockwise

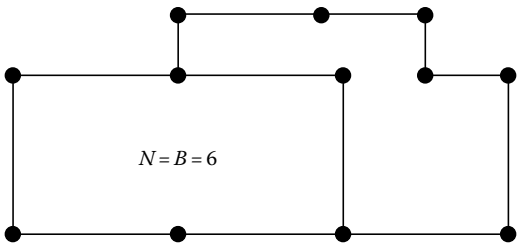


FIGURE 19.34 Two-mesh (series) circuit.

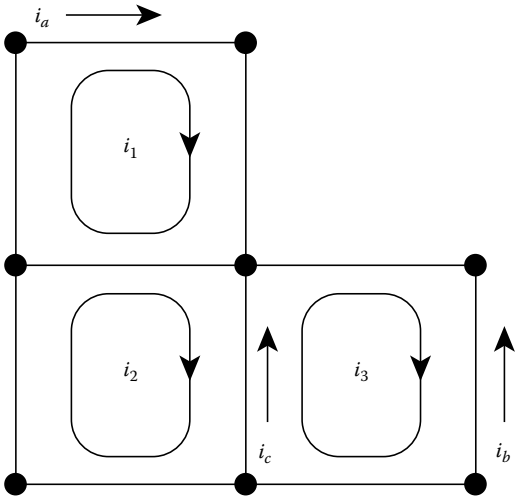


FIGURE 19.35 Mesh currents.

direction, although this is not necessary. We see that i_1 is the only current flowing in the branch in which the element current i_a is defined, therefore, $i_a = i_1$; similarly, i_3 is the only mesh current flowing in the element carrying element current i_b, but the two are defined in opposite directions. Thus, one sees that $i_b = -i_3$. The third element where the current is indicated, however, is seen to carry two mesh currents in opposite directions. Hence, its element current is $i_c = i_3 - i_2$. In general, an element that is shared between two meshes has an element current which is the algebraic sum or difference* of the two adjacent mesh currents.

We used the term ''fictitious'' in our definition of mesh current. In the last example, however, we see that it is possible to make a physical measurement of each mesh current because each flows in an element that is not shared with any other mesh. Thus, one need only insert an ammeter in that element to measure the associated mesh current. Circuits exist, however, in which one or more mesh currents are impossible to measure. Figure 19.36 plots the graph of such a circuit. Each of the meshes is assumed to be carrying a mesh current, although only one has been drawn explicitly, i_k. As readily observed, each of the other mesh currents appears in a nonshared branch. For the mesh where the current is shown, however, it is impossible to find an element or a conductor carrying only that current. For this reason, i_k is merely a fiction, though a useful one.

It is easy to see that mesh currents automatically satisfy KCL because they form a complete loop. Observe the stylized picture in Figure 19.37, which shows three meshes represented for simplicity as small, closed ovals. M_1 lies entirely within the arbitrary closed surface S; thus, its current does not appear in KCL for that surface. M_2 lies entirely outside S, so the same thing is true for its current. Finally, we note that, regardless of the shape of S, M_3 penetrates it an even number of times. Thus, its current will appear in KCL for S an even number of times, and half of its appearances will carry a positive sign and half a negative sign. Thus, we have shown that any mesh current automatically satisfies KCL for any closed surface.

Because KCL is automatically satisfied, we must turn to KVL for the solution of a network in terms of its mesh currents. Figure 19.38 is an example circuit. Just as we assumed at the outset of Section 19.1.1 that any circuit under consideration contained only resistors and current sources, we will assume at first that any circuit under consideration contains only resistors and voltage sources. The one shown in Figure 19.38 has this property.

* Always the difference if all mesh currents are defined in the same direction: clockwise or counterclockwise.

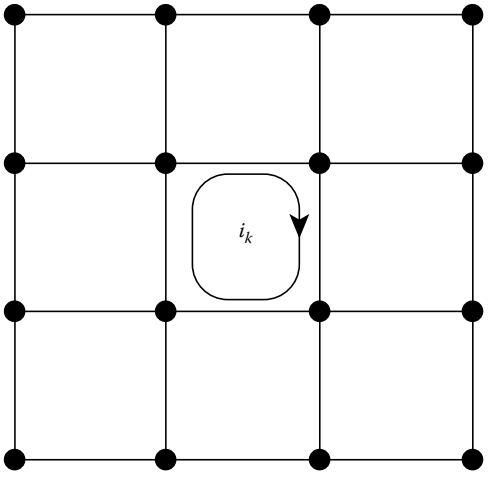


FIGURE 19.36 Fictitious mesh current.

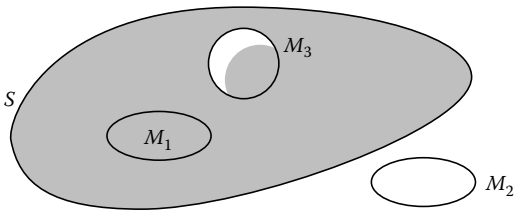


FIGURE 19.37 Illustration of KCL for mesh currents.

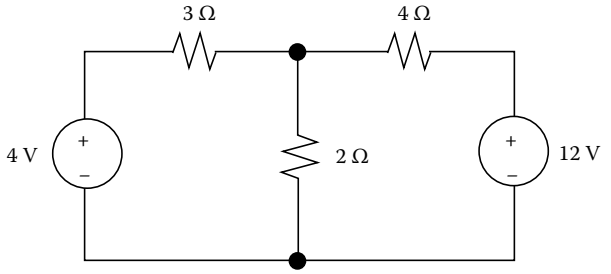


FIGURE 19.38 Example circuit.

The first step is to identify the meshes and assign a mesh current to each. Identification of the meshes is easy, and this is the primary reason for its effectiveness in hand analysis of circuits. The mesh currents can be assigned in arbitrary directions, but for circuits of the sort considered here, it is more convenient to assign them all in the same direction, as in Figure 19.39. Writing one KVL equation for each mesh results in

$$3i_1 + 2(i_1 - i_2) = 4 \tag{19.61}$$

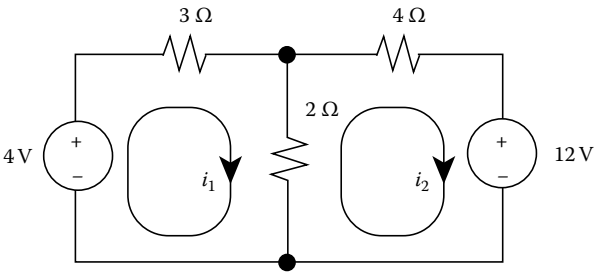


FIGURE 19.39 Example circuit prepared for mesh analysis.

and

$$2(i_2 - i_1) + 4i_2 = -12 \tag{19.62}$$

In matrix form,

$$\begin{bmatrix} 5 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} 4 \\ -12 \end{bmatrix} \tag{19.63}$$

The solution is $i_1 = 0$ A and $i_2 = 2$ A. The same procedure holds for any planar circuit of an arbitrary number of meshes.

Suppose, now, that the circuit being considered has one or more current sources, such as the one in Figure 19.40. The meshes are readily determined; one need only to look for the ''window panes,'' as meshes have been called. The only problem is this: When we write our mesh equations, what values do we use for the voltages across the current sources? These voltages are not known.

Thus, we could ascribe their voltages as unknowns, but this would lead to a hybrid form of analysis in which the unknowns are both element voltages and mesh currents; however, a more straightforward way is available. Consider this question: should the variables we use or the loops around which we decide to write KVL change if we alter the values of any of the independent sources? The answer, of course, is no. Thus, let us test the circuit by deactivating it, that is, by reducing all sources to zero. Recalling that a zero-valued voltage source is a short circuit and a zero-valued current source is an open circuit, we obtain the test circuit in Figure 19.41.

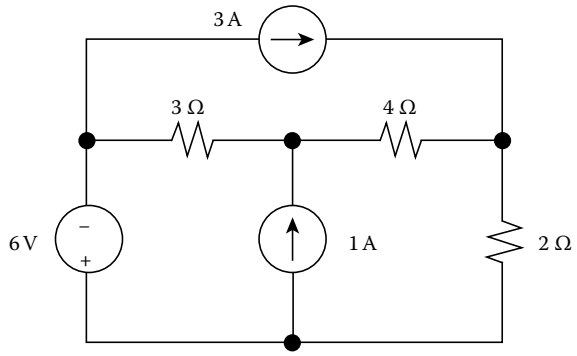


FIGURE 19.40 Example with current sources.

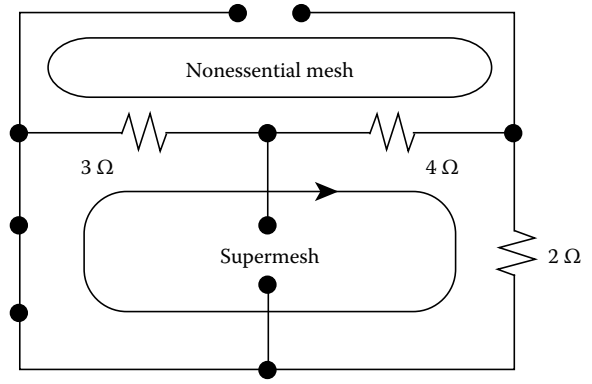


FIGURE 19.41 Deactivated current.

Notice what has happened. The two bottom meshes merge, thus forming one larger mesh in the deactivated circuit. The top mesh disappears (as a mesh or loop). For this reason, we refer to the former as a supermesh and the latter as a nonessential mesh. Observe also that it was the deactivation of the current sources that altered the topology; in fact, deactivation of the voltage sources has no effect on the mesh structure at all. Thus, we see that only one KVL equation is required to solve the deactivated circuit. (Reactivation of the source(s) is necessary, otherwise all voltage and currents will have zero values.) The conclusion relative to our example circuit is this: To solve the original circuit in terms of mesh currents, only one equation (KVL around the supermesh) is necessary.

The original circuit, with its three mesh currents arbitrarily defined, is redrawn in Figure 19.42. Notice that the isolated (nonshared) current source in the top (nonessential) mesh defines the associated mesh current as having the same value as the source itself. On the other hand, the 1 A current source shared by the bottom two meshes introduces a more general constraint: the difference between the two mesh currents must be the same as the source current. This constraint has been used to label the mesh current in the right-hand mesh with a value such that it, minus the left-hand mesh current, is equal to the source current. The nice feature of this approach is that one can clearly see which mesh currents are unknown and which are dependent upon the unknowns. Exactly one independent mesh current is always associated with a supermesh. Recalling our test circuit in Figure 19.41, we see that we need to write only one KVL equation around the supermesh. It is

$$3(i-3)+4(i+1-3)+2(i+1)=-6 \tag{19.64}$$

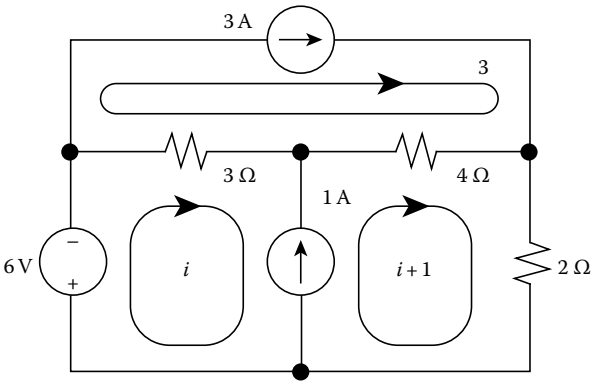


FIGURE 19.42 Assigning the mesh currents.

or

$$9i - 9 - 8 + 2 = -6 \tag{19.65}$$

The solution is $i = 1$ A. From this, one can compute the mesh current on the bottom right to be $i + 1 = 2$ A and the one in the top loop is already known to be 3 A. With these known mesh currents, we can solve for any circuit variable desired.

The development of mesh analysis seems at first glance to be the complete analog of nodal. This is not quite the case, however, because nodal will work for nonplanar circuits, while mesh works only for planar circuits; furthermore, no global reference exists for mesh currents as it does for node voltages.

Analyzing the problem, we observe that each current source, when deactivated, reduces the number of meshes by one. (A given element can be shared only by two meshes). Combining this fact with Equation 19.60, we see that the required number of mesh equations is

$$N_{\text{me}} = B - N + 1 - N_{\text{I}}, \tag{19.66}$$

where

B is the number of branches
N is the number of nodes
N_{I} is the number of current sources

Note that mesh analysis is undertaken for circuits containing dependent sources in exactly the same manner as in nodal analysis, that is, by first taping the dependent sources, writing the mesh equations as above, and then untaping the dependent sources and expressing their controlled variables in terms of the unknown mesh currents. Figure 19.43 shows an example of such a circuit; in fact, it is the same figure investigated with nodal analysis in the preceding section.

The first step is to tape the dependent sources, thus placing them on the same footing as their more independent relatives. Then the circuit is tested for complexity by deactivating it as shown in Figure 19.44. All element labels have been removed merely to avoid obscuring the ideas being discussed. We see one supermesh, one essential mesh, and no nonessential mesh. Therefore, two KVL equations must be written in the original circuit, which is shown with the dependent sources taped in Figure 19.45. Notice that only two unknowns exist, and also that the dependent sources have been taped. For the moment, we have turned them into independent sources (albeit with unknown values).

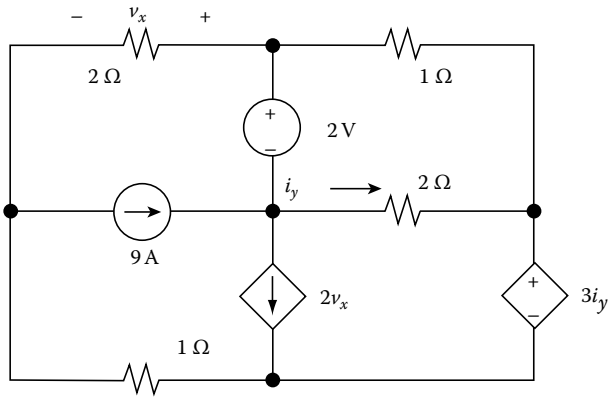


FIGURE 19.43 Example circuit.

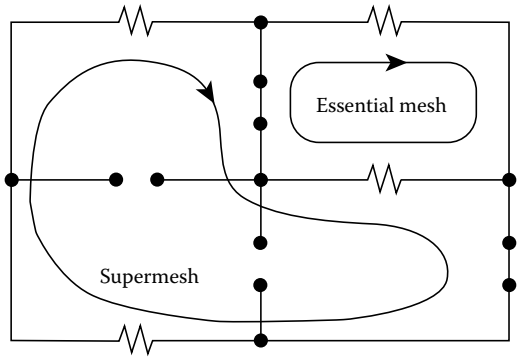


FIGURE 19.44 Testing the example circuit.

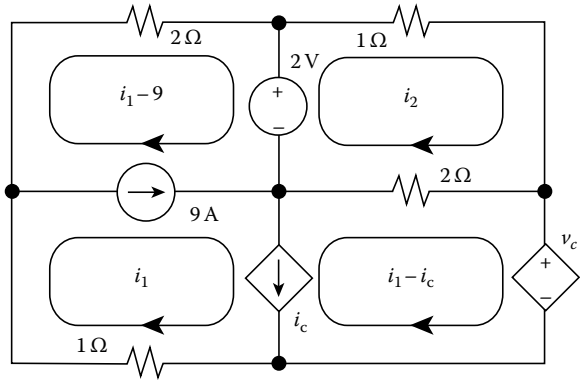


FIGURE 19.45 Example circuit prepared for mesh analysis.

We are now in a position to write KVL equations:

$$2(i_1 - 9) + 2(i_1 - i_c - i_2) + 1i_1 = -2 - v_c \quad \text{(supermesh)} \tag{19.67}$$

and

$$1i_2 + 2(i_2 - i_1 + i_c) = 2 \quad \text{(essential mesh)} \tag{19.68}$$

Observe that i_c and v_c are not known quantities, as would be the case where they are the values of independent sources. Thus, at this point, we must untape the dependent sources and express their values in terms of the mesh currents. We find that

$$i_c = 2v_x = 2x2x(-i_1 + 9) = -4i_1 + 36 \tag{19.69}$$

and

$$v_c = 3i_y = 3(i_1 - i_c - i_2) = 15i_1 - 3i_2 + 36 \tag{19.70}$$

Inserting the last two results in Equations 19.67 and 19.68 results in the matrix equation

$$\begin{bmatrix} 28 & -5 \\ -10 & 3 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} 196 \\ -70 \end{bmatrix} \tag{19.71}$$

The coefficient matrix is no longer symmetric now that dependent sources have been introduced. (This is also the case with nodal analysis. The example treating this same circuit is found in Section 19.1.3 and should be checked to verify this point.) However, the solution is found, as usual, to be $i_1 = 7$ A and $i_2 = 0$ A.

A careful consideration of what we have done up to this point reveals that the mesh equations can be written in the form

$$A\bar{i}_M = B\bar{i}_s + C\bar{v}_s \tag{19.72}$$

In this general formulation, A is a square $n_M \times n_M$ matrix, where M is the number of meshes, and B and C are rectangular matrices whose dimensions depend upon the number of independent voltage and current sources, respectively. The variables $\bar{v}_s$ and $\bar{i}_s$ are the column matrices of independent voltage and current source values, respectively. A is symmetric if the circuit contains only resistors and independent sources. As was the case for nodal analysis, the elucidation of conditions under which the A matrix is nonsingular is difficult. Certainly, it can be singular for circuits with dependent sources; surprisingly, circuits also exist with only resistors and independent sources for which A is singular as well.

Finally, the mesh analysis procedure for circuits with dynamic elements should be clear. The algebraic process closely follows that for nodal analysis. For this reason, that topic is not discussed here.

19.1.3 Fundamental Cutset-Loop Circuit Analysis

As effective as nodal and mesh analysis are in treating circuits by hand, particular circuits exist for which they fail. Consider, for example, the circuit in Figure 19.46. If we were to blindly perform nodal analysis on this circuit, we would perhaps prepare it for analysis as shown in Figure 19.47. We have three nonreference nodes, hence, we have three nodal equations:

$$\frac{v_1}{2} = 2 \tag{19.73}$$

$$\frac{v_2 - v_3}{2} = 2 \tag{19.74}$$

$$\frac{v_3 - v_2}{2} = -2 \tag{19.75}$$

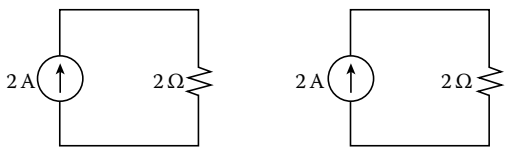


FIGURE 19.46 Example circuit.

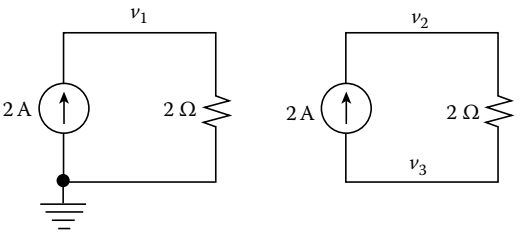


FIGURE 19.47 Example circuit prepared for nodal analysis.

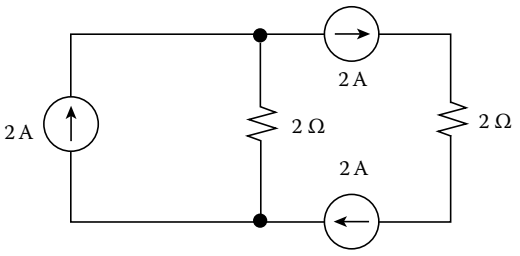


FIGURE 19.48 Another example circuit.

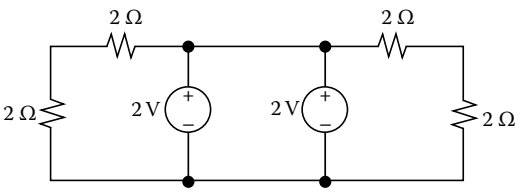


FIGURE 19.49 Another example of a singular circuit.

The third equation is simply the negative of the second; hence, the set of equations is linearly dependent and does not have a unique solution. The reason is quite obvious: the circuit is not connected.* It actually consists of two circuits considered as one. Therefore, one should actually select two reference nodes rather than one. A bit of reasoning along this line indicates that the number of nodal equations should be

$$N_{ne} = N - 1 - P \tag{19.76}$$

where P is the number of separate parts, and hence the number of reference nodes required.

Another way nodal analysis can fail is not quite as obvious. Figure 19.48 illustrates the situation. In this case, we have a cutset of current sources. Therefore, in reality, at least one of the current sources cannot be independent for it must have the same value as the other in the cutset. We will leave the writing of the nodal equations as an exercise for the reader. They are, however, linearly dependent. The problem here clearly becomes evident if one deactivates the circuit, because the resulting test circuit is not connected.

Analogous problems can occur with mesh analysis, as the circuit in Figure 19.49 demonstrates. We find a loop of voltage sources and, when the circuit is deactivated, one mesh disappears. Again, it is left as an exercise for the reader to write the mesh equations and show that they are linearly independent (the coefficients of all currents in the KVL equation for the central mesh are zero).

One might question the practically of such circuits because clearly no one would design such networks to perform a useful function. In the computer automation of circuit analysis, however, dynamic elements are often modeled over a small time increment in terms of independent sources, and singular behavior can result. Furthermore, one would like to be able to include more general elements than R, L, C, and voltage and current sources. For such reasons, a general method that does not fail is desirable. We develop this method next. It is related to the modified nodal analysis technique that is described elsewhere in the book.

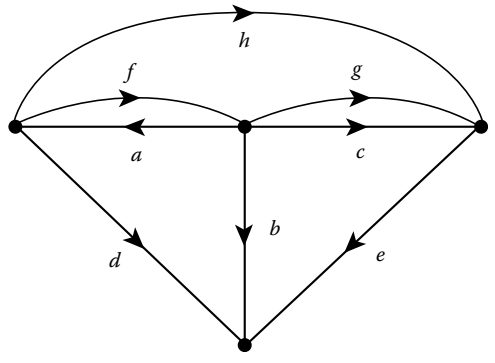


FIGURE 19.50 Example of a circuit graph.

To develop this technique, we will examine the graph of a specific circuit: the one in Figure 19.50. Graph theory is covered elsewhere in the book, but salient points will be reviewed here [7,8]. The graph, of course, is not concerned at all with the v–i characteristics of the elements themselves—only with

* A circuit is connected if at least one path exists between each pair of nodes.

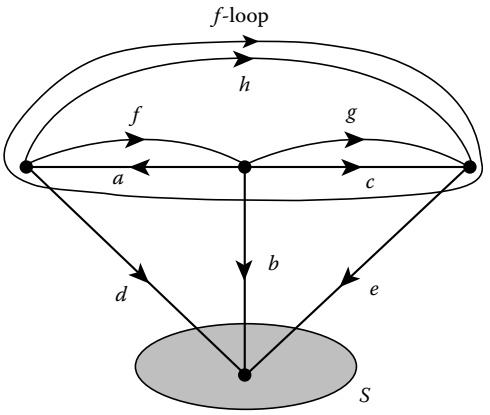


FIGURE 19.51 Fundamental cutsets and loops.

how they are interconnected. The lines (or edges or branches) represent the elements and the dots represent the nodes. The arrows represent the assumed references for voltage and current, the positive voltage at the "upstream" end of the arrow and the current reference in the direction of the arrow. We also recall the definition of a tree: for a connected graph of N nodes, a tree is any subset of edges of the graph that connects all the nodes, but which contains no loops. Such a tree is shown by means of the darkened edges in the figure: a, b, and c. The complement of a tree is called a cotree. Thus, edges d, e, f, g, and h form a cotree in Figure 19.50. If a graph consists of separate parts (that is, it is not connected), then one calls a subset of edges that connects all N nodes, but forms no loops, a forest. The complement of a forest is a coforest. The analysis method presented here is applicable to either connected or nonconnected circuits. However, we will use the terms for a graph that is connected for ease of comprehension; one should go through a parallel development for nonconnected circuits to assure oneself that the generalization holds.

Each edge contained in a tree is called a twig, and each edge contained in a cotree is called a link. (Remember that the set of all nodes in a connected graph is split into exactly two sets of nodes, each of which is individually connected by twigs, when a tree edge is removed). The set of links having one of its nodes in one set and another in the second, together with the associated twig defining the two sets of nodes, is called a fundamental cutset, of f-cutset. If all links associated with a given tree are removed, then the links replaced one at a time, it can be seen that each link defines a loop called a fundamental loop or f-loop. Figure 19.51 is a fundamental cutset and a fundamental loop for the graph shown in Figure 19.50. The closed surface S is placed around one of the two sets of nodes so defined (in this case consisting of a single node) and is penetrated by the edges in the cutset b, d, and e. A natural orientation of the cutset is provided by the direction of the defining twig, in this case edge b. Thus, a positive sign is assigned to b; then, any link in the fundamental cutset with a direction relative to S agrees with that of the twig receives a positive sign, and each with a direction that is opposite receives a negative sign. Similarly, the fundamental loop is given a positive sense by the direction of the defining link, in the case edge h. It is assigned a positive sign; then, each twig in the f-loop is given a positive sign if its direction coincides in the loop with the defining link, and a negative sign if it does not.

The positive and negative signs just defined can be used to write one KCL equation for each f-cutset and one KVL equation for each f-loop, as follows. Consider the example graph with which we are working. The f-cutsets are $\{d, b, e\}$, $\{d, a, f, h\}$, and $\{e, c, g, h\}$. In general, $N-1$ f-cutsets are associated with each tree—exactly the same as the number of twigs in the tree. Using surfaces similar to S in Figure 19.51 for each of the f-cutsets, we have the following set of KCL equations:

$$i_a - i_d - i_f - i_h = 0 \tag{19.77}$$

$$i_b + i_d + i_e = 0 \tag{19.78}$$

$$i_c - i_e + i_g + i_h = 0 \tag{19.79}$$

In matrix form, these equations become

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \\ i_d \\ i_e \\ i_f \\ i_g \\ i_h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{19.80}$$

The coefficient matrix consists of zeroes, and positive and negative ones. It is called the *f*-cutset matrix. Each row corresponds to the KCL equation for one of the *f*-cutsets, and has a zero entry for each edge not in that cutset, $a+1$ for any edge in the cutset with the same orientation as the defining twig, and $a-1$ for each edge in the cutset whose orientation is opposite to the defining twig, and $a-1$ for each edge in the cutset whose orientation is opposite to the defining twig. One often labels the rows and columns,

$$Q = \begin{matrix} a \\ b \\ c \end{matrix} \begin{bmatrix} a & b & c & d & e & f & g & h \\ 1 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \end{bmatrix} \tag{19.81}$$

to emphasize the relation between the matrix and the graph. Thus, the first row corresponds to KCL for the *f*-cutset defined by twig *a*, the second to that defined by twig *b*, and the last to the cutset defined by twig *c*. The columns correspond to each of the edges in the graph, with the twigs occupying the first $N-1$ columns in the same order as that in which they appear in the rows. Notice that a unit matrix of order $N-1 \times N-1$ is located in the leftmost $N-1$ columns. Furthermore, Q has dimensions $(N-1) \times B$, where B is the number of branches (edges). Clearly, Q has maximum rank because of the leading unit matrix. More succinctly, one writes KCL in terms of the *f*-cutset matrix as

$$Q\bar{i} = [U \dot{:} H]\bar{i} = 0 \tag{19.82}$$

where $\bar{i}$ is the column matrix of all the branch currents. Here, the structure of Q appears explicitly with the unit matrix in the first $N-1$ columns and, for our example,

$$H = \begin{bmatrix} -1 & 0 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 \end{bmatrix} \tag{19.83}$$

In general, H will have dimensions $(N-1) \times (b-N+1)$.

Each of the links, all $B-N+1$ of them, have an associated KVL equation. In our example, using the same order for these links and equations that occurs in the KCL equations,

$$\begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \\ v_d \\ v_e \\ v_f \\ v_g \\ v_h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{19.84}$$

We have one row for each link and, therefore, one for each *f*-loop. If a given edge is in the given loop, $a+1$ is in the corresponding column if its direction agrees with that of the defining link, and $a-1$ if it disagrees. Notice that a unit matrix of dimensions $(B-N+1)\times(B-N+1)$ is located in the last $B-N+1$ columns. Even more important, observe that the matrix in the first $N-1$ columns has a familiar form; in fact, it is $-H'$, the negative transpose of the matrix in Equation 19.83.

This is no accident. In fact, the entries in this matrix are strictly due to twigs in the tree. Focus on a given twig and a given link. The twig defines two twig-connected sets of nodes, as mentioned above. If the given link has both its terminal nodes in only one of these sets, the given twig voltage does not appear in the KVL equation for that *f*-loop. If, on the other hand, one of the link nodes is in one of those sets and the other in the alternate set, the given twig voltage will appear in the KVL equation for the given link, with $a+1$ multiplier if the directions of the twig agree relative to the *f*-loop and $a-1$ if they do not. However, a little thought shows that the same result holds for the *f*-cutset equation defined by the twig, except that the signs are reversed. If the link and twig directions agree for the *f*-loop, they disagree for the *f*-cutset, and vice versa. Thus, we can write KVL for the *f*-loops, in general, as

$$B_f\bar{v} = [-H'\vdots U]\bar{v} = 0 \tag{19.85}$$

where

B_f is called the fundamental loop matrix

$\bar{v}$ is the column matrix of all branch voltages

Suppose that we partition the branch voltages and branch currents according to whether they are associated with twigs or links. Thus, we write

$$\bar{v} = \left[\bar{v}_T'\bar{v}_C'\right]' \tag{19.86}$$

and

$$\bar{i} = \left[\bar{i}_T'\bar{i}_C'\right]' \tag{19.87}$$

We use transpose notation to conserve space, and the subscripts T and C represent tree and cotree voltages and currents, respectively. We cannot use Equations 19.82 and 19.85 to write the composite circuit variable vector as

$$\bar{u} = \begin{bmatrix}\bar{v}\\ \bar{i}\end{bmatrix} = \begin{bmatrix} U & 0\\ H' & 0\\ 0 & -H\\ 0 & U\end{bmatrix}\begin{bmatrix}\bar{v}_T\\ \bar{i}_C\end{bmatrix} \tag{19.88}$$

The coefficient matrix is of dimensions $2B\times B$ and has rank B because of the two unit matrices. What we have accomplished is a direct sum decomposition of the $2B$-dimensional vector space consisting of all circuit variables in terms of the $N-1$ dimensional vector space of tree voltages and the $B-N+1$ dimensional vector space of link currents. Furthermore, the tree voltages and link currents form a basis for the vector space of all circuit variables.

We have discussed topology enough for our purposes. We now treat the elements. We are looking for a generalized method of circuit analysis that will succeed, not only for circuits containing *R*, *L*, *C*, and source elements, but ideal transformers, gyrators, nullators, and norators (among others) as well. Thus, we turn to a discussion of elements [9].

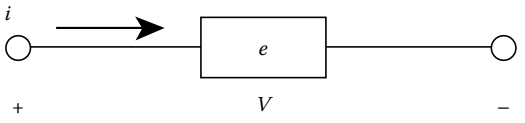


FIGURE 19.52 Two-terminal element.

Consider elements having two terminals only, as shown in Figure 19.52. The most general assumption we can make, assuming that we are ruling out "nonlinear" elements, is that the *v–i* characteristic of each such element be affine, that is, it is defined by an operator equation of the form

$$\begin{bmatrix} a & b \\ 0 & c \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} f(t) \\ g(t) \end{bmatrix} \tag{19.89}$$

where the parameters a, b, and c are operators. It is more classical to assume a scalar form of this equation; that is, with c and $g(t)$ both zero. In a series of papers in the 1960s, however, Carlin and Youla, Belevitch, and Tellegen [10–12] proposed that the *v–i* characteristic be interpreted as a multidimensional relationship. Among other things to come out of the approach was the definition of the nullator and the norator. Now, assuming that this defining characteristic is indeed multidimensional, we see at once that it is not necessary to consider operator matrices of a dimension larger than 2×2. There must be two columns because there are only two scalar terminal variables. If more than two rows were found, the additional equations would be either redundant or inconsistent, depending upon whether row reductions resulted in additional rows of all zeroes or in an inconsistent equation. Finally, the (2, 1) element in the operator matrix clearly can be chosen to be zero as shown, because otherwise it could be reduced to zero with elementary row operations. That is, one could, unless $a = 0$; but here, an exchange of rows produces the desired result shown. Note that any or all of a, b, and c can be the zero operator.

We pause here to remark that a and b can be rather general operators. If they are differential, or Heaviside, operators, (that is, they are real, rational functions of p), a theory of lumped circuits (differential systems) is obtained. On the other hand, they could be rational functions of the delay operator E.* In this case, one would obtain the theory of distributed (transmission line) circuits. Then, if a common delay parameter is used, one obtains a theory of commensurate transmission line circuits; if not, an incommensurate theory results. If the parameters are functions of both p and d, a mixed theory results. We will assume here that a, b, and c are rational functions of p.

Let us suppose that c is the zero operator and that $g(t) = 0$ is the second equality resulting from the stipulation of existence (consistency). This gives the affine scalar relationship

$$av + bi = f(t) \tag{19.90}$$

Special cases are now examined. For instance, if $b = 0$ and $a \neq 0$, one has

$$v(t) = f(t)/a = v_s(t) \tag{19.91}$$

This, of course, is the *v–i* characteristic for an independent voltage source. If, on the other hand, $a = 0$ and $b \neq 0$, one has

$$i(t) = f(t)/b = i_s(t) \tag{19.92}$$

This is an ideal current source. If, in addition $f(t)$ is identically zero, one obtains a short circuit and an open circuit, respectively. These results are shown in Figure 19.53. Now suppose that a and b are both zero. Then, $f(t)$ must be identically zero as well; otherwise, the element does not exist. In this case any arbitrary voltage and current are possible. The resulting element, a "singular one" to be sure, is called a norator. Its symbol is shown in Figure 19.54.

* $Ex(t) = x(t - T)$ for all t and all waveforms $x(t)$.

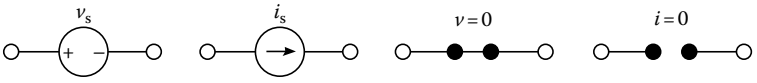


FIGURE 19.53 Conventional two-terminal elements.

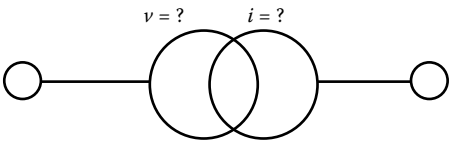


FIGURE 19.54 Norator.

Remaining with the same general case, that is, with $c = 0$ and $g(t) = 0$, we ask what element results if we also assume that neither a nor b are zero, but that $f(t)$ is identically zero. We can solve for either the voltage or the current. In either case, one obtains a passive element, as shown in Figure 19.55. If $-b/a$ is constant, a resistor will result; if $-b/a$ is a constant times the differential operator p, an inductor will result; and if $-b/a$ is reciprocal in p, a capacitor will result. In case the ratio is a more complicated function of p, one would consider the two-terminal object to be a subcircuit, that is, a two-terminal object decomposable into other elements, with $-b/a$ being the driving point impedance operator.

One can, in fact, derive the Thévenin and Norton equivalents from these considerations. Staying with the general case of c and $g(t)$ both zero, but allowing $f(t)$ to be nonzero, we first assume that $a \neq 0$. Then, we obtain

$$v(t) = \frac{f(t)}{a} - \frac{b}{a} i(t) = v_{oc}(t) + Z_{eq}(p) i(t) \tag{19.93}$$

which represents the Thévenin equivalent subcircuit shown in Figure 19.56a. Alternately, if $b \neq 0$ we can write

$$i(t) = \frac{f(t)}{b} - \frac{a}{b} i(t) = i_{sc}(t) + Y_{eq}(p) v(t) \tag{19.94}$$

The latter equation is descriptive of the Norton equivalent shown in Figure 19.56b. The basic assumption is that the two-terminal object has a v–i characteristic (i.e., an affine relationship); if this object contains only elements characterized by affine relationships having a rank property to be given later, one can use the analysis method being presented here to prove that this assumption is true. At this point,

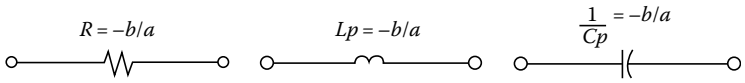


FIGURE 19.55 Passive elements.

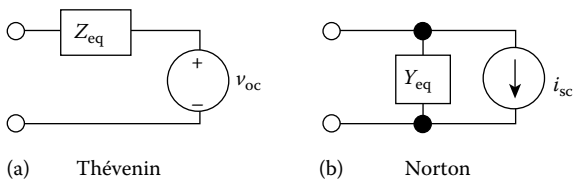


FIGURE 19.56 Two general equivalent subcircuits.

however, we are merely assembling a catalog of elements, so we assume that the two-terminal object is, indeed, a single element (it cannot be decomposed farther).

We have only one other case to consider: that in which c is a nonzero operator. If this is the situation and if, in addition, $a \neq 0$ as well, one can solve Equation 19.90 by inverting the coefficient matrix to obtain

$$\begin{bmatrix} v \\ i \end{bmatrix} = \begin{bmatrix} 1/c & -b/ac \\ 0 & 1/a \end{bmatrix} \begin{bmatrix} f(t) \\ g(t) \end{bmatrix} = \begin{bmatrix} v_s(t) \\ i_s(t) \end{bmatrix} \tag{19.95}$$

Therefore, the voltage and current are independently specified. First, suppose that both $v_s(t)$ and $i_s(t)$ are identically zero. Then, one has $v(t) = 0$ and $i(t) = 0$ for t. The associated element is called a nullator, and has the symbol shown in Figure 19.57a. Finally, if $v_s(t)$ and $i_s(t)$ are nonzero, one can sketch the equivalent subcircuit as in Figure 19.57b.

At this point, we have an exhaustive catalog of two-terminal circuit elements: the independent voltage and current sources, the resistor, the inductor, the capacitor, the norator, and the nullator. We would like to include more complex elements with more than two terminals as well. Figure 19.58a shows a three-terminal element and Figure 19.58b shows a two-port element. For the former, we see at once that only two voltages and two currents can be independently specified because KVL gives the voltage between the left and right terminals in terms of the two shown, while KCL gives the current in the third lead. As for the latter, it is a basic assumption that only the two-port voltages and the two-port currents are required to specify its operation. In fact, one assumes that the currents coming out of the bottom leads are identical to those going into the top leads. We also assume that the v–i characteristic is independent of the voltages between terminals in different ports. Each of the ports will be an edge in the circuit graph that results when such elements are interconnected.

Because four variables are associated with a three-terminal or two-port element, the dimensionality of the resulting vector space is four; thus, we assume that the describing v–i characteristic is

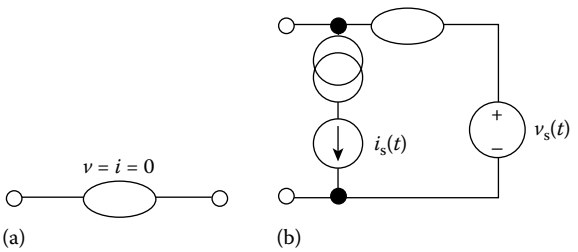


FIGURE 19.57 Nullator element (a) and an equivalent subcircuit (b).

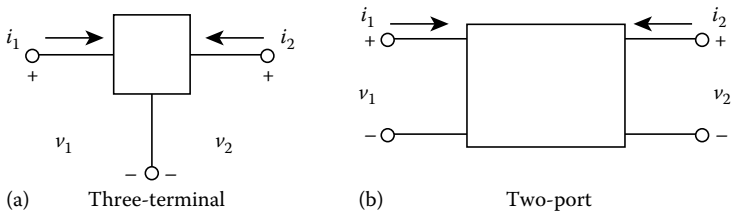


FIGURE 19.58 Three-terminal (a) and two-port elements (b).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} f_1(f) \\ f_2(f) \\ f_3(f) \\ f_4(f) \end{bmatrix} \tag{19.96}$$

We justify this form exactly as for the case of two-terminal elements. We will not exhaustively catalog all of the possible three-terminal/two-port elements for reasons of space; however, note that the usual case is that in which $a_{ij} = 0$ for $i \geq 3$. In this case one must insist that $f_3(t) = f_4(t) = 0$; then one has the 2×2 system of equations

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix} \tag{19.97}$$

If the two forcing functions on the right are not identically zero, a number of different two-port equivalent circuits can be generated—generalized Thévenin and Norton equivalents. If both of these forcing functions are identically zero and if at least one 2×2 submatrix of the coefficient operator matrix on the left side is nonsingular, one can derive a hybrid matrix and a hybrid parameter equivalent circuit. Specialized versions are the impedance parameters, the admittance parameters, and the transmission or chain parameters. Furthermore, one can accommodate controlled sources, transformers, gyrators, and all of the other known two-port elements.

To present just one example, assume that the operator Equation 19.97 has the form

$$\begin{bmatrix} n & -1 & 0 & 0 \\ 0 & 0 & -1 & n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{19.98}$$

The parameter n, assumed to be a real scalar multiplier, is called the turns ratio, and the element is the ideal transformer. The VCVS obeys

$$\begin{bmatrix} \mu & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{19.99}$$

Thus, i_1 is identically zero and $v_2 = \mu v_1$. The quantity μ is the voltage gain.

Similarly, for each element with any number of ports,* we can write

$$A_0 \bar{v} + B_0 \bar{i} = \bar{C}_0 \tag{19.100}$$

where the voltage and current vectors are the terminal variables of the element. We can then represent the element equations for any circuit in the same form by forming A_0 and B_0 as quasidiagonal matrices,

* A two-terminal element is a one-port device.

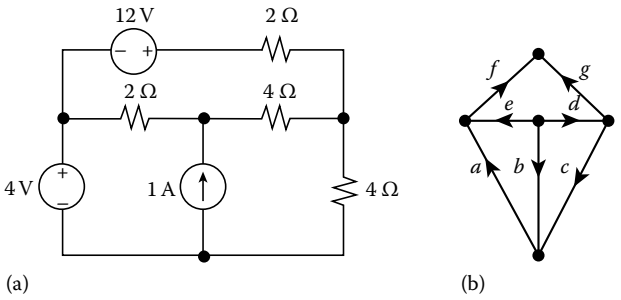


FIGURE 19.59 Example circuit (a) and its graph (b).

each of whose diagonal terms is the corresponding A_0 or B_0 for a given element, and stacking up the individual $\bar{C}_0$ column matrices to form the overall matrix. We then rewrite Equation 19.100 in the form

$$[A_0 B_0]\begin{bmatrix}\bar{v}\\ \bar{i}\end{bmatrix} = \bar{C} \tag{19.101}$$

where the voltage and current vectors are each $B \times 1$ column matrices of the individual element voltages and currents. We make the assumption that the matrix $[A\ B]$, which is of dimension $B \times 2B$ is of maximum rank b. This is the only assumption required for the procedure to be outlined to succeed, as will later be demonstrated.

An example will clarify things. Figure 19.59 is an example circuit. The correspondence between the edge labels and the circuit elements is obvious; that is, for instance, a is the 4 V voltage source and its voltage is -4 V (minus, because of the definition of positive voltage on edge a in the graph). We have shown a tree on the graph. The f-cutset matrix is

$$Q = \begin{matrix}b\\ d\\ e\\ f\end{matrix}\begin{bmatrix} b & d & e & f & a & c & g\\ 1 & 0 & 0 & 0 & -1 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & -1 & -1\\ 0 & 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [U \vdots H] \tag{19.102}$$

Thus,

$$H = \begin{bmatrix} -1 & 1 & 0\\ 0 & -1 & -1\\ 1 & 0 & 1\\ 0 & 0 & 1 \end{bmatrix} \tag{19.103}$$

Although we could construct it from the Q matrix, we can just as easily read off the f-loop matrix from the graph:

$$B = \begin{matrix}a\\ c\\ g\end{matrix}\begin{bmatrix} b & d & e & f & a & c & g\\ 1 & 0 & -1 & 0 & 1 & 0 & 0\\ -1 & 1 & 0 & 0 & 0 & 1 & 0\\ 0 & 1 & -1 & -1 & 0 & 0 & 1 \end{bmatrix} = [-H' \vdots U] \tag{19.104}$$

The element constraint equations are

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} v_b \\ v_d \\ v_e \\ v_f \\ v_a \\ v_c \\ v_g \\ i_b \\ i_d \\ i_e \\ i_f \\ i_a \\ i_c \\ i_g \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ -12 \\ 4 \\ 0 \\ 0 \end{bmatrix} \tag{19.105}$$

In this case, both A_0 and B_0 are diagonal because all the elements are of the two-terminal variety.

The vector of all circuit variables is now expressed in terms of the basis in Equation 19.88, the tree voltages and link currents. We then have

$$[AB]\begin{bmatrix} \bar{v} \\ \bar{i} \end{bmatrix} = [AB]\begin{bmatrix} U & 0 \\ H' & 0 \\ 0 & -H \\ 0 & U \end{bmatrix}\begin{bmatrix} \bar{v}_{\mathrm{T}} \\ \bar{i}_{\mathrm{C}} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ v_d \\ v_e \\ v_f \\ v_a \\ i_c \\ i_g \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ -12 \\ 4 \\ 0 \\ 0 \end{bmatrix} \tag{19.106}$$

After multiplying the two matrices, we have a more compact matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -4 & -4 \\ 0 & 0 & 1 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -4 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} v_b \\ v_d \\ v_e \\ v_f \\ v_a \\ i_c \\ i_g \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ -12 \\ 4 \\ 0 \\ 0 \end{bmatrix} \tag{19.107}$$

We leave it to the reader to show that the solution is given by (in transpose notation):

$$\begin{bmatrix} v_b & v_d & v_e & v_f & i_a & i_c & i_g \end{bmatrix}' = [0 \quad -4 \quad 4 \quad -12 \quad 0 \quad 1 \quad -2] \tag{19.108}$$

In general, one must solve the matrix equation

$$[A \; B] = \begin{bmatrix} U & 0 \\ H' & 0 \\ 0 & -H \\ 0 & U \end{bmatrix} \begin{bmatrix} \bar{v}_T \\ \bar{i}_C \end{bmatrix} = C \tag{19.109}$$

where H is the nonunit submatrix in the f-cutset matrix and C_0 is the $B \times 1$ column matrix of constants (or independent functions of time). Here is the crucial result: $[A_0 \; B_0]$ has dimensions $B \times 2B$; if it has rank B, then the product of it with the next matrix, which also has rank B, will be square (of dimensions $B \times B$) and of rank B by Sylvester's inequality [13]. In this case, the resulting coefficient matrix will be invertible, and a solution is possible.

The procedure just described, although involving more computation than, for example, nodal analysis, is general. If one solves for all element currents and voltages for a general circuit, however, one must anticipate additional complexity. Furthermore, the method outlined is algorithmic and can be computer automated. The element constraint matrices A_0 and B_0 consist of stylized submatrices corresponding to each type of element. These are referred to as ''stamps'' in the modified nodal technique described elsewhere in this volume, and the preceding method is quite similar. The major difference is that is uses node voltage as a basis for the branch voltage space of a circuit instead of the tree voltages described previously.

References

1. P. W. Bridgman, *The Logic of Modern Physics*, New York: Macmillan, 1927.
2. W.-K. Chen, *Linear Networks and Systems*, Monterey, CA: Brooks-Cole, 1983.
3. J. Choma, *Electrical Networks: Theory and Analysis*, New York: Wiley, 1985.
4. L. P. Huelsman, *Basic Circuit Theory*, Englewood Cliffs, NJ: Prentice Hall, 1927.
5. A. M. Davis, A unified theory of lumped circuits and differential system based on Heaviside operators and causality, *IEEE Trans. Circuits Syst.*, 41(11), 712–727, November, 1990.
6. A. M. Davis, *Linear Circuit Analysis*, Boston, MA: PWS Publishing Company, 1998.
7. W.-K. Chen, *Applied Graph Theory: Graphs and Electrical Networks*, New York: North-Holland, 1976.
8. S.-P. Chan, *Introductory Topological Analysis of Electrical Networks*, New York: Holt, Rinehart, & Winston, 1969.
9. A. M. Davis, unpublished notes.

10. H. J. Carlin and D. C. Youla, Network synthesis with negative resistors, *Proc. IEEE*, 49, 907–920, May 1961.
11. V. Belevitch, Four dimensional transformations of 4-pole matrices with applications to the synthesis of reactance 4-poles, *IRE Trans. Circuit Theory*, CT-3, 105–111, June 1956.
12. B. D. H. Tellegen, La Recherche pour une Serie Complete d'Elements de Circuit Ideaux Non-Lineaires, *Rendiconti Del Seminario Mathematico e Fisico di Milano*, 25, 134–144, April 1954.
13. F. R. Gantmacher, *Theory of Matrices*, New York: Chelsea, 1959.

19.2 Network Theorems

Marwan A. Simaan

In Section 19.1, we learned how to determine currents and voltages in a resistive circuit. Methods have been developed, which are based on applying KVL and KCL, to derive a set of mesh or node equations which, when solved, will yield mesh currents or node voltages, respectively. Frequently, and especially if the circuit is complex with many elements, the application of these methods may be considerably simplified if the circuit itself is simplified. For example, we may wish to replace a portion of the circuit consisting of resistors and sources by an equivalent circuit that has fewer elements in order to write fewer mesh or node equations.

In this context, we introduce three important and related theorems known as the superposition, the Thévenin and the Norton theorems. The superposition theorem shows how to solve for a variable in a circuit that has many independent sources, by solving simpler circuits, each excited by only one source. The Thévenin and Norton theorems can be used to replace a portion of a circuit at any two terminals by an equivalent circuit which consists of a voltage source in series with a resistor (i.e., a nonideal voltage source) or a current source in parallel with a resistor (i.e., a nonideal current source). Another important result derived in this section concerns the calculation of power dissipated in a load resistor connected to a circuit. This result is known as the maximum power transfer theorem, and is frequently used in circuit design problems. Finally, a result known as the reciprocity theorem is also discussed.

An important property of linear resistive circuits is the type of relationship that exists between any variable in the circuit and the independent sources. For linear resistive circuits the solution for a voltage or current variable can always be expressed as a linear combination of the independent sources. Let us elaborate on what we mean by this statement through an example.

Consider the circuit in Figure 19.60 and assume that we are interested in the voltage v across R_2. We can solve for v by first applying KCL at node a to get the equation

$$-\frac{v - v_1}{R_1} + \beta i_x + i_1 - \frac{v}{R_2} = 0 \tag{19.110}$$

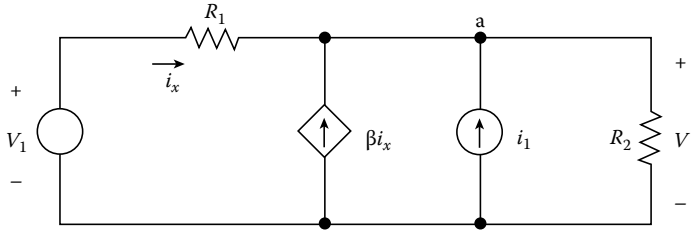


FIGURE 19.60 Example of a linear circuit.

and then by making use of the fact that

$$i_x = \frac{v_1 - v}{R_1} \tag{19.111}$$

This gives

$$v = \frac{(1+\beta)R_2}{R_1 + (1+\beta)R_2} v_1 + \frac{R_1 R_2}{R_1 + (1+\beta)R_2} i_1 \tag{19.112}$$

Here, the voltage v is a linear combination of the independent sources v_1 and i_1.

The preceding observation indeed applies to every linear circuit. In general, if we let y denote a voltage across, or a current in, any element in a linear circuit and if we let $\{x_1, x_2, \ldots, x_N\}$ denote the independent voltage and current sources in that circuit (assuming there is a total of N such sources), then we can write

$$y = \sum_{k=1}^{N} a_k x_k \tag{19.113}$$

where $a_1, a_2, \ldots, a_N$ are constants which depend on the circuit parameters. Thus, in the circuit of Figure 19.60, every current or voltage variable can be expressed as a linear combination of the form

$$y = a_1 V_1 + a_2 i_1$$

where a_1 and a_2 are constants that depend on R_1, R_2, and β. For the voltage v across R_2, this relationship is given by expression (Equation 19.112).

Mathematically, the relationship between y and $\{x_1, x_2, \ldots, x_N\}$ expressed in Equation 19.113 is said to be linear because it satisfies the following two conditions:

1. The *superposition condition*, which requires that

$$\begin{aligned} \text{if} \quad & \hat{y} = \sum_{k=1}^{N} a_k \hat{x}_k \\ \text{and} \quad & \tilde{y} = \sum_{k=1}^{N} a_k \tilde{x}_k \\ \text{then} \quad & \hat{y} + \tilde{y} = \sum_{k=1}^{N} a_k (\hat{x}_k + \tilde{x}_k) \end{aligned}$$

2. The homogeneity condition, which requires that

$$\begin{aligned} \text{if} \quad & \hat{y} = \sum_{k=1}^{N} a_k \hat{x}_k \\ \text{then} \quad & c\hat{y}(t) = \sum_{k=1}^{N} a_k \{c\hat{x}_k\} \end{aligned}$$

for any constant c.

Example 19.1 illustrates how these two conditions are satisfied for the circuit of Figure 19.60.

Example 19.1

For the circuit of Figure 19.60, let $R_1 = 2\ \Omega$, $R_2 = 1\ \Omega$, and $\beta = 2$. Show that the expression for v in terms of v_1 and i_1 satisfies the superposition and homogeneity conditions.

Substituting the values of R_1, R_2, and β in Equation 19.112, the expression for v becomes

$$v = \frac{3}{5}v_1 + \frac{2}{5}i_1 \tag{19.114}$$

To check the superposition property, let $v_1 = \hat{v}_1$ and $i_1 = \hat{i}_1$. Then, the voltage $\hat{v}$ across R_2 is

$$\hat{v} = \frac{3}{5}\hat{v}_1 + \frac{2}{5}\hat{i}_1$$

Similarly, let $V_1 = \tilde{V}_1$ and $i_1 = \tilde{i}_1$. Then, the voltage $\tilde{V}$ across R_2 is

$$\tilde{v} = \frac{3}{5}\tilde{v}_1 + \frac{2}{5}\tilde{i}_1$$

Now, assume that $V_1 = \hat{V}_1 + \tilde{V}_1$ and that $i_1 = \hat{i}_1 + \tilde{i}_1$. Then, according to Equation 19.114 the corresponding voltage V across R_2 is

$$\begin{aligned} V &= \frac{3}{5}\left(\hat{V}_1 + \tilde{V}_1\right) + \frac{2}{5}\left(\hat{i}_1 + \tilde{i}_1\right) \\ &= \left(\frac{3}{5}\hat{V}_1 + \frac{2}{5}\hat{i}_1\right) + \left(\frac{3}{5}\tilde{V}_1 + \frac{2}{5}\tilde{i}_1\right) \\ &= \hat{V} + \tilde{V} \end{aligned}$$

Hence, the superposition condition is satisfied.

To check the homogeneity condition, let $V_1 = c\hat{V}_1$ and $i_1 = c\hat{i}_1$, where c is an arbitrary constant. Then, according to Equation 19.114 the corresponding voltage v across R_2 is

$$\begin{aligned} V &= \frac{3}{5}\left(c\hat{V}_1\right) + \frac{2}{5}\left(c\hat{i}_1\right) \\ &= c\left(\frac{3}{5}\hat{V}_1 + \frac{3}{5}\hat{i}_1\right) \\ &= c\hat{V} \end{aligned}$$

The homogeneity condition is also satisfied.

19.2.1 Superposition Theorem

Let us reexamine expression (Equation 19.112) for the voltage v in the circuit of Figure 19.60. To be more specific, let us use this expression to calculate the voltage across R_2 for the two circuits shown in Figure 19.61a and b, respectively. Observe that the first circuit is obtained from the original circuit by deactivating the current source (i.e., setting $i_1 = 0$) and leaving the voltage source to act alone. The second is obtained by deactivating the voltage source (i.e., setting $V_1 = 0$) and leaving the current source to act alone. If we label the voltages across R_2 in these two circuits as v_a and v_b, respectively, then

$$v_a = v\Big|_{\substack{\text{when} \\ i_1 = 0}} = \frac{(1+\beta)R_2}{R_1 + (1+\beta)R_2}v_1$$

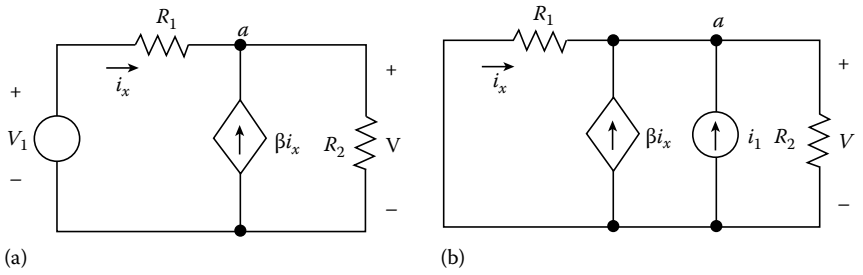


FIGURE 19.61 Circuit of Figure 19.60 with (a) the current source deactivated and with (b) the voltage source deactivated.

and

$$v_b = v\Big|_{\substack{\text{when}\\ v_1=0}} = \frac{R_1 R_2}{R_1 + (1+\beta)R_2} i_1$$

In other words, expression (Equation 19.112), which was used to derive the above two expressions, can itself be written as

$$v = v\Big|_{\substack{\text{when}\\ i_1=0}} + v\Big|_{\substack{\text{when}\\ v_1=0}}$$

or

$$v = v_a + v_b$$

Thus, we conclude that the voltage across R_2 in Figure 19.60 is actually equal to the sum of two voltages across R_2 due to two independent sources in the circuit acting individually.

The preceding result is in fact a direct consequence of the linearity property

$$y = a_1 x_1 + a_2 x_2 + \cdots + a_N x_N$$

expressed in Equation 19.113. Note that, from this expression, we can write

$$\begin{aligned}
a_1 x_1 &= y\big|_{\text{when } x_1 \neq 0, x_2 = 0, x_3 = 0, \ldots, x_N = 0},\\
a_2 x_2 &= y\big|_{\text{when } x_1 = 0, x_2 \neq 0, x_3 = 0, \ldots, x_N = 0},\\
&\vdots \quad\quad \vdots\\
a_N x_N &= y\big|_{\text{when } x_1 = 0, x_2 = 0, x_3 = 0, \ldots, x_N \neq 0}
\end{aligned}$$

This means Equation 19.113 can be rewritten as

$$\begin{aligned}
y = {} & y\big|_{\text{when } x_1 \neq 0, x_2 = 0, x_3 = 0, \ldots, x_N = 0}\\
& + y\big|_{\text{when } x_1 = 0, x_2 \neq 0, x_3 = 0, \ldots, x_N = 0}\\
& \quad \vdots \quad\quad \vdots\\
& + y\big|_{\text{when } x_1 = 0, x_2 = 0, x_3 = 0, \ldots, x_N \neq 0}
\end{aligned}$$

The following theorem, known as the superposition theorem, is therefore directly implied from the previous expression:

> The voltage across any element (or current through any element) in a linear circuit may be calculated by adding algebraically the individual voltages across that element (or currents through that element) due to each independent source acting alone with all other independent sources deactivated.

In this statement, the word deactivated is used to imply that the source is set to zero. In this context, we refer to a deactivated current source as one that is replaced by an open circuit and a deactivated voltage source as one that is replaced by a short circuit. Note that the action of deactivating a source refers only to independent sources. Example 19.2 illustrates how the superposition theorem can be used to solve for a variable in a circuit with more than one independent source.

Example 19.2

For the circuit shown in Figure 19.62, apply superposition to calculate the current *I* in the 4 Ω resistor.

Because we are interested in calculating *I* using the superposition theorem, we need to consider the two circuits shown in Figure 19.63a and b. The first is obtained by deactivating the current source and the second is obtained by deactivating the voltage source. Let I_a be the current in the 4 Ω resistor in the first circuit and I_b be the current in the same resistor in the second. Then, by superposition

$$I = I_a + I_b$$

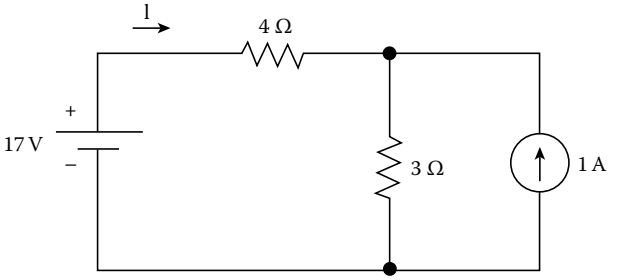


FIGURE 19.62 Circuit for Example 19.2.

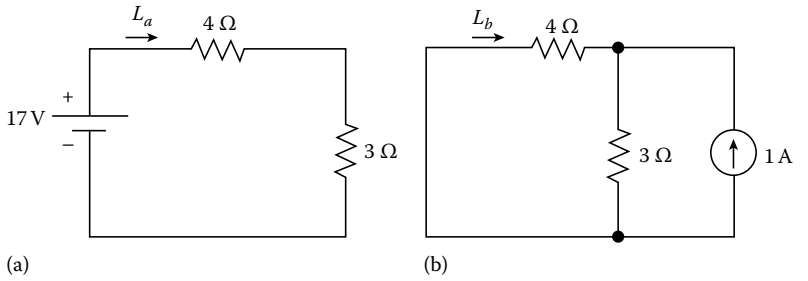


FIGURE 19.63 Circuit for Example 19.2 with (a) the current source deactivated and with (b) the voltage source deactivated.

We can solve for I_a and I_b independently as follows. From Figure 19.63a:

$$I_a = \frac{17}{7}\text{A}$$

and from Figure 19.63b, applying the current divider rule,

$$I_b = -1 \cdot \frac{3}{7}\text{A}$$

Thus,

$$\begin{aligned} I &= \frac{17}{7} - \frac{3}{7} \\ &= 2\text{A} \end{aligned}$$

19.2.2 Thévenin Theorem

In the discussion on the superposition theorem, we interpreted Equation 19.112 for the voltage V in the circuit of Figure 19.60 as a superposition of two terms. Let us now examine a different interpretation of this expression.

Suppose we factor the common term in expression (Equation 19.112), so that it can be written as

$$v = \frac{(1+\beta)R_2}{R_1 + (1+\beta)R_2}\left\{v_1 + \frac{R_1}{1+\beta}i_1\right\} \tag{19.115a}$$

or

$$v = \frac{R_2}{(R_1/(1+\beta)) + R_2}\left\{v_1 + \left(\frac{R_1}{1+\beta}\right)i_1\right\} \tag{19.115b}$$

Now, suppose we define

$$v_0 = v_1 + \frac{R_1}{1+\beta}i_1 \tag{19.116}$$

and

$$R_0 = \frac{R_1}{1+\beta} \tag{19.117}$$

Then, we can write Equation 19.112 in the simple form

$$v = \frac{R_2}{R_0 + R_2}v_0 \tag{19.118}$$

This expression can be interpreted as a voltage divider equation for a two-resistor circuit as shown in Figure 19.64. This circuit has a voltage source v_0 in series with two resistors R_0 and R_2. When this circuit

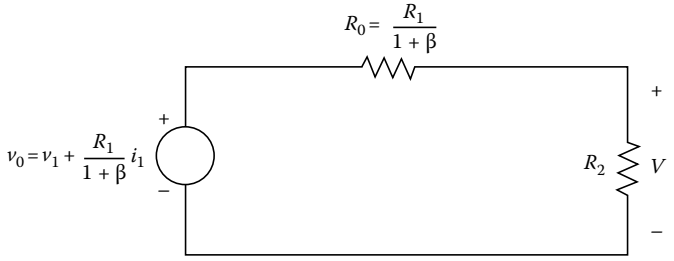


FIGURE 19.64 Voltage divider circuit representing Equation 19.118.

is compared with Figure 19.60, the combination of voltage source v_0 in series with R_0 can be interpreted as an equivalent replacement of all the elements in the circuit connected to R_2. That is, we could remove that portion of the circuit of Figure 19.60 consisting of v_1, R_1, βi_x, and i_1 and replace it with the voltage source v_0 in series with the resistor R_0.

The fact that a portion of a circuit can be replaced by an equivalent circuit consisting of a voltage source in series with a resistor is actually a direct result of the linearity property, and hence is true for linear circuits in general. It is known as Thévenin's theorem* and is stated as follows:

> Any portion of a linear circuit between two terminals a and b can be replaced by an equivalent circuit consisting of a voltage source V_{th} in series with a resistor R_{th}. The voltage V_{th} is determined as the open circuit voltage at terminals $a-b$. The resistor R_{th} is equal to the input resistance at terminals $a-b$ with all the independent sources deactivated.

The various steps involved in the derivation of the Thévenin equivalent circuit are illustrated in Figure 19.65. The Thévenin voltage v_{th} is determined by solving for the voltage at terminals $a-b$ when open circuited, and the Thévenin resistance R_{th} is determined by calculating the input resistance of the circuit at terminals $a-b$ when all the independent sources have been deactivated. Examples 19.3 and 19.4 illustrate the application of this important theorem.

Example 19.3

For the circuit in Figure 19.66, determine the Thévenin equivalent of the portion of the circuit to the left of terminals $a-b$; use it to calculate the current I in the 2 Ω resistor.

First, we determine V_{th} from the circuit of Figure 19.67a. Note that because terminals $a-b$ are open circuited the current in the branch containing the 2 Ω resistor and 9 V source is equal to 3 A in the direction shown. Applying KCL at the upper node of the 1 Ω resistor, we can calculate the current in this resistor to be $3 + 2 = 5$ A as shown. Writing a KVL equation around the inner loop (counterclockwise at terminal b) we have

$$V_{\text{th}} + 9 - (2 \times 3) - (1 \times 5) = 0$$

which yields

$$V_{\text{th}} = 2 \text{ V}$$

* For an interesting, brief discussion on the history of Thévenin's theorem, see an article by James E. Brittain, in *IEEE Spectrum*, p. 42, March 1990.

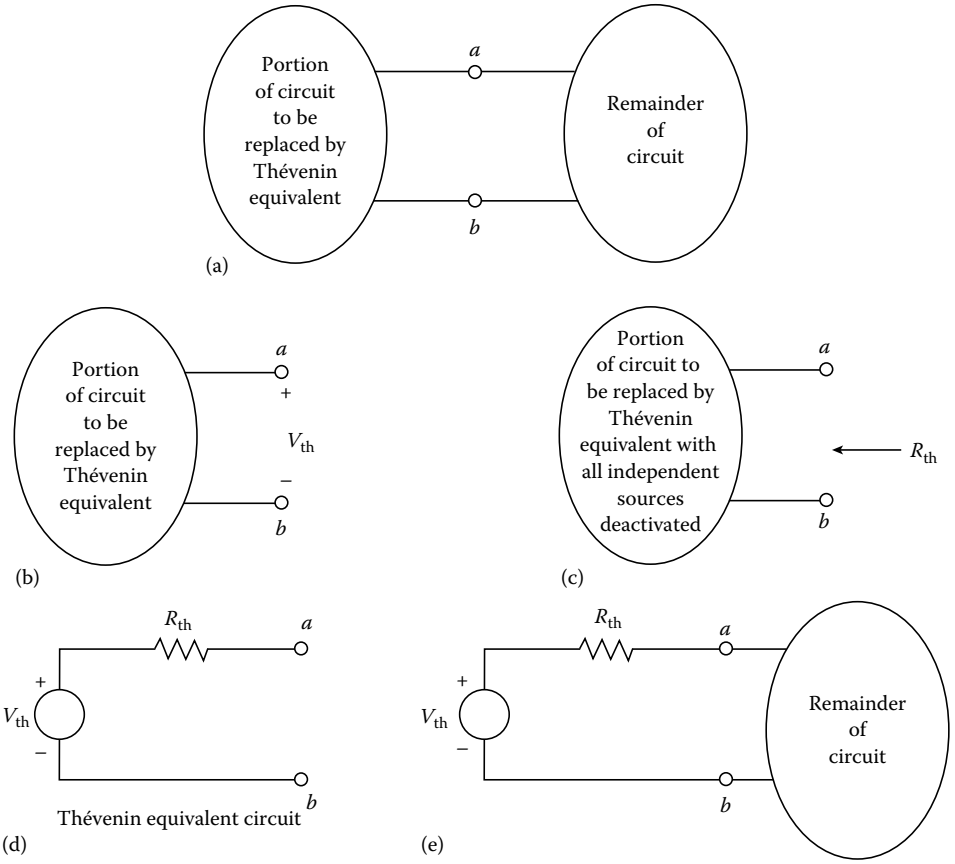


FIGURE 19.65 Steps in determining the Thévenin equivalent circuit.

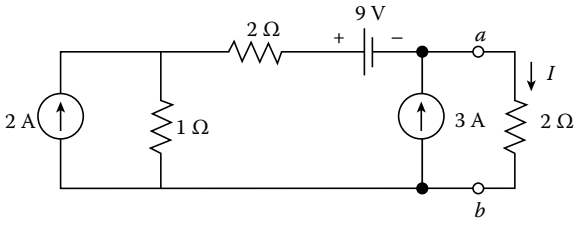


FIGURE 19.66 Circuit for Example 19.3.

Now, for R_{th} the three sources are deactivated to obtain the circuit shown in Figure 19.67b. From this circuit, it is clear that

$$R_{th} = 3\ \Omega$$

The circuit obtained by replacing the portion to the left of terminals $a-b$ with its Thévenin equivalent is shown in Figure 19.67c. The current I is now easily computed as

$$I = \frac{2}{2+3} = 0.4\ \text{A}$$

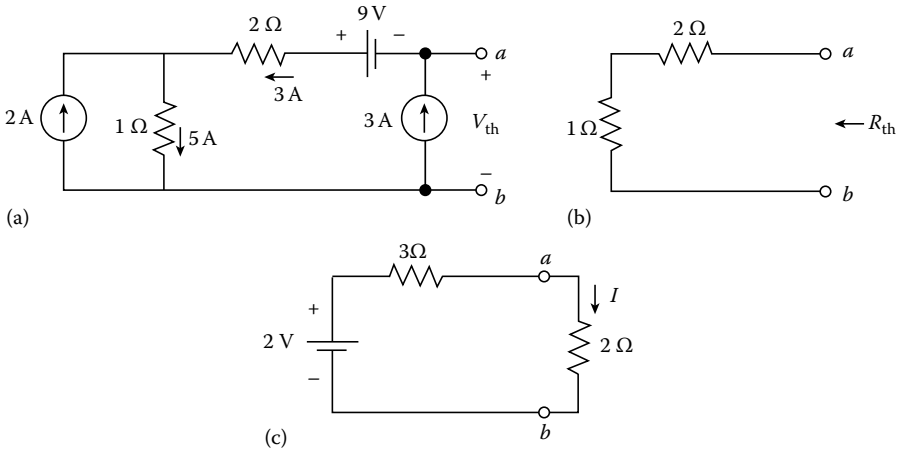


FIGURE 19.67 (a) Calculation of V_{th}, (b) calculation of R_{th}, and (c) the equivalent circuit for Example 19.3.

Example 19.4

For the circuit shown in Figure 19.60, determine the Thévenin equivalent circuit for the portion of the circuit to the left of resistor R_2.

In deriving Equation 19.118 from 19.110, we actually already determined the Thévenin equivalent for the portion of the circuit to the left of R_2. This was shown in Figure 19.64. Of course, this procedure is not the most efficient way to determine the Thévenin equivalent. Let us now illustrate how the equivalent circuit is obtained using the procedure described in Thévenin's theorem. First, we determine v_{th} from the circuit of Figure 19.68a with R_2 removed and terminals $a-b$ left open. Applying KCL at node a, we have

$$i_x + \beta i_x + i_1 = 0$$

or

$$i_x = -\frac{i_1}{1+\beta}$$

Hence,

$$\begin{aligned} v_{th} &= v_1 - R_1 i_x \\ &= v_1 + \frac{R_1}{1+\beta} i_1 \end{aligned}$$

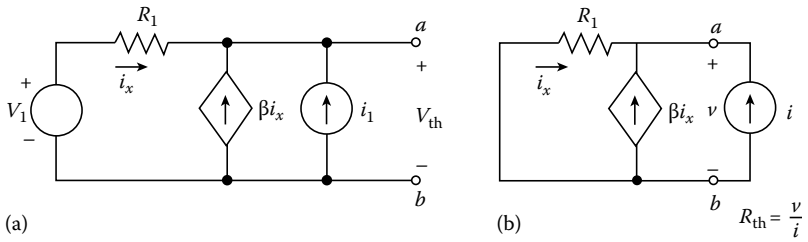


FIGURE 19.68 Calculation of (a) V_{th} and (b) R_{th} for the circuit of Figure 19.60 (Example 19.4).

As for R_{th}, we need to consider the circuit shown in Figure 19.68b, in which the two independent sources were deactivated. Because of the presence of the dependent source βi_x, we determine R_{th} by exciting the circuit with an external source. Let us use a current source i for this purpose and determine the voltage v across it as shown in Figure 19.68b. We stress that i is an arbitrary and completely independent source and is in no way related to i_1, which was deactivated. Applying KCL at node a, we have

$$i_x + \beta i_x + i = 0$$

or

$$i_x = -\frac{1}{1+\beta} i$$

Also, applying Ohm's law to R_1,

$$v = -R_1 i_x$$

or

$$v = \frac{R_1}{1+\beta} i$$

Hence,

$$R_{th} = \frac{v}{i} = \frac{R_1}{1+\beta}$$

Note that v_{th} and R_{th} determined previously are the same as v_0 and R_0 of Equations 19.116 and 19.117.

19.2.3 Norton Theorem

Instead of a voltage source in series with a resistor, it is possible to replace a portion of a circuit by an equivalent current source in parallel with a resistor. This result is formally known as Norton's theorem and is stated as follows:

> Any portion of linear circuit between two terminals a and b can be replaced by an equivalent circuit consisting of a current source i_n in parallel with a resistor R_n. The current i_n is determined as the current that flows from a to b in a short circuit at terminals $a-b$. The resistor R_n is equal to the input resistance at terminals $a-b$ with all the independent sources deactivated.

As in the case of Thévenin's, the preceding theorem provides a procedure for determining the "Norton" current source and "Norton" resistance in the Norton equivalent circuit. The various steps in this procedure are illustrated in Figure 19.69. The Norton current is determined by solving for the current in a short circuit at terminals $a-b$ and the Norton resistance is determined by calculating the input resistance to the circuit at terminals $a-b$ when all independent sources have been deactivated. The Norton's equivalent of a portion of a circuit is, in effect, a nonideal current source representation of that portion. It should be noted that the procedure to determine R_n is exactly the same as that for R_{th}. In other words,

$$R_n = R_{th}$$

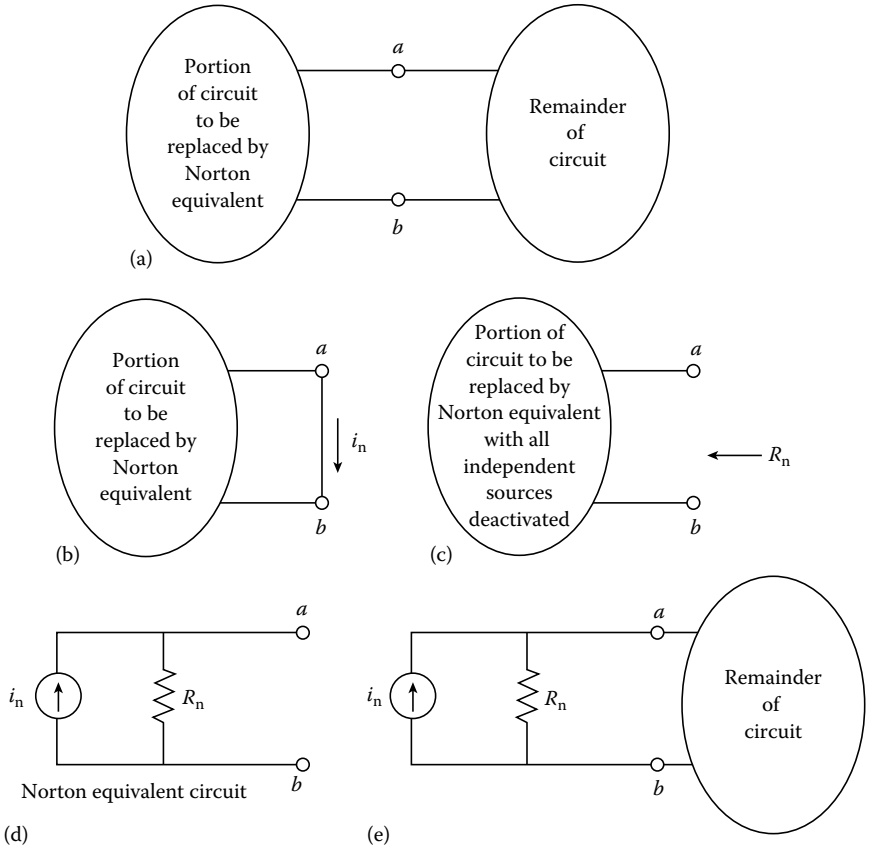


FIGURE 19.69 Steps in determining the Norton equivalent circuit.

Also, if we compare Thévenin's and Norton's equivalent circuits, we see that these are indeed related by the voltage–current source transformation rule discussed earlier in this chapter. Each circuit is a source transformation of the other. For this reason, the Thévenin and Norton equivalent circuits are often referred to as dual circuits, and the two resistance R_{th} and R_n are frequently referred to as the source resistance and denoted by R_s. Clearly, v_{th} and i_n are related by

$$v_{th} = R_s i_n$$

Example 19.5

For the circuit shown in Figure 19.70, determine the Norton equivalent circuit at terminals *a*–*b*.

We determine Norton's current I_n by placing a short circuit between *a* and *b*, as shown in Figure 19.71a, and solving for the current in it with a reference direction going from *a* to *b*. For this circuit, we could use the mesh equation method. In matrix form, the mesh equations are

$$\begin{bmatrix} 8 & -4 \\ -4 & 7 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

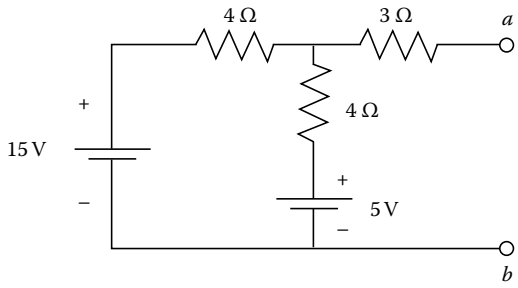


FIGURE 19.70 Circuit for Example 19.5.

and the solution for the mesh currents is

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \frac{1}{56-16}\begin{bmatrix} 7 & 4 \\ 4 & 8 \end{bmatrix} \cdot \begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

From this, we extract I_n as

$$I_n = I_2 = \frac{40+40}{40} = 2 \text{ A}$$

Norton's resistance R_n is determined by deactivating the two voltage sources, as shown in Figure 19.71b, and calculating the input resistance at terminals $a-b$. Clearly,

$$R_n = (4 \parallel 4) + 3 = 5 \ \Omega$$

Thus, the Norton equivalent for the circuit of Figure 19.70 is shown in Figure 19.71c.

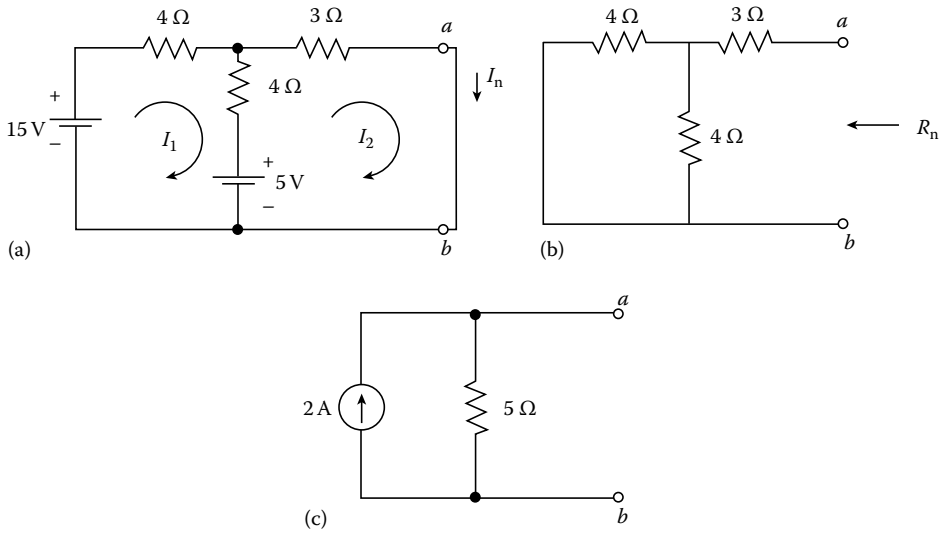


FIGURE 19.71 (a) Calculation of I_n, (b) calculation of R_n and (c) Norton's equivalent for the circuit of Example 19.5.

Example 19.6

For the circuit shown in Figure 19.72 determine the Norton equivalent circuit at terminals *a*–*b* and use it to calculate the current and power dissipated in *R*.

With a short circuit placed at terminals *a*–*b*, as shown in Figure 19.73a, the voltage $V_x = 0$. Hence, the dependent source in this circuit is equal to zero. This means that the 8 A current source has the 9 and 3 Ω resistors in parallel across it, and I_n is the current in the 3 Ω resistor. Using the current divider rule we have

$$I_n = 8\frac{9}{12} = 6 \text{ A}$$

Now, deactivating the independent source to determine R_n, we excite the circuit with a voltage source V at terminals *a*–*b*. Let I be the current in this source as shown in Figure 19.73b. Applying KCL at node a yields the current in the 3 Ω resistor to be $I - (V/4)$ from right to left. Applying KVL around the outer loop and making use of the fact that in this circuit $V_x = V$, we get

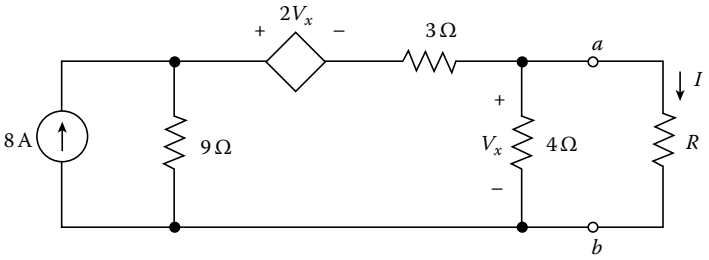


FIGURE 19.72 Circuit for Example 19.6.

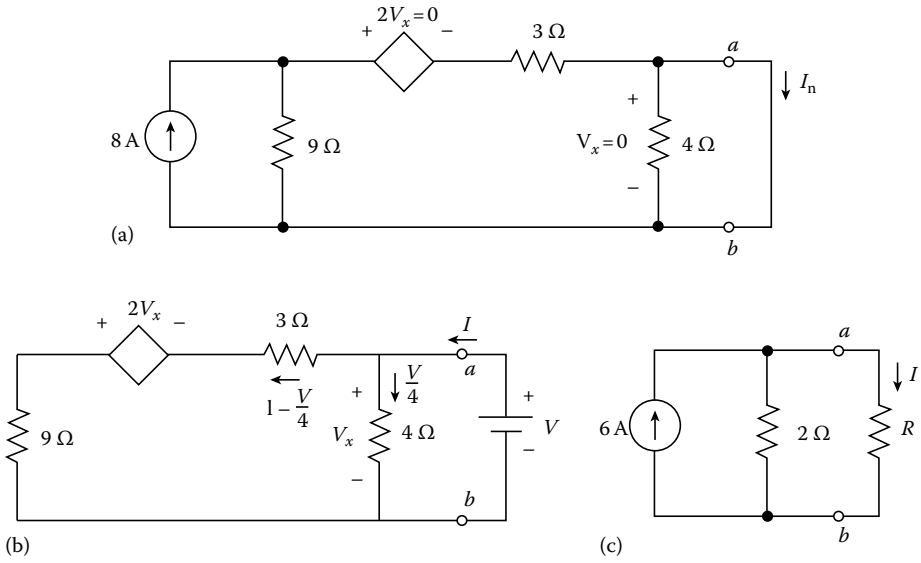


FIGURE 19.73 (a) Calculation of I_n, (b) calculation of R_n, and (c) Norton's equivalent for the circuit of Example 19.6.

$$V - 3\left(I - \frac{V}{4}\right) + 2V - 9\left(I - \frac{V}{4}\right) = 0$$

Solution of this equation yields

$$R_{\mathrm{n}} = \frac{V}{I} = 2\ \Omega$$

The Norton equivalent of the portion of the circuit to the left of terminals $a-b$, connected to the resistor R is shown in Figure 19.73c. Applying the current divider rule gives

$$\begin{aligned} I &= 6\frac{2}{2+R} \\ &= \frac{12}{2+R}\,\mathrm{A} \end{aligned}$$

and the power dissipated in R is

$$\begin{aligned} P &= RI^2 \\ P &= \frac{144R}{(2+R)^2}\,\mathrm{W} \end{aligned} \tag{19.119}$$

19.2.4 Maximum Power Transfer Theorem

In Example 19.6, we replaced the entire circuit connected to the resistor R at terminals $a-b$ by its Norton equivalent in order to calculate the power P dissipated in R. Because R did not have a fixed value, we determined an expression for P in terms of R. Suppose we are now interested in examining how P varies as a function of R. A plot of P versus R as given by Equation 19.119 is given in Figure 19.74.

The first noticeable characteristic of this plot is that it has a maximum. Naturally, we would be interested in the value of R that results in maximum power delivered to it. This information is directly available from the plot in Figure 19.74. To maximize P the value of R should be 2 Ω and the maximum power is $P_{\mathrm{max}} = 18$ W. That is, 18 W is the most that this circuit can deliver at terminals $a-b$, and that occurs when $R = 2\ \Omega$. Any other value of R will result in less power delivered to it.

The problem of finding the value of a load resistor R_{L} such that maximum power is delivered to it is obviously an important circuit design problem. Because it is possible to reduce any linear circuit connected to R_{L} into either its Thévenin or Norton equivalent, as illustrated in Figure 19.75, the problem becomes quite simple. We need to consider only either the circuit of Figure 19.75b or that of Figure 19.75c. Let us first consider the circuit that uses the Thévenin equivalent. In this case, the power P delivered to R_{L} is given by

$$P = \left(\frac{v_{\mathrm{th}}}{R_{\mathrm{th}} + R_{\mathrm{L}}}\right)^2 R_{\mathrm{L}} \tag{19.120}$$

In general, we may not always be able to plot P vs. R_{L}, as we did earlier, therefore, we need to maximize P mathematically. We do this by solving the necessary condition

$$\frac{\mathrm{d}P}{\mathrm{d}R_{\mathrm{L}}} = 0 \tag{19.121}$$

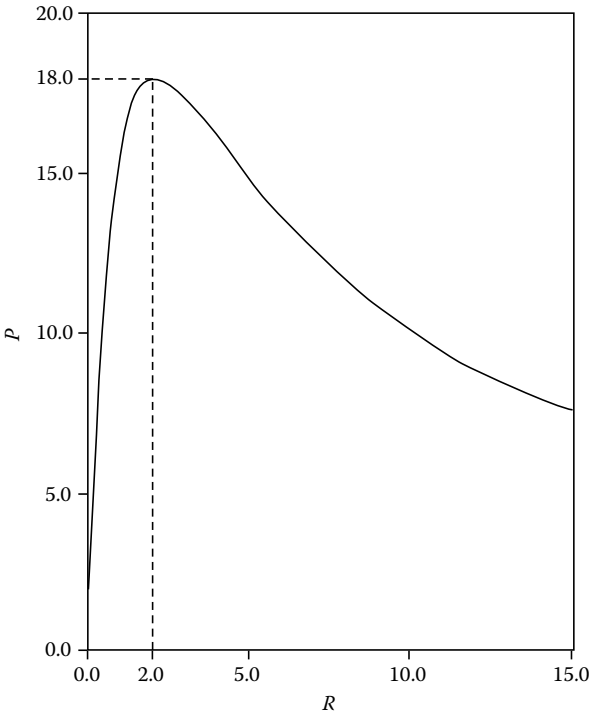


FIGURE 19.74 Plot of P vs. R for the circuit of Example 19.6.

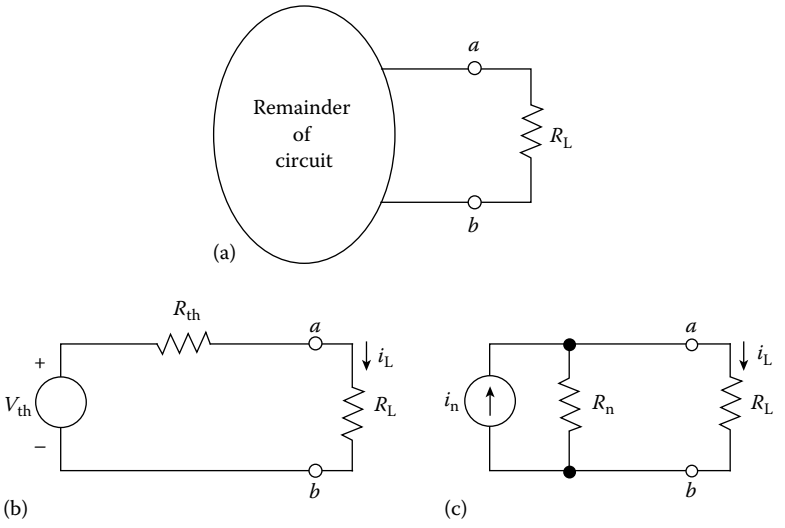


FIGURE 19.75 (a) A load resistance R_L in a circuit. (b) R_L with the remainder of the circuit reduced to a Thévenin equivalent. (c) R_L with the remainder of the circuit reduced to a Norton equivalent.

for R_L. To guarantee that R_L maximizes P, it must also satisfy the sufficiency condition

$$\left.\frac{d^2P}{dR_L^2}\right|_{R_L} < 0 \tag{19.122}$$

Thus, applying these conditions to Equation 19.120, we have

$$\begin{aligned}\frac{dP}{dR_L} &= v_{th}^2\left[\frac{(R_{th}+R_L)^2 - 2R_L(R_{th}+R_L)}{(R_{th}+R_L)^4}\right],\\ &= v_{th}^2\frac{(R_{th}-R_L)}{(R_{th}+R_L)^3}\end{aligned} \tag{19.123}$$

Equating the right-hand side of Equation 19.123 to zero and solving for R_L yields

$$R_L = R_{th}$$

The sufficiency condition (Equation 19.122) yields

$$\frac{d^2P}{dR_L^2} = v_{th}^2\frac{2R_{th}-4R_L}{(R_{th}+R_L)^4}$$

When R_L is replaced with R_{th}, we get

$$\left.\frac{d^2P}{dR_L^2}\right|_{R_L=R_{th}} = -\frac{v_{th}^2}{8R_{th}^3} < 0$$

Thus, $R_L = R_{th}$ satisfies both conditions (Equations 19.121 and 19.122), and hence is the maximizing value. This result is often referred to as the "maximum power transfer theorem." It says

> The maximum power that can be transferred to a load resistance R_L by a circuit represented by its Thévenin equivalent is attained when R_L is equal to R_{th}.

The corresponding value of P_{max} is obtained from Equation 19.120 as

$$P_{max} = \frac{v_{th}^2}{4R_{th}} \tag{19.124}$$

In the case of Norton's equivalent circuit of Figure 19.75b, a similar derivation can be carried out. The power P delivered to R_L is given by

$$P = \left(\frac{R_n i_n}{R_n + R_L}\right)^2 R_L \tag{19.125}$$

This expression has exactly the same form as Equation 19.120. Consequently, its maximum is achieved when

$$R_L = R_n$$

and the corresponding maximum power is

$$P_{\max} = \frac{R_n i_n^2}{4} \tag{19.126}$$

This leads to the following alternate statement of the maximum power transfer theorem:

> The maximum power that can be transferred to a load resistance R_L by a circuit represented by its Norton equivalent is attained when R_L is equal to R_n.

Example 19.7

Consider the circuit of Example 19.3 in Figure 19.66. Determine the value of a load resistor R_L connected in place of the 2 Ω resistor at terminals $a-b$ in order to achieve maximum power transfer to the load.

Solution

The Thévenin equivalent for the circuit of Figure 19.66 already was determined and is shown in Figure 19.67c. Using the results of the maximum power transfer theorem, we should have

$$R_L = 3\Omega$$

The corresponding value of maximum power is

$$P_{\max} = \frac{2^2}{4 \times 3} = \frac{1}{3}\text{W}$$

19.2.5 Reciprocity Theorem

The reciprocity theorem is an important result that applies to circuits consisting of linear resistors and one independent source (either a current source or a voltage source). It does not apply to nonlinear circuits, and, in general, it does not apply to circuits containing dependent sources. The reciprocity theorem is stated as follows:

The ratio of a voltage (or current) response in one part of the circuit to the current (or voltage) source is the same if the locations of the response and the source are interchanged.

It is important to note that the reciprocity theorem applies only to circuits in which the source and response are voltage and current or current and voltage, respectively. It does not apply to circuits in which the source and response are of the same type (i.e., voltage and voltage or current and current).

Example 19.8

Verify the reciprocity theorem for the circuit shown in Figure 19.76a.

Solution

If we interchange the location of the 40 V voltage source and current response I, we obtain the circuit shown in Figure 19.76b. The reciprocity theorem implies that I should be the same in both circuits.

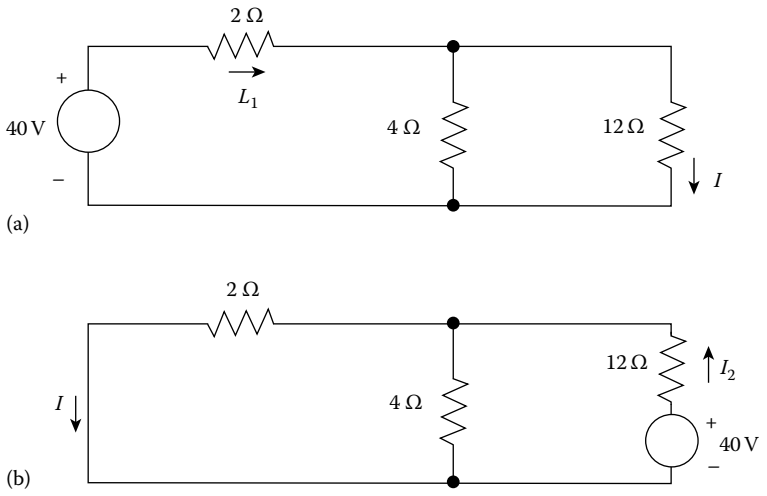


FIGURE 19.76 (a) Circuit for Example 19.8. (b) Circuit with location of voltage source and current response interchanged.

For the circuit in Figure 19.76a, the current I_1 in the 2 Ω resistor is equal to

$$\begin{aligned} I_1 &= \frac{40}{2 + 4 \parallel 12} \\ &= \frac{40}{2 + 3} \\ &= 8\,\text{A} \end{aligned}$$

and the response I can be easily determined, using the current divider rule, as

$$\begin{aligned} I &= 8 \times \frac{4}{4 + 12} \\ &= 2\,\text{A} \end{aligned}$$

For the circuit in Figure 19.76b, the current I_2 in the 12 Ω resistor is equal to

$$\begin{aligned} I_2 &= \frac{40}{12 + 2 \parallel 4} \\ &= \frac{40}{12 + \frac{4}{3}} \\ &= 3\,\text{A} \end{aligned}$$

and the response I can be determined easily, using the current divider rule, as

$$\begin{aligned} I &= 3 \times \frac{4}{2 + 4} \\ &= 2\,\text{A} \end{aligned}$$

Thus, the reciprocity theorem is satisfied.

20

Terminal and Port Representations

20.1 Introduction **20**-1
20.2 Terminal Representations **20**-1
20.3 Port Representations **20**-5
20.4 Port Representations and Feedback Systems **20**-15
20.5 Conclusions **20**-19
References **20**-20

James A. Svoboda
Clarkson University

20.1 Introduction

Frequently, it is useful to decompose a large circuit into subcircuits and to consider the subcircuits separately. Subcircuits are connected to other subcircuits using terminals or ports. Terminal and port representations of a subcircuit describe how that subcircuit will act when connected to other subcircuits [1,2]. Terminal and port representations do not provide the details of mesh or node equations because these details are not required to describe how a subcircuit interacts with other subcircuits.

In this chapter, terminal and port representations of circuits are described. Particular attention is given to the distinction between terminals and ports. Applications show the usefulness of terminal and port representations.

20.2 Terminal Representations

Figure 20.1 illustrates a subcircuit that can be connected to other subcircuits using terminals. The subcircuit is shown symbolically on the left and an example is shown on the right. Nodes *a*, *b*, *c*, and *d* are terminals and may be used to connect this circuit to other circuits. Node *e* is internal to the circuit and is not a terminal. The terminal voltages V_a, V_b, V_c, and V_d are node voltages with respect to an arbitrary reference node. The terminal currents I_a, I_b, I_c, and I_d describe currents that will exist when this subcircuit is connected to other subcircuits. Terminal representations show how the terminal voltages and currents are related. Several equivalent representations are possible, depending on which of the terminal currents and voltages are selected as independent variables.

A terminal representation of the example network in Figure 20.1 can be obtained by writing a node equation for the network. A simpler procedure is available for passive networks, i.e., networks consisting entirely of resistors, capacitors, and inductors. Because the circuit in Figure 20.1 is a passive circuit, the simpler procedure will be used to write terminal equations to represent this circuit.

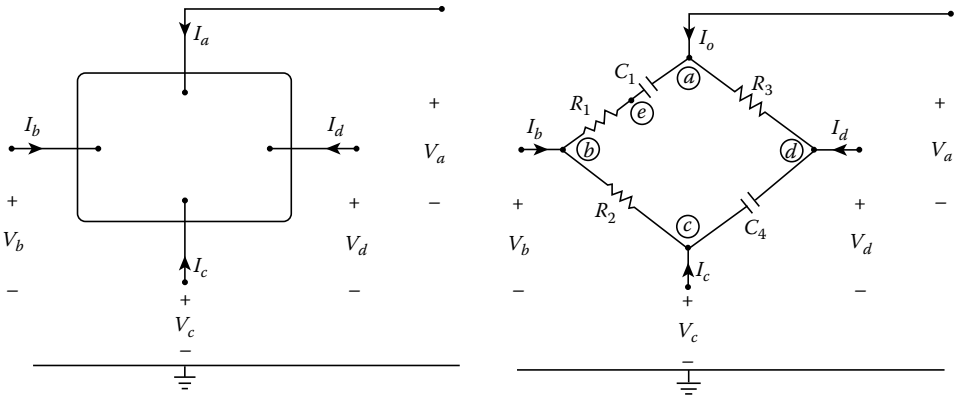


FIGURE 20.1 Four-terminal network.

The nodal admittance matrix of the example circuit will have five rows and five columns. The rows, and also the columns, of this matrix will correspond to the five nodes of the circuit in the order *a*, *b*, *c*, *d*, and *e*. For example, the fourth row of the nodal admittance matrix corresponds to node *d* and the third column corresponds to node *c*. Let y_{ij} denote the admittance of a branch of the network which is incident with nodes *i* and *j* and let Y_{ij} denote the element of the nodal admittance matrix that is in row *i* and column *j*. Then,

$$Y_{ii} = \sum_{\substack{\text{all branches} \\ \text{incident to node } i}} y_{ij} \tag{20.1}$$

i.e., the diagonal element of the nodal admittance matrix in row *i* and column *i* is equal to the sum of the admittances of all branches in the circuit that are incident with node *i*. The off-diagonal elements of the nodal admittance matrix are given by

$$Y_{ij} = -\sum_{\substack{\text{all branches incident} \\ \text{to both nodes } i \text{ and } j}} y_{ij} \tag{20.2}$$

The example circuit is represented by the node equation

$$\begin{pmatrix} I_a \\ I_b \\ I_c \\ I_d \\ 0 \end{pmatrix} = \begin{pmatrix} C_1 s + \dfrac{1}{R_3} & 0 & 0 & -\dfrac{1}{R_3} & -C_1 s \\ 0 & \dfrac{1}{R_1} + \dfrac{1}{R_2} & -\dfrac{1}{R_2} & 0 & -\dfrac{1}{R_1} \\ 0 & -\dfrac{1}{R_2} & C_4 s + \dfrac{1}{R_2} & -C_4 s & 0 \\ -\dfrac{1}{R_3} & 0 & -C_4 s & C_4 s + \dfrac{1}{R_3} & 0 \\ -C_1 s & -\dfrac{1}{R_1} & 0 & 0 & C_1 s + \dfrac{1}{R_1} \end{pmatrix} \begin{pmatrix} V_a \\ V_b \\ V_c \\ V_d \\ V_e \end{pmatrix} \tag{20.3}$$

Suppose that $C_1 = 1$ F, $C_4 = 2$ F, $R_1 = 1/2\ \Omega$, $R_2 = 1/4\ \Omega$, and $R_3 = 1\ \Omega$. Then,

$$\begin{pmatrix} I_a \\ I_b \\ I_c \\ I_d \\ 0 \end{pmatrix} = \begin{pmatrix} s+1 & 0 & 0 & -1 & -1s \\ 0 & 6 & -4 & 0 & -2 \\ 0 & -4 & 2s+4 & -2s & 0 \\ -1 & 0 & -2s & 2s+1 & 0 \\ -s & -2 & 0 & 0 & s+2 \end{pmatrix} \begin{pmatrix} V_a \\ V_b \\ V_c \\ V_d \\ V_e \end{pmatrix}$$

No external current I_e exists because node e is not a terminal. Row 5 of this equation, corresponding to node e, can be solved for V_e and then V_e can be eliminated. Doing so results in

$$\begin{pmatrix} I_a \\ I_b \\ I_c \\ I_d \end{pmatrix} = \begin{pmatrix} \dfrac{3s+2}{s+2} & -\dfrac{2s}{s+2} & 0 & -1 \\ -\dfrac{2s}{s+2} & \dfrac{6s+8}{s+2} & -4 & 0 \\ 0 & -4 & 2s+4 & -2s \\ -1 & 0 & -2s & 2s+1 \end{pmatrix} \begin{pmatrix} V_a \\ V_b \\ V_c \\ V_d \end{pmatrix} \tag{20.4}$$

The terminal voltages were selected to be the independent variables and the entries in the matrix are admittances. Because none of the nodes of the subcircuit was chosen to be the reference node, the rows and columns of the matrix both sum to zero. This matrix is called an indefinite admittance matrix. Because it is singular, Equation 20.4 cannot be solved directly. To see the utility of Equation 20.4, consider Figure 20.2. Here, the four-terminal network has been connected to a voltage source and an amplifier and node c has been grounded. This external circuitry restricts the terminal currents and voltages of the four-terminal network. In particular,

$$V_b = V_s, \quad I_a = 0, \quad V_c = 0, \quad I_d = 0, \quad \text{and} \quad V_o = KV_d \tag{20.5}$$

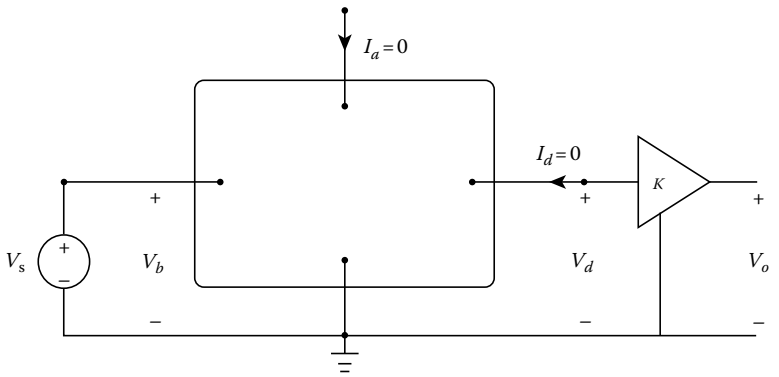


FIGURE 20.2 First application of the four-terminal network.

Under these conditions, Equation 20.4 becomes

$$\begin{pmatrix} 0 \\ I_b \\ I_c \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{3s+2}{s+2} & -\frac{2s}{s+2} & 0 & -1 \\ -\frac{2s}{s+2} & \frac{6s+8}{s+2} & -4 & 0 \\ 0 & -4 & 2s+4 & -2s \\ -1 & 0 & -2s & 2s+1 \end{pmatrix} \begin{pmatrix} V_a \\ V_s \\ 0 \\ \frac{V_o}{K} \end{pmatrix} \tag{20.6}$$

Because I_b and I_c are not of interest, the second and third rows of this equation can be ignored. The first and fourth rows can be solved to obtain the transfer function of this circuit

$$\frac{V_o(s)}{V_s(s)} = \frac{K}{3(s+1)} \tag{20.7}$$

Next, consider Figure 20.3. The four-terminal network is used in a second, different application. In this case,

$$I_a + I_b = -\frac{V_b}{R_b}, \quad V_c = V_s, \quad V_a = V_b, \quad I_d = 0, \quad \text{and} \quad V_d = V_o \tag{20.8}$$

Because $V_a = V_b$, the first two columns of Equation 20.4 can be added together, replacing V_a by V_b. Also, it is convenient to add the first two rows together to obtain $I_a + I_b$. Then, Equation 20.4 reduces to

$$\begin{pmatrix} -\frac{V_b}{R_b} \\ I_c \\ 0 \end{pmatrix} = \begin{pmatrix} 5 & -4 & -1 \\ -4 & 2s+4 & -2s \\ -1 & -2s & 2s+1 \end{pmatrix} \begin{pmatrix} V_b \\ V_s \\ V_o \end{pmatrix} \tag{20.9}$$

Because I_c is not of interest, the second row of this equation can be ignored. The first and third rows of this equation can be solved to obtain the transfer function

$$\frac{V_o(s)}{V_s(s)} = \frac{(10R_b + 2)s + 4R_b}{(10R_b + 2)s + 4R_b + 1} \tag{20.10}$$

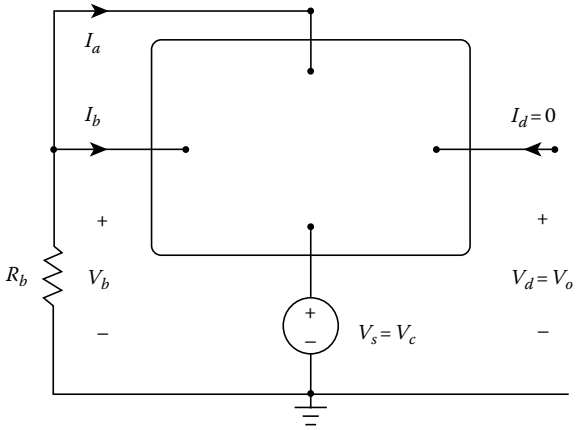


FIGURE 20.3 Second application of the four-terminal network.

These examples illustrate the utility of the terminal equations. In these examples, the problem of analyzing a network was divided into two parts. First, the terminal equation was obtained from the node equation by eliminating the rows and columns corresponding to nodes that are not terminals. Second, the terminal equation is combined with equations describing the external circuitry connected to the four-terminal network. When the external circuit is changed, only the second step must be redone. The advantage of representing a subnetwork by terminal equations is greater when

1. The subnetwork has many nodes that are not terminals.
2. The subnetwork is expected to be a component of many different networks.

20.3 Port Representations

A port consists of two terminals with the restriction that the terminal currents have the same magnitude but opposite sign. Figure 20.4 shows a two-port network. In this case, the two-port network was constructed from the four-terminal network by pairing terminals a and b to form port 1 and pairing terminals d and c to form port 2. The restrictions

$$I_1 = I_a = -I_b \quad \text{and} \quad I_2 = I_d = -I_c \tag{20.11}$$

must be satisfied in order for these two pairs of terminals to be ports.

The behavior of the two-port network is described using four variables. These variables are the port voltages V_1 and V_2 and the port currents I_1 and I_2. Several equivalent representations are possible, depending on which two of the port currents and voltages are selected as independent variables. The left column of Table 20.1 shows the two-port representations corresponding to four of the possible choices of independent variables. These representations are equivalent, and the right column of Table 20.1 shows how one representation can be obtained from another.

In row 1 of Table 20.1, the port voltages are selected to be the independent variables. In this case, the two-port circuit is represented by an equation of the form

$$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \tag{20.12}$$

The elements of the matrix in this equation have units of admittance. They are denoted using the letter y to suggest admittance and are called the "y parameters" or the "admittance parameters" of the two-port network.

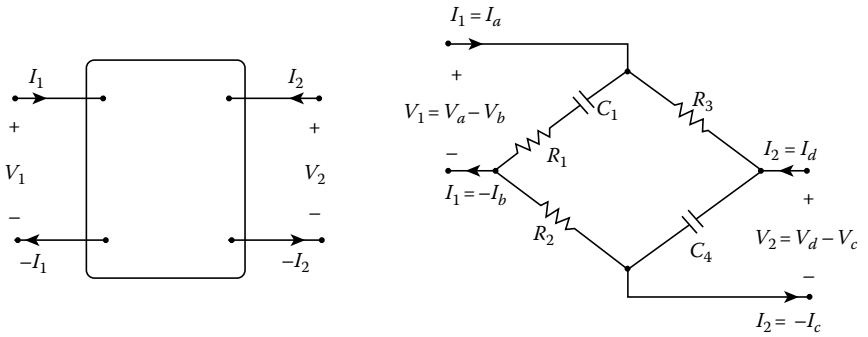


FIGURE 20.4 Two-port network.

TABLE 20.1 Relationships between Two-Port Representations

$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$ $\lvert Y \rvert = y_{11}y_{22} - y_{12}y_{21}$	$\begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} = \begin{pmatrix} \frac{z_{22}}{\lvert Z \rvert} & -\frac{z_{12}}{\lvert Z \rvert} \\ -\frac{z_{21}}{\lvert Z \rvert} & \frac{z_{11}}{\lvert Z \rvert} \end{pmatrix} = \begin{pmatrix} \frac{1}{h_{11}} & -\frac{h_{12}}{h_{11}} \\ \frac{h_{21}}{h_{11}} & \frac{\lvert H \rvert}{h_{11}} \end{pmatrix} = \begin{pmatrix} \frac{D}{B} & -\frac{\lvert T \rvert}{B} \\ -\frac{1}{B} & \frac{A}{B} \end{pmatrix}$
$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}$ $\lvert Z \rvert = z_{11}z_{22} - z_{12}z_{21}$	$\begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} = \begin{pmatrix} \frac{y_{22}}{\lvert Y \rvert} & -\frac{y_{12}}{\lvert Y \rvert} \\ -\frac{y_{21}}{\lvert Y \rvert} & \frac{y_{11}}{\lvert Y \rvert} \end{pmatrix} = \begin{pmatrix} \frac{\lvert H \rvert}{h_{22}} & \frac{h_{12}}{h_{22}} \\ -\frac{h_{21}}{h_{22}} & \frac{1}{h_{22}} \end{pmatrix} = \begin{pmatrix} \frac{A}{C} & -\frac{\lvert T \rvert}{C} \\ \frac{1}{C} & \frac{D}{C} \end{pmatrix}$
$\begin{pmatrix} V_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} I_1 \\ V_2 \end{pmatrix}$ $\lvert H \rvert = h_{11}h_{22} - h_{12}h_{21}$	$\begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} = \begin{pmatrix} \frac{1}{y_{11}} & -\frac{y_{12}}{y_{11}} \\ \frac{y_{21}}{y_{11}} & \frac{\lvert Y \rvert}{y_{11}} \end{pmatrix} = \begin{pmatrix} \frac{\lvert Z \rvert}{z_{22}} & \frac{z_{12}}{z_{22}} \\ -\frac{z_{21}}{z_{22}} & \frac{1}{z_{22}} \end{pmatrix} = \begin{pmatrix} \frac{B}{D} & \frac{\lvert T \rvert}{D} \\ -\frac{1}{D} & \frac{C}{D} \end{pmatrix}$
$\begin{pmatrix} V_1 \\ I_1 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_2 \\ -I_2 \end{pmatrix}$ $\lvert T \rvert = AD - BC$	$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} -\frac{y_{22}}{y_{21}} & -\frac{1}{y_{21}} \\ -\frac{\lvert Y \rvert}{y_{21}} & -\frac{y_{11}}{y_{21}} \end{pmatrix} = \begin{pmatrix} \frac{z_{11}}{z_{21}} & \frac{\lvert Z \rvert}{z_{21}} \\ \frac{1}{z_{21}} & \frac{z_{22}}{z_{21}} \end{pmatrix} = \begin{pmatrix} -\frac{\lvert H \rvert}{h_{21}} & -\frac{h_{11}}{h_{21}} \\ -\frac{h_{22}}{h_{21}} & -\frac{1}{h_{21}} \end{pmatrix}$

In row 2 of Table 20.1, the port currents are selected to be the independent variables. Now the elements of the matrix have units of impedance. They are denoted using the letter z to suggest impedance and are called the "z parameters" or the "impedance parameters" of the two-port network.

In row 3 of Table 20.1, I_1 and V_2 are the independent variables. The elements of the matrix do not have the same units. In this sense, this is a hybrid matrix. They are denoted using the letter h to suggest hybrid and are called the "h parameters" or the "hybrid parameters" of the two-port network. Hybrid parameters are frequently used to represent bipolar transistors.

In the last row of Table 20.1, the independent signals are V_2 and $-I_2$. In this case, the two-port parameters are called "transmission parameters" or "*ABCD* parameters." They are convenient when two-port networks are connected in cascade.

Next, consider the problem of calculating or measuring the parameters of a two-port circuit. Equation 20.12 suggests a procedure for calculating the y parameters of a two-port circuit. The first row of Equation 20.12 is

$$I_1 = y_{11}V_1 + y_{12}V_2 \tag{20.13}$$

Setting $V_1 = 0$ leads to

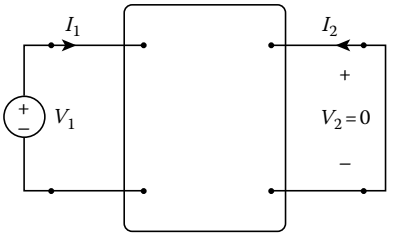
$$y_{12} = \frac{I_1}{V_2} \quad \text{when} \quad V_1 = 0 \tag{20.14}$$

This equation describes a procedure that can be used to measure or calculate y_{12}. A short circuit is connected to port 1 to set $V_1 = 0$. A voltage source having voltage V_2 is connected across port 2 and the current, I_1, in the short circuit is calculated. Finally, y_{12} is calculated as the ratio of I_1 to V_2.

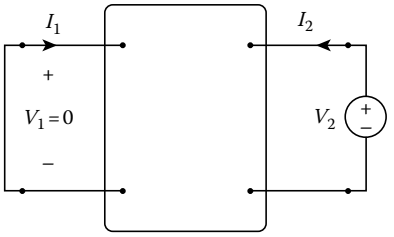
Similar procedures can be used to calculate any of the y, z, h, or transmission parameters. Table 20.2 tabulates these procedures.

As an example of how Table 20.2 can be used, consider calculating the y parameters of the two-port circuit shown in Figure 20.3. (Recall that $C_1 = 1$ F, $C_4 = 2$ F, $R_1 = 1/2\ \Omega$, $R_2 = 1/4\ \Omega$, and $R_3 = 1\ \Omega$.) According to Table 20.2, two cases will have to be considered. In the first, a voltage source is connected

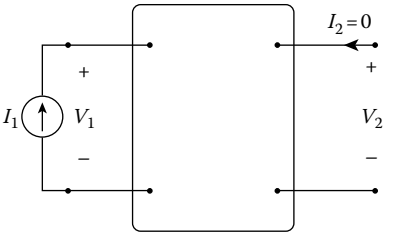
TABLE 20.2 Calculation of Two-Port Parameters



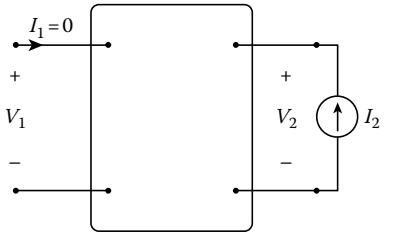
$y_{11} = \frac{I_1}{V_1}$ and $y_{21} = \frac{I_2}{V_1}$ when $V_2 = 0$



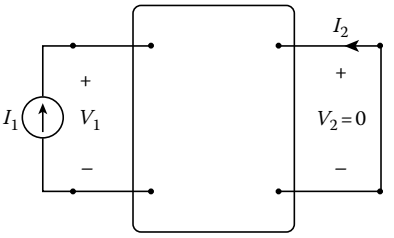
$y_{12} = \frac{I_1}{V_2}$ and $y_{22} = \frac{I_2}{V_2}$ when $V_1 = 0$



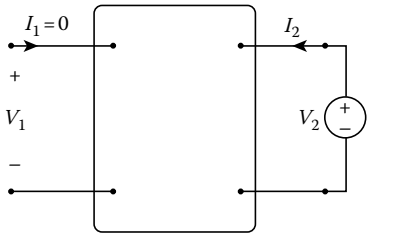
$z_{11} = \frac{V_1}{I_1}$ and $z_{21} = \frac{V_2}{I_1}$ when $I_2 = 0$



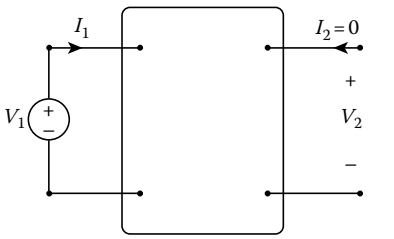
$z_{12} = \frac{V_1}{I_2}$ and $y_{22} = \frac{V_2}{I_1}$ when $I_1 = 0$



$h_{11} = \frac{V_1}{I_1}$ and $h_{21} = \frac{I_2}{I_1}$ when $V_2 = 0$

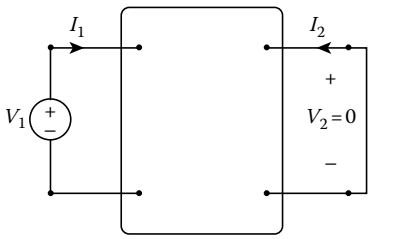


$h_{12} = \frac{V_1}{V_2}$ and $h_{22} = \frac{I_2}{V_2}$ when $I_1 = 0$



$A = \frac{V_1}{V_2}$ when $I_2 = 0$

$C = \frac{I_1}{V_2}$ when $I_2 = 0$



$B = \frac{V_1}{-I_2}$ when $V_2 = 0$

$D = \frac{I_1}{-I_2}$ when $V_2 = 0$

to port 1 and a short circuit is connected to port 2. In the second, a short circuit is connected across port 1 and a voltage source is connected to port 2. The resulting circuits are shown in Figure 20.5. In Figure 20.5a

$$I_1 = \frac{V_1}{R_1 + \frac{1}{C_1 s}} + \frac{V_1}{R_2 + R_3} = \frac{14s + 8}{5s + 10} V_1 = y_{11} V_1 \tag{20.15}$$

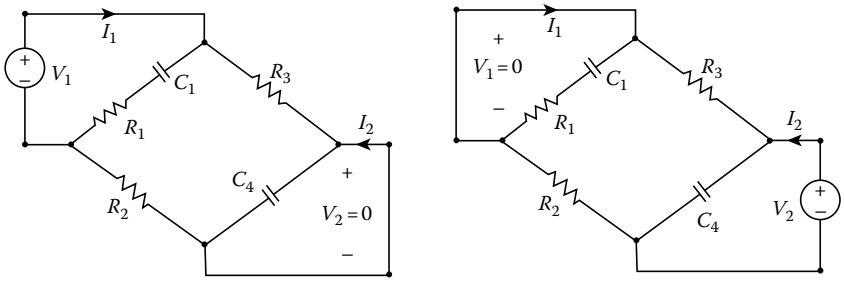


FIGURE 20.5 Test circuits used to calculate the *y* parameters of the example circuit.

and

$$I_2 = -\frac{V_1}{R_2 + R_3} = -\frac{4}{5}V_1 = y_{21}V_1 \tag{20.16}$$

In Figure 20.5b

$$I_1 = -\frac{V_2}{R_2 + R_3} = -\frac{4}{5}V_2 = y_{12}V_2 \tag{20.17}$$

and

$$I_2 = \frac{V_2}{\frac{1}{C_4 s}} + \frac{V_2}{R_2 + R_3} = \frac{10s + 4}{5}V_2 = y_{22}V_2 \tag{20.18}$$

Finally, the two-port network is represented by

$$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} \dfrac{14s + 8}{5s + 10} & -\frac{4}{5} \\ -\frac{4}{5} & \dfrac{10s + 4}{5} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \tag{20.19}$$

To continue the example, row 3 of Table 20.1 illustrates how to convert these admittance parameters to hybrid parameters. From Table 20.1,

$$|Y| = \frac{14s + 8}{5s + 10}\left[\frac{10s + 4}{5}\right] - \left(-\frac{4}{5}\right)\left(-\frac{4}{5}\right) = \frac{28s^2 + 24s}{5(s + 2)} \tag{20.20}$$

$$h_{11} = \frac{1}{y_{11}} = \frac{5s + 10}{14s + 8} \tag{20.21}$$

$$h_{12} = -\frac{y_{12}}{y_{11}} = \frac{2s + 4}{7s + 4} \tag{20.22}$$

$$h_{21} = \frac{y_{21}}{y_{11}} = -\frac{2s + 4}{7s + 4} \tag{20.23}$$

$$h_{22} = \frac{|Y|}{y_{11}} = \frac{(14s + 12)s}{7s + 4} \tag{20.24}$$

To complete the example, notice that Table 20.2 shows how to calculate the hybrid parameters directly from the circuit. According to Table 20.2, h_{22} is calculated by connecting an open circuit across port 1 and a voltage source having voltage V_2 across port 2. The resulting circuit is depicted in Figure 20.6. Now, h_{22} is determined from Figure 20.6b by calculating the port current I_2.

$$I_2 = \frac{V_2}{R_1 + R_2 + R_3 + \frac{1}{C_1 s}} + \frac{V_2}{\frac{1}{C_4 s}} = \frac{14s^2 + 12s}{7s + 4} V_2 = h_{22} V_2 \tag{20.25}$$

Of course, this is the same expression as was calculated earlier from the y parameters of the circuit.

Next, consider the problem of analyzing a circuit consisting of a two-port network and some external circuitry. Figure 20.7 depicts such a circuit. The currents in the resistors R_s and R_L are given by

$$I_1 = \frac{V_s - V_1}{R_s} \quad \text{and} \quad I_2 = -\frac{V_2}{R_L} \tag{20.26}$$

Suppose the two-port network used in Figure 20.7 is the circuit shown in Figure 20.4 and represented by y parameters in Equation 20.19. Combining the previous expressions for I_1 and I_2 with Equation 20.19 yields

$$\begin{pmatrix} \frac{V_s}{R_s} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{14s+8}{5s+10} + \frac{1}{R_s} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{10s+4}{5} + \frac{1}{R_L} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \tag{20.27}$$

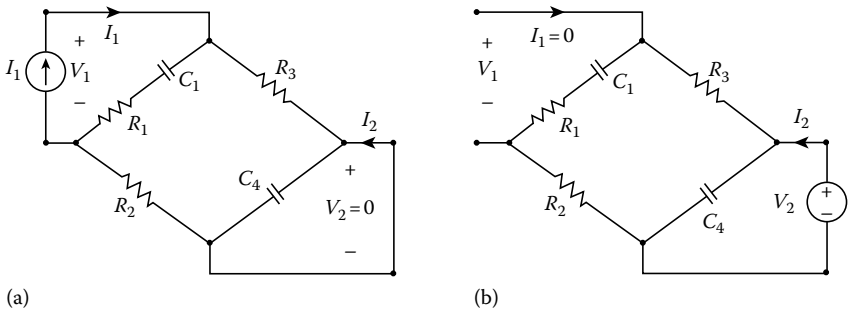


FIGURE 20.6 Test circuits used to calculate the h parameters of the example circuit.

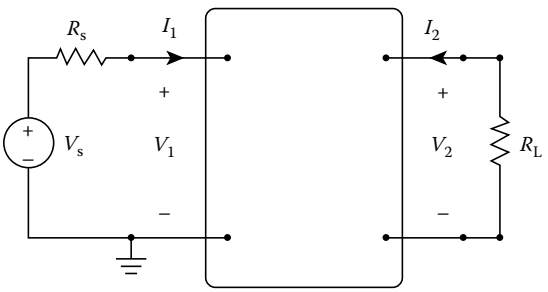


FIGURE 20.7 Application of the two-port network.

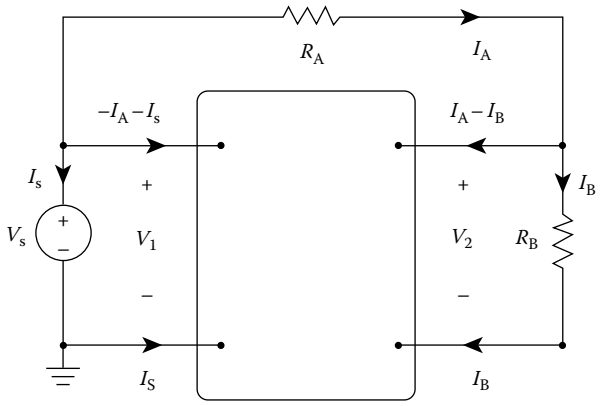


FIGURE 20.8 Incorrect application of the two-port representation.

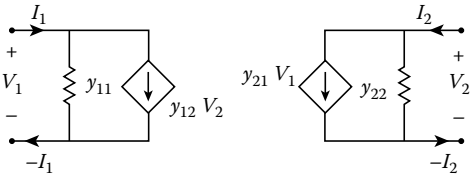
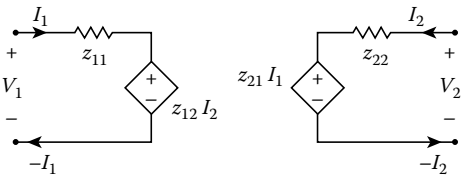
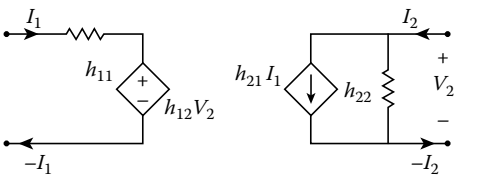
This equation can then be solved, e.g., using Cramer's rule, for the transfer function

$$\frac{V_2}{V_s}(s) = \frac{-\left(-\frac{4}{5}\right)\left(\frac{1}{R_s}\right)}{\left(\frac{14s+8}{5s+10} + \frac{1}{R_s}\right)\left(\frac{10s+4}{5} + \frac{1}{R_L}\right) - \left(-\frac{4}{5}\right)^2} = \frac{4(s+2)R_L}{(28s^2+24s)R_L R_s + (10s+4)(s+2)R_L + (14s+8)R_s + 5(s+2)} \tag{20.28}$$

Next consider Figure 20.8. This circuit illustrates a caution regarding use of the port convention. The use of ports assumes that the currents in the terminals comprising a port are equal in magnitude and opposite in sign. This assumption is not satisfied in Figure 20.8 so port equations cannot be used to represent the four-terminal network.

Table 20.3 presents three circuit models for two-port networks. These three models are based on *y*, *z*, and *h* parameters, respectively. Such models are useful when analyzing circuits that contain subcircuits

TABLE 20.3 Models of Two-Port Networks

$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$	
$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}$	
$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \end{pmatrix}$	

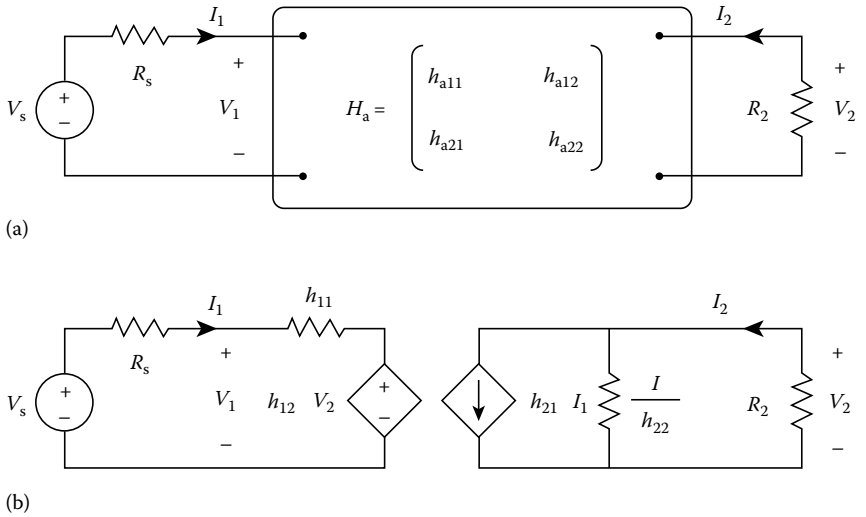


FIGURE 20.9 Application of the circuit model associated with *H* parameters.

that are represented by port parameters. As an example, consider the circuit shown in Figure 20.9a. This circuit contains a two-port network represented by *h* parameters. In Figure 20.9b the two-port network has been replaced by the model corresponding to *h* parameters. Analysis of Figure 20.9b yields

$$\begin{aligned} V_s &= (R_s + h_{11})I_1 + h_{12}V_2 \\ V_2 &= h_{21}\frac{-R_2}{R_2h_{22}+1}I_1 \end{aligned} \tag{20.29}$$

After some algebra,

$$\frac{V_2}{V_s} = \frac{h_{21}R_2}{h_{12}h_{21}R_2 - (h_{11}+R_2)(h_{22}R_2+1)} \tag{20.30}$$

The circuits in Figure 20.9a and b are equivalent, so this is the voltage gain of the circuit in Figure 20.9a.

H parameters are frequently used to describe bipolar transistors. Table 20.4 shows the popular methods for converting this three terminal device into a two-port network. The three configurations shown in Table 20.4 are called "common emitter," "common collector," and "common base" to indicate which terminal is common to both ports. Table 20.4 also presents the notation that is commonly used to name the *h* parameters when they are used to represent a transistor.

For example, h_{fe} is the forward gain when the transistor is connected in the common emitter configuration. Comparing with Table 20.3, it can be observed that $h_{fe} = h_{21}$.

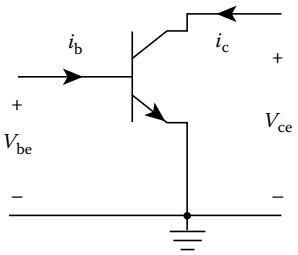
Figure 20.10 depicts a popular model of a bipolar transistor. Suppose the *h* parameters are calculated for the common emitter configuration. The result is

$$\begin{pmatrix} h_{ie} & h_{re} \\ h_{fe} & h_{oe} \end{pmatrix} = \begin{pmatrix} r_\pi & 0 \\ \beta & r_0 \end{pmatrix} \tag{20.31}$$

This calculation makes a connection between the parameters of the circuit model of the transistor and the *h* parameters used to describe the transistor, e.g., $h_{fe} = \beta$.

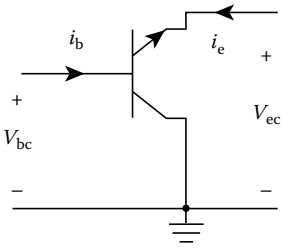
Figures 20.11 through 20.13 illustrate some common interconnections of two-port networks. In Figure 20.11, the two-port networks labeled A and B are connected in cascade. As illustrated in Figure 20.11, the

TABLE 20.4 Using *H* Parameters to Specify Bipolar Transistors



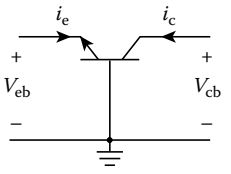
Common emitter

$$\begin{pmatrix} v_{be} \\ i_c \end{pmatrix} = \begin{pmatrix} h_{ie} & h_{re} \\ h_{fe} & h_{oe} \end{pmatrix} \begin{pmatrix} i_b \\ v_{ce} \end{pmatrix}$$



Common collector

$$\begin{pmatrix} v_{bc} \\ i_e \end{pmatrix} = \begin{pmatrix} h_{ic} & h_{rc} \\ h_{fc} & h_{oc} \end{pmatrix} \begin{pmatrix} i_b \\ v_{ec} \end{pmatrix}$$



Common base

$$\begin{pmatrix} v_{eb} \\ i_c \end{pmatrix} = \begin{pmatrix} h_{ib} & h_{rc} \\ h_{fb} & h_{oc} \end{pmatrix} \begin{pmatrix} i_b \\ v_{cb} \end{pmatrix}$$

Note: i, input impedance or admittance; o, output impedance or admittance; f, forward gain; r, reverse gain; e, common emitter; b, common base; c, common collector.

transmission matrix of the composite network is given by the product of the transmission matrices of the subnetworks. This simple relationship makes it convenient to use transmission parameters to represent a composite network that consists of the cascade of two subnetworks.

In Figure 20.12, the two-port networks labeled A and B are connected in parallel. As shown in Figure 20.12, the admittance matrix of the composite network is given by the sum of the admittance matrices of the subnetworks. This simple relationship makes it convenient to use admittance parameters to represent a composite network that consists of parallel subnetworks.

In Figure 20.13, the two-port networks labeled A and B are connected in series. As illustrated in Figure 20.13, the impedance matrix of the composite network is given by the sum of the impedance matrices of the subnetworks. This simple relationship makes it convenient to use impedance parameters to represent a composite network that consists of series subnetworks.

Figure 20.14 is a circuit that consists of three two-port networks. Two-port parameters representing the entire network can be calculated from the two-port parameters of the subnetworks. Let *c* denote the two-port network consisting of network *b* connected in parallel with network *a*. Represent network *c* with *y* parameters by converting the *h* parameters

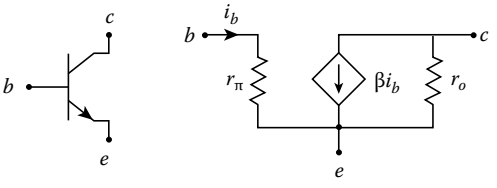
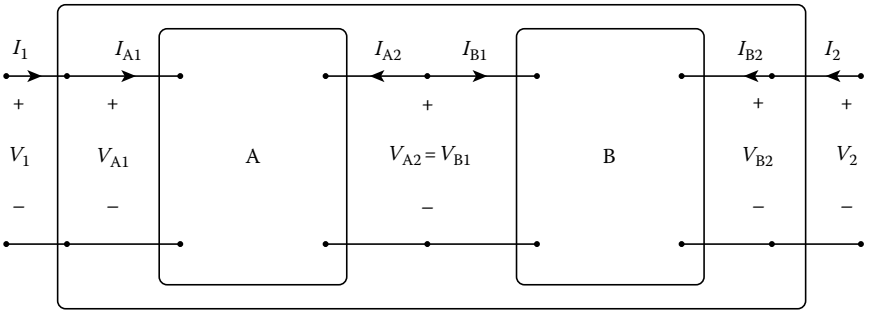
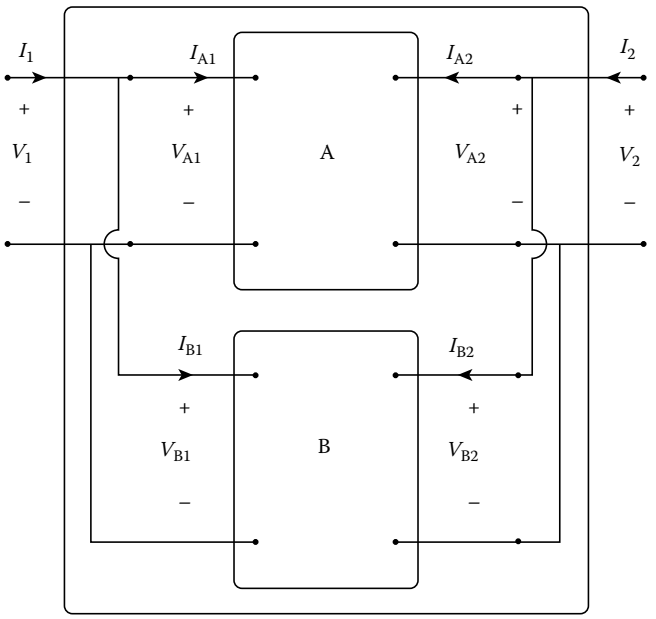


FIGURE 20.10 Hybrid pi model of a transistor.



$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} A_a & B_a \\ C_a & D_a \end{pmatrix} \begin{pmatrix} A_b & B_b \\ C_b & D_b \end{pmatrix}$$

FIGURE 20.11 Cascade connection of two-port networks.



$$\begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix} = \begin{pmatrix} y_{A11} & y_{A12} \\ y_{A21} & y_{A22} \end{pmatrix} + \begin{pmatrix} y_{B11} & y_{B12} \\ y_{B21} & y_{B22} \end{pmatrix}$$

FIGURE 20.12 Parallel connection of two-port networks.

representing network *a* to *y* parameters and adding these *y* parameters to the *y* parameters representing network *b*.

$$Y_c = \begin{pmatrix} \dfrac{1}{h_{a11}} + y_{b11} & -\dfrac{h_{a12}}{h_{a11}} + y_{b12} \\ \dfrac{h_{a21}}{h_{a11}} + y_{b21} & \dfrac{|H_a|}{h_{a11}} + y_{b22} \end{pmatrix} \triangleq \begin{pmatrix} y_{c11} & y_{c12} \\ y_{c21} & y_{c22} \end{pmatrix} \tag{20.32}$$

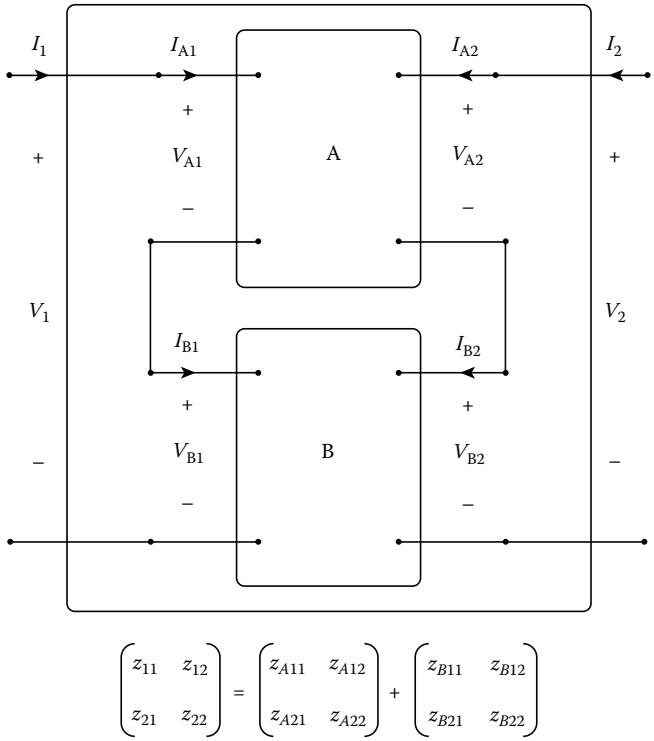


FIGURE 20.13 Series connection of two-port networks.

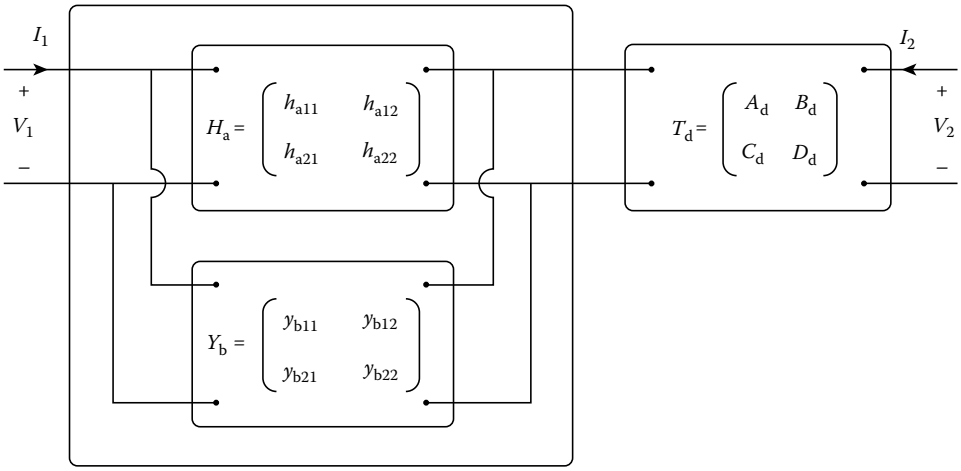


FIGURE 20.14 Circuit consisting of three two-port networks.

Next, network *c* is connected in cascade with network *d*. Represent network *d* with transmission parameters by converting the *y* parameters representing network *c* to transmission parameters and multiplying these transmissions parameters by the transmission parameters representing network *d*.

$$T = \begin{pmatrix} -\dfrac{y_{c22}}{y_{c21}} & -\dfrac{1}{y_{c21}} \\ -\dfrac{|Y_c|}{y_{c21}} & -\dfrac{y_{c11}}{y_{c21}} \end{pmatrix} \begin{pmatrix} A_d & B_d \\ C_d & D_d \end{pmatrix} \triangleq \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{20.33}$$

Finally, the circuit in Figure 20.14 is represented by

$$\begin{pmatrix} V_1 \\ I_1 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} V_2 \\ -I_2 \end{pmatrix} \tag{20.34}$$

20.4 Port Representations and Feedback Systems

Port representations of networks can be used to decompose a network into subnetworks. In this section, the decomposition will be done in such a way as to establish a connection between the network and a feedback system represented by the block diagram shown in Figure 20.15. Having established this connection, the theory of feedback systems can be applied to the network. Here, the problem of determining the relative stability of the network will be considered.

The theory of feedback systems [4] can be used to determine relative stability such as the one shown in Figure 20.15. The transfer function of this system is

$$T(\omega) = D(\omega) + \frac{C_1(\omega)A(\omega)C_2(\omega)}{1 + A(\omega)\beta(\omega)} \tag{20.35}$$

The phase and gain margins are parameters of a system that are used to measure relative stability. These parameters can be calculated from $A(\omega)$ and $\beta(\omega)$. To calculate the phase margin, first identify ω_m as the frequency at which

$$\begin{aligned} |A(\omega_m)||\beta(\omega_m)| = 1 &\Rightarrow |A(\omega_m)| = \frac{1}{|\beta(\omega_m)|} \\ &\Rightarrow 20\log_{10}|A(\omega_m)| = -20\log_{10}|\beta(\omega_m)| \end{aligned} \tag{20.36}$$

The phase margin is then given by

$$\begin{aligned} \phi_m &= 180^\circ + \angle(A(\omega_m)\,\beta(\omega_m)) \\ &= 180^\circ + (\angle A(\omega_m) + \angle\beta(\omega_m)) \end{aligned} \tag{20.37}$$

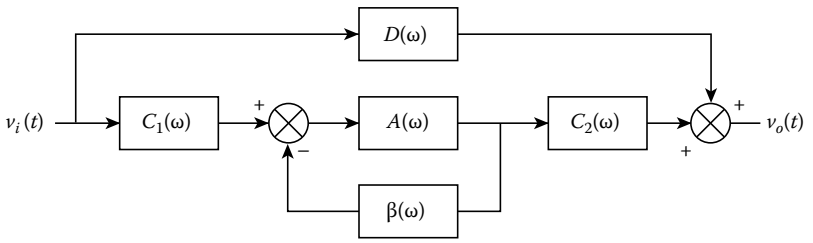


FIGURE 20.15 Feedback system.

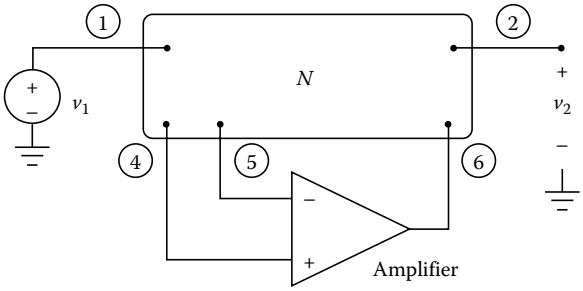


FIGURE 20.16 Identifying the network *N*.

The gain margin is calculated similarly. First, identify ω_p as the frequency at which

$$180^\circ = \angle\big(A(\omega_p)\beta(\omega_p)\big) \tag{20.38}$$

The gain margin is given by

$$\text{GM} = \frac{1}{|A(\omega_p)||\beta(\omega_p)|} \tag{20.39}$$

Next, consider an active circuit which can be represented as shown in Figure 20.16. For convenience, it is assumed that the input and output of the circuit are both node voltages. An amplifier has been selected and separated from the rest of the circuit. The rest of the circuit has been denoted as *N*.

Suppose that the amplifier is a voltage-controlled voltage source (VCVS), e.g., an inverting or a noninverting amplifier or an op-amp. Replacing the VCVS by a simple model yields the circuit shown in Figure 20.17. (The VCVS model accounts for input and output resistance and frequency-dependent gain.) Figure 20.17 shows how to identify the network *N*β which consists of the network *N* from Figure 20.16 together with the input and output resistances of the VCVS. The network *N*β has been called the "Beta Network" [5].

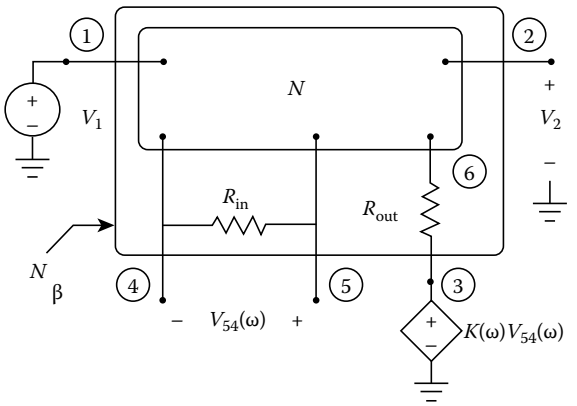


FIGURE 20.17 Identifying the Beta Network, *N*β.

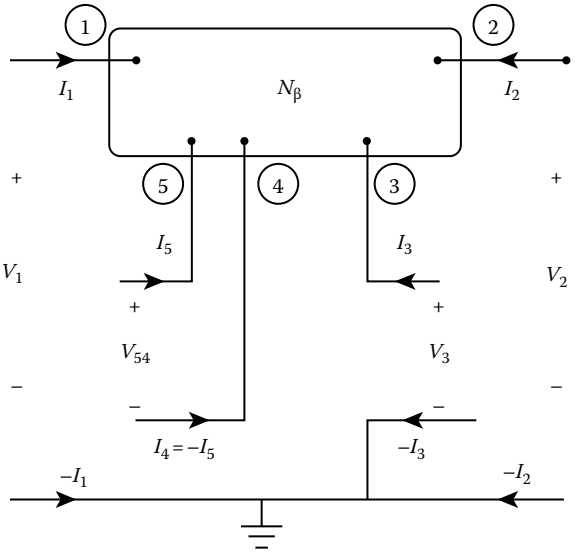


FIGURE 20.18 Identifying the ports of the Beta Network.

Figure 20.18 illustrates a way of grouping the terminals of the five-terminal network $N\beta$ to obtain a four-port network. This four-port network can be represented by the hybrid equation

$$\begin{pmatrix} I_1 \\ V_2 \\ I_3 \\ V_{54} \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{pmatrix} \begin{pmatrix} V_1 \\ I_2 \\ V_3 \\ I_5 \end{pmatrix} \tag{20.40}$$

Because I_1 and I_3 are not of interest, the first and third rows of this equation can be set aside. Then setting I_2 and I_5 equal to zero yields

$$\begin{pmatrix} V_2 \\ V_{54} \end{pmatrix} = \begin{pmatrix} H_{21} & H_{23} \\ H_{41} & H_{43} \end{pmatrix} \begin{pmatrix} V_1 \\ V_3 \end{pmatrix} \tag{20.41}$$

where, for example,

$$H_{43}(\omega) = \frac{V_{54}(\omega)}{V_3(\omega)} \quad \text{when} \quad V_1(\omega) = 0 \tag{20.42}$$

and $H_{21}(\omega)$, $H_{23}(\omega)$, and $H_{41}(\omega)$ are defined similarly.

The amplifier model requires that

$$V_3(\omega) = K(\omega)V_{54}(\omega) \tag{20.43}$$

Combining these equations yields

$$T(\omega) = \frac{V_2(\omega)}{V_1(\omega)} = H_{21}(\omega) + \frac{K(\omega)H_{23}(\omega)H_{41}(\omega)}{1 - K(\omega)H_{43}(\omega)} \tag{20.44}$$

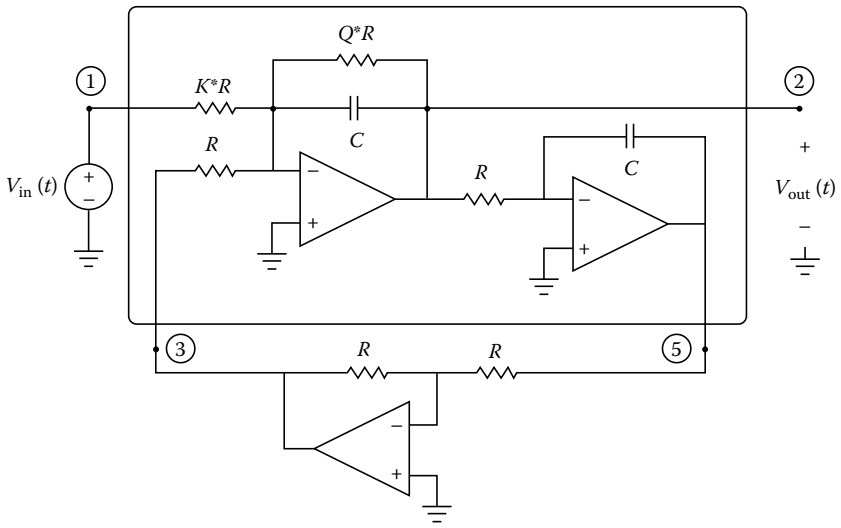


FIGURE 20.19 Tow–Thomas biquad.

Comparing this equation with the transfer function of the feedback system yields

$$\begin{aligned} A(\omega) &= -K(\omega) \\ \beta(\omega) &= H_{43}(\omega) \\ C_1(\omega) &= H_{23}(\omega) \\ C_2(\omega) &= H_{41}(\omega) \\ D(\omega) &= H_{21}(\omega) \end{aligned} \tag{20.45}$$

These equations establish a correspondence between the feedback system in Figure 20.15 and the active circuit in Figure 20.16. This correspondence can be used to identify $A(s)$ and $\beta(s)$ corresponding to a particular circuit. Once $A(s)$ and $\beta(s)$ are known, the phase or gain margin can be calculated.

Figure 20.19 shows a Tow–Thomas bandpass biquad [3]. When the op-amps are ideal devices the transfer function of this circuit is

$$T(s) = \frac{V_{in}(s)}{V_{out}(s)} = \frac{-\frac{1}{KRC}s}{s^2 + \frac{s}{QRC} + \frac{1}{R^2C^2}} \tag{20.46}$$

Figure 20.20 illustrates circuits that can be used to identify $A(s)$ and $\beta(s)$.

$$\begin{aligned} A(s) &= -1 \\ \beta(s) &= \frac{Q}{QR^2C^2s^2 + RCs} \end{aligned} \tag{20.47}$$

Next,

$$1 = |A(\omega_m)||\beta(\omega_m)| \Rightarrow (RC\omega_m)^2 = \frac{\frac{1}{Q^2} \pm \sqrt{\frac{1}{Q^4} + 4}}{2} \tag{20.48}$$

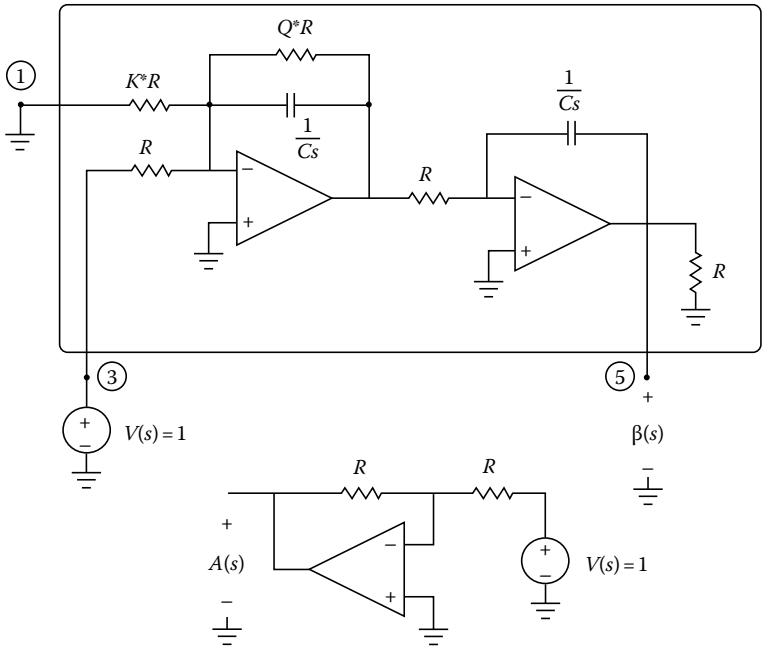


FIGURE 20.20 Calculating $A(s)$ and $\beta(s)$ for the Tow–Thomas biquad.

The phase margin is given by

$$\begin{aligned}\phi_m &= 180^\circ + \angle A(\omega_m) + \angle\beta(\omega_m)\\ &= 180^\circ + 180^\circ - \tan^{-1}\frac{-1}{QRC\omega_m}\\ &= \tan^{-1}\frac{1}{QRC\omega_m}\end{aligned} \tag{20.49}$$

When Q is large,

$$\omega_m \approx \frac{1}{RC} \Rightarrow \phi_m \approx \tan^{-1}\frac{1}{Q} \tag{20.50}$$

When the op-amps are not ideal, it is not practical to calculate the phase margin by hand. With computer-aided analysis, accurate amplifier models, such as macromodels, can easily be incorporated into this analysis [5].

20.5 Conclusions

It is frequently useful to decompose a large circuit into subcircuits. These subcircuits are connected together at ports and terminals. Port and terminal parameters describe how the subcircuits interact with other subcircuits but do not describe the inner workings of the subcircuit itself.

This chapter has presented procedures for determining port and terminal parameters and for analyzing networks consisting of subcircuits which are represented by port or terminal parameters. Port equations were used to establish a connection between electronic circuits and feedback systems.

References

1. W.-K. Chen, *Active Network and Feedback Amplifier Theory*, New York: McGraw-Hill, 1980.
2. R. C. Dorf and J. A. Svoboda, *Introduction to Electric Circuits*, New York: John Wiley & Sons, 2006.
3. L. P. Huelsman and P. E. Allen, *Theory and Design of Active Filters*, New York: McGraw-Hill, 1980.
4. R. C. Dorf, *Modern Control Systems*, Reading, MA: Addison-Wesley, 1989.
5. J. A. Svoboda and G. M. Wierzba, Using PSPICE to determine the relative stability of RC active filters, *International Journal of Electronics*, 74, 4, 593–604, 1993.

21

Signal Flow Graphs in Filter Analysis and Synthesis

21.1 Formulation of Signal Flow Graphs for Linear Networks **21**-1
21.2 Synthesis of Active Filters Based on Signal Flow Graph Associated with a Passive Filter Circuit **21**-3
21.3 Synthesis of Active Filters Based on Signal Flow Graph Associated with a Filter Transfer Function **21**-11
References **21**-21

Pen-Min Lin
Purdue University

21.1 Formulation of Signal Flow Graphs for Linear Networks

Any lumped network obeys three basic laws: Kirchhoff's voltage law (KVL), Kirchhoff's current law (KCL), and the elements' laws (branch characteristics). For filter applications, we write the frequency-domain instead of the time-domain network equations. Three general methods for writing network equations are described in Chapter 19. They are the node equations, loop equations, and hybrid equations. This section outlines another method, the signal flow graph (SFG) method of characterizing a linear network. The basic definitions of terms and theorems related to SFGs are presented in Section 8.4. The concepts of tree, cotree, loop, and cutset, required for the present discussion, are introduced in Chapter 7. Note that the terms loop, tree, and cotree refer to directed circuit, spanning tree, and cospanning tree defined in Chapter 7.

Consider first the construction of SFGs for linear networks without controlled sources. For all practical networks, the independent voltage sources (E) contain no loops, and the independent current sources (J) contain no cutsets. Under these conditions, it is always possible to select a tree T, such that all voltage sources are included in the tree and all current sources are included in the cotree. The network branches are divided into four sets (each set may be empty) indicated by subscripts as follows:

E: independent voltage sources
J: independent current sources
Z: passive branches in the tree, characterized by impedances
Y: passive branches in the cotree, characterized by admittances

A step-by-step procedure for constructing an SFG is given below.

Procedure 1 (for linear networks without controlled sources)

Step 1: Apply KVL to express each V_Y (voltage of a passive branch in the cotree) in terms of V_E and V_Z.

Step 2: Apply KCL to express each I_Z (current of a passive branch in the tree) in terms of I_J and I_Y.

Step 3: For each passive tree branch, consider its voltage as the product of impedance and current, i.e., $V_Z = Z_Z I_Z$.

Step 4: For each passive cotree branch, consider its current as the product of admittance and voltage, i.e., $I_Y = Y_Y V_Y$.

Example 21.1

Construct a SFG for the low-pass filter network shown in Figure 21.1a, and use Mason's formula to find the voltage gain function $H(s) = V_o(s)/V_i(s)$.

Solution

The graph associated with the network is shown in Figure 21.1b in which the branch numbers and reference directions (passive sign convention) have been assigned. The complexity of the SFG depends on the choice of the tree. In the case of a ladder network, a good tree to use is a star tree which has all tree branches connected to a common node. For the present network, we choose the tree to be $T = \{1, 2, 3, 4\}$, shown in heavy lines in Figure 21.1b.

Step 1 yields: $V_5 = V_i - V_2$, $V_6 = V_2 - V_3$, $V_7 = V_3 - V_4$

Step 2 yields: $I_2 = I_5 - I_6$, $I_3 = I_6 - I_7$, $I_4 = I_7$

$$V_2 = \frac{1}{0.31s} I_2$$

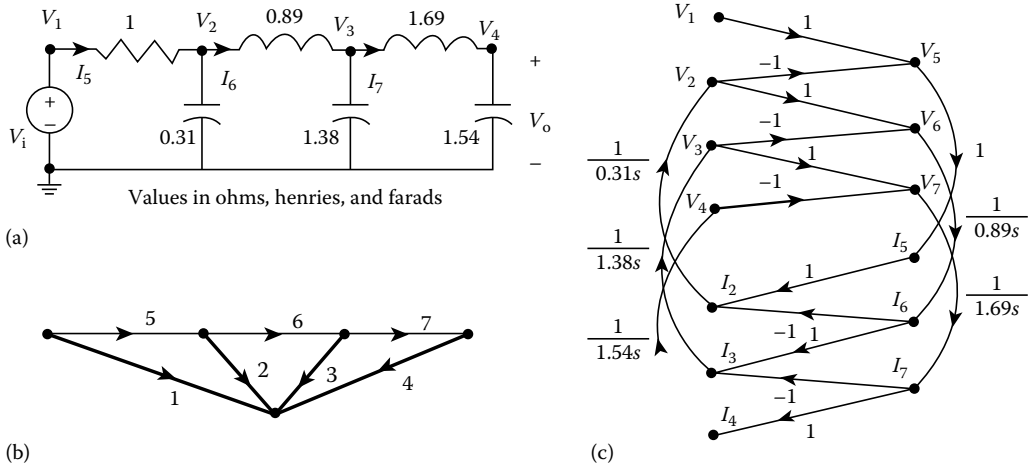


FIGURE 21.1 (a) A low-pass filter network. (b) Directed graph for the network and a chosen tree. (c) SFG based on the chosen tree.

$$\textbf{Step 3 yields: } V_3 = \frac{1}{1.38s} I_3$$
$$V_4 = \frac{1}{1.54s} I_4$$
$$I_5 = V_5$$
$$\textbf{Step 4 yields: } I_6 = \frac{1}{0.89s} V_6$$
$$I_7 = \frac{1}{1.69s} V_7$$

The SFG of Figure 21.1c displays all the preceding relationships.

Applying Mason's gain formula to the SFG of Figure 21.1c, we find

$$H(s) = \frac{V_o}{V_i} = \frac{V_4}{V_1} = \frac{1}{s^5 + 3.24s^4 + 5.24s^3 + 5.24s^2 + 3.24s + 1}$$

Next, consider linear networks containing controlled sources. All four types of controlled sources may be present. Our strategy is to utilize procedure 1 described previously with some preanalysis manipulations. The following is a step-by-step procedure.

Procedure 2 (for linear networks containing controlled sources)

Step 1: Temporarily replace each controlled voltage source by an independent voltage source, and each controlled current source by an independent current source, while retaining their original reference directions. The resultant network has no controlled sources.

Step 2: Construct the SFG for the network obtained in step 1 using procedure 1.

Step 3: Express the desired outputs and all controlling variables, if they are not present in the SFG, in terms of the quantities already present in the SFG.

Step 4: Reinstate the constraints of all controlled sources.

Example 21.2

Construct an SFG for the amplifier circuit depicted in Figure 21.2a.

Solution

We first replace the controlled voltage source μV_g by an independent voltage source V_x. A tree is chosen to be (V_s, R_a, V_x). The result of step 1 of procedure 2 is depicted in Figure 21.2b where dashed lines indicate cotree branches.

For the links R_b and R_c, we have $I_b = G_b V_b = G_b\ (V_s - V_a + V_x)$, and $I_c = G_c V_c = -G_c V_x$. For the tree branch R_a we have $V_a = R_a I_a = R_b I_b$. The result of step 2 of procedure 2 is depicted in Figure 21.2c. Note that the simple relationships $V_b = (V_s - V_a + V_x)$, $V_c = -V_x$ and $I_a = I_b$ have been used to eliminate the variables V_b, V_c and I_a. As a result, these variables do not appear in Figure 21.2c.

The desired output is $V_o = -V_x$ and the controlling voltage is $V_g = V_s - V_a$. After expressing these relationships in the SFG, step 3 of procedure 2 results in Figure 21.2d.

Finally, we reinstate the constraint of the controlled source, namely, $V_x = \mu V_g$. The result of step 4 of Procedure 21.2, in Figure 21.2e, is the desired SFG.

21.2 Synthesis of Active Filters Based on Signal Flow Graph Associated with a Passive Filter Circuit

The preceding section demonstrates that the equations governing a linear network can be described by an SFG in which the branch weights (or transmittances) are either real constants or simple expressions of the form K_s or K/s. All the cause–effect relationships displayed in such an SFG can, in turn, be

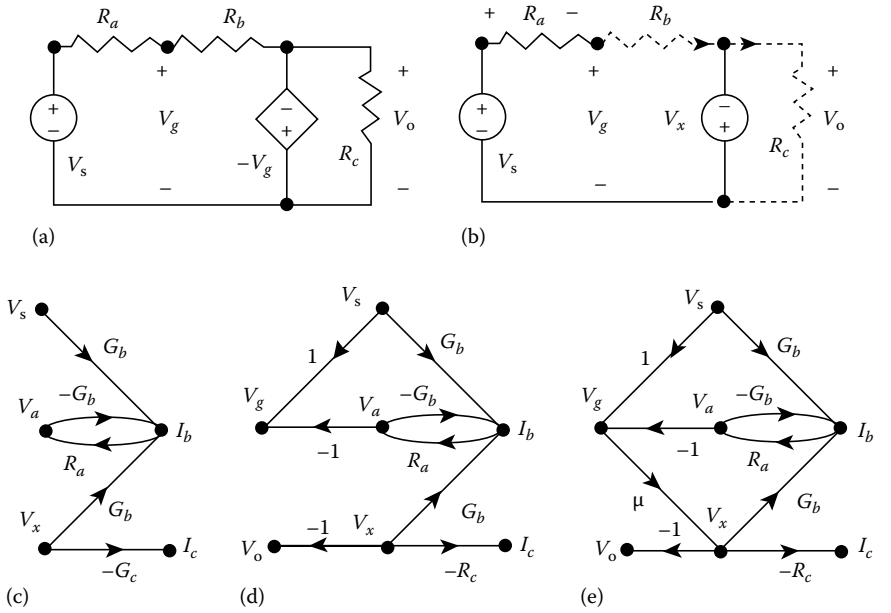


FIGURE 21.2 (a) A linear active network. (b) Result of step 1, Procedure 21.2. (c) Result of step 2, Procedure 21.2. (d) Result of step 3, Procedure 21.2. (e) The desired SFG.

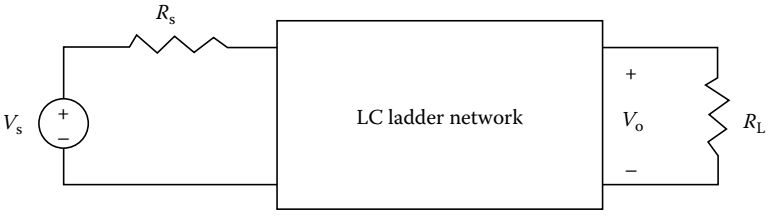


FIGURE 21.3 A doubly terminated passive filter.

implemented with resistors, capacitors, and ideal operational amplifiers. The inductors are not needed in the implementation. Whatever frequency response prevailing in the original linear circuit appears exactly in the RC-op-amp circuit.

In active filter synthesis, the method described in Section 21.1 is applied to a passive filter in the form of an *LC* (inductor–capacitor) ladder network terminated in a resistance at both ends as illustrated in Figure 21.3. The reason is that this type of filter, with $R_s = R_L$, has been proved to have the best sensitivity property [1, p. 196]. By this, we mean that the frequency response is least sensitive with respect to the changes in element values, when compared to other types of filter circuits. Because magnitude scaling (i.e., multiplying all impedances in the network by a factor K_m) does not affect the voltage gain function, we always normalize the prototype passive filter network so that the source resistance becomes 1 Ω. The advantage of this normalization will become evident in several examples given in this section.

The SFG illustrated in Figure 21.1c has many branches crossing each other. For a ladder network, with a proper choice of the tree and a rearrangement of the SFG nodes, all crossings can be eliminated. To achieve this, we first label a general ladder network as shown in Figure 21.4.

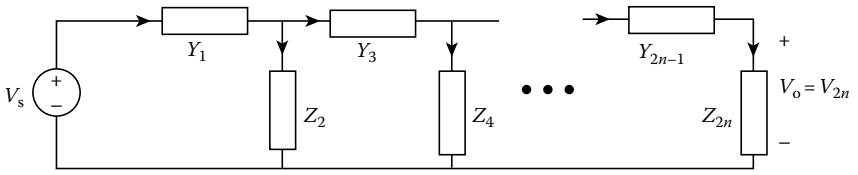


FIGURE 21.4 A general ladder network.

The following conventions are used in the labels of Figure 21.4:

1. All series branches are numbered odd and characterized by their admittances.
2. All shunt branches are numbered even and characterized by their impedances.
3. A single arrow is used to indicate the reference directions of both the voltage and the current of each network branch. Passive sign convention is used.
4. If the *LC* ladder in Figure 21.4 has a series element at the source end, then Y_1 represents that element in series with R_s.
5. If the *LC* ladder in Figure 21.4 has a shunt element at the load end, then Z_{2n} represents that element in parallel with R_L.

For constructing the SFG, choose a tree to consist of the voltage source and all shunt branches. The SFG for the circuit may be constructed using procedure 1 of Section 21.1. First, list the equations obtained in each step.

Step 1: $V_1 = V_s - V_2,\ V_3 = V_2 - V_4, \ldots,\ V_{2n-1} = V_{2n-2} - V_{2n}$
Step 2: $I_2 = I_1 - I_3,\ I_4 = I_3 - I_5, \ldots,\ I_{2n} = I_{2n-1}$
Step 3: $V_2 = Z_2 I_2,\ V_4 = Z_4 I_4, \ldots,\ V_{2n} = Z_{2n} I_{2n}$
Step 4: $I_1 = Y_1 V_1,\ I_3 = Y_3 V_3, \ldots,\ I_{2n-1} = Y_{2n-1} V_{2n-1}$

These relationships are represented by the SFG in Figure 21.5 for the case of a four-element ladder network. Note that the SFG graph nodes have been arranged in such a way that there are no branch crossings. The pattern displayed in this SFG suggests the children's game of "leapfrog." Consequently, an active filter synthesis based on the SFG of Figure 21.5 is called a leapfrog realization. The transmittance of each SFG branch indicates the type of mathematical operation performed. For example, $1/s$ means integration and is implemented by an op-amp integrator. Likewise, $1/(s+a)$ is implemented by a lossy op-amp integrator. It is well known that inverting integrators and inverting summers can be designed with singled-ended op-amps (i.e., the noninverting input terminal of each op-amp is grounded), [2–5]. Noninverting integrators and noninverting summers can also be designed, but require differential-input op-amps and more complex circuitry. Therefore, there is an advantage in using the inverting types. To this end, we multiply all Z's and Y's in Figure 21.5 by -1, with the result shown in Figure 21.6. Note that in Figure 21.6 we have removed the labels of internal SFG nodes because they are of no consequence in determining the transfer function. The transfer function V_o/V_s is the same for both Figures 21.5 and 21.6. This is quite obvious from Mason's gain formula, as all path weights and loop weights are not affected by

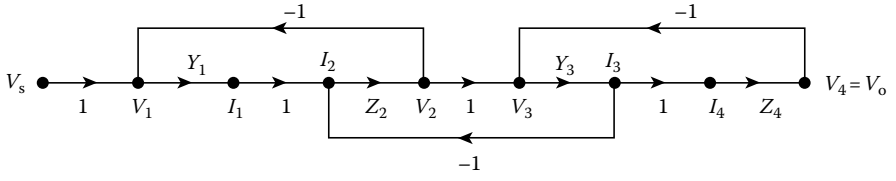


FIGURE 21.5 SFG for a four-element ladder network.

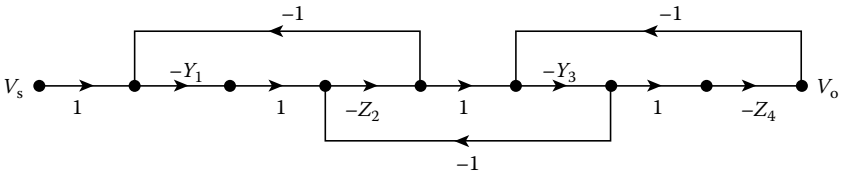


FIGURE 21.6 Inverting integrators are used in this modified SFG.

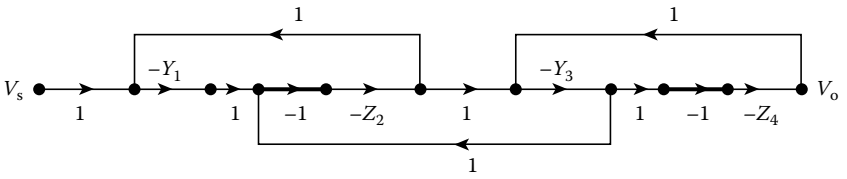


FIGURE 21.7 Modification to reduce the number of inverting amplifiers.

the modification. A branch transmittance of -1 indicates an inverting amplifier. In the interest of reducing the total number of op-amps used, we want to reduce the number of branches in the SFG that have weight -1. This can be achieved by inserting branches weighted -1 in some strategic places. Each insertion will lead to the change of the signs of one or two feedback branches. The rules are (a) inserting a branch weighted -1 in a forward path segment shared by two feedback loops changes the signs of the two feedback branch weights; (b) inserting a branch weighted -1 in a forward path segment belonging to one feedback loop only changes the signs of that feedback branch weight. Figure 21.6 is modified this way and the result is shown in Figure 21.7. The inserted branches are shown in heavy lines.

Comparing Figure 21.6 with Figure 21.7, we see that there is no change in path weights and loop weights. Therefore, Mason's gain formula assures that both SFG have the same transfer function. For a six-element ladder network, three branches weighted -1 must be inserted. This leads to a sign change of the single forward path weight in the SFG, and the output node variable now becomes $-V_o$. For filter applications this change of sign in the transfer function is acceptable as we are concerned mainly with the magnitude response.

An implementation of the SFG of Figure 21.7 may be accomplished easily by referring to Table 21.1 and picking the component op-amp circuits for realizing the SFG transmittances -1, $-Y_1$, $-Z_2$, etc. Figure 21.7 dictates how these component op-amp circuits are interconnected to produce the desired voltage gain function. An example will illustrate the procedure.

Example 21.3

Figure 21.8 shows a normalized Butterworth fourth-order, low-pass filter, where the 1 Ω source resistance has been included in Y_1, and the 1 Ω load resistance included in Z_4.

The leapfrog-type SFG for this circuit, after suitable modifications, is shown in Figure 21.7, where

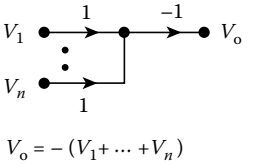
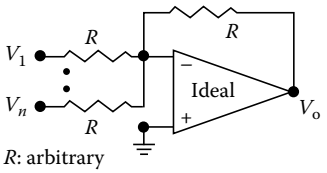
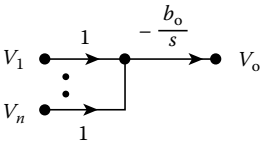
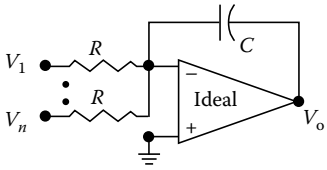
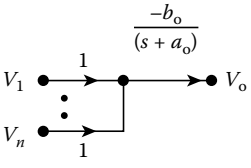
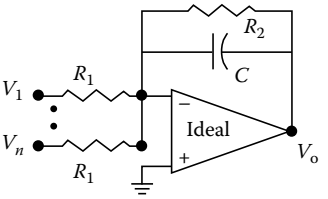
$$-Y_1 = -\frac{1}{0.7654s + 1}$$

$$-Z_2 = -\frac{1}{1.848s}$$

$$-Y_3 = -\frac{1}{1.848s}$$

$$-Z_4 = -\frac{1}{0.7654s + 1}$$

TABLE 21.1 Component Op-Amp Circuits for Synthesizing Active Low-Pass Filters by the Leapfrog Technique

(1)	 $V_o = -(V_1 + \cdots + V_n)$	Inverting summer  *R*: arbitrary
(2)	 $V_o = -\frac{b_o}{s}(V_1 + \cdots + V_n)$	Inverting summing integrator  *C*: arbitrary, $R = \frac{1}{b_o C}$
(3)	 $V_o = \frac{-b_o}{(s + a_o)}(V_1 + \cdots + V_n)$	Inverting summing lossy integrator  *C*: arbitrary, $R_1 = \frac{1}{b_o C}$, $R_2 = \frac{1}{a_o C}$

Note: Each component RC-op-amp circuit in the right column may be magnitude-scaled by an arbitrary factor.

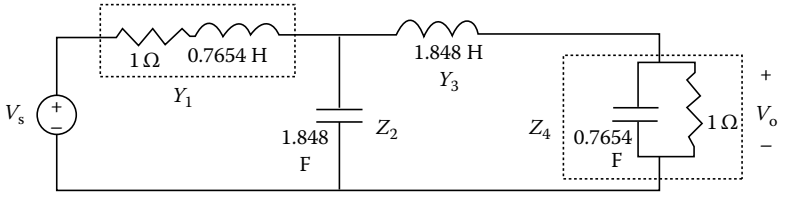


FIGURE 21.8 A fourth-order, Butterworth low-pass filter.

The SFG branch transmittance $-Z_2$ and $-Y_3$ are realized using item (2) of Table 21.1, while $-Y_1$ and $-Z_4$ use item (3). The two SFG branches with weight -1 in Figure 21.7 require item (1). The SFG branches with weight 1 merely indicate how to feed the inputs to each component network. No additional op-amps are needed for such SFG branches. Thus, a total of six op-amps are required. The interconnection of these component circuits is described by Figure 21.7. The complete circuit is shown in Figure 21.9. One-farad capacitances have been used in the circuit. Recall that the original passive low-pass filter has a 3 dB frequency of 1 rad/s. By suitable magnitude scaling and frequency scaling, all element values in the active filter of Figure 21.9 can be made practical. For example, if the 3 dB frequency

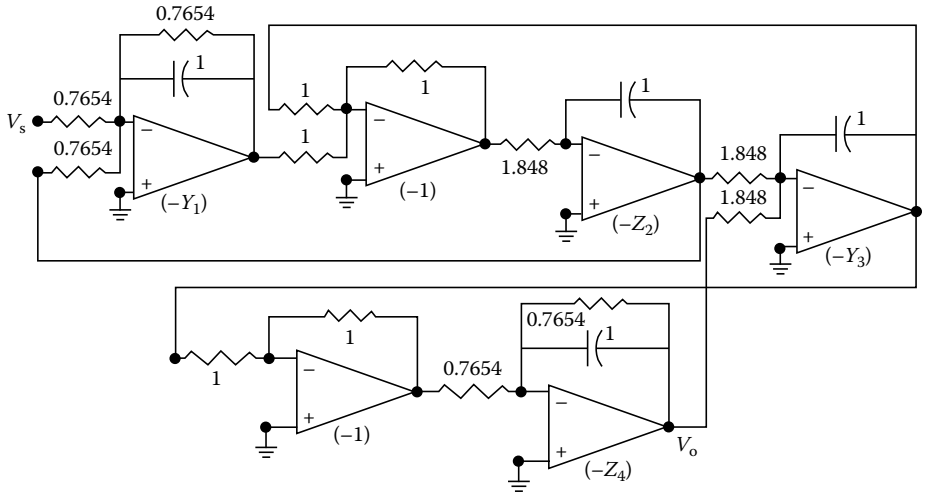


FIGURE 21.9 Leapfrog realization of passive filter of Figure 21.8. For the normalized case of $\omega_{3\ dB} = 1$ rad/s, values are in Ω and F. For a practical case of $\omega_{3\ dB} = 10^6$ rad/s, values are in kΩ and nF.

is changed to 10^6 rad/s, then the capacitances in Figure 21.9 are divided by 10^6. We may arbitrarily magnitude scale the resultant circuit by a factor of 10^3. Then, all resistances are multiplied by 10^3 and all capacitances are further divided by 10^3. The final circuit is still the one shown in Figure 21.9, but with resistances in kΩ and capacitance in nF. The parenthetical quantity beside each op-amp indicates the type of transfer function it produces.

If a doubly terminated passive filter has a shunt reactance at the source end and a series reactance the load end, then its dual network has a series reactance at the source end and a shunt reactance at the load end. The voltage gain functions of the original network and its dual differ at most by a multiplying constant. We can apply the method to the dual network.

For doubly terminated Butterworth and Chebyshev low-pass filters of odd orders, the passive filter either has series reactances or shunt reactances at both ends. The next example shows the additional SFG manipulations needed to construct the RC-op-amp circuit.

Example 21.4

Obtain a leapfrog realization of the third-order Butterworth low-pass filter shown in Figure 21.10a.

Solution

The network is again a four-element ladder network with a modified SFG in terms of the series admittances and shunt impedances as depicted in Figure 21.6. Note that the 1 Ω source resistance alone constitutes the element Y_1. Inserting a branch weighted -1 in front of $-Y_3$ changes the weights of two feedback branches from -1 to 1, and the output from V_o to $-V_o$. Figure 21.10b gives the result. Next, apply the node absorption rule to remove nodes V_A and V_B in Figure 21.10b. The result is Figure 21.10c.

Finally, we recognize that the left-most branch weight -1 is not contained in any loop weights, and appears in the single forward path weight. Therefore, if this branch weight is changed from -1 to 1, the output will be changed from $-V_o$ to V_o. When this change is made, and all specific branch weights are used, the final SFG is given in Figure 21.10d. The circuit implementation is now a simple matter of

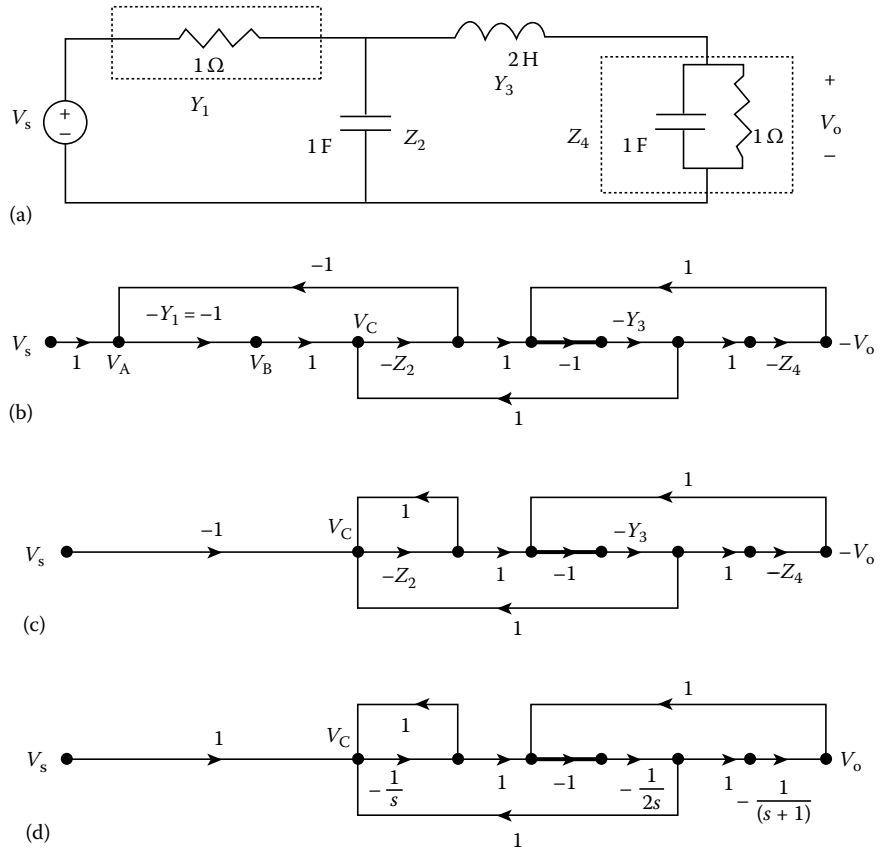


FIGURE 21.10 Leapfrog realization of a third-order, Butterworth low-pass filter. (a) The passive prototype. (b) Leapfrog SFG simulation. (c) Absorption of SFG nodes. (d) Final SFG for active filter realization.

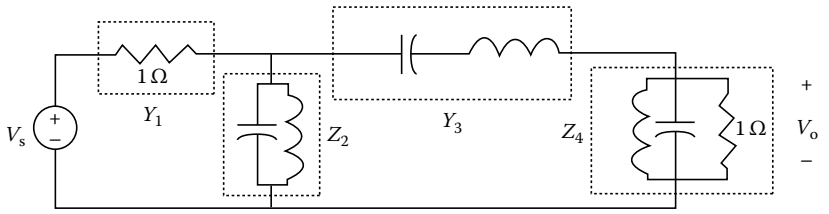


FIGURE 21.11 A bandpass passive filter derived from the circuit of Figure 21.10a.

picking component networks from Table 21.1 and connecting them as in Figure 21.10d. A total of four op-amps are required, one each for the branch transmittance $-1/s$, $-1/(2s)$, $-1/(s+1)$, and -1.

Passive bandpass filters may be derived from low-pass filters using the frequency transformation technique described in Chapter 3 of *Passive, Active and Digital Filters*. The configuration of a bandpass filter derived from the third-order Butterworth filter of Figure 21.10a is given in Figure 21.11.

The impedance and admittance functions Z_2, Y_3, and Z_4 are of the form

$$\frac{s}{a_2 s^2 + a_1 s + a_0}$$

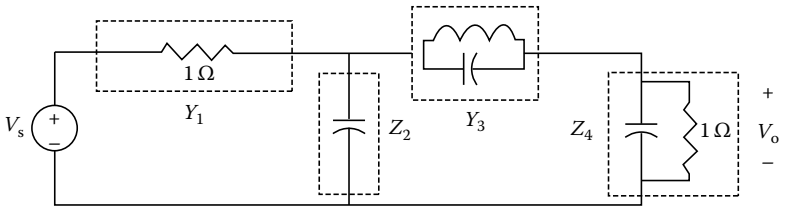


FIGURE 21.12 Network configuration of a doubly terminated filter having a third-order, elliptic or inverse Chebyshev low-pass response.

The SFG thus contains quadratic branch transmittances. Several single op-amp realizations of the quadratic transmittances are discussed in Chapter 14 of *Passive, Active and Digital Filters*, while some multiple op-amp realizations are presented in the next section. The interconnection of the component networks, however, is completely specified by an SFG similar to Figure 21.7 or Figure 21.10d. Complete design examples of this type of bandpass active filter may be found in many books [2–5].

The previous example shows the application of the leapfrog technique to low-pass and bandpass filters of the Butterworth or Chebyshev types. The technique, when applied to a low-pass filter having an elliptic response or an inverse Chebyshev response will require the use of some differentiators. The configuration of a third order low-pass elliptic filter or inverse Chebyshev filter is depicted in Figure 21.12. Notice that Y_3 has the expression

$$Y_3 = a_2 s + \frac{a_0}{s} = \frac{a_2 s^2 + a_0}{s}$$

The term $a_2 s$ in the voltage gain function of the component network clearly indicates the need of a differentiator. An example of such a design may be found in Ref. [1, pp. 382–385].

As a final point in the leapfrog technique, consider the problem of impedance normalization. In all the previous examples, the passive prototype filter has equal terminations and has been magnitude-scaled so that $R_s = R_L = 1$. Situations occur where the passive filter has unequal terminations. For example, the passive filter may have $R_s = 100\ \Omega$ and $R_L = 400\ \Omega$ in a four-element ladder network in Figure 21.8. Three possibilities will be considered.

1. No impedance normalization is done on the passive filter. Then,

$$-Y_1 = -\frac{1}{L_1 s + 100}$$
$$-Z_4 = -\frac{1}{C_4 s + \frac{1}{400}}$$

   From Table 21.1, the lossy integrator realizing $-Y_1$ has a resistance ratio of 100, and the resistance ratio for the $-Z_4$ circuit is 400. Such a large ratio is undesirable.
2. An impedance normalization is done with $R_o = 100\ \Omega$ so that R_s becomes 1 and R_L becomes 4. Then

$$-Y_1 = -\frac{1}{L_1 s + 1}$$
$$-Z_4 = -\frac{1}{C_4 s + \frac{1}{4}}$$

   The resistance ratio in the lossy integrator now becomes 1 for the $-Y_1$ circuit, and 4 for the $-Z_4$ circuit—an obvious improvement over the nonnormalized case.

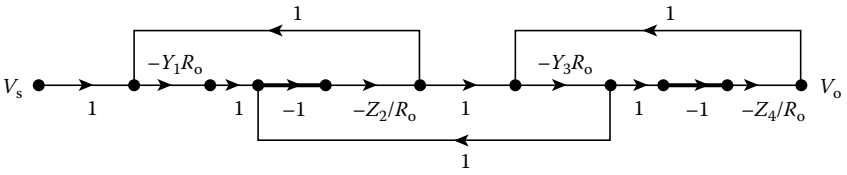


FIGURE 21.13 Result of normalization of the SFG of Figure 21.7.

3. An impedance normalization is done with $R_o = \sqrt{R_s R_L} = 200$. Then $R_s = 0.5$, $R_L = 2$, and

$$-Y_1 = -\frac{1}{L_1 s + 0.5}$$
$$-Z_4 = -\frac{1}{C_4 s + 0.5}$$

The resistance ratio in the lossy integrator is now 2 for both the $-Y_1$ circuit and the $-Z_4$ circuit—a further improvement over case (2) using $R_o = R_s$.

The conclusion is that, in the interest of reducing the spread of resistance values, the best choice of R_o for normalizing the passive filter is $R_o = \sqrt{R_s R_L}$. For the case of equal terminations, this choice leads to $R_s = R_L = 1$.

Instead of starting with a normalized passive filter, one can also construct a leapfrog-type SFG based on the unnormalized passive filter. For a four-element ladder network, the result is given in Figure 21.7. We now perform the following SFG manipulation, which has the same effect as the impedance normalization of the passive filter: Select a normalization resistance, R_o and divide all Z's in the SFG by R_o, and multiply all Y's by R_o. The resultant SFG is given in Figure 21.13.

It is easy to see that the SFG in both Figures 21.7 and 21.13 have the same loop weights and single forward path weight. Therefore, the voltage gain function remains unchanged with the normalization process. One advantage of using the normalized SFG is that the branch transmittances $Y_k R_o$ and Z_k/R_o are dimensionless, and truly represent voltage gain function of component op-amp circuits [2, p. 288].

21.3 Synthesis of Active Filters Based on Signal Flow Graph Associated with a Filter Transfer Function

The preceding section describes one application of the SFG in the synthesis of active filters. The starting point is a passive filter in the form of a doubly terminated *LC* ladder network. In this section, we describe another way of using the SFG technique to synthesize an active filter. The starting point in this case is a filter transfer function instead of a passive network.

Let the transfer voltage ratio function of a filter be

$$\frac{V_o}{V_i} = H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0}, \quad m \le n \tag{21.1}$$

In Chapters 2 and 3 of *Passive, Active, and Digital Filters*, it is demonstrated that, by properly selecting the coefficient a's and b's, all types of filter characteristics can be obtained: low-pass, high-pass, bandpass, band elimination, and all-pass. We assume that these coefficients have been determined. Our problem is how to realize the transfer function using SFG theory and RC-op-amp circuits.

Recall Mason's gain formula of an SFG described in Section 8.4. For the present application, we impose two constraints on the SFG:

1. No second-order or higher-order loops are present. In other words, all loops in the SFG touch each other.
2. Every forward path from the source node to the output node touches all loops.

For such a special kind of SFG, Mason's gain formula reduces to

$$\frac{V_o}{V_i} = H(s) = \frac{\sum P_k}{1 - (L_1 + L_2 + \cdots + L_n)} \tag{21.2}$$

where

L_n is the nth loop weight

P_k is the kth forward path weight

Summations are over all forward paths and all loops

Our strategy is to manipulate Equation 21.1 into the form of Equation 21.2, and then construct an SFG to have the desired loops and paths, meeting constraints (1) and (2). Integrators are preferred over differentiators in actual circuit implementation, therefore, we want $1/s$ instead of s to appear as the SFG branch transmittances. This suggests the division of both the numerator and denominator of Equation 21.1 by s^n, the highest degree term in the denominator.

The result is

$$\begin{aligned}\frac{V_o}{V_i} &= H(s) \\ &= \frac{b_m\left(\frac{1}{s}\right)^{n-m} + b_{m-1}s^{n-m+1} + \cdots + b_1\left(\frac{1}{s}\right)^{n-1} + b_0\left(\frac{1}{s}\right)^n}{1 + a_{n-1}\left(\frac{1}{s}\right) + \cdots + a_1\left(\frac{1}{s}\right)^{n-1} + a_0\left(\frac{1}{s}\right)^n}, \quad m \le n\end{aligned} \tag{21.3}$$

Comparing Equation 21.3 with Equation 21.2, we can identify the loop weights

$$\begin{aligned} L_1 &= -a_{n-1}(1/s) \\ L_2 &= -a_{n-2}(1/s)^2 \\ &\cdots \\ L_n &= -a_0(1/s)^n \end{aligned} \tag{21.4}$$

and the forward path weights

$$b_m\left(\frac{1}{s}\right)^{n-m},\ b_{m-1}\left(\frac{1}{s}\right)^{n-m+1},\ \ldots\ b_1\left(\frac{1}{s}\right)^{n-1},\ b_0\left(\frac{1}{s}\right)^n \tag{21.5}$$

Many SFGs may be constructed to have such loop and path weights, and the touching properties stated previously in (1) and (2). Two simple ones are given in Figure 21.14a and b for the case $n = m = 3$. The extension to higher-order transfer functions is obvious. In control theory, the system represented by Figure 21.14a is said to be of the controllable canonical form, and Figure 21.14b the observable canonical form. In a filter application, we need not be concerned about the controllability and observability of the system. The terms are used here merely for the purpose of circuit identification. Our major concern here is how to implement the SFG by an RC-op-amp circuit.

An SFG branch having transmittance $1/s$ indicates an integrator. If the terminating node of the $1/s$ branch has no other incoming branches (as in Figure 21.14a), then that node variable represents the output of the integrator. On the other hand, if $1/s$ is the transmittance of only one of several incoming branches incident at the node V_k (as in Figure 21.14b), then V_k is not the output of an integrator. In order

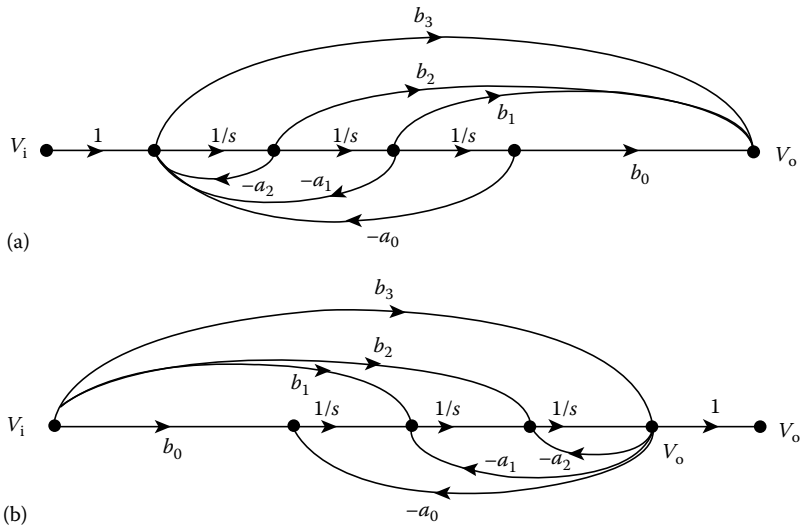


FIGURE 21.14 Two simple SFGs having a gain function given by Equation 21.3. (a) Controllable canonical form. (b) Observable canonical form.

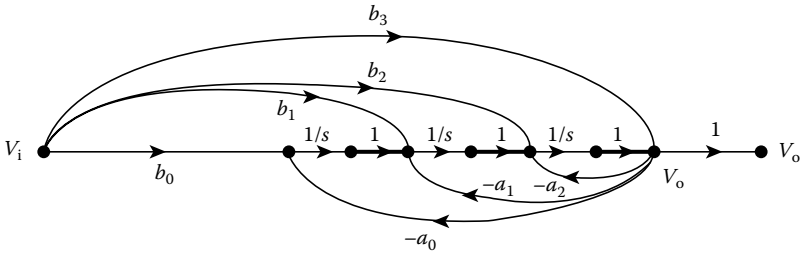


FIGURE 21.15 Insertion of dummy branches to identify integrator outputs.

to identify the integrator outputs clearly for the purpose of circuit interconnection, we insert some dummy branches with weight 1 in series with the branches weighted $1/s$. When this is done to Figure 21.14b, the result is Figure 21.15 with the inserted dummy branches shown in heavy lines. An SFG branch with weight $-1/s$ represents an inverting integrator. As pointed out in Section 21.2, the circuitry of an inverting integrator is simpler than that of a noninverting integrator. To have an implementation utilizing inverting integrators, we replace all SFG branch weights $1/s$ in Figure 21.14 by $-1/s$. In order to maintain the original path and loop weights, the signs of some feedback branches and forward path branches must be changed accordingly. When this is done, Figures 21.14a and 21.15 become those shown in Figure 21.16a and b, respectively. Our next goal is to implement these SFGs by RC-op-amp circuits. Because SFGs of the kind described in this section are widely used in the study of linear systems by the state variable approach, the active filters based such SFGs are called state variable filters [6].

Example 21.5

Synthesize a state variable active filter to have a third order Butterworth low-pass response having 3 dB frequency $\omega_o = 10^6$ rad/s. All op-amps used are single-ended.

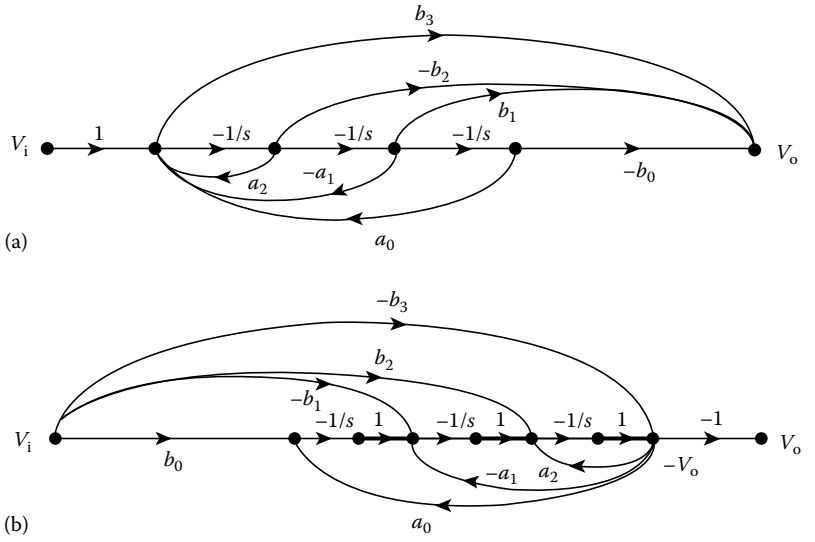


FIGURE 21.16 Simulation of $H(s)$ by an SFG containing inverting integrators.

Solution

As usual in filter synthesis, we first construct the filter for the normalized case, i.e., $\omega_o = 1$ rad/s, and then perform frequency scaling to obtain the required filter. From Chapter 2 of *Passive, Active, and Digital Filters*, the normalized voltage gain function of the filter is

$$H(s) = \frac{V_o}{V_i} = \frac{1}{s^3 + 2s^2 + 2s + 1} \tag{21.6}$$

and the two SFGs in Figure 21.16 become those depicted in Figure 21.17.

Because we are concerned with the magnitude response only, $-V_o$ instead of V_o can be accepted as the desired output. Therefore, in Figure 21.17, the rightmost SFG branch with gain (-1) need not be implemented. The implementation of the SFG as RC-op-amp circuits is now just a matter of looking up Table 21.2, selecting proper component networks and connecting them as specified by Figure 21.17. The results are given in Figure 21.18a and b. These circuits, with element values in ohms and farads,

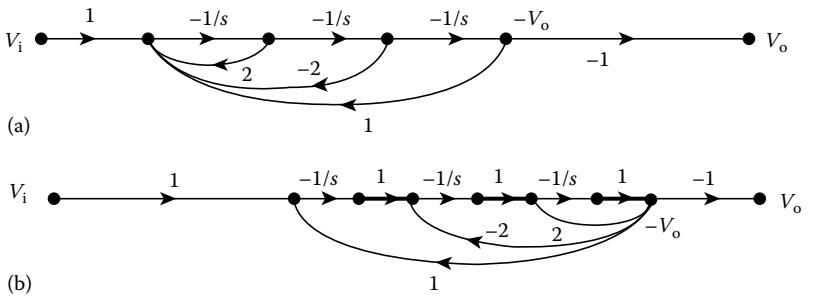
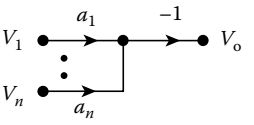
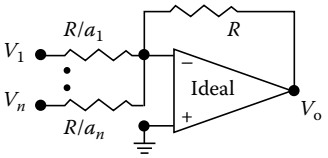
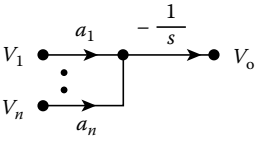
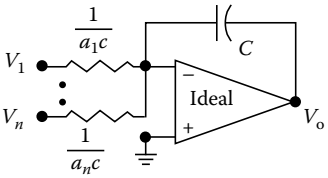
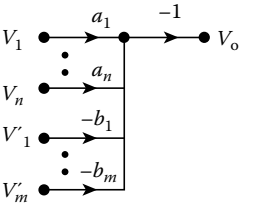
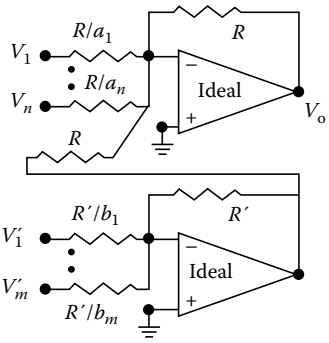


FIGURE 21.17 Two SFG representations of Equation 21.6.

TABLE 21.2 Single-Ended Op-Amp Circuits for Implementing State Variable Active Filters

	SFG	RC-op-amp circuit
(1)		Inverting scaled summer 
	$V_o = -(a_1V_1 + \cdots + a_nV_n)$	R: arbitrary
(2)		Inverting scaled summing integrator 
	$V_o = -\frac{1}{s}(a_1V_1 + \cdots + a_nV_n)$	C: arbitrary
(3)		Bi-polarity summer 
	$V_o = -(a_1V_1 + \cdots + a_nV_n) + (b_1V'_1 + \cdots + b_mV_n)$	R and R': arbitrary

realize the normalized transfer function having $\omega_c = 1$ rad/s. To meet the original specification of $\omega_c = 10^6$ rad/s, we frequency-scale the circuits by a factor 10^6 (i.e., divide all capacitances by 10^6). To have practical resistance values, we further magnitude-scale the circuits by a factor of, say, 1000. The resistances are multiplied by 1000, and the capacitances are further divided by 1000. The final circuits are still given by Figure 21.18, but now with element values in kΩ and nF.

In Example 21.5, both realizations require four op-amps. In general, for an nth order transfer function given by Equation 21.1 with all coefficients nonzero, a synthesis based on Figure 21.16a (controllable canonical form) requires $n + 3$ single-ended op-amps. The breakdown is as follows (refer to Figure 21.16a):

n inverting scaled integrators (item 2, Table 21.2) for the n SFG branches with weight $-1/s$
Two op-amps for the bipolarity summer (item 3, Table 21.2) to obtain V_o
One inverting scaled summer (item 1, Table 21.2) to invert and add up signals from branches with weights $-a_1$, $-a_3$, etc., before applying to the left-most integrator

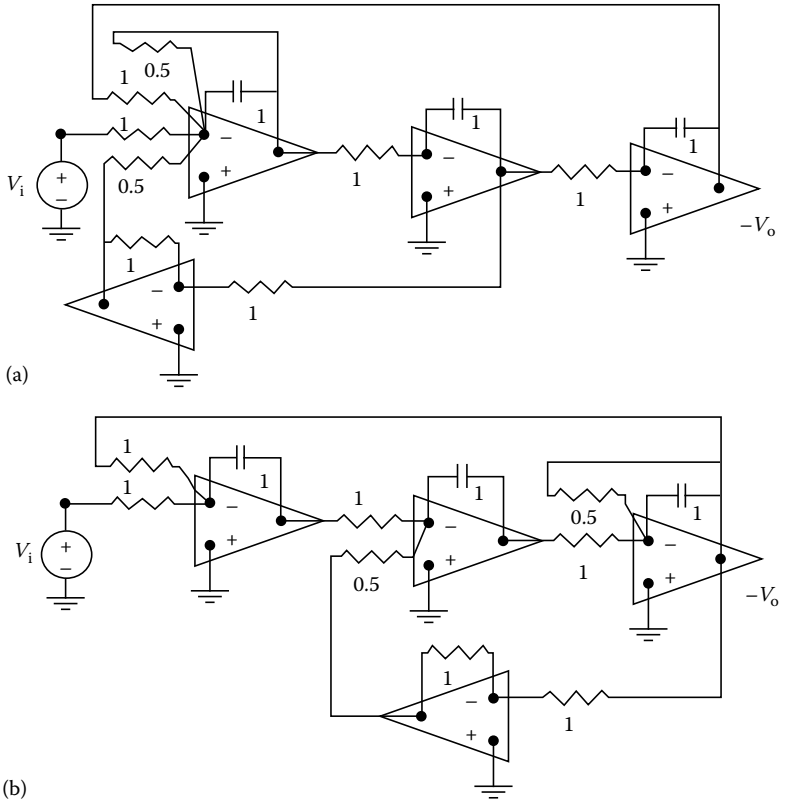


FIGURE 21.18 Two op-amp circuit realizations of $H(s)$ given by Equation 21.6.

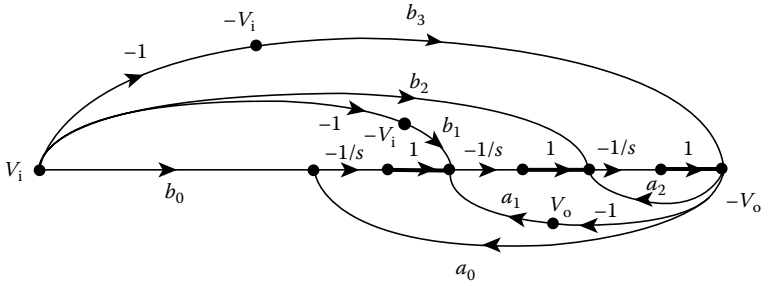


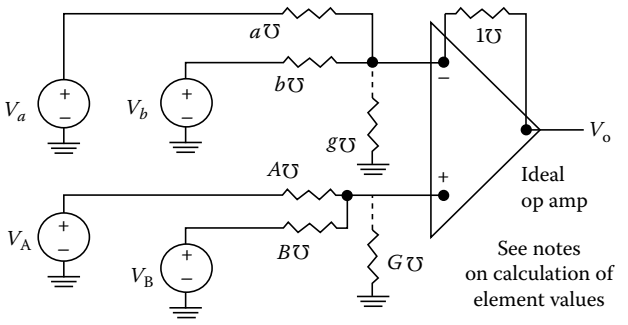
FIGURE 21.19 Modification of Figure 21.16b to use all positive a's and b's.

On the other hand, a synthesis based on Figure 21.16b (observable canonical form) requires only $n+2$ single-ended op-amps. To see this, we redraw Figure 21.16b as Figure 21.19 by inserting branches with weight -1, and making all literal coefficients positive. The breakdown is as follows (referring to Figure 21.19, extended to nth order $H(s)$):

n inverting scaled integrators (item 2, Table 21.2) for the n SFG branches with weight $-1/s$
One inverting amplifier at the input end to provide $-V_i$
One inverting amplifier at the output end to make available both V_o and $-V_o$

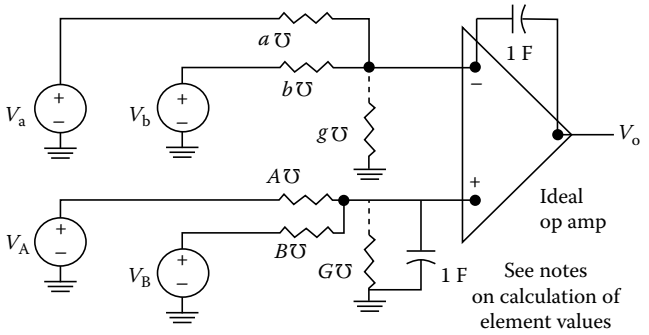
TABLE 21.3 Differential-Input Op-Amp Circuit

(1) Bi-polarity scaled summer $V_o = -(aV_a + bV_b) + (AV_A + BV_B)$



(2) Bi-polarity scaled summing integrator

$V_o = \frac{1}{s}[-(aV_a + bV_b) + (AV_A + BV_B)$



Note: Calculation of element values in Table 21.3 [7].

(i) The initial design uses 1 Ω resistance or 1 F capacitance as the feedback element.

(ii) Either the g mho conductance or the G mho conductance (not both) is connected.

Choose the values of g or G such that the sum of all conductances connected to the inverting input terminal equals the sum of all conductances connected to the noninverting input terminal.

(iii) Starting with the initial design, one may magnitude-scale all elements connected to the inverting input terminal by one factor, and all elements connected to the noninverting input terminal by the same or a different factor.

The number of op-amps can be reduced if the restriction of using single-ended op-amp is removed. Table 21.3 describes several differential-input op-amp circuits suitable for use in the state variable active filters.

If differential-input op-amps are used, then the number of op-amps required for the realization of Equation 21.1 (with $m = n$) is reduced to $(n+1)$ for Figure 21.16a and n for Figure 21.16b. The breakdowns are as follows:

For the controllable canonical form SFG of Figure 21.16a:

$n-1$ inverting integrators (item 2, Table 21.2 with one input) for the n SFG branches with weight $-1/s$, except the leftmost

One bipolarity-scaled summing integrator (item 2, Table 21.3) for the leftmost SFG branch with weight $-1/s$

One bipolarity-scaled summer (item 1, Table 21.3) to obtain V_o

For the observable canonical form SFG of Figure 21.16b:

n bipolarity scaled summing integrator (item 2, Table 21.3), one for each SFG branch with weight $-1/s$

To construct the op-amp circuit, one refers to the SFG of Figure 21.14 and obtains the expression relating the output of each op-amp to the outputs of other op-amps. After that is done, refer to Table 21.3, pick the appropriate component circuits, and connect them as specified by the SFG. The next example outlines the procedure of utilizing differential-input type op-amps to reduce the total number of op-amps to $(n+1)$ or n.

Example 21.6

Design a state-variable active low-pass filter to meet the following requirements: magnitude response is of the inverse Chebyshev type

$$\alpha_{max} = 0.5\,\text{dB}, \quad \alpha_{min} = 20\,\text{dB}, \quad \alpha(\omega_s) = \alpha_{min}$$
$$\omega_p = 333.33\,\text{rad/s}, \quad \omega_s = 1000\,\text{rad/s}$$

Solution

Using the method described in Chapter 3 of *Passive, Active, and Digital Filters*, the normalized transfer function (i.e., $\omega_s = 1$ rad/s) is found to be

$$H(s) = \frac{V_o}{V_i} = \frac{K(s^2 + 1.33333)}{s^3 + 1.40534s^2 + 0.94200s + 0.40196} \tag{21.7}$$

The SFGs for this $H(s)$ are simply obtained from Figure 21.16 by removing the two branches having weights b_3 and b_1.The results are shown in Figure 21.20a for the case $K = 1$, and in Figure 21.20b for the case $K = -1$. A four op-amp circuit for the normalized $H(s)$ may be constructed in accordance with the SFG of Figure 21.20a. The component op-amp circuits are selected from Tables 21.2 and 21.3 in the following manner:

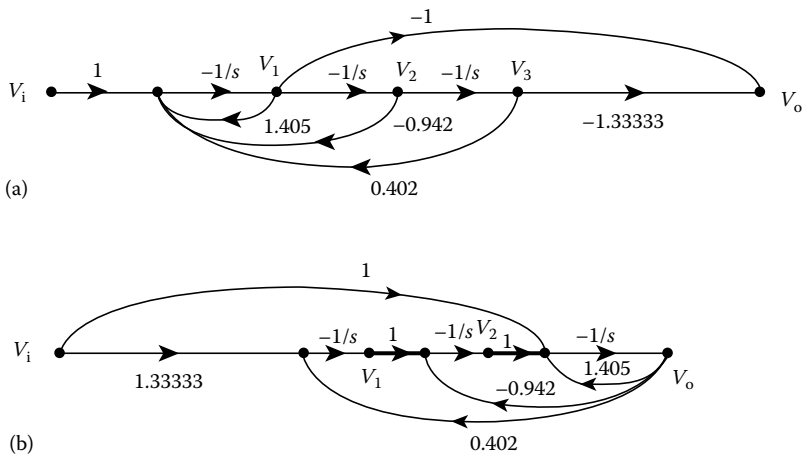


FIGURE 21.20 Two SFGs realizing the transfer function of Equation 21.7.

Relationship from SFG	component op-amp circuit
$V_1 = -\frac{1}{s}(V_i + 1.405V_1 + 0.402V_3 - 0.942V_2)$	item 2, Table 21.3
$V_2 = -\frac{1}{s}V_1$	item 2, Table 21.2
$V_3 = -\frac{1}{s}V_2$	item 2, Table 21.2
$V_o = -V_1 - 1.33333\ V_3$	item 1, Table 21.2

After connecting these four-component op-amp circuits as described in Figure 21.20a, and frequency-scaling the whole circuit by 1000, and magnitude-scaling by 1000, the final op-amp circuit meeting the low-pass filter specifications is shown in Figure 21.21a.

In a similar manner, a three-op-amp circuit for the normalized $H(s)$ may be constructed in accordance with the SFG of Figure 21.20b. The final op-amp circuit meeting the lowpass filter specifications is shown in Figure 21.21b. Both circuits in Figure 21.21 achieve a gain constant $|K| = 1$ in Equation 21.7. Should a different value of $|K| = 1/\alpha$ be desired, it is only necessary to multiply the values of all resistors connected to the input V_i by α.

When the method of this section is applied to a second order transfer function, the resultant op-amp circuit is called a "state variable biquad." Biquads and first order op-amp circuits are used as the basic building blocks in the synthesis of a general nth order transfer function by the "cascade" approach. Single op-amp biquads are discussed in Section II of *Passive, Active, and Digital Filters*. Depending on the SFGs chosen and the types of op-amps allowed (single-ended or differential-input), a state variable biquad may require from two to five op-amps. Some special but useful state variable biquads are listed in Table 21.4 for reference purposes.

All the SFGs used in the previous examples are of the two types (controllable and observable canonical forms) illustrated in Figure 21.16; however, many other possible SFGs produce the same transfer function. For example, a third-order, low-pass Butterworth or Chebyshev filter has an all-pole transfer function.

$$H(s) = \frac{V_o}{V_i} = \frac{K}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{21.8}$$

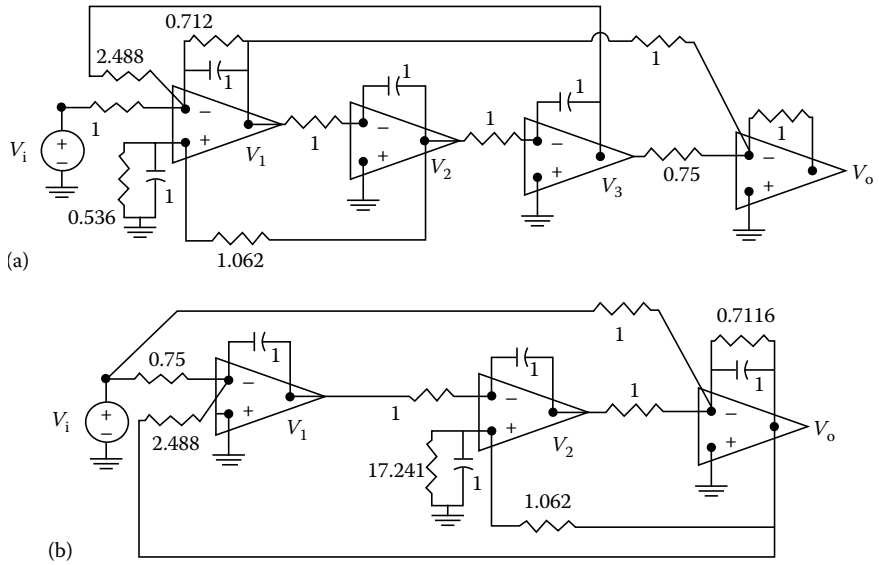


FIGURE 21.21 Two realizations of the third-order, inverse Chebyshev low-pass filter of Example 21.6. Element values are in kΩ and μF.

TABLE 21.4 Some Special State-Variable Biquads

Normalized transfer functions

(1) Lowpass

$$\frac{V_{LP}}{V_i} = \frac{1}{s^2 + \frac{1}{Q}s + 1}$$

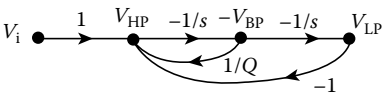
(2) Bandpass

$$\frac{V_{BP}}{V_i} = \frac{s}{s^2 + \frac{1}{Q} + 1}$$

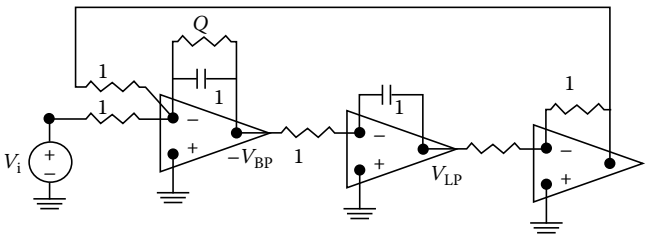
(3) Highpass

$$\frac{V_{HP}}{V_i} = \frac{s^2}{s^2 + \frac{1}{Q}s + 1}$$

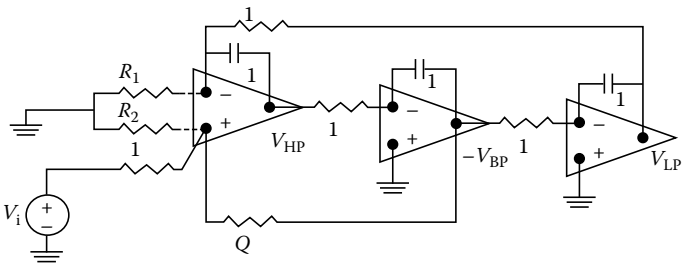
Signal flow graph for transfer functions (1)–(3)



Op amp circuits for (1)–(2). Available outputs: LP and BP



Op amp circuits for (1)–(3). Available outputs: LP, BP, and HP



Either R_1 or R_2 is connected. See notes in Table 21.3 for the determination of their values.

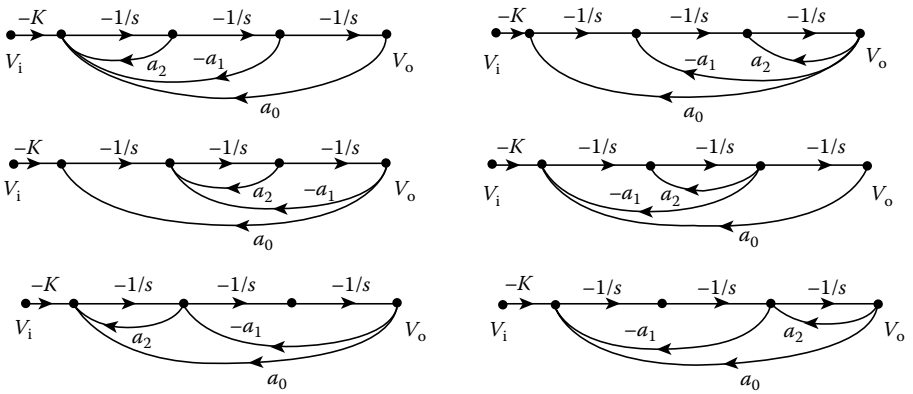


FIGURE 21.22 Six SFGs realizing a third-order, all-pole transfer function.

A total of six SFGs may be constructed in accordance with Equation 21.2 to produce the desired $H(s)$. These are illustrated in Figure 21.22. Among these, six SFGs only two have been chosen for consideration in this section.

Similarly, for a fourth-order, low-pass Butterworth or Chebyshev filter, a total of 20 SFGs may be constructed. The reader should consult Refs. [8,9] for details.

References

1. G. Daryanani, *Principles of Active Network Synthesis and Design*, New York: John Wiley & Sons, 1976.
2. G. C. Temes and J. W. LaPatra, *Introduction to Circuit Synthesis and Design*, New York: McGraw-Hill, 1977.
3. M. E. Van Valkenburg, *Analog Filter Design*, New York: Holt, Rinehart & Winston, 1982.
4. W. K. Chen, *Passive and Active Filters*, New York: John Wiley & Sons, 1986.
5. L. P. Huelsman, *Active and Passive Analog Filter Design*, New York: McGraw-Hill, 1993.
6. W. J. Kerwin, L. P. Huelsman, and R. W. Newcomb, State variable synthesis for insensitive integrated circuit transfer functions, *IEEE J. Solid Circuits*, SC-2, 87–92, Sep. 1967.
7. P. M. Lin, Simple design procedure for a general summer, *Electron. Eng*., 57(708), 37–38, Dec. 1985.
8. N. Fliege, A new class of canonical realizations for analog and digital circuits, *IEEE Proc. 1984 International Symposium on Circuits and Systems*, Montreal, Canada, pp. 405–408, May 1984.
9. P. M. Lin, Topological realization of transfer functions in canonical forms, *IEEE Trans. Automatic Control*, AC-30, 1104–1106, Nov. 1985.

22

Analysis in the Frequency Domain

22.1 Network Functions **22**-1
Definition of Network Functions • Poles and Zeros • Network Stability • Initial and Final Value Theorems
22.2 Advanced Network Analysis Concepts **22**-11
Introduction • Fundamental Network Analysis Concepts • Kron's Formula • Engineering Application of Kron's Formula • Generalization of Kron's Formula • Return Ratio Calculations • Evaluation of Driving Point Impedances • Sensitivity Analysis
References **22**-39

Jiri Vlach
University of Waterloo

John Choma, Jr.
University of Southern California

22.1 Network Functions

Jiri Vlach

22.1.1 Definition of Network Functions

Network functions can be defined if the following constraints are satisfied:

1. Network is linear.
2. It is analyzed in the frequency domain using the Laplace transform.
3. All initial voltages and currents are zero (zero state conditions).

This chapter demonstrates how the various functions can be derived, but first we introduce some explanations and definitions. If we analyze any linear network, we can take as output any nodal voltage, or a difference of any two nodal voltages; denote such as output voltage by V_{out}. We can also take as the output a current through any element of the network; we call it output current, I_{out}. If the network is excited by a voltage source, E, then we can also calculate the current delivered into the network by this source; this is the input current, I_{in}. If the network is excited by a current source, J, then the voltage across the current source is the input voltage, V_{in}.

Suppose that we analyze the network and keep the letter E or J in our derivations. Then, we can define the following network functions:

$$\begin{aligned}
&\text{Voltage transfer function,} && T_{\mathrm{v}} = \frac{V_{\mathrm{out}}}{E}\\
&\text{Input admittance,} && Y_{\mathrm{in}} = \frac{I_{\mathrm{in}}}{E}\\
&\text{Transfer admittance,} && Y_{\mathrm{tr}} = \frac{I_{\mathrm{out}}}{E}\\
&\text{Current transfer function,} && T_{\mathrm{i}} = \frac{I_{\mathrm{out}}}{J}\\
&\text{Input impedance,} && Z_{\mathrm{in}} = \frac{V_{\mathrm{in}}}{J}\\
&\text{Transfer impedance,} && Z_{\mathrm{tr}} = \frac{V_{\mathrm{out}}}{J}
\end{aligned} \tag{22.1}$$

Output impedance or output admittance are also used, but the concept is equivalent to the input impedance or admittance. The only difference is that, for calculations, the source is placed temporarily at a point from which the output normally will be taken. In the Laplace transform, it is common to use capital letters, V for voltages and I for currents. We also deal with impedances, Z, and admittances, Y. Their relationships are

$$V = ZI, \quad I = YV$$

The impedance of a capacitor is $Z_{\mathrm{C}} = 1/sC$, the impedance of an inductor is $Z_{\mathrm{L}} = sL$, and the impedance of a resistor is R. The inverse of these values are admittances: $Y_{\mathrm{C}} = sC$, $Y_{\mathrm{L}} = 1/sL$, and the admittance of a resistor is $G = 1/R$.

To demonstrate the derivations of the above functions two examples are used. Consider the network in Figure 22.1, with input delivered by the voltage source, E. By Kirchhoff's current law (KCL), the sum of currents flowing away from node 1 must be zero:

$$(G_1 + sC_1 + G_2)V_1 - G_2V_2 - EG_1 = 0$$

Similarly, the sum of currents flowing away from node 2 is

$$-V_1G_2 + (G_2 + sC_2 + G_3)V_2 = 0$$

The independent source is denoted by the letter E, and is assumed to be known. In mathematics, we transfer known quantities to the right. Doing so and collecting the equations into one matrix equation results in

$$\begin{bmatrix} G_1 + G_2 + sC_1 & -G_2 \\ -G_2 & G_2 + G_3 + sC_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} EG_1 \\ 0 \end{bmatrix}$$

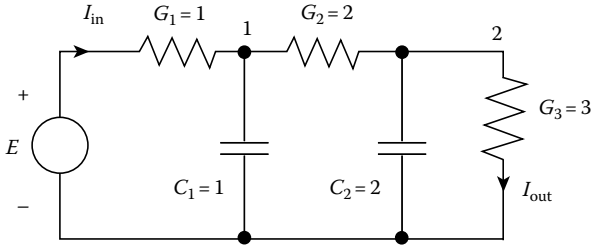


FIGURE 22.1 Network used to illustrate the derivations of network functions.

If numerical values from the figure are used, this system simplifies to

$$\begin{bmatrix} s+3 & -2 \\ -2 & 2s+5 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} E \\ 0 \end{bmatrix}$$

or

$$\boldsymbol{YV} = \boldsymbol{E}$$

Any method can be used to solve this system, but for the sake of explanation it is advantageous to use Cramer's rule. First, find the determinant of the matrix,

$$D = 2s^2 + 11s + 11$$

To obtain the solution for the variable $V_1(V_2)$, replace the first (second) column of $\boldsymbol{Y}$ by the right-hand side and calculate the determinant of such a modified matrix. Denoting such a determinant by the letter N with an appropriate subscript, evaluate

$$N_1 = \begin{vmatrix} E & -2 \\ 0 & 2s+5 \end{vmatrix} = (2s+5)E$$

Then,

$$V_1 = \frac{N_1}{D} = \frac{2s+5}{2s^2+11s+11}E$$

Now, divide the equation by E, which results in the voltage transfer function

$$T_{\mathrm{v}} = \frac{V_1}{E} = \frac{2s+5}{2s^2+11s+11}$$

To find the nodal voltage V_2, replace the second column by the elements of the vector on the right-hand side:

$$N_2 = \begin{vmatrix} s+3 & E \\ -2 & 0 \end{vmatrix} = 2E$$

The voltage is

$$V_2 = \frac{N_2}{D} = \frac{2}{2s^2+11s+11}E$$

and another voltage transfer function of the same network is

$$T_{\mathrm{v}} = \frac{V_2}{E} = \frac{2}{2s^2+11s+11}$$

Note that many network functions can be defined for any network. For instance, we may wish to calculate the currents I_{in} or I_{out}, marked in Figure 22.1. Because the voltages are already known, they are used: For the output current $I_{out} = G_3 V_2$ and divided by E

$$Y_{tr} = \frac{I_{out}}{E} = \frac{3V_2}{E} = \frac{6}{2s^2 + 11s + 11}$$

The input current $I_{in} = E - G_1 V_1 = E - V_1 = E(2s^2 + 9s + 6)/(2s^2 + 11s + 11)$ and dividing by E

$$Y_{in} = \frac{I_{in}}{E} = \frac{2s^2 + 9s + 6}{2s^2 + 11s + 11}$$

In order to define the other possible network functions, we must use a current source, J, as in Figure 22.2, where we also take the current through the inductor as an output variable. This method of formulating the network equations is called modified nodal. The sum of currents flowing away from node 1 is

$$(G_1 + sC_1)V_1 + I_L - J = 0$$

from node 2 it is

$$G_2 V_2 - I_L = 0$$

and the properties of the inductor are expressed by the additional equation

$$V_1 - V_2 - sLI_L = 0$$

Inserting numerical values and collecting in matrix form:

$$\begin{bmatrix} s+1 & 0 & 1 \\ 0 & 2 & -1 \\ 1 & -1 & -s \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} = \begin{bmatrix} J \\ 0 \\ 0 \end{bmatrix}$$

The determinant of the system is

$$D = -(2s^2 + 3s + 3)$$

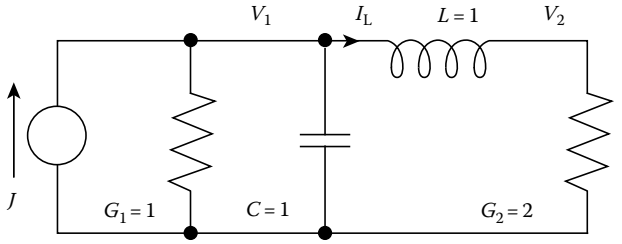


FIGURE 22.2 Network used to calculate other possible network functions.

To solve for V_1, we replace the first column by the right-hand side and evaluate the determinant

$$N_1 = \begin{vmatrix} J & 0 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & -s \end{vmatrix} = -(2s+1)J$$

Then, $V_1 = N_1/D$ and dividing by J we obtain the network function

$$Z_{\text{tr}} = \frac{V_1}{J} = \frac{2s+1}{2s^2+3s+3}$$

To obtain the inductor current, evaluate the determinant of a matrix in which the third column is replaced by the right-hand side: $N_3 = -2J$. Then, $I_{\text{L}} = N_3/D$ and

$$T_{\text{i}} = \frac{I_{\text{L}}}{J} = \frac{2}{s^2+3s+3}$$

In general,

$$F = \frac{\text{Output variable}}{E \text{ or } J} = \frac{\text{Numerator polynomial}}{\text{Denominator polynomial}} \tag{22.2}$$

Any method that may be used to formulate the equations will lead to the same result. One example shows this is true. Reconsider the network in Figure 22.2, but use the admittance of the inductor, $Y_{\text{L}} = 1/sL$, and do not consider the current through the inductor. In such a case, the nodal equations are

$$\left\{1 + s + \frac{1}{s}\right\} V_1 - \frac{1}{s} V_2 = J$$
$$-\frac{1}{s} V_1 + \left\{\frac{1}{s} + 2\right\} V_2 = 0$$

We can proceed in two ways:

1. We can multiply each equation by s and thus remove the fractions. This provides the system equation

$$\begin{bmatrix} s^2+s+1 & -1 \\ -1 & 2s+1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} sJ \\ 0 \end{bmatrix}$$

   The determinant of this matrix is $D = 2s^3 + 3s^2 + 3s$. To calculate V_1, find $N_1 = s(2s+1)J$. Their ratio is the same as before because one s in the numerator can be canceled against the denominator.
2. If we do not remove the fractions and go ahead with the solution, we have the matrix equation

$$\begin{bmatrix} s+1+1/s & -1/s \\ -1/s & 1/s+2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix}$$

   The determinant is $D = 2s + 3 + 3/s$ and the numerator for V_1 is $N_1 = (1/s+2)J$. Taking their ratio

$$V_1 = \frac{(1/s+2)J}{2s+3+3/s} = \frac{2s+1}{2s^2+3s+3} J$$

   which is the same result as before.

We conclude that it does not matter which method is used to formulate the equations. The result is always a ratio of two polynomials in the variable s.

Many additional conclusions can be drawn from these examples. The most important result so far is that all network functions of any given network have the same denominator. It was easy to discover this property because we used Cramer's rule, with its evaluation by the ratio of two determinants. It should be mentioned at this point that we may have network functions in which some terms of the numerator can cancel against the same terms of the denominator. Such a cancellation represents a mathematical simplification which does not change the validity of the above statement.

Occasionally, the network may have more than one source. In such cases, we apply the superposition principle of linear networks. The contribution to the output can be calculated separately for each source and the results added. All that must be done is to correctly remove those sources which are not considered at the moment. All unused independent voltage sources must be replaced by short circuits. All unused independent current sources are replaced by open circuits (removed from the network). Although we did not use dependent sources in our examples, it is necessary to stress that such removal must not take place for dependent sources.

Network functions can be used to find responses to any given input signal. First, multiply the network function by E or J; this will give the expression for the output. Afterward, the letter E or J is replaced by the Laplace transform of the signal. For instance, if the signal is a unit step, then the source is replaced by $1/s$. If it is cost ωt, then the source is replaced by the Laplace transform, $s/(s^2 + \omega^2)$, and so on.

In the Laplace transform, one special signal exists, the Dirac impulse, commonly denoted by $\delta(t)$. It can be represented as a rectangular pulse having width T and height $1/T$. The area of the pulse is always 1, even if we go to lim $T \to 0$, which is the Dirac impulse. Its Laplace transform is 1. Because multiplication by 1 does not change the network function, we conclude that any network function is also the Laplace transform of the network response to the Dirac impulse.

A word of caution: In the network function always divide by the independent voltage (current) source. We cannot take two analysis results, for instance V_1 and V_2, derived for Figure 22.1, and take their ratio. This will not be a network function.

22.1.2 Poles and Zeros

Networks with lumped elements have network functions which are always ratios of two polynomials with real coefficients. For some applications the polynomials may be expressed as functions of some (or all) elements, but the principle is unchanged.

Because we have a ratio of two polynomials, the network function can be written in two forms:

$$F = \frac{\sum_{i=0}^{M} a_i s^i}{\sum_{i=0}^{N} b_i s^i} = K \frac{\prod_{i=1}^{M} (s - z_i)}{\prod_{i=1}^{N} (s - p_i)} \tag{22.3}$$

The middle form is what we obtain from analyses similar to those in the examples. Algebraically, a polynomial of order N has exactly N roots. This leads to the form on the right. The multiplicative constant, K, is the ratio

$$K = \frac{a_M}{b_N}$$

and is obtained by dividing each polynomial by the coefficient of its highest power.

It is easy to find roots of a first- and second-order polynomial because formulas are available, but in all other cases iterative methods and a computer are utilized. However, even without actually finding the roots, we can draw a number of important conclusions.

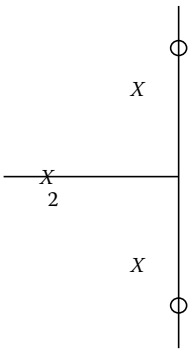


FIGURE 22.3 Network function with two complex conjugate zeros on the imaginary axis, two complex conjugate poles, and one double real pole.

If the highest power of the polynomial is odd, then at least one real root will exist. The other roots may be either real or complex, but if they are complex, then they always appear in complex conjugate pairs. The roots of the numerator are called zeros, and those of the denominator are called poles. We denote the zeros by

$$z_i = a_i + jb_i$$

where $j = \sqrt{-1}$. Either a or b may be zero. For the poles, we have similarly

$$p_i = c_i + jd_i$$

The polynomial also may have multiple roots. For instance, the polynomial $P(s) = (s+1)^2(s+2)^3$ has a double root at $s = -1$ and a triple root at $s = -2$. The positions of the poles and zeros, with the constant K, completely define the network function and also all network properties. The positions can be plotted in a complex plane, the zeros indicated by small circles and poles by crosses. A multiple pole (zero) is indicated by a number appearing at the cross (circle). Figure 22.3 shows a network function with two complex conjugate zeros on the imaginary axis, two complex conjugate poles, and one double real pole.

As derived previously, all network functions of any given network have the same poles. Their positions depend only on the structure of the network and are independent of the signal or where the signal is applied. Because of this fundamental property, the poles are also called natural frequencies of the network. The zeros depend on the place at which we attach the source and also on the point where we take the output.

It is possible to have networks in which a pole is in exactly the same position as a zero; mathematically, such terms cancel. Figure 22.4 is an example. Writing the sum of currents at nodes 1, 2, and 3, we obtain

$$\begin{aligned}(2s+2)V_1 - (s+3)V_3 &= sE\\ (2s+2)V_2 - V_3 &= E\\ -sV_1 - V_2 + (s+3)V_3 &= 0\end{aligned}$$

and in matrix form

$$\begin{bmatrix} 2s+2 & 0 & -(s+3) \\ 0 & 2s+2 & -1 \\ -s & -1 & s+3 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} sE \\ E \\ 0 \end{bmatrix}$$

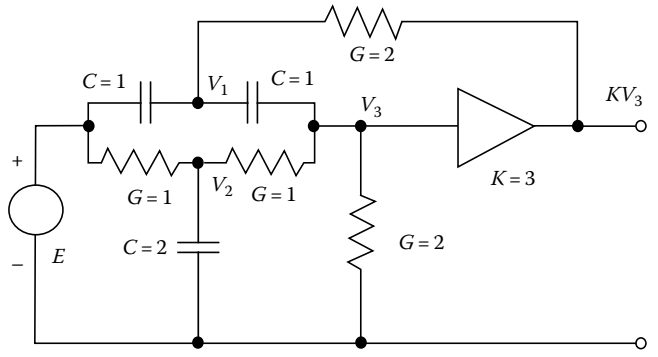


FIGURE 22.4 Network in which a pole is in exactly the same position as a zero.

By carefully evaluating the determinant we discover that we can keep the term $(2s+2)$ separate and get $D=(2s+2)(s^2+2s+5)$. Replacing the third column by the right-hand side, we calculate the numerator $N_3=(2s+2)(s^2+1)E$. Because the output is KV_3, the voltage transfer function is

$$T_{\mathrm{v}}=\frac{3(2s+2)(s^2+1)}{(2s+2)(s^2+2s+5)}$$

Mathematically, the term $(2s+2)$ cancels and the network function is sometimes written as

$$T_{\mathrm{v}}=\frac{3(s^2+1)}{s^2+2s+5}$$

Such cancellation makes the denominator different from other network functions that we might derive for the same network, but it is not a correct way to describe the properties of the network. The cancellation gives the impression that we have a second-order network, while it is actually a third-order network.

22.1.3 Network Stability

Stability of the network depends entirely on the positions of its poles. The following is a list of the conditions in order for the network to be stable, with subsequent explanation of the reasons:

1. Network is stable if all its poles are in the left half of the complex plane.
2. Network is unstable if at least one of its poles is in the right-half plane.
3. Network is marginally stable if all its poles are simple and exactly on the imaginary axis.
4. Network is unstable if it has all poles on the imaginary axis, but at least one of them has multiplicity two or more.

Courses on mathematics teach the process of decomposing a rational function into partial fractions. We show an example with one simple real pole and a pair of simple complex conjugate poles,

$$F(s)=\frac{3s^2+8s+6}{(s+1)(s^2+2s+2)}=\frac{1}{s+1}+\frac{1+j}{s+1+j}+\frac{1-j}{s+1-j}$$

The poles are $p_1=-1$ and $p_{2,3}=-1\pm j$, all with negative real parts and all lying in the left-half plane. Partial fraction decomposition is on the right of the preceding equation. It is always true, for any lumped network, that the decomposition for a real pole has a real constant in the numerator. Complex poles always appear in complex conjugate pairs and the decomposition constants, if complex, also are complex conjugate. Once such a decomposition is available, tables can be used to invert the functions into time domain. The decomposition may be quite a laborious process, however, only a few types of terms need be considered for lumped networks. All are collected in Table 22.1. Each time domain expression is multiplied by unit step, $u(t)$, which is zero for $t<0$ and is one for $t\geq 0$. Such multiplication correctly expresses the fact that the time functions start at $t=0$.

Formula one in Table 22.1 shows that a real, single pole in the left-half plane will lead to a time-domain function which decreases as e^{-ct}. This response is called stable. If $c=0$, then the response becomes $u(t)$. Should the pole be in the right-half plane, the exponent will be positive and e^{ct} will grow rapidly and without bound. This network is said to be unstable.

Formula two shows what happens if the pole is real, with multiplicity n. If it is in the left-half plane, then t^{n-1} is a growing function, but e^{-ct} decreases faster, and for large t the result tends to zero. The function is still stable.

TABLE 22.1 Laplace Transform Pairs

Formula	Laplace Domain	Time Domain
1	$\frac{K}{s+c}$	$Ke^{-ct}u(t)$
2	$\frac{K}{(s+c)^n}$	$K\frac{t^{n-1}}{(n-1)!}e^{-ct}u(t)$
3	$\frac{A+jB}{s+c+jd}+\frac{A-jB}{s+c-jd}$	$2e^{-ct}(A\cos dt+B\sin dt)u(t)$
4	$\frac{A+jB}{(s+c+jd)^n}+\frac{A-jB}{(s+c-jd)^n}$	$\frac{2t^{n-1}}{(n-1)!}e^{-ct}(A\cos dt+B\sin dt)u(t)$

Formula three considers the case of two simple complex conjugate poles. Their real parts influence the exponent, and the imaginary parts contribute to oscillations. If the real part is negative, the oscillations will be damped, the response will become zero for large t, and the network will be stable. If the real part is zero, then the oscillations continue indefinitely with constant amplitude. For the positive real part, the network becomes unstable.

Formula four considers a pair of multiple complex conjugate poles. As long as the real part is negative, the oscillations will decrease with time and the network will be stable. If a real part is zero or positive, the network is unstable because the oscillations will grow.

22.1.4 Initial and Final Value Theorems

Finding the poles and evaluating the time domain response is a complicated process, which normally requires the use of a computer. It is, therefore, advisable to use all possible steps that may provide information about the network behavior without actually finding the full time-domain response.

Two Laplace transform theorems help in finding how the network behaves at $t=0$ and at $t\to\infty$. Both theorems are derived from the Laplace transform formula for differentiation,

$$\int_{0^-}^{\infty} f'(t)e^{-st}\,dt = sF(s)-f(0^-) \tag{22.4}$$

where 0^- indicates that we are considering the instant just before the signal is applied. If we let $s\to 0$, then $e^0=1$, and the integral of the derivative becomes the function itself. Inserting the integrating limits we get

$$f(\infty)-f(0^-)=\lim_{s\to 0}[sF(s)-f(0^-)]$$

Canceling $f(0^-)$ on both sides, we arrive at the "final value theorem"

$$\lim_{t\to\infty} f(t)=\lim_{s\to 0} sF(s) \tag{22.5}$$

Another possibility is to let $s\to\infty$; then e^{-st} in Equation 22.4 will be zero and the whole left side becomes zero. This can be written as

$$0=\lim_{s\to\infty}[sF(s)-f(0^-)]$$

and because $f(0^-)$ is nothing but the limit of $f(t)$ for $t \to 0^-$, we obtain the initial value theorem

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s) \tag{22.6}$$

Note the similarity of the two theorems; we will apply them to the function used in Section 22.1.3. Consider

$$sF(s) = \frac{3s^3 + 8s^2 + 6s}{s^3 + 3s^2 + 4s + 2}$$

If we take any large value of s, the highest powers will dominate and in limit, for $s \to \infty$, we get 3. This is the value of the time-domain response at $t = 0$. The limit for $s = 0$ is zero, and from the final value theorem we know that $f(t)$ will be zero for $t \to \infty$.

To extract still more information, use the example collected in Table 22.2. Scrolling down the table, each Laplace domain function is s times that above it. Each multiplication by s means differentiation in the time domain, as follows from Equation 22.4. Scrolling down the second column of Table 22.2, each function is the derivative of that above it. To apply the limiting theorems, take the Laplace domain formula, which is one level lower, and insert the limits. The limiting is also shown and is confirmed by inserting either $t = 0$ or $t \to \infty$ into the time functions.

Although the two theorems are useful, the final value theorem is valid only if the function is stable. Consider the unstable function with two poles in the right-half plane

$$F_1(s) = \frac{1}{(s+1)(s-1+j)(s-1-j)} = \frac{1}{s^3 - s^2 + 2}$$

Its time-domain response is

$$f_1(t) = \frac{1}{5}[e^{-t} + e^{+t}(2\sin^t - \cos^t)]u(t)$$

and the term e^{+t} will cause the function to grow for large t. If the final value theorem is applied, we consider

$$sF_1(s) = \frac{s}{s^3 + s^2 + 2}$$

Inserting $s = 0$, the theorem predicts that the time function will approach zero for large t. This is disappointing, but some additional simple rules can be applied. The function is unstable if some coefficients of the denominator are missing, or if the denominator coefficients do not all have the

TABLE 22.2 Additional Laplace Transform Pairs

Laplace Domain $D = s^3 + 3s^2 + 4s + 2$	Time Domain	$s \to \infty$ $t = 0$	$s = 0$ $t \to \infty$
$F(s) = 1/D$	$f(t) = e^{-t}(1 - \cos t)u(t)$	0	0
$G(s) = sF(s) = s/D$	$g(t) = f'(t) = e^{-t}(-1 + \cos t + \sin t)u(t)$	0	0
$H(s) = s^2F(s) = s^2/D$	$h(t) = f''(t) = e^{-t}(1 - 2\sin t)u(t)$	1	0
$K(s) = s^3/D$	$k(t) = f'''(t) = \delta(t) + e^{-t}(2\sin t - 2\cos t - 1)u(t)$	$\delta(t)$	0

same sign (all + or all −). Such situations are easily detected, but if all coefficients have the same sign, nothing can be said about stability. Additional theorems exist (e.g., Hurwitz theorem), but if in doubt, it is probably simplest to go to the computer and find the poles.

22.2 Advanced Network Analysis Concepts

John Choma, Jr.

22.2.1 Introduction

The systematic analysis of an electrical or electronic network entails formulating and solving the relevant Kirchhoff equations of equilibrium. The analysis is conducted to acquire a theoretically sound understanding of circuit responses. Such an understanding minimally delineates the dynamical effects of topology, controllable circuit branch variables, and observable parameters for active devices embedded in the circuit. It also illuminates circuit node and branch impedances to which the relevant responses of the circuit undergoing investigation are especially sensitive. Unfortunately, the complexity of modern networks, and particularly integrated analog electronic circuits, often inhibits the mathematical tractability that underpins an engineering understanding of circuit behavior. It is therefore not surprising that when mathematical analyses accompany a computer-assisted circuit design venture, the subcircuits identified for manual study are simplified representations of the corresponding subcircuits in the draft design solution. Unless care is exercised, these approximations can mask a satisfying understanding, and they can even lead to erroneous results.

Analytical and modeling approximations notwithstanding, the key to assimilating a satisfying understanding of the electrical characteristics of complex circuits is appropriate studies of simpler partitions of these circuits. To this end, Kron [1,2] has provided, and others have explained and reinforced [3–5], an elegant theory that allows the circuit response solutions of these network partitions to be coalesced so that the desired response of the interconnected circuit is reconstructed exactly. Aside from formalizing an analytical mechanism for studying complicated circuits in terms of the solutions gleaned for more manageable subcircuits of the composite network [6], Kron's work allows for a computationally efficient study of feedback network responses. The theory also allows for the investigation of the sensitivity of overall network performance with respect to both small and large parametric changes [7]. In view of the exclusive focus on linear circuits in this section, it is worth interjecting that a form of Kron's partitioning theory is also applicable to certain classes of nonlinear circuits [8].

22.2.2 Fundamental Network Analysis Concepts

The derivation of Kron's formula, as well as the development of a general methodology for applying Kron's partitioning mechanism to the analyses of complex circuits, requires a fundamental understanding of the classical techniques exploited in the analysis of linear networks. Such an understanding begins by considering the $(n+1)$ node, b branch linear network abstracted in Figure 22.5a. The input port, which is defined by the node pair, 1–2, is excited by a signal source whose Thévenin voltage is V_S and whose Thévenin impedance is Z_S. In response to this excitation, a load voltage, V_L, is developed across a load impedance, Z_L, which shunts the output port consisting of the node pair, 3–4. Two other nodes, nodes m and p, are explicitly delineated for future reference. In response to the applied signal source, the voltage across the input port is V_I, while the voltage established across the node pair, m–p is V_k. In addition, the reference, or ground, node is labeled node 0. Either node 2, node 4, or both of these nodes can be incident with the ground node; that is, the signal source and/or the load impedance can be terminated to the network ground. The diagram in Figure 22.5b is identical to

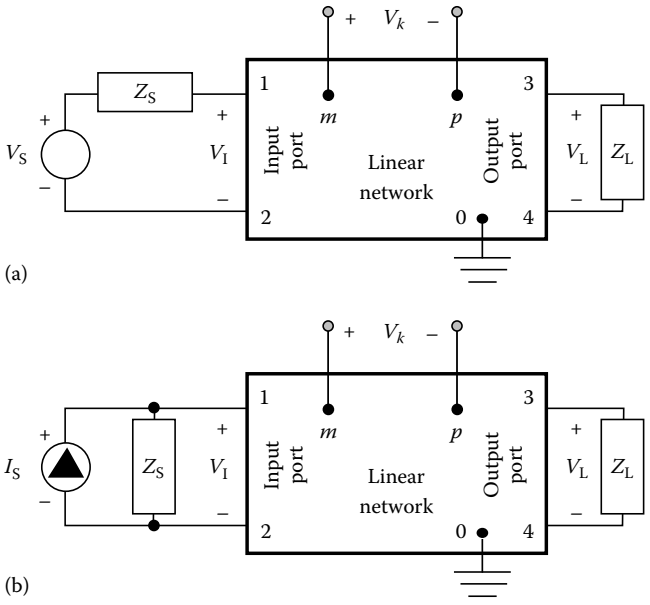


FIGURE 22.5 (a) Generalized linear network driven by a voltage source. (b) The network of (a), but with the signal excitation represented by its Norton equivalent circuit.

that of Figure 22.5a except for the fact that the applied signal source is represented by its Norton equivalent circuit, where the Norton signal current, I_S, is

$$I_S = \frac{V_S}{Z_S} \tag{22.7}$$

Assuming that a nodal admittance matrix exists for the linear $(n+1)$ node network at hand, the n equilibrium Kirchhoff's current law (KCL) equations can be expressed as the matrix relationship

$$\mathbf{J} = \mathbf{YE} \tag{22.8}$$

where

J is an n-vector whose ith entry
J_i is an independent current flowing into the ith circuit node
E is an n-vector of node voltages such that its ith entry
E_i is an ith node voltage referenced to network ground
Y is a square matrix of order n, is the nodal admittance matrix of the circuit

If **Y** is nonsingular, the node voltages follow as

$$\mathbf{E} = \mathbf{Y}^{-1}\mathbf{J} \tag{22.9}$$

Note that Equation 22.9 is useful symbolically, but not necessarily computationally. In particular, Equation 22.9 shows that the n node voltages of the $(n+1)$ node, b branch network of Figure 22.5 can be straightforwardly computed in terms of the known independent current source vector and the parameters embedded in the network nodal admittance matrix. In an actual analytical environment, however, the nodal admittance matrix is rarely formulated and inverted. Instead, some or all of the n

node voltages of interest are determined merely by algebraically manipulating and solving either the n independent KCL equations or the $(b-n)$ independent Kirchhoff's voltage law (KVL) equations that are required to establish the equilibrium conditions of the subject network.

If the n vector, **E**, is indeed evaluated, all n independent node voltages are known, because

$$\mathbf{E}^{\mathrm{T}} = [E_1, E_2, E_3, \ldots, E_m, \ldots, E_p, \ldots, E_n] \tag{22.10}$$

where the superscript T indicates the operation of matrix transposition. In general, E_i, for $i = 1, 2, \ldots, n$, is the voltage developed at node i with respect to ground. It follows that the voltage between any two nodes derives directly from the network solution inferred by Equation 22.9. For example, the input port voltage, V_{I}, is $(E_1 - E_2)$, the output port voltage, V_{L}, is $(E_3 - E_4)$, and the voltage, V_k, from node m to node p is $V_k = (E_m - E_p)$.

The calculation of the voltage appearing between any two circuit nodes can be formalized with the help of the generalized network diagrammed in Figure 22.6 and through the introduction of the connection vector concept. In particular, let $\mathbf{A}_{ij}$ denotes the $(n \times 1)$ connection vector for the port defined by the node pair, i–j. Moreover, let the voltage, V, at node i be taken as positive with respect to node j, and allow a current, I (which may be zero), to flow into node i and out of node j, as indicated in the diagram. Then, the elements of the connection vector, $\mathbf{A}_{ij}$, are all zero except for $a+1$ in its ith row and $a-1$ in its jth row. If node j is the reference node, all elements of $\mathbf{A}_{ij}$, which in the case can be written simply as $\mathbf{A}_i$, are zero except for the ith row element, which remains $+1$. Thus, $\mathbf{A}_{ij}$ has the form

$$\mathbf{A}_{ij}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & \cdots & \underset{\substack{\downarrow \\ i\text{th column}}}{+1} & \cdots & \underset{\substack{\uparrow \\ j\text{th column}}}{-1} & \cdots & \underset{\substack{\downarrow \\ n\text{th column}}}{0} \end{bmatrix} \tag{22.11}$$

For the special case in which a circuit branch element interconnects every pair of circuit nodes, $\mathbf{A}_{ij}$ is the appropriate column of the node to branch incidence matrix, which is a rectangular matrix of order $(n \times b)$, for the $(n+1)$ node, b branch network at hand [9].

Returning to the calculation of V_{I}, V_{L}, and V_k, it follows from Equations 22.9 through 22.11 that

$$V_{\mathrm{I}} = E_1 - E_2 = \mathbf{A}_{12}^{\mathrm{T}}\mathbf{E} = \mathbf{A}_{12}^{\mathrm{T}}\mathbf{Y}^{-1}\mathbf{J} \tag{22.12}$$

$$V_{\mathrm{L}} = E_3 - E_4 = \mathbf{A}_{34}^{\mathrm{T}}\mathbf{E} = \mathbf{A}_{34}^{\mathrm{T}}\mathbf{Y}^{-1}\mathbf{J} \tag{22.13}$$

and

$$V_k = E_m - E_p = \mathbf{A}_{mp}^{\mathrm{T}}\mathbf{E} = \mathbf{A}_{mp}^{\mathrm{T}}\mathbf{Y}^{1}\mathbf{J} \tag{22.14}$$

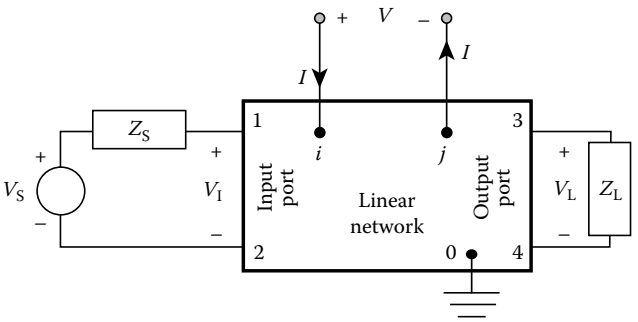


FIGURE 22.6 Generalized network diagram used to define the connection vector concept.

Assuming that I_S is the only independent source of excitation in the network of Figure 22.5

$$\mathbf{J} = \mathrm{A}_{12} I_S \tag{22.15}$$

which is the mathematical equivalent of the observation that the Norton source current, I_S, is entering node 1 of the network and leaving node 2. Accordingly,

$$V_I = \left(\mathbf{A}_{12}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) I_S \tag{22.16}$$

$$V_L = \left(\mathbf{A}_{34}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) I_S \tag{22.17}$$

and

$$V_k = \left(\mathbf{A}_{mp}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) I_S \tag{22.18}$$

Several noteworthy features are implicit to the foregoing three relationships. First, each of the three parenthesized matrix products on the right-hand sides of the equations is a scalar. This observation follows from the facts that a transposed connection vector is a row matrix of order $(1 \times n)$, the inverse nodal admittance matrix is an n-square, and a connection vector is an n-vector. Second, these scalar products represent transimpedances from the input port to the port at which the voltage of interest is extracted. In the case of Equation 22.16, the ratio, $V_I\, I_S$, is actually the impedance, Z_{SS}, seen by the Norton current, I_S; that is,

$$Z_{SS} \triangleq \frac{V_S}{I_S} = \left(\mathbf{A}_{12}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) \tag{22.19}$$

where, as asserted previously, I_S is presumed to be the only source of energy applied to the network undergoing study. Similarly,

$$Z_{LS} \triangleq \frac{V_L}{I_S} = \left(\mathbf{A}_{34}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) \tag{22.20}$$

is the transimpedance from the input port to the output port, while

$$Z_{kS} \triangleq \frac{V_k}{I_S} = \left(\mathbf{A}_{mp}^T \mathbf{Y}^{-1} \mathbf{A}_{12}\right) \tag{22.21}$$

is the transimpedance from the input port to the port defined by the node pair, m–p.

The impedance in Equation 22.19 and the transimpedances given by Equations 22.20 and 22.21 are cast as explicit algebraic functions of the inverse of the network nodal admittance matrix. However, similar to the node voltages in Equations 22.19 and 22.10, network transimpedances are rarely calculated manually through an actual delineation and inversion of the nodal admittance matrix. Instead, they usually derive from a straightforward analysis of the considered network, subject to the proviso that all excitations applied to the subject network, save for the single test current source, are reduced to zero. For example, in the abstraction shown in Figure 22.7, the transimpedance, Z_{ij}, from any port j to any port i is

$$Z_{ij} \triangleq \left.\frac{V_{\text{test}}}{I_{\text{test}}}\right|_{\text{all independent sources} = 0} \tag{22.22}$$

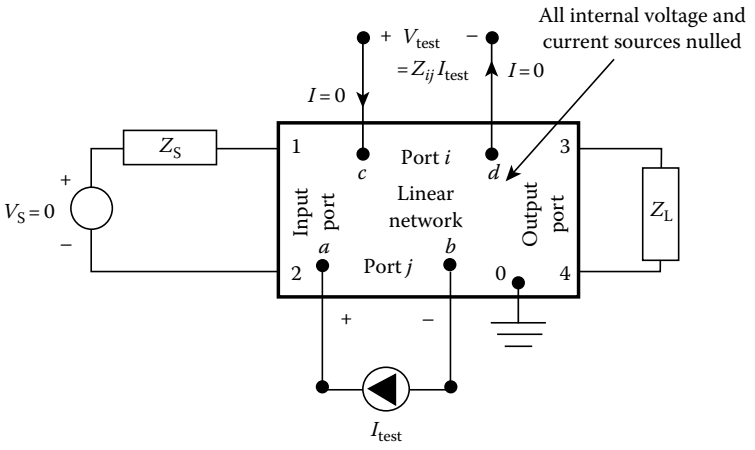


FIGURE 22.7 An illustration of a practical manual technique for computing the transimpedance between any port j to any port i of a linear network.

For the case of $j = i$, this transimpedance becomes the effective impedance seen at port i by the test current source. In view of the preceding discussion, and the node pairs indicated in Figure 22.7, the transimpedance (or impedance) quantity that derives from Equation 22.22 is identical to the matrix relationship

$$Z_{ij} = \left(\mathbf{A}_{cd}^{\mathrm{T}} \mathbf{Y}^{-1} \mathbf{A}_{ab}\right) \tag{22.23}$$

The last result highlights the fact that all network transimpedances are directly related to the inverse of the nodal admittance matrix. Hence, these transimpedances are inversely proportional to the determinant, $\Delta Y(s)$, of the admittance matrix, $\mathbf{Y}$. It follows that the poles of all transimpedances and effective port impedances are the roots of the characteristic polynomial

$$\det(\mathbf{Y}) \triangleq \Delta Y(s) = 0 \tag{22.24}$$

Note from Equations 22.7, 22.17, and 22.20 that the voltage gain of the considered linear network is

$$\frac{V_{\mathrm{L}}}{V_{\mathrm{S}}} = \frac{Z_{\mathrm{LS}}}{Z_{\mathrm{S}}} \tag{22.25}$$

Thus, if the source impedance in Figure 22.5 is a real number, $Z_S = R_S$, the roots of Equation 22.24 also comprise the poles of the voltage transfer function and, indeed, of the linear network.

22.2.3 Kron's Formula

Assume now that the network depicted in Figure 22.5 has been analyzed in the sense that all network node voltages developed in response to the signal source have been determined. Assume further that subsequent to this analysis, an impedance, Z_k, appended to nodes m and p, as shown in Figure 22.8. In addition to causing a current, I, to flow into node m and out of node p, this additional branch element is likely to perturb the values of all of the originally computed circuit node and circuit branch voltages. The matrix, $\mathbf{E}'$, of new node voltages can be evaluated for the modified topology in Figure 22.8 by determining the new nodal admittance matrix $\mathbf{Y}'$, and the reapplying Equation 22.9. The tedium associated with a

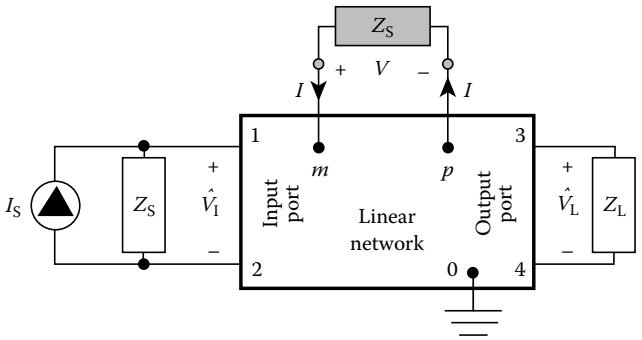


FIGURE 22.8 The inclusion of an impedance, Z_k, between nodes m And p, subsequent to the analysis of the network in Figure 22.5.

second network analysis, along with the inefficiency of discarding the results of a study performed on a network whose topology differs only modestly from that of the original configuration, can be circumvented through the use of Kron's theorem. As illuminated next, this theorem derives from a methodical application of such classical concepts as the theories of superposition, substitution, and Thévenin. In addition to providing a computationally efficient mechanism for determining $\mathbf{E}'$, Kron's technique allows for a direct comparison of $\mathbf{E}'$ to the matrix, $\mathbf{E}$, of original node voltages. It therefore allows for a convenient response sensitivity analysis with respect to the appended branch element.

It is appropriate to interject that the problem postulated previously possesses more than mere academic interest. It is, in fact, a problem that is commonly encountered, for example, in the analysis of electronic circuits. In order to linearize these circuits around specified quiescent operating points, it is necessary to supplant the utilized active devices by small signal equivalent circuits. Such models are invariably simplified, often through the tacit neglect of presumably noncritical branch elements, to mitigate analytical complexity and tedium. Thus, while the circuit properly identified for investigation might be of the topological form appearing in Figure 22.8, the circuit actually subjected to manual circuit analysis is likely the reduced structure depicted in Figure 22.5; that is, the ostensibly noncritical impedance, Z_k is removed in the interest of analytical tractability. Questions naturally arise in regard to the degree of error incurred by the invoked circuit simplification. Kron's method, as developed next, answers these questions in terms of the results already deduced for the approximate network and without requiring explicit analytic results for the ''exact'' network.

The process of evaluating the perturbation on network node voltages incurred by the action of shunting nodes m and p in the circuit of Figure 22.5 by the impedance Z_k begins by determining the Thévenin equivalent circuit that drives the appended branch. To this end, Z_k is removed in the diagram of Figure 22.8, thereby collapsing the network to Figure 22.5a. The relevant Thévenin voltage, V_{th}, at the node pair, mp, is, in fact, V_k, as defined by Equation 22.18. Recalling Equation 22.21, this voltage is

$$V_{\text{th}} \equiv V_k = \left(\mathbf{A}_{mp}^{\text{T}}\mathbf{Y}^{-1}\mathbf{A}_{12}\right)I_{\text{S}} = Z_{k\text{S}}I_{\text{S}} \tag{22.26}$$

The corresponding Thévenin impedance, Z_{th}, derives from a study of the test configuration of Figure 22.9, in which the independent source current, I_{S}, is nulled, the impedance, Z_k, in Figure 22.8 is replaced by a test current of value I_{test}, and the ratio of the resultant port voltage, V_{test}, to I_{test} is understood to be the desired Thévenin impedance. For this configuration, the network nodal admittance matrix, $\mathbf{Y}$, is unchanged, but the independent network current vector, $\mathbf{J}$, becomes $\text{A}_{mp}\,I_{\text{test}}$. Thus, the resultant n-vector, $\mathbf{E}''$, of nodal voltages is

$$\mathbf{E}'' = \mathbf{Y}^{-1}\mathbf{A}_{mp}I_{\text{test}} \tag{22.27}$$

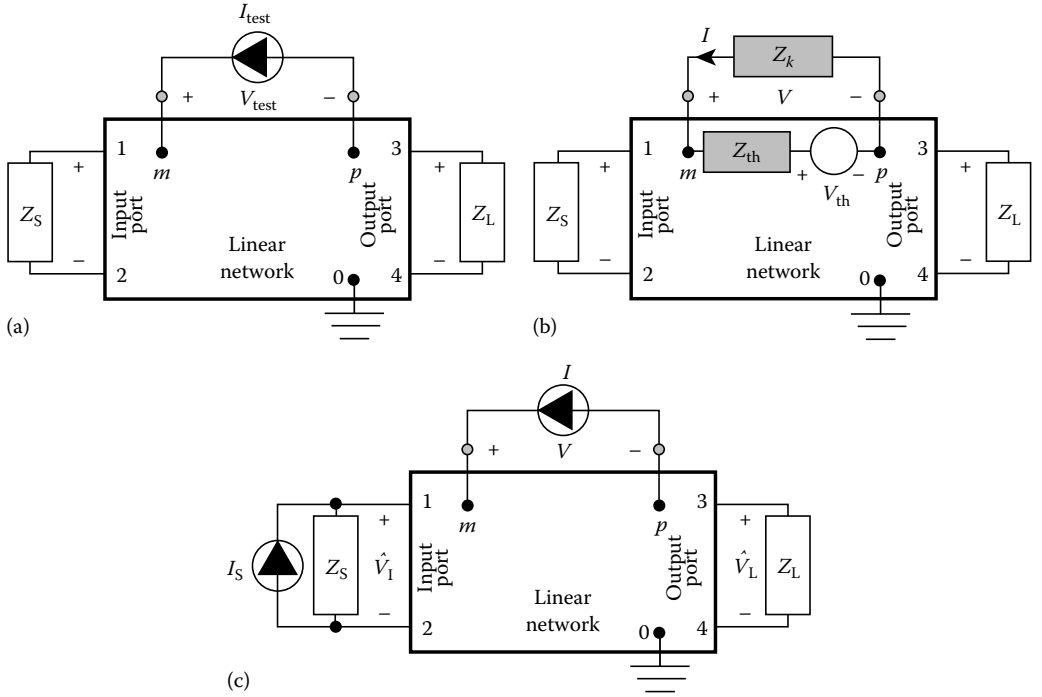


FIGURE 22.9 (a) Circuit diagram for evaluating the Thévenin impedance seen by the appended impedance Z_k. (b) Circuit diagram used to compute the current, I, conducted by Z_k. (c) The application of the substitution theorem with respect to Z_k.

and by Equation 22.8, the voltage, V_{test}, is

$$V_{\text{test}} = \left(\mathbf{A}_{mp}^{\text{T}}\mathbf{Y}^{-1}\mathbf{A}_{mp}\right)I_{\text{test}} \tag{22.28}$$

It follows that the requisite Thévenin impedance, Z_{th}, is

$$Z_{\text{th}} = \frac{V_{\text{test}}}{I_{\text{test}}} = \left(\mathbf{A}_{mp}^{\text{T}}\mathbf{Y}^{-1}\mathbf{A}_{mp}\right) \tag{22.29}$$

Insofar as the appended impedance, Z_k, is concerned, the network in Figure 22.8 behaves in accordance with the circuit abstraction of Figure 22.9. The current, I, conducted by Z_k is, without approximation,

$$I = -\frac{V_{\text{th}}}{Z_{\text{th}} + Z_{\text{k}}} = -\left(\frac{Z_{k\text{S}}}{Z_{\text{th}} + Z_k}\right)I_{\text{S}} \tag{22.30}$$

where Equation 22.26 has been used, and Z_{th} is understood to be given by Equation 22.29. However, by the substitution theorem, the impedance, Z_k in Figure 22.8 can be supplanted by an independent current source of value I, as suggested in Figure 22.9c. Specifically, this substitution of the impedance of interest with a current source with a value that is dictated by Equation 22.30 guarantees that the n-vector of node voltages for the modified circuit in Figure 22.9c is identical to the n-vector, E', of node voltages for the topology given in Figure 22.8.

The circuit of Figure 22.9c now has two independent excitations: the original current sources, I_S, and the current, I, substituted for the appended impedance, Z_k. Accordingly, the current source vector for the subject circuit superimposes two current components and is given by

$$\mathbf{J} = \mathbf{A}_{12} I_S + \mathbf{A}_{mp} I = \mathbf{A}_{12} I_S - \mathbf{A}_{mp} \left(\frac{Z_{kS}}{Z_{th} + Z_k} \right) I_S \tag{22.31}$$

The corresponding vector of node voltages is, by Equation 22.9,

$$\mathbf{E}' = \mathbf{Y}^{-1} \left[\mathbf{A}_{12} - \mathbf{A}_{mp} \left(\frac{Z_{kS}}{Z_{th} + Z_k} \right) \right] I_S \tag{22.32}$$

If analytical attention focuses on the general output voltage, $\hat{V}_{ij}$, developed between nodes i and j in the circuit of Figure 22.8,

$$\hat{V}_{ij} = \mathbf{A}_{ij}^{T} \mathbf{E}' \tag{22.33}$$

where

$$\hat{V}_{ij} = \left[\left(\mathbf{A}_{ij}^{T} \mathbf{Y}^{-1} \mathbf{A}_{12} \right) - \left(\mathbf{A}_{ij}^{T} \mathbf{Y}^{-1} \mathbf{A}_{mp} \right) \left(\frac{Z_{kS}}{Z_{th}} + Z_k \right) \right] I_S \tag{22.34}$$

The result in Equation 22.34 is one of many possible versions of Kron's formula. It states that when an impedance, Z_k, is appended between nodes m and p of a linear network whose nodal admittance matrix is $\mathbf{Y}$, the perturbed voltage established between any two nodes, i and j, can be determined as a function of the parameters indigenous to the original network (prior to the inclusion of Z_k). In particular, the evaluation is executed on the original network (with Z_k absent) and exploits the original nodal admittance matrix, $\mathbf{Y}$, the original transimpedance, Z_{kS}, between the input port and the port to which Z_k is ultimately appended, and the Thévenin impedance, Z_{th}, is observed when looking into the terminal pair to which Z_k is connected.

22.2.4 Engineering Application of Kron's Formula

The engineering utility of Kron's formula, Equation 22.34, is best demonstrated by examining the voltage transfer function of the network in Figure 22.8 in terms of the companion gain for the network depicted in Figure 22.5a. Using Equation 22.7 and noting that the perturbed output voltage is developed from node 3 to node 4, the perturbed voltage gain, $\hat{A}_v$, is

$$\hat{A}_v \triangleq \frac{\hat{V}_L}{V_S} = \frac{\mathbf{A}_{34}^{T} \mathbf{Y}^{-1} \mathbf{A}_{12}}{Z_S} - \left(\frac{\mathbf{A}_{34}^{T} \mathbf{Y}^{-1} \mathbf{A}_{mp}}{Z_S} \right) \left(\frac{Z_{kS}}{Z_{th} + Z_k} \right) \tag{22.35}$$

The first matrix product on the right-hand side of this relationship represents the transimpedance, Z_{LS}, from the input port to the output port of the original network, as given by Equation 22.20. Moreover, the resultant impedance ratio, Z_{LS}/Z_S, is the voltage gain, A_v, of the original ($Z_k = \infty$) network, as delineated in Equation 22.25. The second matrix product symbolizes the transimpedance, Z_{Lk}, from the port to which the appended impedance, Z_k, is connected to the output port; that is,

$$Z_{Lk} = \mathbf{A}_{34}^{T} \mathbf{Y}^{-1} \mathbf{A}_{mp} \tag{22.36}$$

Assuming $A_v \neq 0$, Equation 22.35 can then be reduced to

$$\hat{A}_v = A_v\left[1 - \left(\frac{Z_{Lk}}{Z_{LS}}\right)\left(\frac{Z_{kS}}{Z_{th} + Z_k}\right)\right] \tag{22.37}$$

This result expresses the perturbed voltage gain as a function of the original voltage gain, A_v, the input to output transimpedance, Z_{LS}, the transimpedance, Z_{kS}, from the input port to the port at which Z_k is appended, and Z_{Lk}, the transimpedance from the port to which Z_k is incident to the output port. Observe that when the appended impedance is infinitely large, the perturbed gain reduces to the original voltage gain, as expected.

In an actual analytical situation, however, all of the transimpedances indicated in Equation 22.37 need not be calculated. In order to demonstrate this contention, rewrite Equation 22.37 in the form

$$\hat{A}_v = A_v\left[\frac{1 + Y_k(Z_{th} - (Z_{Lk}Z_{kS}/Z_{LS}))}{1 + Y_k Z_{th}}\right] \tag{22.38}$$

where

$$Y_k = \frac{1}{Z_k} \tag{22.39}$$

is the admittance of the appended impedance, Z_k. Now consider the test structure of Figure 22.10a, which is the modified circuit shown in Figure 22.8, but with the appended branch supplanted by a test current source, I_{test}. With two sources, I_S and I_{test}, activating the network, superposition yields a resultant output port voltage, V_{LL}, of

$$V_{LL} = Z_{LS}I_S + Z_{Lk}I_{test} \tag{22.40}$$

and a test port voltage, V_{test}, of

$$V_{test} = Z_{kS}I_S - Z_{kk}I_{test} \tag{22.41}$$

For $I_S = 0$, the network in Figure 22.10a reduces to the configuration in Figure 22.9a, and Equation 22.41 delivers $V_{test}/I_{test} = Z_{kk}$. It follows that the impedance parameter, Z_{kk}, is the Thévenin impedance seen by

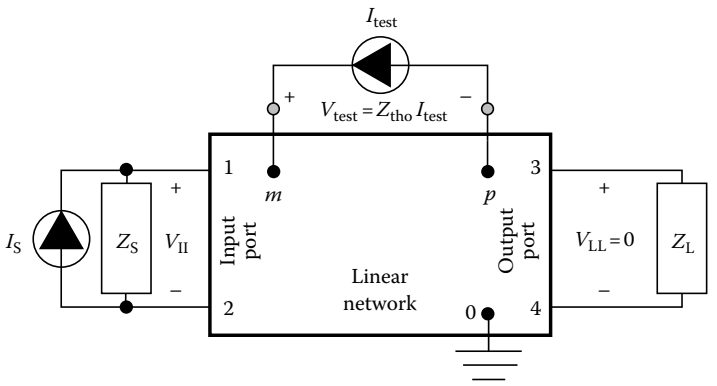


FIGURE 22.10 Network diagram pertinent to the computation of the null Thévenin impedance seen by the impedance appended to the node pair, *m*–*p*.

Z_k, as determined in conjunction with an analytical consideration of Figure 22.5a; that is, Equation 22.41 is equivalent to the expression

$$V_{\text{test}} = Z_{k\text{S}} I_{\text{S}} + Z_{\text{th}} I_{\text{test}} \tag{22.42}$$

Consider the case, suggested in Figure 22.10, in which the output port voltage, V_{LL}, is constrained to zero for any and all values of the load impedance, Z_{L}. From Equation 22.40, this case requires a source excitation that satisfies

$$I_{\text{S}} = -\left(\frac{Z_{\text{L}k}}{Z_{\text{LS}}}\right) I_{\text{test}} \tag{22.43}$$

If this result is substituted into Equation 22.42, the ratio, $V_{\text{test}}/I_{\text{test}}$, is found to be

$$\frac{V_{\text{test}}}{I_{\text{test}}} = Z_{\text{th}} - \frac{Z_{\text{L}k} Z_{k\text{S}}}{Z_{\text{LS}}} \tag{22.44}$$

which mirrors the parenthesized numerator term on the right-hand side of Equation 22.38. The ratio in Equation 22.44 might rightfully be termed a null Thévenin impedance, Z_{tho}, seen by Z_k, in the sense that it is indeed the Thévenin impedance witnessed by Z_k, but under the special circumstance of a nonzero source excitation selected to null the output response variable of the network undergoing investigation. Thus, in Figure 22.10,

$$\left.\frac{V_{\text{test}}}{\text{I}_{\text{test}}}\right|_{\substack{\text{Source}\neq 0 \\ \text{Output}=0}} \triangleq Z_{\text{tho}} = Z_{\text{th}} - \frac{Z_{\text{L}k} Z_{k\text{S}}}{Z_{\text{LS}}} \tag{22.45}$$

Equation 22.38 now reduces to the simpler result

$$\hat{A}_{\text{v}} = A_{\text{v}} \left(\frac{1 - Y_k Z_{\text{tho}}}{1 + Y_k Z_{\text{th}}}\right) = A_{\text{v}} \left(\frac{1 + (Z_{\text{tho}}/Z_k)}{1 + (Z_{\text{th}}/Z_k)}\right) \tag{22.46}$$

Equation 22.46 is both computationally useful and philosophically important. From a computational viewpoint, it allows for an efficient evaluation of the voltage transfer function of a linear network, perturbed by the addition of an impedance element between two extant nodes of the network, in terms of the voltage gain of the original, unperturbed circuit. As expected, this original voltage gain, A_{v}, is the voltage gain of the perturbed network for the special case of a perturbing impedance where the admittance is zero (or whose impedance value is infinitely large). Only two other parameters are required to complete the evaluation of the perturbed gain. The first is the Thévenin impedance, Z_{th}, seen by the appended impedance element. This Thévenin impedance is calculated traditionally by nulling all independent sources applied to the subject network. The second parameter is the null Thévenin impedance, Z_{tho}, which is the value of Z_{th} for the special circumstance of a test current source and independent source excitations selected to constrain the output response variable to zero. Once Z_{tho} and Z_{th} are determined, the degree to which the voltage transfer function is dependent on the appended impedance is easily determined. For example, the per-unit change in gain owing to the addition of Z_k between nodes m and p in the network of Figure 22.8 is

$$\frac{\Delta A_{\text{v}}}{A_{\text{v}}} = \frac{\hat{A}_{\text{v}} - A_{\text{v}}}{A_{\text{v}}} = \frac{Z_{\text{tho}} - Z_{\text{th}}}{Z_k + Z_{\text{th}}} \tag{22.47}$$

it is important to note that the transfer function sensitivity implied by Equation 22.47 imposes no *a priori* restrictions on the value of Z_k. In particular, Z_k can assume any value from that of a short circuit to that of the opposite extreme of an open circuit.

From a philosophical perspective, when analytical attention focuses explicitly on feedback circuits, Equation 22.46 can be derived from signal flow theory [10,11] and is actually Bode's classical gain equation [12]. In the context of Bode's theory Y_k is referred to as a reference parameter, or a critical parameter, of the feedback circuit. The product, Y_kZ_{th}, is termed the return ratio with respect to the critical parameter, while the product Y_kZ_{tho}, is identified as the null return ratio with respect to the critical parameter. Finally, when Equation 22.46 is applied as Bode's equation, the transfer function, A_v is termed the null transfer function, in the sense that it is the actual transfer function of the network at hand, under the special case of a critical parameter constrained to zero.

Example 22.1

In an attempt to demonstrate the engineering utility of the foregoing theoretical disclosures, consider the problem of determining the voltage gain of the common emitter amplifier—a schematic diagram is offered in Figure 22.11a. Without detracting from the primary intent of this example, the schematic diagram at hand has been simplified in that requisite biasing subcircuitry is not shown. Assuming that the bipolar junction transistor embedded in the amplifier operates in its linear regime, the pertinent small signal equivalent circuit is the topology depicted in Figure 22.11b.

Assume that the amplifier source resistance, R_S, is 600 Ω and that the load resistance, R_K, is 10 kΩ. Assume further that the model parameters for the transistor are as follows: r_b (internal emitter resistance) = 2.5 Ω, r_0 (forward early resistance) = 18 kΩ, β (forward short circuit current gain) = 90, and r_c (internal collector resistance) = 70 Ω. Determine the voltage gain of the amplifier and the effect exerted on the gain by neglecting the early resistance, R_0.

Solution

1. Analytical simplicity traditionally dictates the tacit neglect of the forward Early resistance, r_0. This commonly invoked approximation reduces the model given in Figure 22.11b to the equivalent circuit in Figure 22.12a. By inspection of the latter diagram, the approximate gain of the common emitter voltage is

$$A_v = \frac{V_O}{V_S} = -\frac{\beta R_L}{R_S + r_b + r_\pi + (\beta + 1)r_e} = -388.3 \text{ v/v}$$

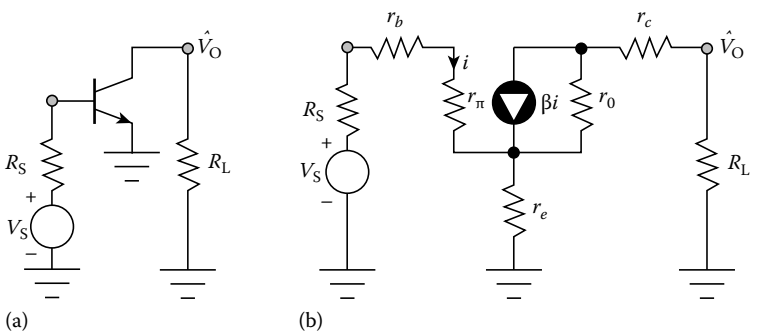


FIGURE 22.11 (a) Simplified schematic diagram of a common emitter amplifier. (b) The small signal equivalent circuit of the common emitter amplifier.

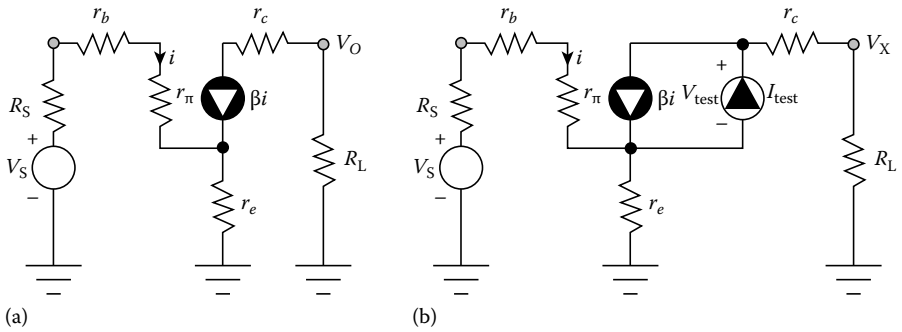


FIGURE 22.12 (a) The approximate small signal equivalent circuit of the common emitter amplifier in Figure 22.11a. The approximation entails the tacit neglect of the forward Early resistance, r_0. (b) The test equivalent circuit used to compute the Thévenin and the null Thévenin resistances seen by r_0 in (a).

2. In order to determine the impact that r_0 has on this voltage gain, r_0 is removed from the equivalent circuit and replaced by a test current source, I_{test}, as depicted in Figure 22.12b. With the independent input voltage, V_S, set to zero, the Thévenin resistance seen by r_0, which is the ratio, V_{test}/I_{test}, is easily shown to be

$$R_{th} = r_e \left\| \left(\frac{R_S + r_b + r_n}{\beta + 1} \right) + \frac{r_c + R_L}{1 + \dfrac{\beta r_e}{R_S + r_b + r_\pi + r_e}} = 9.10 \text{ k}\Omega \right.$$

On the other hand, if V_X is constrained to zero, no current flows through the load resistance branch, and therefore, I_{test} is necessarily βi. This condition gives a null Thévenin resistance of

$$R_{tho} = -\frac{r_e}{\beta} = -27.78 \times 10^{-3} \ \Omega$$

3. With $A_v = -388.3$ v/v, $Z_k = r_0 = 18$ kΩ, $Z_{th} = R_{th} = 9.10$ kΩ, and $Z_{tho} = R_{tho} = -27.78 \times 10^{-3}$ Ω, Equation 22.46 produces a corrected voltage gain of

$$\hat{A}_v = A_v \left(\frac{1 + \dfrac{R_{tho}}{r_0}}{1 + \dfrac{R_{th}}{r_0}} \right) = -258.0 \text{ v/v}$$

From Equation 22.47, the presence of r_0, as opposed to its absence, decreases the voltage gain of the subject amplifier by almost 34%.

Example 22.2

As a second example, consider the series-shunt feedback amplifier whose schematic diagram, neglecting requisite biasing circuitry, appears in Figure 22.13a. The analysis of this circuit is simplified by the removal of the connection of the feedback resistance, R_F, at the emitter of transistor Q_1, as shown in Figure 22.13b. If the voltage gain of the simplified topology is denoted as A_v, the voltage gain of the closed loop configuration in Figure 22.13a derives from Equation 22.46, provided that the impedance, Z_k, between the indicated node pair, m–p, is taken as a short circuit; that is, $Z_k = 0$.

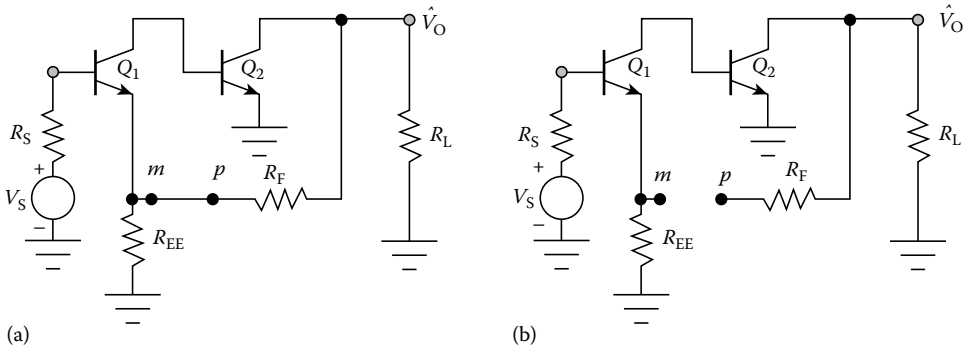


FIGURE 22.13 (a) Simplified schematic diagram of a series-shunt feedback bipolar junction transistor amplifier. The biasing subcircuitry is not shown. (b) The amplifier in (a), but with the feedback resistance connection to the emitter of transistor Q_1 removed.

Assume that the amplifier source resistance, R_S, is 300 Ω, the load resistance, R_L, is 3.5 kΩ, the feedback resistance, R_F, is 1.5 kΩ, and the emitter degeneration resistance, R_{EE}, is 100 Ω. The transistor model invoked for small signal analysis is identical to that used in the preceding example, save for the proviso that the Early resistance, r_0, is ignored herewith. Both transistors are presumed to have identical small signal model parameters as follows: $r_b = 90\ \Omega$, $r_\pi = 1.4$ kΩ, $r_e = 2.5\ \Omega$, $\beta = 90$, and $r_c = 70\ \Omega$. Use Kron's theorem to determine the voltage gain of the closed loop series-shunt feedback amplifier.

Solution

1. The voltage gain of the pertinent equivalent circuit in Figure 22.14 is straightforwardly derived as

$$A_v = \frac{V_O}{V_S} = \frac{\beta^2 R_L}{R_S + r_b + r_\pi + (\beta + 1)(r_e + R_{EE})} = 2550 \text{ v/v}$$

   Observe that the feedback resistance, R_F, does not enter into this calculation because of its disconnection at the emitter of transistor Q_1. Furthermore, the internal collector resistance, r_c, is inconsequential for both transistor stages because the neglect of the forward Early resistance, r_0, places r_c in series with a controlled current source.
2. The Thévenin resistance, R_{th}, seen by the ultimately appended short circuit between nodes m and p is now calculated through use of the model in Figure 22.14b. For this calculation, the signal voltage, V_S, is reduced to zero. With $V_S = 0$,

$$\frac{i_1}{i_{test}} = -\frac{R_{EE}}{R_S + r_b + r_\pi + (\beta + 1)(r_e + R_{EE})}$$

   Noting that $i_2 = -\beta i_1$, KVL yields

$$V_{test} = (R_{EE} + R_L + R_F)I_{test} + \left[(\beta + 1)R_{EE} - \beta^2 R_L\right] i_1$$

   Using the preceding result, introducing the resistance variable, R_X, such that

$$R_X \triangleq r_e + \frac{R_S + r_b + r_\pi}{\beta + 1} = 22.17\ \Omega$$

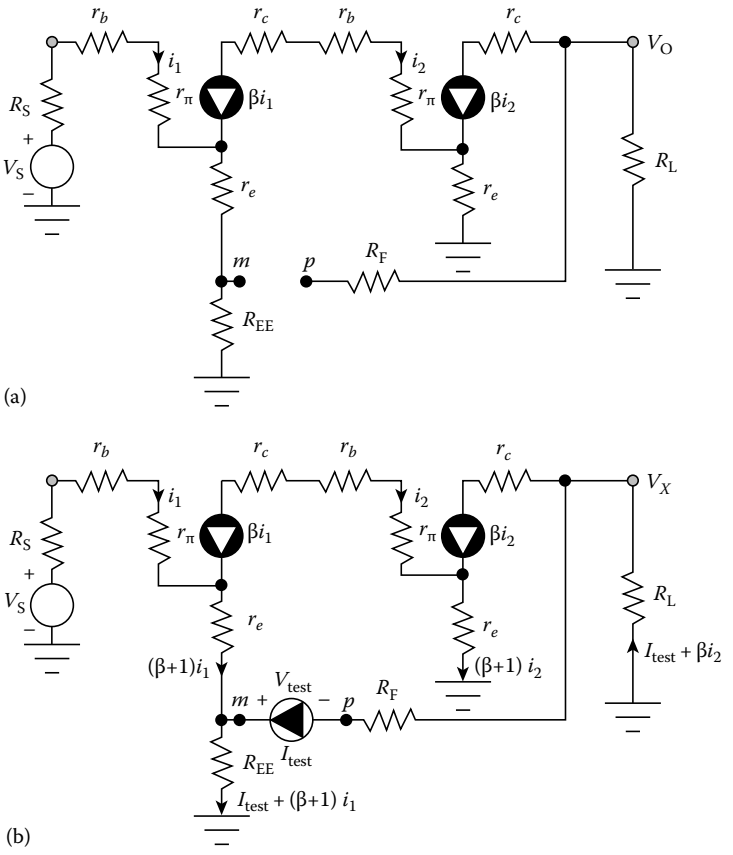


FIGURE 22.14 (a) Small signal equivalent circuit of the feedback amplifier in Figure 22.13b. This circuit is used to compute the voltage gain with the feedback resistance disconnected at the emitter of transistor Q_1. (b) The small signal model used to compute the Thévenin and the null Thévenin resistances seen by the short circuit that is ultimately appended to the node pair, m–p, in Figure 22.13b.

and letting

$$\alpha \triangleq \frac{\beta}{\beta + 1} = 0.989$$

R_{th}, which is the ratio V_{test}/I_{test}, is found to be

$$R_{th} = R_F + (R_{EE}|R_X)\left(1 + \frac{\alpha\beta R_{EE}}{R_{EE} + R_X}\right)R_L = 260.0\ \text{k}\Omega$$

3. For the evaluation of null Thévenin resistance, R_{tho}, the output voltage variable, V_X, in Figure 22.14b is nulled, thereby forcing the current relationship, $I_{test} = -\beta i_2 = +\beta^2 i_1$. Accordingly,

$$R_{tho} = R_F + \left(1 + \frac{1}{\alpha\beta}\right)R_{EE} = 1601\ \Omega$$

4. With $Z_k = 0$, $Z_{th} = R_{th} = 260$ kΩ, and $Z_{tho} = R_{tho} = 1601$ Ω, Equation 22.46 provides, after reconnection of the feedback element, an amplifier gain of

$$\hat{A}_v = A_v\left(\frac{R_{tho}}{R_{th}}\right) = 15.70 \text{ v/v}$$

It is interesting to note that, if the transistors utilized in the feedback amplifier have very large β, which is tantamount to very small R_X and $a \approx 1$, the voltage gain with the feedback element disconnected is

$$a_v \approx \frac{\beta R_L}{R_{EE}}$$

Moreover,

$$R_{th} \approx \beta R_L$$

and

$$R_{tho} \approx R_F + R_{EE}$$

It follows from Equation 22.46 that the approximate voltage gain, subsequent to the reconnection of the feedback resistance, R_F, to the emitter of transistor Q_1 is

$$\hat{A}_v = A_v\left(\frac{R_{tho}}{R_{th}}\right) \approx \left(\frac{\beta R_L}{R_{EE}}\right)\left(\frac{R_F + R_{EE}}{\beta R_L}\right) = 1 + \frac{R_F}{R_{EE}} = 16 \text{ v/v}$$

which is within 2% of the accurately estimated voltage gain.

22.2.5 Generalization of Kron's Formula

The Kron–Bode equation in Equation 22.46 was derived expressly for investigating the voltage transfer function of a linear network to which an impedance element is appended between two network nodes. This equation also can be adapted to the problem of determining the explicit dependence of any type of transfer relationship on any parameter within any linear network.

To this end, consider any linear network, such as the generalization shown in Figure 22.15, whose, load impedance is Z_L and whose source impedance is Z_S. Identify a critical network parameter, say P, to which the dependence on, and sensitivity to, the overall transfer performance of the network undergoing study

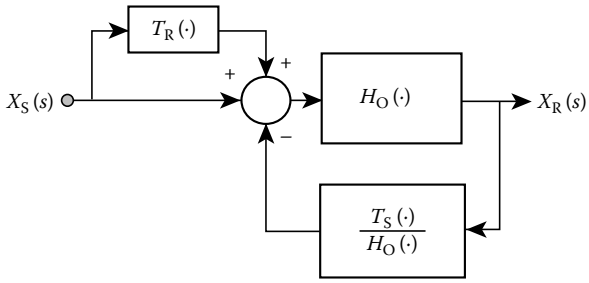


FIGURE 22.15 Generalized block diagram nodel of the I–O transfer characteristics of a linear network.

is of particular interest. This parameter can be, for example, a circuit branch impedance or an active element gain factor where numerical values cannot be determined accurately or controlled adequately in view of potentially unacceptable manufacturing tolerances or device fabrication uncertainties. Let the transfer function of interest be

$$H(P, Z_S, Z_L) = \frac{X_R(s)}{X_S(s)} \tag{22.48}$$

where

$X_R(s)$ denotes the transform of the voltage or current response variable
$X_S(s)$ is the transform of the voltage or current input variable

The functional notation, $H(P, Z_S, Z_L)$, underscores the observation that the transfer function of the liner network is likely to be dependent on the critical parameter, P, the source impedance, Z_S, and the load impedance, Z_L. The corresponding extension of the Kron–Bode relationship is

$$H(P, Z_S, Z_L) = \frac{X_R(s)}{X_S(s)} = H(0, Z_S, Z_L)\left[\frac{1 + PQ_R(Z_S, Z_L)}{1 + PQ_S(Z_S, Z_L)}\right] \tag{22.49}$$

where $H(0, Z_S, Z_L)$, termed the null gain or zero parameter gain, signifies the value of the network transfer function, $H(P, Z_S, Z_L)$, when P is set to zero. This null gain must be finite and nonzero. With reference to the appended impedance formulation in Equation 22.46, observe that the critical parameter, P, is Y_k, the admittance of the appended impedance element, while $H(0, Z_S, Z_L)$ is the gain, A_v, of the network, under the condition of an absent impedance element ($Y_k = 0$).

The product, $PQ_S(Z_S, Z_L)$, is termed the return ratio with respect to parameter P, $T_S(P, Z_S, Z_L)$, and the product, $PQ_R(Z_S, Z_L)$, is referred to as the null return ratio with respect to P, $T_R(P, Z_S, Z_L)$, that is,

$$T_R(P, Z_S, Z_L) \triangleq PQ_R(Z_S, Z_L) \tag{22.50a}$$

$$T_S(P, Z_S, Z_L) \triangleq PQ_S(Z_S, Z_L) \tag{22.50b}$$

It is to be understood that both $Q_S(Z_S, Z_L)$ and $Q_R(Z_S, Z_L)$ are independent of the critical parameter, P. With reference once again to Equation 22.46, note that $Q_S(Z_S, Z_L)$ is the Thévenin impedance seen by the appended admittance, Y_k, while $Q_S(Z_S, Z_L)$ is the null Thévenin impedance facing Y_k.

Equation 22.49 can now be rewritten as

$$H(P, Z_S, Z_L) = \frac{X_R(s)}{X_S(s)} = H(0, Z_S, Z_L)\left[\frac{1 + T_R(P, Z_S, Z_L)}{1 + T_S(P, Z_S, Z_L)}\right] \tag{22.51}$$

Alternatively,

$$H(P, Z_S, Z_L) = \frac{X_R(s)}{X_S(s)} = H(0, Z_S, Z_L)\left[\frac{F_R(P, Z_S, Z_L)}{F_S(P, Z_S, Z_L)}\right] \tag{22.52}$$

where

$$F_S(P, Z_S, Z_L) \triangleq 1 + T_S(P, Z_S, Z_L) \tag{22.53a}$$

$$F_R(P, Z_S Z_L) \triangleq 1 + T_R(P, Z_S, Z_L) \tag{22.53b}$$

respectively, denote the return difference with respect to P and the null return difference with respect to P.

An initial appreciation of the engineering significance of the zero parameter gain, $H(0, Z_S, Z_L) + H_0(\cdot)$, the return ratio, $T_S(P, Z_S, Z_L) + T_S(\cdot)$, and the null return ratio, $T_R(P, Z_S, Z_L) + T_R(\cdot)$, is gleaned by using Equation 22.51 to write

$$X_R(s) = H_0(\cdot)[1 + T_R(\cdot)]X_S(s) - T_S(\cdot)X_R(s) \tag{22.54}$$

In view of the generality of the Kron–Bode formula, this algebraic manipulation of Equation 22.51 implies that the dynamical input/output transfer relationship of all linear networks can be symbolically represented by the block diagram offered in Figure 22.15. This block diagram makes clear that because $T_S(\cdot)$ and $T_R(\cdot)$ are zero for $P = 0$, $H_0(\cdot)$ is the gain afforded by the network as a result of input–output electrical paths that exclude the parameter, P. The diagram also suggests that $T_S(\cdot)$ is a measure of the amount of feedback incurred by parameter P around that part of the circuit that excludes parameter P. Finally, the diagram at hand implies that $T_R(\cdot)$ is a measure of the amount of feedforward incurred by parameter P. In particular, if feedback is removed, two paths remain for the transmission of a signal from the input port to the output port of a linear network. One of these paths, the transmittance of which is measured by $H_0(\cdot)$, is direct and entails the processing of the input signal by that part of the circuit that excludes P. The other nonfeedback path has an effective transmittance of $T_R(\cdot)\, H_0(\cdot)$. The latter path is the feedforward path in the sense that signal is processed through a signal path that is divorced from feedback and is not a result of direct source coupling through the topological part of the network that excludes parameter P.

22.2.6 Return Ratio Calculations

In the generalized transfer relationship of Equation 22.49, the critical parameter, P, can assume one of only six possible forms: circuit branch admittance, circuit branch impedance, transimpedance, transadmittance, gain associated with a current-controlled current source (CCCS), and gain associated with a voltage-controlled voltage source (VCVS) [13]. The methodology underlying the computation of the return ratio and the null return ratio with respect to each of these critical parameter possibilities is given below. The case of $P = Y_k$, a circuit branch admittance, was investigated in the context of Kron's partitioning theorem. Nevertheless, it is reinvestigated next for the purpose of establishing an analytical common denominator for return ratio calculations with respect to the five other reference parameter possibilities.

22.2.6.1 Circuit Branch Admittance

Consider the network abstraction of Figure 22.16a, which identifies a branch admittance, Y_k, as a critical parameter for analysis; that is, $P = Y_k$ in Equation 22.49. The input excitation can be either a voltage source, or a current source and is therefore indicated as a general transformed input variable, $X_S(s)$. Similarly, the output or response variable can be either a voltage or a current, thereby encouraging the generalized transformed response notation, $X_R(s)$. The source and load impedances (or admittances) are absorbed into the network. In order to evaluate the zero parameter gain, $H(0, Z_S, Z_L)$, Y_k is set to zero by removing it from the network. An analysis is then conducted to determine the ratio $X_R(s)/X_S(s)$, of output to input variables, as suggested in Figure 22.16b.

As demonstrated for the case of $P = Y_k$, a circuit branch admittance, the function, $Q_S(Z_S, Z_L)$, in Equation 22.49 is the Thévenin impedance, Z_{th}, facing Y_k. This impedance is computed by determining the ratio, V_x/I_x, with the signal source, $X_S(s)$, nulled, as indicated in Figure 22.16c. Note in Figure 22.16a, that the volt–ampere relationship of the branch housing Y_k is $I_k = Y_k V_k$, where the direction of the branch current, I_k, coincides with the direction of the test current source, I_x, used in the determination of Z_{th}.

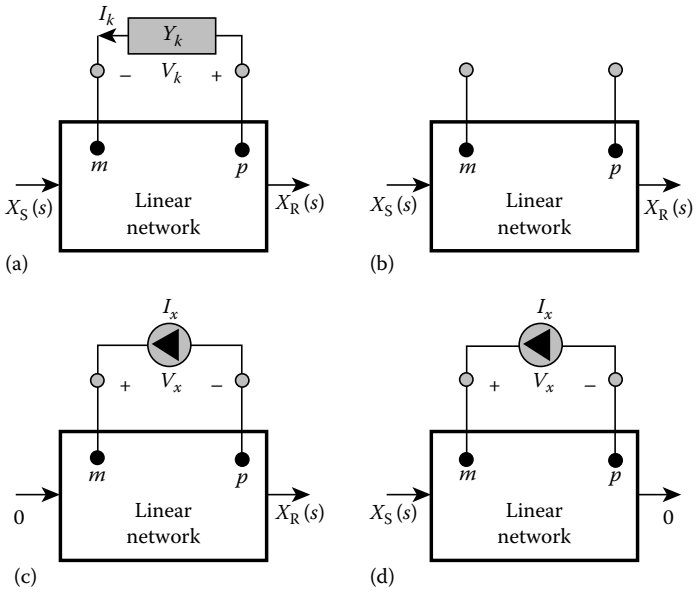


FIGURE 22.16 (a) Linear circuit for which the identified critical parameter is a branch admittance, Y_k. (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain $H(0, Z_S, Z_L)$. (c) The ratio, V_x/I_x, is the function $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, V_x/I_x, is the function, $Q_R(Z_S, Z_L)$, in Equation 22.49.

A comparison Figure 22.16c and alludes to the methodology of replacing the admittance branch by a source of excitation (a current source) where the electrical nature is identical to the dependent electrical variable (a current, I_k) of the branch volt–ampere characteristic. Note that the polarity of the voltage, V_x, used in the determination of the test ratio, V_x/I_x, is opposite to that of the original branch voltage V_k. This is to say that although I_k and V_k are in associated reference polarity in Figure 22.16a, I_x and V_x in the test cell of Figure 22.16c are in disassociated polarity.

The computation of the function, $Q_R(Z_S, Z_L)$, in Equation 22.49 mirrors the computation of $Q_S(Z_S, Z_L)$, except for the fact that instead of setting the source excitation to zero, the output response, $X_R(s)$, is nulled. The source excitation, $X_S(s)$, remains at some computationally unimportant nonzero value, such that its effects, when superimposed over those of the test current, I_x, forces $X_R(s)$ to zero. The situation at hand is diagrammed in Figure 22.16d.

Example 22.3

Return to the series-shunt feedback amplifier of Figure 22.29a. Evaluate the voltage gain of the circuit, but, take the conductance, G_F, of the feedback resistance, R_F, as the critical parameter. The circuit and device model parameters remain the same as in Example 22.2: $R_S = 300\ \Omega$, $R_L = 3.5\ k\Omega$, $R_F = 1.5\ k\Omega$, $R_{EE} = 100\ \Omega$, $r_b = 90\ \Omega$, $r_\pi = 1.4\ k\Omega$, $r_e = 2.5\ \Omega$, $\beta = 90$, and $r_c = 70\ \Omega$.

Solution

1. The zero parameter voltage gain, A_{vo}, of the subject amplifier is the voltage gain of the circuit with $G_F = 0$. But $G_F = 0$ amounts to a removal of the feedback resistance, R_F. Such removal is electrically equivalent to open circuiting the indicated node pair, *m–p*, as diagrammed in the small signal

model of Figure 22.14a. Thus, A_{vo} is identical to the gain, computed in Step (1) of Example 22.2. In particular,

$$A_{vo} = \frac{\beta^2 R_L}{R_S + r_b + r_\pi + (\beta + 1)(r_e + R_{EE})} = 2550 \text{ v/v}$$

2. The model pertinent to computing the functions, $Q_S(Z_S, Z_L)$, and $Q_R(Z_S, Z_L)$, is offered in Figure 22.17. Note that the test current source, I_x, which replaces the critical conductance element, G_F, and the resultant test response voltage, V_x, are in disassociated reference polarity. As in Example 22.2, let

$$R_X \triangleq r_e + \frac{R_S + r_b + r_\pi}{\beta + 1} = 22.17$$

and

$$\alpha \triangleq \frac{\beta}{\beta + 1} = 0.989$$

Then, with $V_S = 0$, and writing $Q_S(Z_S, Z_L)$ as $Q_S(R_S, R_L)$ because of the lack of energy storage elements in the circuit undergoing study,

$$Q_S(R_S, R_L) = \left.\frac{V_x}{I_x}\right|_{V_S=0} = R_{EE} \parallel R_X + \left(1 + \frac{\alpha\beta R_{EE}}{R_{EE} + R_X}\right) R_L' = 258.5 \text{ k}\Omega$$

On the other hand,

$$Q_R(R_S, R_L) = \left.\frac{V_x}{I_x}\right|_{V_O=0} = \left(1 + \frac{1}{\alpha\beta}\right) R_{EE} = 101.1 \ \Omega$$

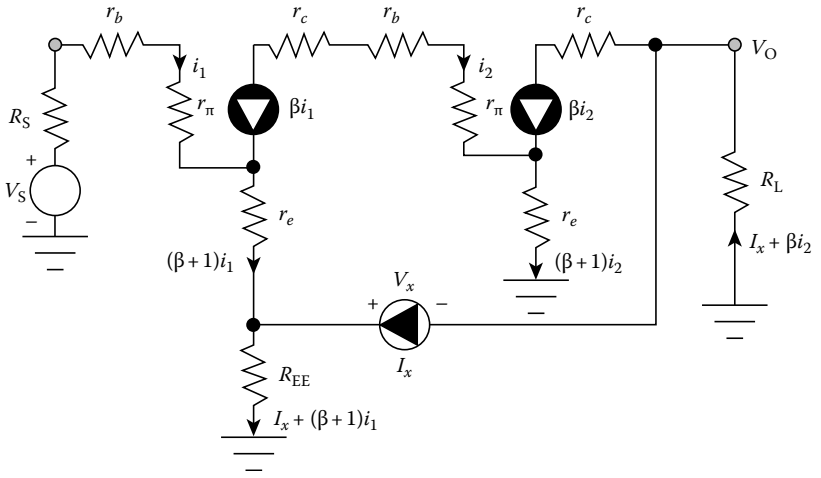


FIGURE 22.17 Circuit used to computer the return ratio and the null return ratio with respect to the conductance, G_F, in the series-shunt feedback amplifier of Figure 22.13a.

3. Substituting the preceding results into Equation 22.49, and recalling that $G_F = 1/R_F$, the voltage gain of the series-shunt feedback amplifier is found to be

$$A_v = \frac{V_O}{V_S} = A_{vo}\left[\frac{1 + \dfrac{Q_R(R_S, R_L)}{R_F}}{1 + \dfrac{Q_S(R_S, R_L)}{R_F}}\right] = 15.7 \text{ v/v}$$

which is the gain result deduced previously.

22.2.6.2 Circuit Branch Impedance

In the circuit of Figure 22.18a, a branch impedance, Z_k, is selected as a critical parameter for analysis; that is $P = Z_k$ in Equation 22.49. The zero parameter gain, $H(0, Z_S, Z_L)$, is evaluated by replacing Z_k with a short circuit, as suggested in Figure 22.18b.

The volt–ampere characteristic equation of the critical impedance branch is $V_k = Z_k I_k$, where, of course, the branch voltage, V_k, and the branch current, I_k, are in associated reference polarity. Because the dependent variable in this volt–ampere expression is a branch voltage, the return and null return ratios are calculated by replacing the subject branch impedance by a test voltage source, V_x. As suggested in Figure 22.18c, the ratio, I_x/V_x, under the condition of nulled independent sources, gives the function $Q_S(Z_S, Z_L)$ in Equation 22.49. On the other hand, and as depicted in Figure 22.18d, the ratio I_x/V_x, with a nulled response, yields $Q_R(Z_S, Z_L)$. Observe that in the present situation, the functions, $Q_S(Z_S, Z_L)$ and $Q_R(Z_S, Z_L)$ are, respectively, the Thévenin and the null Thévenin admittances facing the branch impedance, Z_k.

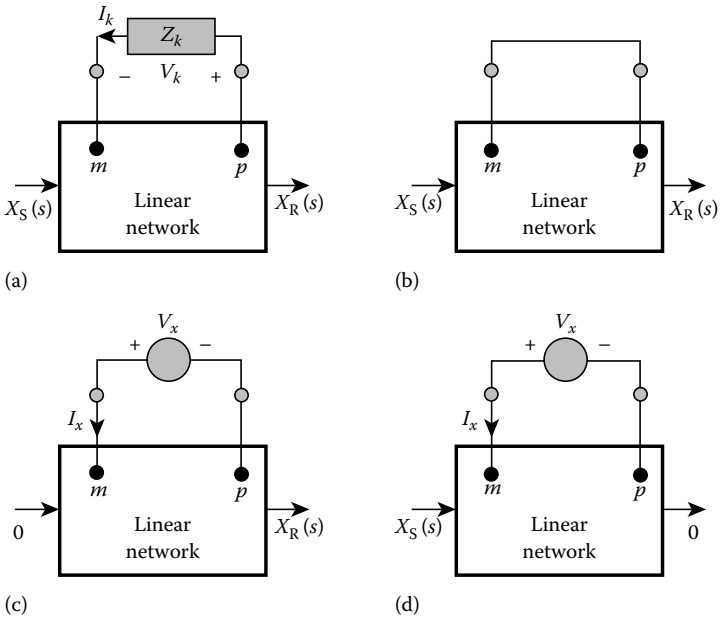


FIGURE 22.18 (a) Linear circuit for which the identified critical parameter is a branch impedance, Z_k. (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain, $H(0, Z_S, Z_L)$. (c) The ratio, I_x/V_x, is the function, $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, I_x/V_x, is the function, $Q_R(Z_S, Z_L)$, in Equation 22.49.

22.2.6.3 Circuit Transimpedance

In the circuit of Figure 22.19a, a circuit transimpedance, Z_t, is selected as a critical parameter for analysis; that is, $P = Z_t$ in Equation 22.49. The zero parameter gain, $H(0, Z_S, Z_L)$, is evaluated by replacing the current-controlled voltage source (CCVS) by a short circuit, as shown in Figure 22.19b.

The volt–ampere characteristic equation of the critical transimpedance branch is $V_k = Z_t I_k$, where I_k is the controlling current for the controlled source branch. Because the dependent variable in this volt–ampere expression is a branch voltage, the return and null return ratios are calculated by replacing the CCVS with a test voltage source, V_x. However, as indicated in Figures 22.19c and d, the polarity of V_x mirrors that of the voltage, V_k, developed across the controlled branch. With I_x taken as a current flowing in the controlling branch in a direction opposite to the polarity of the original controlling current, I_k, the ratio, I_x/V_x, under the condition of nulled independent sources, gives the function, $Q_S(Z_S, Z_L)$ in Equation 22.49. On the other hand, and as depicted in Figure 22.19d, the ratio, I_x/V_x, with a nulled response, yields $Q_R(Z_S, Z_L)$.

22.2.6.4 Circuit Transadmittance

In the network of Figure 22.20a, a circuit transadmittance, Y_t, is selected as the critical parameter. The zero parameter gain, $H(0, Z_S, Z_L)$, is evaluated by replacing the voltage-controlled current source (VCCS) with an open circuit, as shown in Figure 22.20b.

The volt–ampere characteristic question of the critical transadmittance branch is $I_k = Y_t V_k$, where V_k is the controlling voltage for the VCCS. Because the dependent variable in this volt–ampere expression is a branch current, the return and null return ratios are calculated by replacing the VCCS with a test current source, I_x, where, as indicated in Figures 22.20c and d, the polarity of I_x mirrors that of the current, I_k, flowing through the controlled branch. With V_x taken as a voltage developed across the controlling branch in a direction opposite to the polarity of the original controlling voltage, V_k, the ratio, V_x/I_x, under

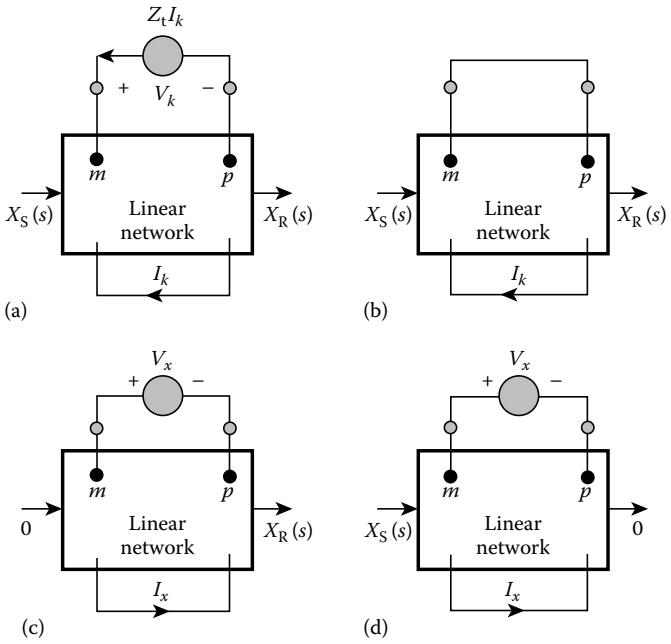


FIGURE 22.19 (a) Linear circuit for which the identified critical parameter is a circuit transimpedance, Z_t. (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain, $H(0, Z_S, Z_L)$. (c) The ratio, I_x/V_x, is the function, $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, I_x/V_x, is the function, $Q_R(Z_S, Z_L)$, in Equation 22.49.

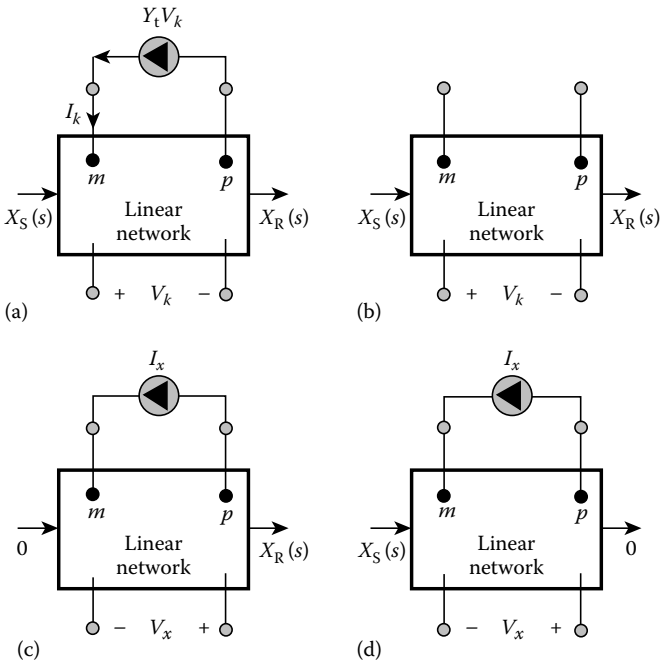


FIGURE 22.20 (a) Linear circuit for which the identified critical parameter is a circuit transadmittance, Y_t, (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain, $H(0, Z_S, Z_L)$. (c) The ratio, V_x/I_x, is the function $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, V_x/I_x, is the function $Q_R(Z_S, Z_L)$, in Equation 22.49.

the condition of nulled independent sources, gives the function, $Q_S(Z_S, Z_L)$ in Equation 22.43. On the other hand, and as offered in Figure 22.20d, the ratio, V_x/I_x, under the condition of a nulled response, yields $Q_R(Z_S, Z_L)$.

Example 22.4

The circuit in Figure 22.21a is a low-frequency, small-signal model of a voltage feedback amplifier. With the transconductance, g_m, selected as the reference parameter of interest, derive a general expression for the voltage gain, $A_v = V_O/V_S$. Approximate the final result for the special case of very large g_m.

Solution

1. The zero parameter voltage gain, A_{vo}, derives from an analysis of the circuit structure given in Figure 22.21b. The diagram differs from Figure 22.21a in that the current conducted by the controlled source branch has been nulled by open circuiting said branch. By inspection of the subject model,

$$A_{vo} = \frac{R_L}{R_L + R_F + R_S}$$

2. The diagram given in Figure 22.21c is appropriate to the computation of the return ratio, $T_S(g_m, Z_S, Z_L) = g_m Q_S(R_S, R_L)$ with respect to the critical transconductance, g_m. A comparison of the model at hand with the diagram in Figure 22.21a confirms that the controlled source, $g_m V$, is replaced by an independent current source, I_x, which flows in a direction identical to that of the controlled source

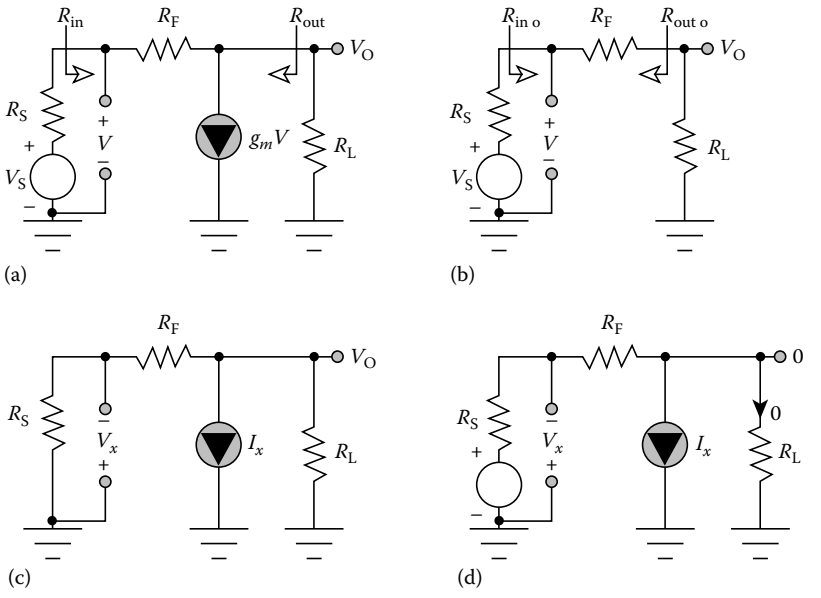


FIGURE 22.21 (a) The low-frequency, small-signal model of a voltage feedback amplifier. (b) The circuit used to evaluate the zero parameter ($g_m = 0$) gain. (c) The circuit used to evaluate the return ratio with respect to g_m. (d) The circuit used to computer the null return ratio with respect to g_m.

it supplants. The ratio, V_x/I_x, is to be computed, where V_x is developed, antiphase to V, across the branch that supports the original controlling voltage for the VCCS. A straightforward analysis produces

$$Q_S(R_S, R_L) = \frac{V_x}{I_x}\left(\frac{R_L}{R_L + R_F + R_S}\right)R_S \equiv A_{vo}R_S$$

3. The null return ratio, $T_R(g_m, Z_S, Z_L) = g_m Q_R(R_S, R_L)$, with respect to g_m is obtained from an analysis of the circuit in Figure 22.21d. Observe a nulled output voltage, with zero current flow through the load resistance, R_L. Observe further that the signal source voltage is nonzero. The specific value of this source voltage is not crucial, and is therefore not delineated. An analysis reveals

$$Q_R(R_S, R_L) = \frac{V_x}{I_x} = -R_F$$

4. Using Equation 22.49, the voltage gain of the circuit undergoing study is found to be

$$A_v = \frac{V_O}{V_S} = A_{vo}\left[\frac{1 + g_m(-R_F)}{1 + g_m R_S A_{vo}}\right]$$

which, for large g_m, reduces to

$$A_v \approx -\frac{R_F}{R_S}$$

22.2.6.5 Gain of a Current-Controlled Current Source

For the network in Figure 22.22a, the reference parameter is α_k, the gain associated with a CCCS. The zero parameter gain, $H(0, Z_S, Z_L)$, is evaluated by replacing this CCCS with an open circuit, as depicted in Figure 22.22b.

The volt–ampere characteristic equation of the branch in which the reference parameter is embedded is $I_k = \alpha_k I_j$, where I_j is the controlling current for the CCCS. Because the dependent variable in this volt-ampere characteristic is a branch current, the return and null return ratios are calculated by replacing the CCCS with a test current source, I_x. As indicated in Figure 22.22c and d, the polarity for I_x mirrors that of the current, I_k, flowing through the controlled branch. Let I_y be the resultant current conducted by the controlling branch, and let this current flow a direction opposite to the polarity of the original controlling current. Then, the current ratio, I_y/I_x, computed under the condition of nulled independent sources, is the function, $Q_S(Z_S, Z_L)$ in Equation 22.49. Similarly, and as suggested in Figure 22.22d, the ratio, I_y/I_x, under the condition of a nulled response, yields $Q_R(Z_S, Z_L)$.

22.2.6.6 Gain of a Voltage-Controlled Voltage Source

In the network of Figure 22.23a, the selected reference parameter is μ_k, the gain corresponding to a VCVS. The zero parameter gain, $H(0, Z_S, Z_L)$, is evaluated by replacing this VCVS with a short circuit, as per Figure 22.23b.

The volt–ampere characteristic equation of the dependent generator branch is $V_k = \mu_k V_j$, where V_j is the controlling voltage for the VCVS. Because the dependent variable in this volt–ampere expression is a branch voltage, the return and null return ratios are calculated by replacing the VCVS with a test voltage source, V_x, where as indicated in Figure 22.23c and d, the polarity of V_x is identical to that of the voltage,

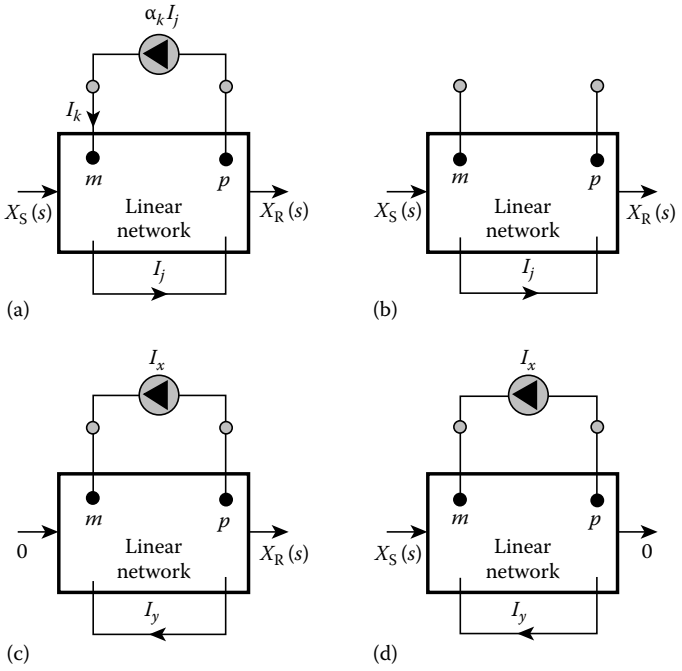


FIGURE 22.22 (a) Linear circuit for which the identified critical parameter is the current gain α_k associated with a CCCS. (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain, $H(0, Z_S, Z_L)$. (c) The ratio, I_y/I_x, is the function, $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, I_y/I_x, is the function, $Q_R(Z_S, Z_L)$, in Equation 22.49.

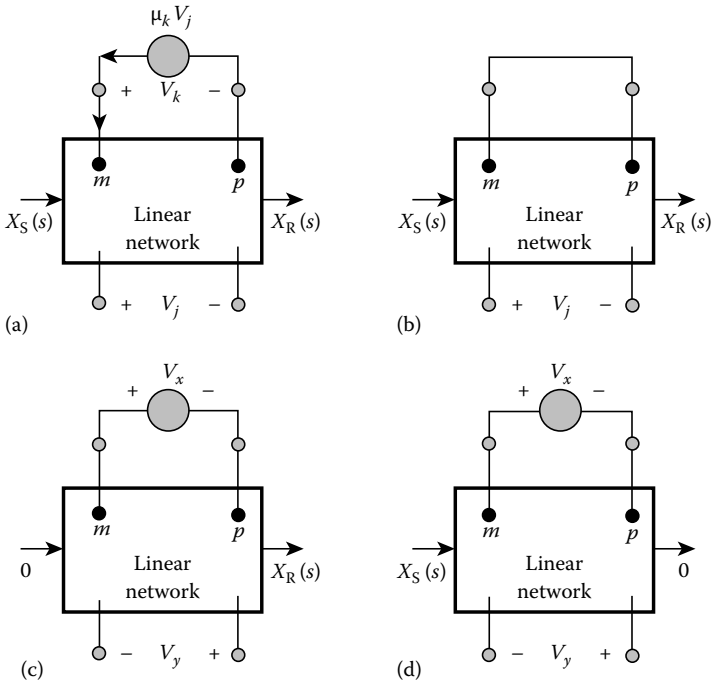


FIGURE 22.23 (a) Linear circuit for which the identified critical parameter is the voltage gain, μ_{k1} associated with a VCVS. (b) The ratio, $X_R(s)/X_S(s)$, is the zero parameter gain, $H(0, Z_S, Z_L)$. (c) The ratio, V_y/V_x, is the function $Q_S(Z_S, Z_L)$, in Equation 22.49. (d) The ratio, V_y/V_x, is the function, $Q_R(Z_S, Z_L)$, in Equation 22.49.

V_k, developed across the controlled branch. Let V_y be the resultant voltage established across the controlling branch, and let the polarity of this voltage be in a direction opposite to that of the original controlling voltage. Then, the voltage ratio, V_y/V_x, computed under the condition of nulled independent sources, is the function, $Q_S(Z_S, Z_L)$, in Equation 22.49. As suggested in Figure 22.23d, the voltage ratio, V_y/V_x, under the condition of a nulled response, yields $Q_R(Z_S, Z_L)$.

22.2.7 Evaluation of Driving Point Impedances

Having formulated generalized techniques for computing the return ratio and the null return ratio with respect to any of the six possible types of critical circuit parameters, the application of Equation 22.49 is established as a powerful and computationally expedient vehicle for evaluating any transfer function of any linear network. The only restriction limiting the utility of Equation 22.49 is that parameter P must be selected in such a way as to ensure that the zero parameter transfer function is finite and nonzero.

Equation 22.49 is commonly used to evaluate the voltage gain, current gain, transimpedance gain, or transadmittance gain of feedback and other types of complex circuitry. However, the expression is equally well suited to determining the driving point input impedance seen by the source impedance, as well as the driving point output impedance seen by the terminating load impedance. In fact, once the return ratios relevant to the gain of interest are found, these I–O impedances can be determined with minimal additional analysis.

Without loss of generality, the foregoing contention is explicitly demonstrated in conjunction with a transimpedance amplifier whose reference parameter is selected to be a branch impedance, Z_k. To this end, consider the circuit abstracted in Figure 22.24, for which the driving point input impedance, Z_{in}, is

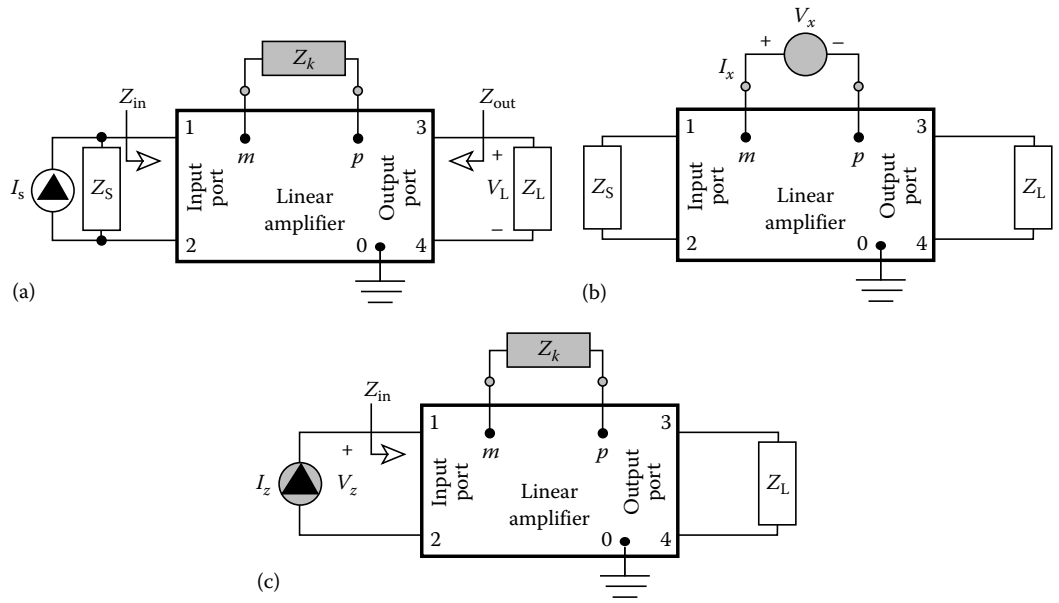


FIGURE 22.24 (a) A liner amplifier for which the input impedance is to be determined. (b) The circuit used for calculating the return ratio with respect to Z_k. (c) The circuit used for calculating the driving point input impedance.

to be determined. The input excitation is a current, I_S, and in response to this input, a signal voltage, V_L, is developed across the load impedance, Z_L. Using Equation 22.49, the I–O transimpedance, $Z_T(Z_k, Z_S, Z_L)$, is

$$Z_T(Z_k, Z_S, Z_L) = \frac{V_L(s)}{I_S(s)} = Z_T(0, Z_S, Z_L)\left[\frac{1 + Z_k Q_R(Z_S, Z_L)}{1 + Z_k Q_S(Z_S, Z_L)}\right] \tag{22.55}$$

where

$Z_T(0, Z_S, Z_L)$ is the circuit transimpedance for $Z_k = 0$, $Z_k Q_R(Z_S, Z_L)$ is the null return ratio with respect to Z_k

$Z_k Q_S(Z_S, Z_L)$ is the return ratio with respect to Z_k

For future reference, the circuit appropriate to the calculation of the function, $Q_S(Z_S, Z_L)$, is drawn in Figure 22.24b.

The input impedance derives from an analytical consideration of the cell depicted in Figure 22.24c, in which the Norton representation of the signal source has been supplanted by a test current source of value I_z. Note that the load impedance remains as the terminating element for the output port. The transfer relationship of interest is the ratio, V_z/I_z, which is the desired driving point input impedance, Z_{in}. Taking care to choose Z_k, the reference parameter for the gain enumeration, as the reference parameter for the input impedance determination, Equation 22.49 gives

$$Z_{in} = \frac{V_z}{I_z} = Z_{in\ o}\left[\frac{1 + Z_k Q_{RR}(Z_S, Z_L)}{1 + Z_k Q_{SS}(Z_S, Z_L)}\right] \tag{22.56}$$

In this expression, $Z_{in\ o}$ generally derives straightforwardly because it is the $Z_k = 0$ value of Z_{in}; that is, Z_{in} is evaluated for the special case of a nulled reference parameter. Such a null in the present situation is

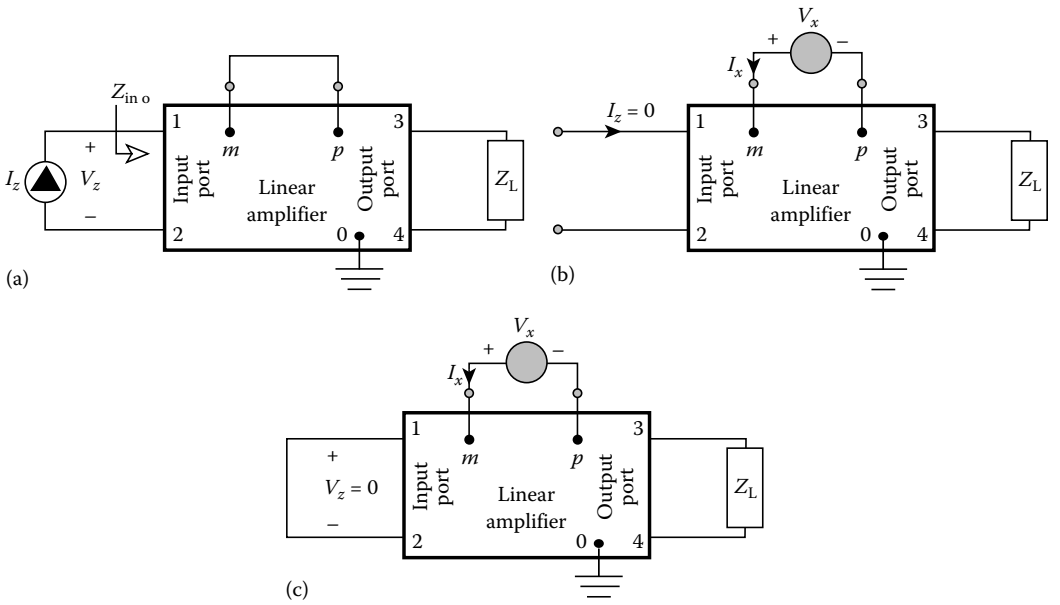


FIGURE 22.25 (a) The circuit used to evaluate the zero parameter driving point input impedance. (b) The computation, relative to input impedance, of the return ratio with respect to Z_k. (c) The computation, relative to input impedance, of the null return ratio with respect to Z_k.

equivalent to short-circuiting Z_k, as indicated in Figure 22.25a. The function, $Q_{SS}(Z_S, Z_L)$, is the delineated.

I_x/V_x ratio, for the case of a source excitation (I_z in the present case) set to zero. The pertinent circuit diagram is the structure in Figure 22.25b. This last circuit differs from the circuit, shown in Figure 22.24b, exploited to find $Q_S(Z_S, Z_L)$ in the gain relationship of Equation 22.50 in only one way: Z_S has been removed, and thus, effectively, Z_S has been set to an infinitely large value. It follows that

$$Q_{SS}(Z_S, Z_L) \equiv Q_S(\infty, Z_L) \tag{22.57}$$

In other words, a circuit analysis aimed toward determining $Q_{SS}(Z_S, Z_L)$ is unnecessary. Instead, $Q_{SS}(Z_S, Z_L)$ is found by evaluating $Q_S(Z_S, Z_L)$, which is already known from the gain analysis, at $Z_S = \infty$.

To evaluate $Q_{RR}(Z_S, Z_L)$, the foregoing I_x/V_x ratio is calculated for the case of zero response. In the present situation the response is the voltage, V_z, and accordingly, the appropriate circuit is depicted in Figure 22.25c. However, a comparison of the circuit at hand with the structure in Figure 22.24, which is exploited to evaluate the return ratio in the gain equation, indicates that it differs only in that Z_S is now constrained to zero to ensure $V_z = 0$. It is therefore apparent that in Equation 22.56

$$Q_{RR}(Z_S, Z_L) \equiv Q_S(0, Z_L) \tag{22.58}$$

Equation 22.56 is now expressible as

$$Z_{in} = \frac{V_z}{I_z} = Z_{in\,o}\left[\frac{1 + Z_k Q_S(0, Z_L)}{1 + Z_k Q_S(\infty, Z_L)}\right] \tag{22.59}$$

which is occasionally referred to as Blackman's formula [14].

Analogous considerations at the output port in the circuit of Figure 22.24a dictate a driving point output impedance, Z_{out}, of

$$Z_{out} = Z_{out\,o}\left[\frac{1 + Z_k Q_S(Z_S, 0)}{1 + Z_k Q_S(Z_S, \infty)}\right] \tag{22.60}$$

where, similar to $Z_{in\,o}$, $Z_{out\,o}$, the $Z_k = 0$ value of Z_{out}, must be finite and nonzero. Although the preceding two relationships were derived for the case in which the selected reference parameter is a branch impedance, both expressions are applicable for any reference parameter, P. In general,

$$Z_{in} = Z_{in\,o}\left[\frac{1 + PQ_S(0, Z_L)}{1 + PQ_S(\infty, Z_L)}\right] \tag{22.61a}$$

$$Z_{out} = Z_{out\,o}\left[\frac{1 + PQ_S(Z_S, 0)}{1 + PQ_S(Z_S, \infty)}\right] \tag{22.61b}$$

Example 22.5

Use the pertinent results of Example 22.4 to derive expression for the driving point input resistance, R_{in}, and the driving point output resistance, R_{out}, of the feedback amplifier in Figure 22.21a.

Solution

1. With g_m set to zero, an inspection of the circuit diagram in Figure 22.21b delivers

$$R_{in\,o} = R_F + R_L$$
$$R_{out\,o} = R_F + R_S$$

2. From the second step in the solution to Example 22.4, the function, $Q_S(R_S, R_L)$, to which the return ratio, $T_S(g_m, R_S, R_L)$ is directly proportional, was found to be

$$Q_S(R_S, R_L) = \left(\frac{R_L}{R_L + R_F + R_S}\right) R_S$$

It follows that

$$Q_S(0, R_L) = 0$$
$$Q_S(\infty, R_L) = R_L$$

Moreover,

$$Q_S(R_S, 0) = 0$$
$$Q_S(R_S, \infty) = R_S$$

3. Equations 22.61a and b resultantly yield

$$R_{in} = R_{in\,o}\left[\frac{1 + g_m Q_S(0, Z_L)}{1 + g_m Q_S(\infty, Z_L)}\right] = \frac{R_R + R_L}{1 + g_m R_L}$$

for the driving point input resistance and

$$R_{\text{out}} = R_{\text{out o}}\left[\frac{1 + g_m Q_S(R_S, 0)}{1 + g_m Q_S(R_S, \infty)}\right] = \frac{R_F + R_S}{1 + g_m R_S}$$

for the driving point output resistance.

22.2.8 Sensitivity Analysis

Yet another advantage of the Kron–Bode formula is its amenability to evaluating the impact exerted on a circuit transfer relationship by potentially large fluctuations in the reference parameter P. This convenience stems from the fact that parameter P is isolated in Equation 22.49; that is, $H(0, Z_S, Z_L)$, $Q_R(Z_S, Z_L)$, and (Z_S, Z_L) are each independent of P. A quantification of this impact is achieved by exploiting the sensitivity function,

$$S_P^H \triangleq \frac{\Delta H/H}{\Delta P/P} \tag{22.62}$$

which compares the per unit change in transfer function, $\Delta H/H$, resulting from a specified per unit change $\Delta P/P$ in a critical parameter. In particular, the notation in this definition is such that if H designates the transfer characteristic, $H(P_0, Z_S, Z_L)$, at the nominal parameter setting, $P = P_0$, $(H + \Delta H)$ signifies the perturbed characteristic, $H(P_0 + \Delta P, Z_S, Z_L)$ where P_0 is altered by an amount ΔP_0. Using Equation 22.49 and dropping the functional notation in Equations 22.53a and 22.53b, it can be demonstrated that

$$S_P^H \triangleq \frac{F_R - F_S}{F_R\left[F_S + (F_S - 1)\left(\dfrac{\Delta P}{P_0}\right)\right]} \tag{22.63}$$

where F_S and F_R are understood to be evaluated at the nominal parameter setting, $P = P_0$. It should be emphasized that unlike a more traditional sensitivity analysis, such as that predicated on the adjoint network [15], Equation 22.63 is easy to use manually and does not rely on an *a priori* assumption of small parametric changes.

References

1. G. Kron, *Tensor Analysis of Networks*, New York: John Wiley & Sons, 1939.
2. G. Kron, A set of principals to interconnect the solutions of physical systems, *J. Appl. Phys.*, 24, 965–980, Aug. 1953.
3. F. H. Branin, Jr., The relation between Kron's method and the classical methods of circuit analysis, *IRE Conv. Rec.*, part 2, 3–28, 1959.
4. F. H. Branin, Jr., A sparse matrix modification of Kron's method of piecewise analysis, *Proc. IEEE Int. Symp. Circuits Systems*, Boston, MA, pp. 383–386, 1975.
5. R. A. Rohrer, Circuit partitioning simplified, *IEEE Trans. Circuits Syst.*, 35, 2–5, Jan. 1988.
6. N. B. Rabbat and H. Y. Hsieh, A latent macromodular approach to large-scale sparse networks, *IEEE Trans. Circuits Syst.*, CAS-23, 745–752, Dec. 1976.
7. J. Choma, Jr., Signal flow analysis of feedback networks, *IEEE Trans. Circuits Syst.*, 37, 455–463, April 1990.
8. N. B. Rabbat, A. L. Sangiovanni-Vincentelli, and H. Y. Hsieh, A multilevel Newton algorithm with macromodeling and latency for the analysis of large-scale nonlinear circuits in the time domain, *IEEE Trans. Circuits Syst.*, CAS-26, 733–741, Sep. 1979.

9. N. Balabanian and T.A. Bickart, *Electrical Network Theory*, New York: John Wiley & Sons, 1969, pp. 73–77.
10. S. J. Mason, Feedback theory—Some properties of signal flow graphs, *Proc. IRE*, 41, 1144–1156, Sep. 1953.
11. S. J. Mason, Feedback theory—Further properties of signal flow graphs, *Proc. IRE*, 44, 920–962, July 1956.
12. H. W. Bode, *Network Analysis and Feedback Amplifier Design*, New York: D. Van Nostrand Company, 1945 (reprinted, 1953), chap. 4.
13. J. Choma, Jr., *Electrical Networks: Theory and Analysis*, New York: Wiley Interscience, 1985, pp. 590–598.
14. S. K. Mitra, *Analysis and Synthesis of Linear Active Networks*, New York: John Wiley & Sons, 1969, p. 206.
15. S. W. Director and R. A. Rohrer, The generalized adjoint network and network sensitivities, *IEEE Trans. Circuit Theory*, CT-16, 318–328, 1969.

23

Tableau and Modified Nodal Formulations

23.1 Introduction **23**-1
23.2 Nodal and Mesh Formulations **23**-1
23.3 Graphs and Tableau Formulation **23**-5
23.4 Modified Nodal Formulation **23**-8
23.5 Nonlinear Elements **23**-14
23.6 Nodal Analysis of Active Networks **23**-17
Conclusion **23**-20
References **23**-20

Jiri Vlach
University of Waterloo

23.1 Introduction

Network analysis is based on formulation of the relevant equations and on their solutions. Various approaches are possible. If we wish to get as much theoretical information as possible, we may resort to hand analysis and keep the elements as variables (literal parameters). In such cases, it is an absolute necessity to use a method that leads to the smallest possible number of equations. If we plan to use a computer, then we can accept methods which lead to larger systems, but the methods must be relatively easy to program. The purpose of this chapter is to give an overview of the various possibilities and point out advantages and disadvantages. Many more details are available in Refs. [1,2].

Section 23.2 is a summary of the nodal and mesh formulations. We review them because they are the best ones for analysis of small networks. Tableau formulation, given in Section 23.3, is very general, but requires special solution routines, probably not available to the reader. Section 23.4 describes the best method for computerized solutions; it is used in many commercial simulators. If nonlinear elements are involved, then iterative solution methods must be used; an introduction on how to deal with nonlinear elements is given in Section 23.5. Finally, Section 23.6 presents a method that is suitable for hand solutions of active networks and which automatically leads to the smallest system of equations.

23.2 Nodal and Mesh Formulations

Classical methods use two types of network equations formulation: the nodal and the mesh. The first one is based on Kirchhoff's current law (KCL): the sum of currents flowing away from a node is equal to zero. The mesh method is based on Kirchhoff's voltage law (KVL): the sum of voltages around any loop is equal to zero.

For simple problems, both methods are about equivalent, but nodal formulation is more general. The mesh formulation is suitable only for planar networks: It must be possible to draw the network without

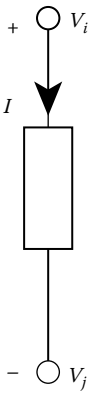


FIGURE 23.1 Definition of positive current direction with respect to the voltage across the element.

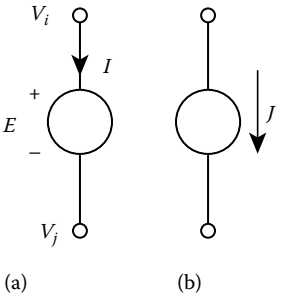


FIGURE 23.2 (a) Direction of positive current through an independent voltage source. (b) Direction of the current through an independent current source.

any element crossing over any other element. Many practical networks are planar, but it is not always easy to see that it is actually the case.

We first introduce some definitions. A positive current flows from a terminal with a higher potential to a terminal with a lower potential. This is sketched on the two-terminal element in Figure 23.1. If we use this definition, then the product of the current and of the voltage across the element, $V_i - V_j$, expresses the power consumed by the element. If the current flows in opposite direction, then the element is delivering power.

We will use the previous definition of a positive current for all elements of the network, irrespective of what their role eventually may be. Thus, a positive current through an independent voltage source will flow from the more positive terminal to the less positive terminal, as sketched in Figure 23.2a. The current flowing through an independent current source is indicated on its symbol, Figure 23.2b, but the voltage across it is not defined; it depends on the network.

The nodal formulation uses the principle that the sum of currents at any node must be equal to zero at any instant of time. To apply this rule in an efficient way, we realize that before we solve the equations, we do not know which way the currents will actually flow. All we know is that if a node is more positive than all the other nodes, then all currents must flow away from this node.

In nodal formulation, the unknowns are nodal voltages and the equations express the sum of currents flowing away from the node. To write the equations we use element admittances: in Laplace transform $Y_C = sC$ for a capacitor, $Y_L = 1/sL$ for an inductor, and $Y_G = G = 1/R$ for a resistor. It is advantageous to use G, because we thus avoid fractions in the equations.

We demonstrate how to set up the equations by considering the network in Figure 23.3. The nodal voltages are denoted V_1 and V_2 and ground (the lower line) is considered to be at zero potential. We do

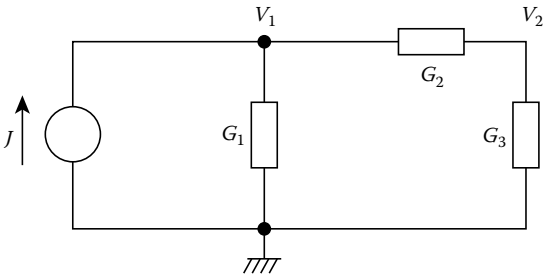


FIGURE 23.3 Example for nodal formulation.

not know which of these nodes is more positive, but we can assume that any node we consider at a given moment is the most positive one. This assumption has the consequence that all currents must flow away from this node. For the given network, the current through G_1 will flow down and its value will be $I_{G_1} = G_1 V_1$, current through G_2 will flow from left to right and will be $I_{G_2} = G_2(V_1 - V_2)$. Current from the independent current source flows into the node and thus must be subtracted. Together, the sum of the currents at node 1 will be zero:

$$G_1 V_1 + G_2(V_1 - V_2) - J = 0$$

Moving to the second node, we still do not know which node is more positive, but we can still make the assumption that now it is this node that is the most positive one. If we make such an assumption, then all currents must flow away from this node: The current through G_3 will be $I_{G_3} = G_3 V_2$ and will flow down, the current through G_2 will flow from right to left and will be $I_{G_2} = G_2(V_1 - V_2)$. In this expression, the first voltage within the parentheses must be the assumed higher potential. This is expressed by the equation

$$G_3 V_2 + G_2(V_2 - V_1) = 0$$

It is advantageous to put the equations into matrix form, with the known independent source transferred to the right side:

$$\begin{bmatrix} G_1 + G_2 & -G_2 \\ -G_2 & G_2 + G_3 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \begin{bmatrix} J \\ 0 \end{bmatrix}$$

The preceding steps were simple because we selected elements that can be handled by this formulation. Unfortunately, many practical elements are not expressed in terms of currents. For instance, a voltage source connected between nodes i and j, with its positive reference on node i, is described by the equation

$$V_i - V_j = E$$

A positive current does flow through such an element from i to j, but is not available in its defining equation. In fact, all voltage sources, independent or dependent, will create this problem. Another element which cannot be handled directly is a short circuit. It is described by the equation

$$V_i - V_j = 0$$

and current is not a part of its definition.

We can always use transformations by applying various theorems such as the Thévenin and Norton transformations or source splitting, and eventually arrive at a network in which all elements have voltage as the independent variable. Such transformations are practical for hand analysis, but are not advantageous for computerized solutions. This is the reason why other formulations have been invented.

Consider next the mesh equations where we use the KVL and impedances of the elements: $Z_L = sL$ for the inductor, $Z_C = 1/sC$ for the capacitor, and R for the resistor. In this formulation, we sum the voltages across the elements in a given closed loop. Because this method is suitable for planar networks only, we usually use the concept or circulating mesh currents, indicated on the network in Figure 23.4. The currents I_1 and I_2 create voltage drops across the resistors. When considering the first mesh, we take the current I_1 as a positive one. The voltage across R_1 is $V_{R_1} = R_1 I_1$. The voltage across R_2 is $V_{R_2} = R_2(I_1 - I_2)$ and the voltage source contributes a value E to the equation. According to our earlier definition, a positive current flows from plus to minus, but I_1 actually goes in the opposite direction through the voltage source. Thus, the voltage across E must be taken with a negative sign and the sum of voltages around the first mesh is

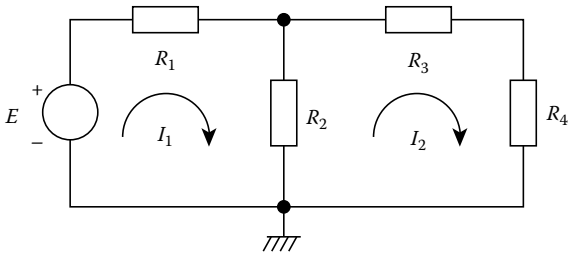


FIGURE 23.4 Example for mesh formulation.

$$R_1 I_1 + R_2(I_1 - I_2) - E = 0$$

When we move to the second mesh, we consider the current I_2 as positive and the sum of voltage drops around the second mesh is

$$R_2(I_2 - I_1) + R_3 I_2 + R_4 I_2 = 0$$

The equations can be collected into one matrix equation

$$\begin{bmatrix} R_1 + R_2 & -R_2 \\ -R_2 & R_2 + R_3 + R_4 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \begin{bmatrix} E \\ 0 \end{bmatrix}$$

Each of these fundamental formulations has its problems.

In nodal formulation, we can deal directly with the following elements:
- Current source, J
- Conductance, $G = 1/R$
- Capacitor admittance, sC
- Voltage controlled current source, VC
- Inductor admittance, $1/sL$

In mesh formulation, we can deal directly with the elements:
- Voltage source, E
- Resistor, R
- Inductor impedance, sL
- Current controlled voltage source, CV
- Capacitor impedance, $1/sC$.

All other elements create problems and must be dealt with by the Thévenin and Norton theorems and/or source splitting.

As an example, we take the network in Figure 23.5a. It is directly suitable for mesh formulation, but we demonstrate both. For simplicity, all resistors have unit values.

The mesh formulation, with the indicated circulating currents I_1 and I_2, leads to the equations

$$\begin{aligned} (R_1 + R_2)I_1 - R_2 I_2 &= E \\ -R_2 I_1 + (R_2 + R_3)I_2 + 3I_1 &= 0 \end{aligned}$$

Inserting numerical values

$$\begin{aligned} 2I_1 - I_2 &= E \\ 2I_1 + 2I_2 &= 0 \end{aligned}$$

The solution is $I_1 = E/3$, $I_2 = -E/3$ and $V_1 = R_2(I_1 - I_2) = 2E/3$.

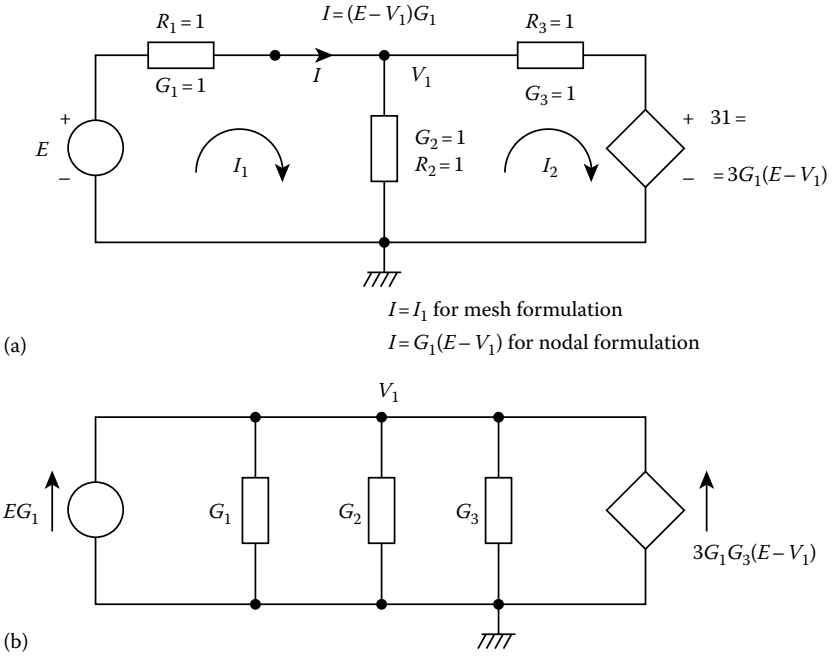


FIGURE 23.5 (a) Example of a network suitable for mesh formulation. (b) Modification of the network in Figure 23.5a to be suitable for nodal formulation.

To use nodal formulation, we must apply several transformation steps. First, we must express the controlling currents as $I = G_1(E - V_1) = E - V_1$ and replace I in the definition of the current controlled voltage source. This has been done in the figure. Afterward, applying Thévenin–Norton transformation we change the voltage sources, in series with resistors, into current sources, in parallel with the same resistors. We get the network in Figure 23.5b. It has only one node for which the balance of current is

$$(G_1 + G_2 + G_3)V_1 - EG_1 - 3G_1G_3(E - V_1) = 0$$

Inserting numerical values and solving, we get the same V_1 as previously.

If we have a mixture of elements, such transformations will always be lengthy, will require redrawings, and can lead to errors. To reduce the chance of such errors to a minimum, in computer applications we need formulations that avoid transformations and use descriptions of the elements as they are given. This is done in both the tableau and nodal formulations, the subjects of the following sections.

Although the nodal and mesh formulations are not always easy to apply, we must stress that they are the best ones for hand solutions. They may require several steps of transformations and redrawings, but ultimately they lead to the smallest possible systems of equations.

23.3 Graphs and Tableau Formulation

Tableau is the most general formulation because the solution simultaneously provides the voltages across all elements, the currents through all elements, and all nodal voltages. The difficulty is that tableau leads to such large systems of equations that complicated sparse matrix solvers are an absolute necessity. Most readers will not have access to such routines, therefore, we will explain its properties only to the extent necessary for understanding. We advise the reader not to use it.

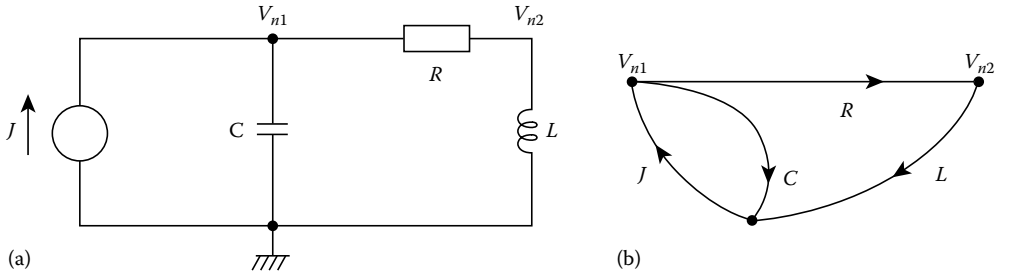


FIGURE 23.6 (a) A simple network and (b) its graph.

Tableau formulation needs for its construction the concept of graphs and the concept of the incidence matrix. Consider the network in Figure 23.6a. A graph of the network replaces each element by a line. We will use oriented graphs, with arrows, because they can be identified with the flow of currents. In all passive elements, the current can flow in any direction and the orientation of the graph is entirely our choice. We do not have such freedom when we consider sources. The direction of the current through the current source is given by the arrow marked at its symbol and we use the same direction in the graph. For the voltage source, the direction of the graph will be from plus to minus, in agreement with our previous explanations. Each node is marked by the node voltage and the line representing the element is given the name of the element. Following these rules, we have constructed the graph in Figure 23.6b. Because the directions of the arrows are the assumed directions of currents, we can write KCL for the two nodes: positive direction is away from the node, negative is into the node. The sums of currents for the nodes are

$$\begin{aligned} -I_{\mathrm{J}} + I_{\mathrm{C}} + I_{\mathrm{R}} &= 0 \\ -I_{\mathrm{R}} + I_{\mathrm{L}} &= 0 \end{aligned}$$

This can also be summarized in one matrix equation

$$\begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} I_{\mathrm{J}} \\ I_{\mathrm{C}} \\ I_{\mathrm{R}} \\ I_{\mathrm{L}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

or

$$\mathbf{AI} = 0 \tag{23.1}$$

The matrix **A** is called the incidence matrix. It has as many rows as there are ungrounded nodes, and as many columns as the number of elements. Note that $+1$ in any given row indicates that we expect the current to flow away from the node, -1 means the opposite.

Still more information can be extracted from this matrix. Denote the nodal voltages by subscripts n in V_{n1} and V_{n2}, as done in Figure 23.6. The voltages across the elements will have as subscripts the names of the elements. We can write the following set of equations which couple the voltages across the elements with the nodal voltages:

$$\begin{aligned} V_{\mathrm{J}} &= -V_{n1} \\ V_{\mathrm{C}} &= V_{n1} \end{aligned}$$

$$V_{\mathrm{R}} = V_{n1} - V_{n2}$$
$$V_{\mathrm{L}} = V_{n2}$$

In matrix form, this is equivalent to

$$\begin{bmatrix} V_{\mathrm{J}} \\ V_{\mathrm{C}} \\ V_{\mathrm{R}} \\ V_{\mathrm{L}} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{n1} \\ V_{n2} \end{bmatrix}$$

The matrix is the transpose of the incidence matrix and we can generalize

$$\mathbf{V}_{\mathrm{el}} - \mathbf{A}^{I}\mathbf{V}_{n} = 0 \tag{23.2}$$

Complete formulation needs expressions that couple the element currents and the element voltages. Writing them in the same sequence as for the graph, and using Laplace transformation, we have

$$I_{\mathrm{J}} = J$$
$$I_{\mathrm{C}} = sCV_{\mathrm{C}}$$
$$V_{\mathrm{R}} = RI_{\mathrm{R}}$$
$$V_{\mathrm{L}} = sLI_{\mathrm{L}}$$

For matrix notation, we need an expression that, with a proper choice of entries, will cover all possible elements. Such an expression is

$$YV_{\mathrm{el}} + ZI_{\mathrm{el}} = W$$

For instance, if we consider the current source, we set $Y = 0$, $Z = 1$ and $W = J$, which gives the preceding equation. Similar choices can be made for the other elements.

The KCL equation, $\mathbf{AI} = \mathbf{0}$, the KVL equation, $\mathbf{V}_{\mathrm{el}} - \mathbf{A}^{\mathrm{T}}\mathbf{V}_{n} = \mathbf{0}$, and the previous equation $\mathbf{YV}_{\mathrm{el}} + \mathbf{ZI}_{\mathrm{el}} = \mathbf{W}$ are collected in one matrix equation. Any sequence can be used; we have chosen

$$\begin{aligned} \mathbf{V}_{\mathrm{el}} - \mathbf{A}^{\mathrm{T}}\mathbf{V}_{n} &= 0 \\ \mathbf{YV}_{\mathrm{el}} + \mathbf{ZI}_{\mathrm{el}} &= \mathbf{W} \\ \mathbf{AI}_{\mathrm{el}} &= 0 \end{aligned} \tag{23.3}$$

and in matrix form

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} & -\mathbf{A}^{\mathrm{T}} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\mathrm{el}} \\ \mathbf{I}_{\mathrm{el}} \\ \mathbf{V}_{n} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{W} \\ \mathbf{0} \end{bmatrix} \tag{23.4}$$

Once the incidence matrix is available, writing this matrix equation is actually quite simple. First determine its size: it will be twice the number of elements plus the number of nodes. For our example, it will be 10. The system equation is in Figure 23.7 where all zero entries were omitted to clearly show the structure. In the top partition is a unit matrix and the negative of the transpose of the **A** matrix. In the bottom partition is the incidence matrix, **A**. The middle portion is filled, element by element, using

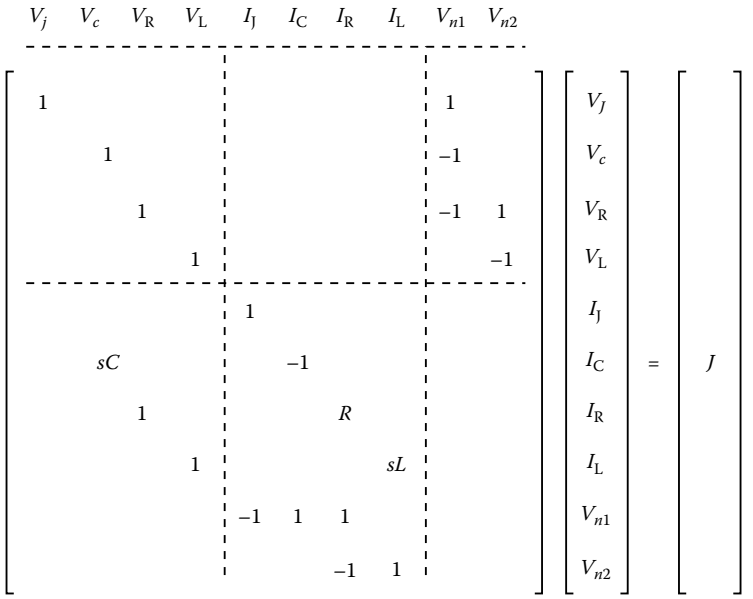


FIGURE 23.7 Tableau formulation for the network in Figure 23.6.

the previous element equation. For better understanding, it is a good idea to write the variables above the matrix, as shown, because each column of the matrix is multiplied by the variable which appears above it.

We have used this simple example to point out the main difficulty of the tableau formulation: the system becomes very large. In nodal formulation, this problem would lead to only two equations. However, the tableau system matrix has many zeros and is said to be sparse. Sparse systems are always solved by special routines which, roughly speaking, do not store the zeros and do not operate on them. Such codes are quite difficult to write, and in tableau we have the additional difficulty that the matrix has a complicated structure. We discussed this formulation more as a warning instead of a recommendation. Unless a suitable sparse matrix solver is already available, this formulation should be avoided.

23.4 Modified Nodal Formulation

Modified nodal formulation is an extension of the nodal formulation and is the method of choice for computerized analysis. It is used in most commercial simulators, and we will explain it in considerable detail.

When nodal formulation is taught in schools, inductors are usually taken as admittances, $Y_L = 1/sL$. This is fine, as long as we work by hand and derive the network function. For instance, nodal equations for the network in Figure 23.8 would be written in the form

$$\left(G_1 + sC_1 + \frac{1}{sL}\right)V_1 - \frac{1}{sL}V_2 = J$$
$$-\frac{1}{sL}V_1 + \left(G_2 + sC_2 + \frac{1}{sL}\right)V_2 = 0$$

However, this creates a problem for computerized solutions. Multiplication by s represents differentiation in the Laplace transform and $1/s$ represents integration. As a result, the two equations are actually a set of two integro-differential equations, and we do not normally have methods to solve them directly in such a

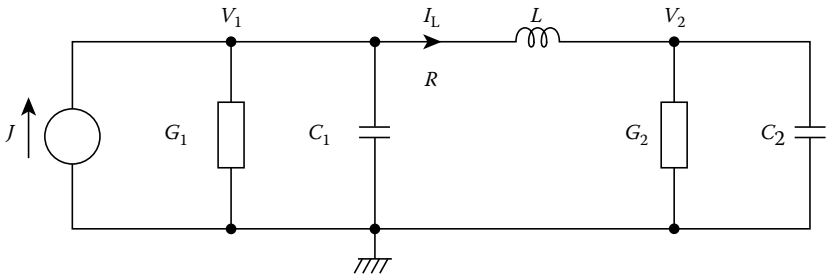


FIGURE 23.8 Example for modified nodal formulation.

form. In all computerized methods, we use integration of systems of first-order differential equations and the preceding equations cannot be arranged into such a form. What we need is a method which will keep all frequency-dependent elements in the form sC or sL, with the variable s in the numerator. Such possibility exists if we take into account a new variable, the current through the inductor. Writing KCL for the two nodes of Figure 23.8

$$(G_1 + sC_1)V_1 + I_L = J$$
$$(G_2 + sC_2)V_2 - I_L = 0$$

We now have two equations but three variables. What we have not used yet is an expression which couples the voltages across the inductor with the current through it. The relationship is $V_1 - V_2 = sLI_L$, but because we do not know any of these three variables, we transfer everything to the left and write the last equation

$$V_1 - V_2 = sLI_L = 0 \tag{23.5}$$

All three can be put into a matrix form

$$\begin{bmatrix} G_1 + sC_1 & 0 & 1 \\ 0 & G_2 + sC_2 & -1 \\ 1 & -1 & -sL \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_L \end{bmatrix} = \begin{bmatrix} J \\ 0 \\ 0 \end{bmatrix}$$

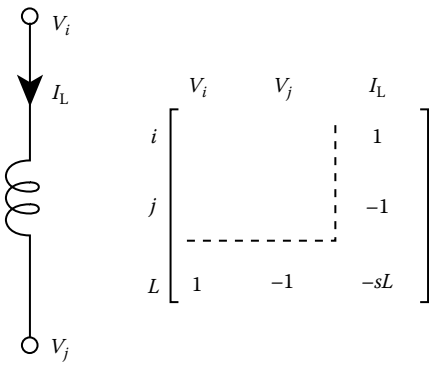


FIGURE 23.9 Stamp of an inductor.

In this equation, the nodal portion is the 2×2 matrix in the upper left corner and information about the inductor is collected in the right-most column and the lowest row. A larger network will have a larger nodal portion, but we still increase the matrix by one row and column for each inductor. This can be prepared as a general stamp as shown in Figure 23.9. The previous matrix is the empty box, separated by the dashed lines, the voltages and the current above the stamp indicate the variables relevant to the inductor, while on the left the letters i and j give the rows (node numbers) where L is connected. Should we have a network with two inductors, we add one row and one column for each.

The next element we take is the voltage-controlled current source; it was already mentioned is Section 23.2 as an

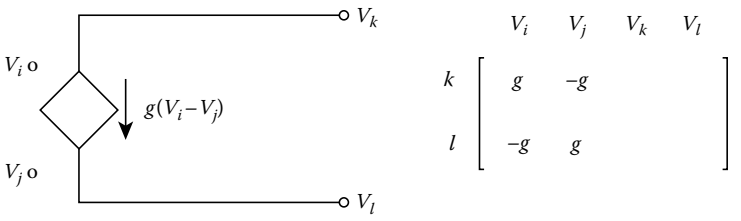


FIGURE 23.10 Stamp for a voltage-controlled current source.

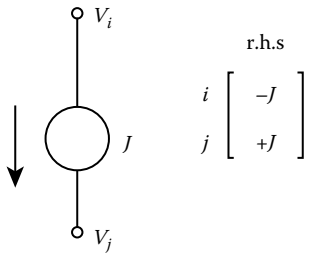


FIGURE 23.11 Stamp for an independent current source.

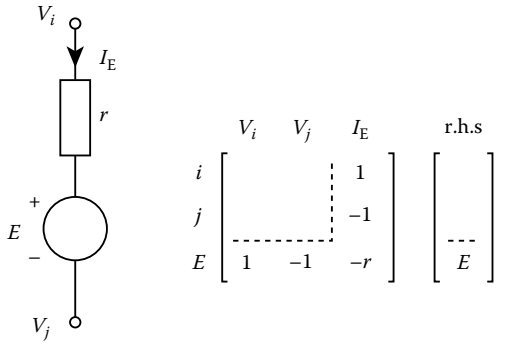


FIGURE 23.12 Stamp for an independent voltage source. The resistor value can be zero.

element which can be taken into the nodal formulation. The *VC* is shown in Figure 23.10. It adds the current $g(V_i - V_j)$ to node k and subtracts the same current at node l. It permits us to write the stamp, Figure 23.10. The element influences the balance of currents at node i and j, and the variables which multiply its transconductance are V_i and V_j.

Network theory defines two independent sources: the current and the voltage source. The current source can be taken into consideration in the right-hand side (r.h.s.) of the nodal portion; its stamp is in Figure 23.11. The independent voltage source cannot be taken directly and we must add its current as a new variable. Consider the source with a resistor in series, shown in Figure 23.12. The voltage relationships are

$$V_i - V_j - rI_E = E \tag{23.6}$$

The current I_E adds to the balance of currents at node i, because it flows away from it. It is subtracted from the balance of currents at node j. The current is taken as a new variable and the equation is attached to previous equations. The stamp is as shown. It is, in fact, a combined stamp for several elements. If we set $r = 0$, we have an ideal voltage source. If we set both r and E equal to zero, we have a stamp for a short circuit. For better understanding, consider the example in Figure 23.13. The network has two ungrounded nodes for which we can write the nodal equations:

$$(G_1 + sC_1)V_1 + I_E = 0$$
$$(G_2 + sC_2)V_2 - I_E = 0$$

To these equations, we must add the equation describing the properties of the ideal voltage source,

$$V_i - V_j = E \tag{23.7}$$

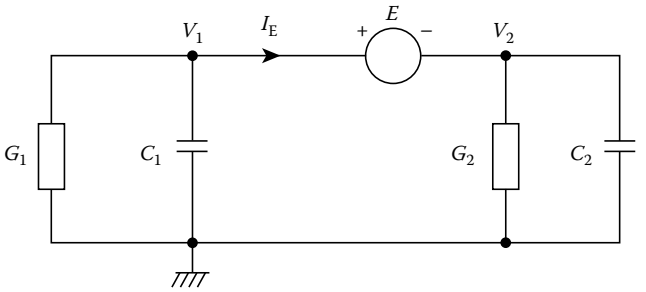


FIGURE 23.13 Example with a floating voltage source.

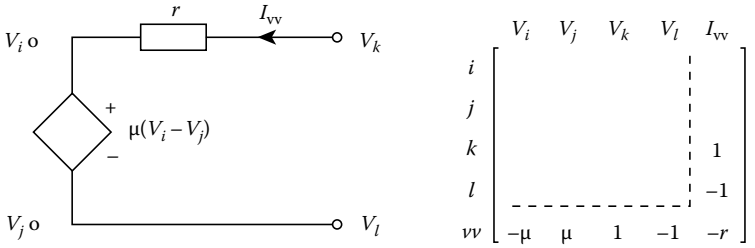


FIGURE 23.14 Stamp for a voltage-controlled voltage source. The resistor value can be zero.

Collecting into matrix form

$$\begin{bmatrix} G_1 + sC_1 & 0 & 1 \\ 0 & G_2 + sC_2 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_E \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix}$$

We still need stamps for the remaining three dependent sources. The voltage controlled voltage source, *VV*, is shown in Figure 23.14. For generality, we added the internal resistor. The output is described by the equation

$$\mu(V_i - V_j) + rI_{VV} = V_k - V_l \tag{23.8}$$

None of the voltages is known, so we transfer everything to the left side. Input terminals do not influence the balance of currents, but the output terminals do. At node k we must add I_{VV}, and subtract the same at node l. The stamp is in Figure 23.14.

The current controlled current source, *CC*, is shown in Figure 23.15. The input terminals are short circuited and

$$V_i - V_j = 0 \tag{23.9}$$

No information is available about the output voltages, but we know that the current is α times the input current, and thus only one additional variable is needed. Balances of currents are influenced at all four nodes: positive I at node i, negative at node j, positive current αI at node k, and negative αI at node l. The stamp is in Figure 23.15, where the added column takes care of the currents and the additional row describes the properties of the short circuit.

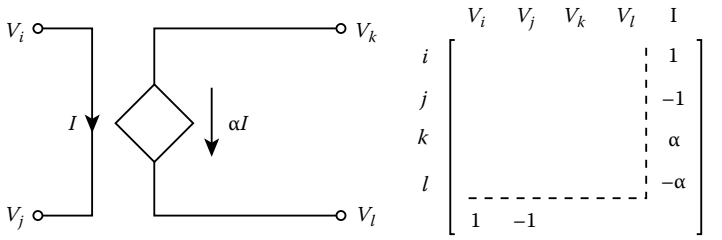


FIGURE 23.15 Stamp for a current-controlled current source.

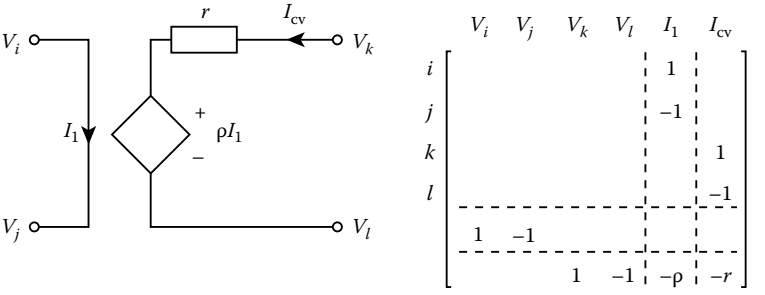


FIGURE 23.16 Stamp for a current-controlled voltage source. The resistor value can be zero.

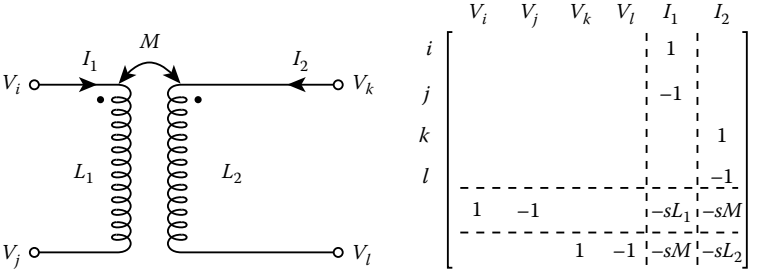


FIGURE 23.17 Stamp for a transformer.

The most complicated dependent source is the current-controlled voltage source, *CV*, shown in Figure 23.16. We can consider it as a combination of a short circuit and a voltage source. The equation for the short circuit is the same as for the *CC*. The output is defined by the equation

$$V_k - V_l = \rho I_1 + r I_{CV} \tag{23.10}$$

and because none of the variables is known, we transfer everything to the left. This element adds two rows and two columns to the previously defined matrix. Its stamp is in Figure 23.16. As before, the internal resistor r can be set equal to zero.

Modified nodal formulation easily takes into account a transformer (see Figure 23.17). It is described by the equations

$$\begin{aligned} V_i - V_j &= sL_1 I_1 + sMI_2 \\ V_k - V_l &= sL_1 I_1 + sL_2 I_2 \end{aligned} \tag{23.11}$$

None of the variables is known, and we transfer everything to the left. The currents influence the balance of currents at all nodes. The stamp of the transformer is in Figure 23.17.

The last element we consider is an ideal operational amplifier. It is sometimes taken as a voltage-controlled voltage source with very high gain, but it is preferable to have a stamp that can take into account ideal properties as well. The element is shown in Figure 23.18. The terminal l is usually grounded, but we will keep it floating to make the stamp more general. No current flows into the device at the input terminals. The output equation is

$$V_k - V_l = A(V_i - V_j) \tag{23.12}$$

Because a computer cannot handle infinity, it is advantageous to introduce the inverted gain

$$B = -1/A \tag{23.13}$$

and modify the previous equation to

$$V_i - V_j + BV_k - BV_l = 0 \tag{23.14}$$

This equation is attached to the set of equations and the balance of currents is influenced at nodes k and l. This leads to the stamp in Figure 23.18. If we set $B = 0$, the operational amplifier becomes ideal, with no approximation.

An example will show how the stamps are used. Consider the network in Figure 23.19. It has no practical application, but serves well for the demonstration of how to set up the modified nodal

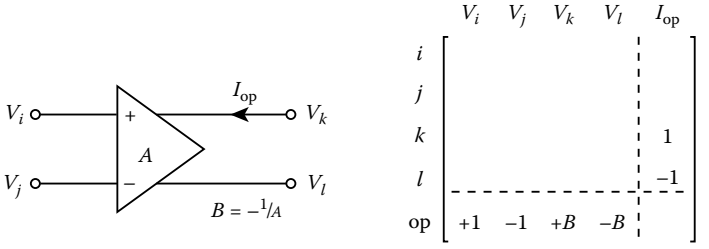


FIGURE 23.18 Stamp for an ideal operational amplifier.

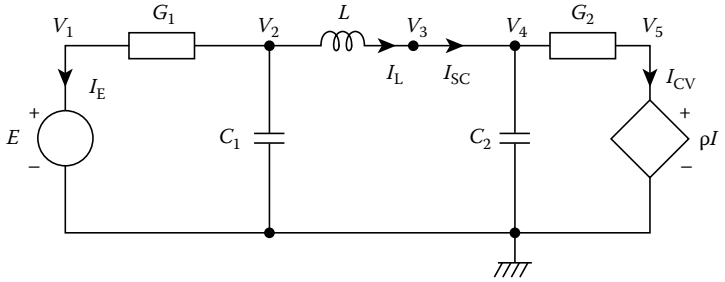


FIGURE 23.19 Example showing the use of modified nodal formulation.

matrix. A short circuit, indicating the controlling current of the current-controlled voltage source is taken into account by increasing the number of nodes. The network has five nongrounded nodes, and thus the dimension of its nodal portion will be 5. The voltage source will increase the system matrix by one row and one column, the inductor also, and the *CV* will need two more rows and columns; altogether the matrix will be 9×9. Write first the nodal portion by disregarding entirely the other elements. This creates the upper left partition. Using the stamps we add first the voltage source, then the inductor, next the short circuit, and finally the current-controlled voltage source. The system matrix is

Modified nodal formulation is the most important method for computer applications. The reader can find additional information in the books [1,2].

$$\left[\begin{array}{ccccc|cccc}
G_1 & -G_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-G_1 & G_1+sC_1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & sC_2+G_2 & -G_2 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & -G_2 & G_2 & 0 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 1 & -1 & 0 & 0 & 0 & -sL & 0 & 0 \\
\hline
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -\rho & 0
\end{array}\right]
\left[\begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ I_E \\ I_L \\ I_{SC} \\ I_{CV} \end{array}\right]
=
\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ E \\ 0 \\ 0 \\ 0 \end{array}\right]$$

23.5 Nonlinear Elements

In previous sections, we used the Laplace transform to explain the various methods of formulation. Because we dealt with linear elements, the systems of equations were linear and it was possible to cast them into matrix forms.

If we must consider nonlinear elements, we face many restrictions. The Laplace transform cannot be used. Various concepts based on it, like the network functions, the poles and the zeros, cannot be applied. Only two types of analysis are available:

The dc solution (operating point)

Time-domain solution for a given input signal

Once we have nonlinear elements, we cannot write the equations in matrix form; all we can do is write KCL equations. We must also find another method for the solution of such nonlinear equations.

Consider two differentiable equations in two unknowns

$$\begin{aligned} f_1(v_1, v_2) &= 0 \\ f_2(v_1, v_2) &= 0 \end{aligned} \tag{23.15}$$

The functions can be expanded into Taylor series and the series truncated after the linear terms:

$$f_1(v_1 + \Delta v_1,\, v_2 + \Delta v_2) = f_1(v_1,\, v_2) + \frac{\partial f_1}{\partial v_1}\Delta v_1 + \frac{\partial f_1}{\partial v_2}\Delta v_2 + \cdots = 0$$

$$f_2(v_1 + \Delta v_1,\, v_2 + \Delta v_2) = f_2(v_1,\, v_2) + \frac{\partial f_2}{\partial v_1}\Delta v_1 + \frac{\partial f_2}{\partial v_2}\Delta v_2 + \cdots = 0$$

We are not considering higher-order terms, so the equation will not be exactly zero, but we can still try to find Δv_1 and Δv_2. Transferring the known function values to the right

$$\frac{\partial f_1}{\partial v_1}\Delta v_1 + \frac{\partial f_1}{\partial v_2}\Delta v_2 = -f_1(v_1, v_2)$$
$$\frac{\partial f_2}{\partial v_1}\Delta v_1 + \frac{\partial f_2}{\partial v_2}\Delta v_2 = -f_2(v_1, v_2)$$

This is a system of linear equations, and we can rewrite it in matrix form

$$\begin{bmatrix} \dfrac{\partial f_1}{\partial v_1} & \dfrac{\partial f_1}{\partial v_2} \\ \dfrac{\partial f_2}{\partial v_1} & \dfrac{\partial f_2}{\partial v_2} \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} = \begin{bmatrix} -f_1(v_1, v_2) \\ -f_2(v_1, v_2) \end{bmatrix} \tag{23.16}$$

The matrix on the left is called the Jacobian; on the right is the negative of the functions. Once this linear system is solved, we can get new values of the variables by writing

$$v_1^{(i+1)} = v_1^{(i)} + \Delta v_1^{(i)}$$
$$v_2^{(i+1)} = v_2^{(i)} + \Delta v_2^{(i)} \tag{23.17}$$

In this equation, we added the superscript to indicate iteration. The process is repeated until all Δv_i become sufficiently small. This iterative method is usually referred to as the Newton–Raphson iteration and is written in the form

$$\mathbf{J}(\mathbf{v}^{(i)})\ \Delta\mathbf{v}^{(i)} = -\mathbf{f}(\mathbf{v}^{(i)})$$
$$\mathbf{v}^{(i+1)} = \mathbf{v}^{(i)} + \Delta\mathbf{v}^{(i)} \tag{23.18}$$

Suppose that we now take the network in Figure 23.3, consider the conductances G_1 and G_3 as linear and replace G_2 by a nonlinear function

$$i = g(v_{\text{el}})$$

where v_{el} is the voltage across this element,

$$v_{\text{el}} = v_1 - v_2$$

and g represents a nonlinear function. The two KCL equations are

$$G_1 v_1 + g(v_{\text{el}}) - J = 0$$
$$-g(v_{\text{el}}) + G_3 v_2 = 0$$

For the Newton–Raphson equation, we need the derivatives with respect to v_1 and v_2. Consider now only the nonlinear element. Using the chain rule of differentiation we can write

$$\frac{\partial g(v_{\text{el}})}{\partial v_1} = \frac{\partial g(v_{\text{el}})}{\partial v_{\text{el}}}\ \frac{\partial v_{\text{el}}}{\partial v_1} = +\frac{\partial g(v_{\text{el}})}{\partial v_{\text{el}}}$$
$$\frac{\partial g(v_{\text{el}})}{\partial v_2} = \frac{\partial g(v_{\text{el}})}{\partial v_{\text{el}}}\ \frac{\partial v_{\text{el}}}{\partial v_2} = -\frac{\partial g(v_{\text{el}})}{\partial v_{\text{el}}}$$

With these preliminary steps, we can now write the Newton–Raphson equation:

$$\begin{bmatrix} G_1 + \dfrac{\partial g(v_{el})}{\partial v_{el}} & -\dfrac{\partial g(v_{el})}{\partial v_{el}} \\ -\dfrac{\partial g(v_{el})}{\partial v_{el}} & G_3 + \dfrac{\partial g(v_{el})}{\partial v_{el}} \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} = - \begin{bmatrix} G_1 v_1 + g(v_{el}) - J \\ -g(v_{el}) + G_3 v_3 \end{bmatrix}$$

Comparing the Jacobian of the Newton–Raphson equation with the nodal formulation, we reach the important conclusion, valid for networks of any size:

1. Linear elements will be in the Jacobian in the same position as they were in the linear system matrix.
2. Nonlinear elements will have entries in the same positions as if they were linear; only their numerical values will be equal to the derivative $\partial g/\partial v_{el}$, evaluated with already available variables.

This conclusion will also be true for the other formulations, similar to the tableau or the modified nodal. So far, we considered only nonlinear resistive elements and the operating point.

Nonlinear storage elements (capacitors and inductors) contribute to the equations with their fluxes and charges. The current through the nonlinear capacitor is defined by

$$i_c = \frac{dq(v_c)}{dt} \tag{23.19}$$

where $q(v_c)$ is the charge and v_c is the voltage across the capacitor. The voltage across the nonlinear inductor is given by

$$v_L = \frac{d\phi(i_L)}{dt} \tag{23.20}$$

with ϕ denoting the flux.

Integration of systems with storage elements is always done by first replacing the derivative by a suitable algebraic expression and then solving the resulting nonlinear algebraic system by the Newton–Raphson method derived previously.

Many methods are available to replace the time domain derivatives by algebraic expressions. Books on numerical analysis usually describe the Runge–Kutta method. We mention it here because it is not suitable for solution of networks. There are several reasons for this, the main one being that the preferred modified nodal formulation does not lead to systems of differential, but rather to systems of algebraic-differential equations. Only two methods are widely used, the trapezoidal formula and a family of backward differentiation formulas (BDFs). Among the BDFs, the simplest is the backward Euler, and we will base our explanations on this formula. It replaces the derivative by the difference of the previous and new value, divided by the step size, h,

$$\frac{dq(v_c)}{dt} \sim \frac{q_{new}(v_c) - q_{old}}{h}$$
$$\frac{d\phi(i_L)}{dt} \sim \frac{\phi_{new}(i_L) - \phi_{old}}{h}$$

Consider the network in Figure 23.20 with nonlinear storage elements and with a linear conductance, G. It can be described by three equations:

$$f_1 = \frac{dq(v_c)}{dt} + G(v_1 - v_2) - j(t) = 0$$

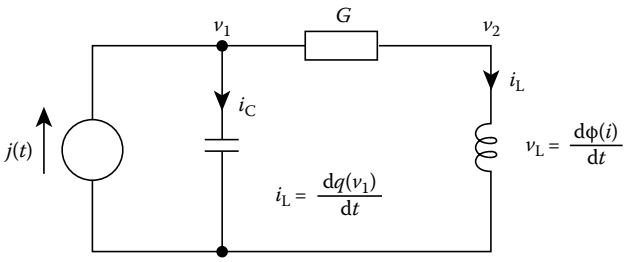


FIGURE 23.20 Network with a nonlinear capacitor and inductor.

$$f_2 = -G(v_1 - v_2) + i_L = 0$$
$$f_3 = v_2 - \frac{d\phi(I_L)}{dt} = 0$$

The time derivatives are replaced by the backward Euler formula

$$\frac{q_{new}(v_1) - q_{old}}{h} + G(v_1 - v_2) - j(t) = 0$$
$$-G(v_1 - v_2) + i_L = 0$$
$$v_2 - \frac{\phi_{new}(i_L) - \phi_{old}}{h} = 0$$

thus changing the system into an algebraic one. If we now differentiate with respect to the variables v_1, v_2, and i_L, we obtain the Jacobian

$$\begin{bmatrix} \frac{1}{h}\frac{\partial q_{new}(v_1)}{\partial v_1} + G & -G & 0 \\ -G & +G & 1 \\ 0 & 1 & -\frac{1}{h}\frac{\partial \phi_{new}(i_L)}{\partial i} \end{bmatrix}$$

It can be observed that values of the derivatives are in the same places as would be the values of $C(L)$ of linear capacitors (inductors). In addition, the variable s from the Laplace domain is replaced by $1/h$.

The example used a grounded capacitor and a grounded inductor. Figure 23.21 gives the stamps for floating nonlinear elements and for the Newton–Raphson iteration, based on the backward Euler formula.

23.6 Nodal Analysis of Active Networks

Low-frequency analog filters are often built with active RC networks and the active elements are almost always operational amplifiers. We have seen in Section 23.4 that each such element adds one row and one column to the modified nodal system matrix, thus making the system too large for hand solutions. We need a method that can reduce the size of the matrix to the minimum. Such reduction is possible [1,2], and becomes extremely simple if the voltage sources (dependent or independent) have one of their terminals grounded. Almost all practical networks meet this condition.

To introduce the method, consider the network in Figure 23.22. If we are not interested in the current through the voltage source, we can write only one nodal equation for the node on the right:

$$(G_1 + G_2)V_1 - EG_1 = 0$$

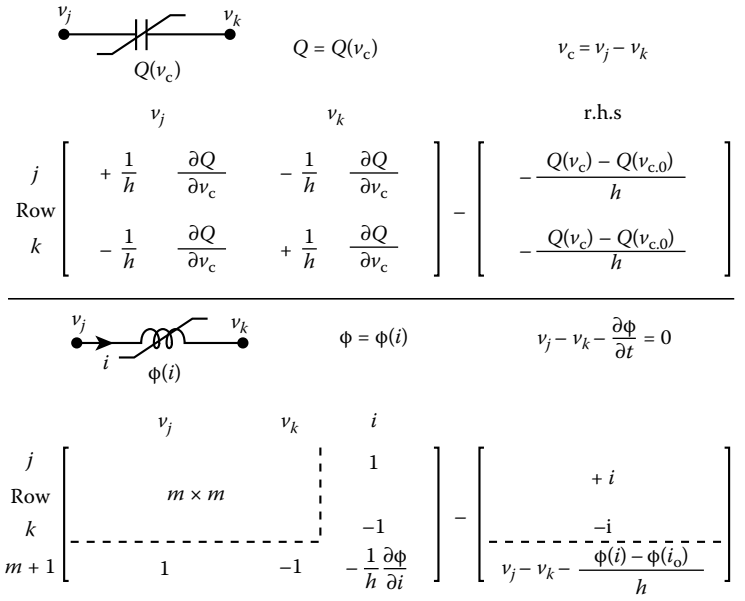


FIGURE 23.21 Stamp for a nonlinear capacitor and inductor.

The source is known, the term in which it appears is transferred to the right side and, instead of three equations of the modified nodal formulation, we must solve only one. Consider next the network in Figure 23.23 with a voltage source and an ideal operational amplifier. One of the output terminals of the operational amplifier is grounded. Such amplifier is described by the equation

$$(V_+ - V_-)A = V_{\text{out}} \tag{23.21}$$

where $A \to \infty$. Divide first by A and then substitute

$$B = -1/A \tag{23.22}$$

This changes the equation into

$$V_+ - V_- + BV_{\text{out}} = 0 \tag{23.23}$$

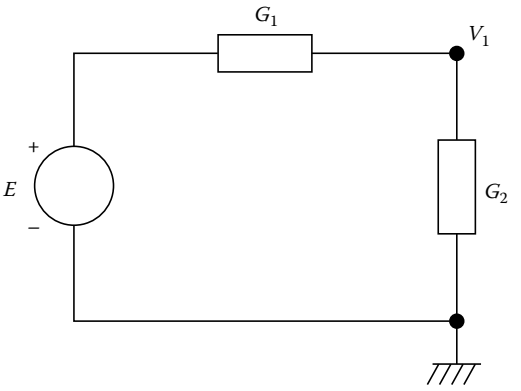


FIGURE 23.22 Example showing how to reduce the number of nodal equations.

If the operational amplifier is ideal, set $B = 0$ and in such case $V_+ = V_-$; the operational amplifier will have the same voltages at its input terminals. We can take this into consideration, by simply writing the voltage with the same subscript to both input terminals of the operational amplifier, as was done in Figure 23.23. We are not interested in the current of the voltage source, nor in the current flowing into the operational amplifier. We mark our lack of interest by crossing out the nodes that have grounded

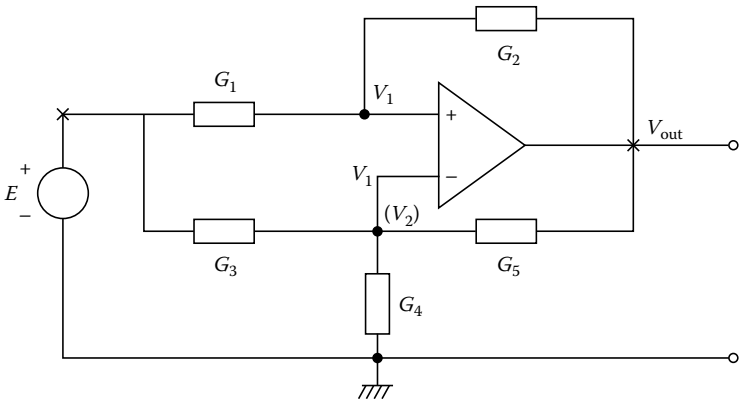


FIGURE 23.23 Nodal analysis of a network with one ideal operational amplifier.

voltage source; this was also done in Figure 23.23. Each node is given a voltage, but we write the nodal equations only at nodes that were not crossed out. For our example:

$$\begin{aligned}(G_1 + G_2)\ V_1 - G_2 V_{\text{out}} - EG_1 &= 0\\ (G_3 + G_4 + G_5)V_1 - G_5 V_{\text{out}} - EG_3 &= 0\end{aligned}$$

Terms with the known source voltage are transferred to the right and we have the system

$$\begin{bmatrix} G_1 + G_2 & -G \\ G_3 + G_4 + G_5 & -G_5 \end{bmatrix} \begin{bmatrix} V_1 \\ V_{\text{out}} \end{bmatrix} = \begin{bmatrix} G_1 E \\ G_3 E \end{bmatrix}$$

A modified modal formulation would have required six equations.

This method can be used for any network if one node of each voltage source, dependent or independent, is grounded. All we have to do is assign every node a voltage, cross out nodes with voltage sources, and write nodal equations for the rest. It is advantageous to use conductances for resistors, because this way we avoid the fractions.

The method remains valid if the operational amplifier is not ideal and has the inverted gain B. The only difference is that for a nonideal amplifier we cannot make any assumptions on the voltages at its input terminals and the subscripts of such voltages must be different. This second case is also illustrated in Figure 23.23 by the voltage V_2 (in brackets) at the lower node. We still write nodal equations for the two input terminals, but to complete the system we must also attach the equation of the operational amplifier. The result is

$$\begin{aligned}V_1(G_1 + G_2) - G_2 V_{\text{out}} &= G_1 E\\ V_1(G_3 + G_4 + G_5)V_2 - G_5 V_{\text{out}} &= G_3 E\\ V_1 - V_2 + BV_{\text{out}} &= 0\end{aligned}$$

and in matrix from

$$\begin{bmatrix} G_1 + G_2 & 0 & -G_2 \\ 0 & G_3 + G_4 + G_5 & -G_5 \\ 1 & -1 & B \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_{\text{out}} \end{bmatrix} \begin{bmatrix} G_1 E \\ G_3 E \\ 0 \end{bmatrix}$$

where B can be set zero for an ideal operational amplifier.

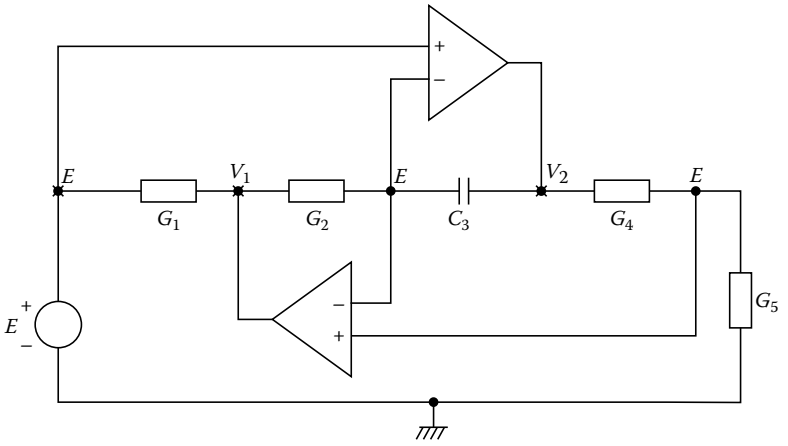


FIGURE 23.24 Nodal analysis of a network with two ideal operational amplifiers.

We will give one example of a practical network, Figure 23.24. The operational amplifiers are ideal and thus the input voltage, E, appears at three terminals of the network. The other terminal voltages are marked by V_1 and V_2. The terminals with voltage sources are marked by crosses and only nodes 3 and 5, counting from left, remain for writing the KCL. They are

$$(G_2 + sC_3)E - G_2V_1 - sC_3V_2 = 0$$
$$(G_4 + G_5)E - G_4V_2 = 0$$

Transferring terms containing the independent voltage source, E, to the other side of the equation, we arrive at the system

$$\begin{bmatrix} G_2 & sC_3 \\ 0 & G_4 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} (G_2 + sC_3)E \\ (G_4 + G_5)E \end{bmatrix}$$

Conclusion

It was demonstrated that hand calculations should use nodal or mesh formulations. Computer applications should be based on modified nodal formulation. For active networks, it is advantageous to use the method of Section 23.6.

References

1. J. Vlach, *Basic Network Theory with Computer Applications*, New York: Van Nostrand Reinhold, 1992.
2. J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd edn., New York: Van Nostrand Reinhold, 1994.

24

Frequency-Domain Methods

24.1 Network Functions **24**-1
Properties of LLFT Network Functions
24.2 Network Theorems **24**-12
Source Transformations • Dividers • Superposition • Thévenin's Theorem • Norton's Theorem • Thévenin's and Norton's Theorems and Network Equations • π–T Conversion • Reciprocity • Middlebrook's Extra Element Theorem • Substitution Theorem
24.3 Sinusoidal Steady-State Analysis and Phasor Transforms **24**-45
Sinusoidal Steady-State Analysis • Phasor Transforms • Inverse Phasor Transforms • Phasors and Networks • Phase Lead and Phase Lag • Phasor Diagrams • Resonance • Power in AC Circuits
Acknowledgments **24**-74
References **24**-74

Peter B. Aronhime
University of Louisville

24.1 Network Functions

Network functions are employed to characterize linear, time-invariant networks in the zero state for a single excitation. As described in Chapter 22 and other sections, network functions contain information concerning a network's stability and natural modes. They allow a designer to focus on obtaining a desired output signal for a given input signal.

In this section, it is shown that the concept of network functions is obtained as an extension of the (transformed) element defining equations for resistors, capacitors, and inductors. The relationships of network functions to transformed loop and node equations are also described. As a result of these relationships, a list of properties of network functions can be generated, which is useful in the analysis of linear networks. Much is known about a network function for a given network even before an analysis is performed and the function itself is obtained.

Ohm's law, $v_R(t) = Ri_R(t)$ where R is in ohms, describes the relationship between the voltage across the resistor and the current through the resistor. These variables and their reference polarity and direction are depicted in Figure 24.1. If elements of the equation for Ohm's law are transformed, we obtain $V(s) = RI(s)$ because R is a constant. Thus, we obtain an Ohm's law-like expression in the frequency domain.

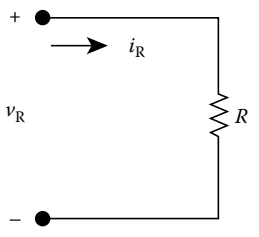


FIGURE 24.1 Reference polarity and direction for Ohm's law.

However, the capacitor's voltage and current are related by the integral

$$v_c(t) = \frac{1}{C}\int_{-\infty}^{t} i_c(t)\mathrm{d}t = \frac{1}{C}\int_{-\infty}^{0} i_C(t)\mathrm{d}t + \frac{1}{C}\int_{0}^{t} i_C(t)\mathrm{d}t$$

$$= V_o + \frac{1}{C}\int_{0}^{t} i_c(t)\mathrm{d}t \tag{24.1}$$

where the voltage reference polarity and the current reference direction are illustrated in Figure 24.2 and the capacitance C is given in farads (F). Unlike the resistor, the relation between the capacitor voltage and the capacitor current is not a simple Ohm's law-like expression in the time domain. In addition, the voltage across the capacitor at any time t, is dependent on the entire history of the current through the capacitor.

The integral expression for the voltage across the capacitor can be split into two terms where the first term is the initial voltage across the capacitor, $V_o = v_c(0)$. If the elements of the equation are transformed, we obtain

$$V_c(s) = \frac{V_o}{s} + \frac{1}{C}\frac{I_c(s)}{s} \tag{24.2}$$

Equation 24.2 shows that if $V_o = 0$, then the expression for the transform of the capacitor voltage becomes more nearly Ohm's law-like in its form. Furthermore, if we associate the s that arises because of the integral of the current with the capacitor C, and define the impedance of the capacitor as $Z(s) = V_c(s)/I_c(s) = 1/(sC)$, then the equation becomes Ohm's law-like in the frequency domain.

A similar process can be applied to the inductor. The current through the inductor is expressed as

$$i_L(t) = \frac{1}{L}\int_{-\infty}^{t} v_L(t)\mathrm{d}t = I_o + \frac{1}{L}\int_{0}^{t} v_L(t)\mathrm{d}t \tag{24.3}$$

where

L is expressed in henries (H)

$I_o = i_L(0)$ is the initial current through the inductor

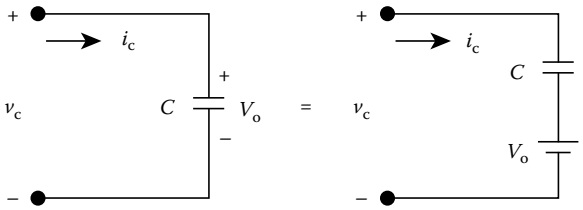


FIGURE 24.2 Capacitor representation showing reference polarities for voltages and reference direction for current.

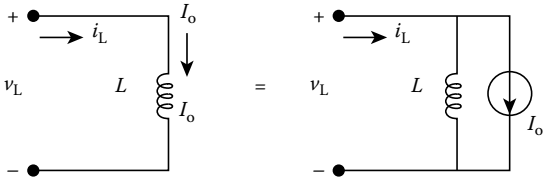


FIGURE 24.3 Inductor representation showing reference directions for currents and reference polarity for voltage.

Figure 24.3 depicts the reference polarity and direction for the inductor voltage and current. If the expression for the current through the inductor is transformed, the result is

$$I_L(s) = \frac{I_o}{s} + \frac{1}{L}\frac{V_L(s)}{s} \tag{24.4}$$

Again, as with the capacitor, if $I_o = 0$ and if the s that is included because of the integral of $v_L(t)$ is considered as associated with L, then the expression for the transform of the current through the inductor has an Ohm's law-like form if we define the impedance of the inductor as $Z(s) = V_L(s)/I_L(s) = sL$.

The impedance concept is an important one in network analysis. It allows us to combine dissimilar elements in the frequency domain—something we cannot do in the time domain. In fact, impedance is a frequency-domain concept. It is the ratio of the transform of the voltage across the port of the network to the transform of the current through the port with all independent sources within the network properly removed and with all initial voltages across capacitors and initial currents through inductors set to zero. Thus, when we indicate that independent sources are to be removed, we mean that initial conditions are to be set to zero as well.

The concept of impedance can be extended to linear, lumped, finite, time-invariant, one-port networks in general. We denote these networks as lumped, linear, finite, time-invariant (LLFT) networks. These networks are linear. That is, they are composed of elements including resistors, capacitors, inductors, transformers, and dependent sources with parameters that are not functions of the voltage across the element or the current through the element. Thus, the differential equations describing these networks are linear.

These networks are lumped and not distributed. That is, LLFT networks do not contain transmission lines as network elements, and the differential equations describing these networks are ordinary and not partial differential equations.

LLFT networks are finite, meaning that they do not contain infinite networks and require only a finite number of network elements in their representation. Infinite networks are sometimes useful in modeling such things as ground connections in the surface of the earth, but we exclude the discussion of them here.

LLFT networks are time-invariant or constant instead of time-varying. Thus, the ordinary, linear differential equations describing LLFT networks have constant coefficients.

The steps for finding the impedance of an LLFT one-port network are

1. Properly remove all independent sources in the network. By "properly" removing independent sources, we mean that voltage sources are replaced by short circuits and current sources are replaced by open circuits. Dependent sources are not removed.
2. Excite the network with a voltage source or a current source at the port, and find an equation or equations to solve for the other port variable.
3. Form $Z(s) = V(s)/I(s)$.

Simple networks do not need to be excited in order to determine their impedance, but in the general case an excitation is required. The next example illustrates these concepts.

Example 24.1

Find the impedances of the one-port networks in Figure 24.4.

Solution

The network in Figure 24.4a is composed of three elements connected in series. No independent sources are present, and there is zero initial voltage across the capacitor and zero initial current through the inductor. The impedance is determined as

$$Z(s) = R + sL + \frac{1}{sC} = L\left(\frac{s^2 + s\frac{R}{L} + \frac{1}{LC}}{s}\right)$$

The network in Figure 24.4b includes a dependent source that depends on the voltage across R_1. The impedance of this network is not obvious, and so we should excite the port. Also, the capacitor has an initial voltage V_c across it. This voltage is set to zero to find the impedance. Figure 24.5 is the network in Figure 24.4b prepared for finding the impedance. Using the impedance concept and two loop equations or two node equations, we obtain

$$Z(s) = \frac{V(s)}{I(s)} = \frac{R_1R_2\left(s + \frac{1}{CR_1}\right)}{[R_1(1-K) + R_2]\left(s + \frac{1}{C[R_1(1-K)+R_2]}\right)}$$

The expressions for impedance found in the previous example are rational functions of s, and the coefficients of s are functions of the elements of the network including the coefficient K of the dependent source in the network in Figure 24.4b. We will demonstrate that these observations are general for LLFT networks; but first we will extend the impedance concept in another direction.

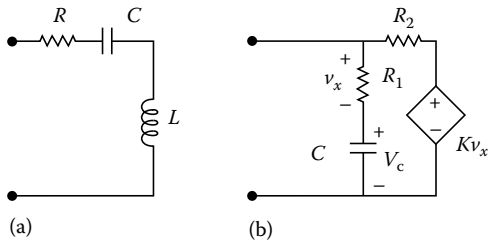


FIGURE 24.4 (a) Simple network. (b) A network containing a dependent source.

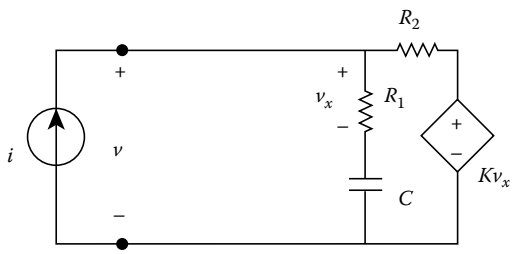


FIGURE 24.5 Network in Figure 24.4b prepared for analysis.

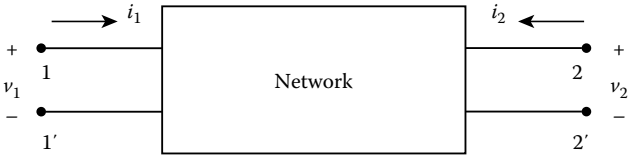


FIGURE 24.6 Reference polarities and reference directions for port variables of a two-port network.

We have defined the impedance of a one-port LLFT network. We can also define another network function—the admittance $Y(s)$. The admittance of a one-port LLFT network is the quotient of the transform of the current through the port to the transform of the voltage across the port with all independent sources within the network properly removed. One-port networks have only two linear network functions, impedance and admittance. Furthermore, $Z(s) = 1/Y(s)$ because both network functions concern the same port of the network, and the impedance or admittance relating the response to the excitation is the same whether a current excitation causes a voltage response or a voltage excitation causes a current response. An additional implication of this observation is that either network function can be determined with either type of excitation, voltage source or current source, applied to the network.

Figure 24.6 depicts a two-port network with the reference polarities and reference directions indicated for the port variables. Port one of the two-port network is formed from the two terminals labeled 1 and 1′. The two terminals labeled 2 and 2′ are associated to form port two. A two-port network has 12 network functions associated with it instead of only two, and so we will employ the following notation for these functions:

$$N_{\mathrm{RE}}(s) = R(s)/E(s)$$

where

$N_{\mathrm{RE}}(s)$ is a network function, the subscript "R" is the port at which the response variable exists, the subscript "E" is the port at which the excitation is applied

$R(s)$ is the transform of the response variable

$E(s)$ is the transform of the excitation that may be a current source or a voltage source depending on the particular network function

For example, for the two-port networks shown in Figure 24.7

$$Z_{21}(s) = \frac{V_2(s)}{I_1(s)} \quad \text{and} \quad G_{12}(s) = \frac{V_1(s)}{V_2(s)} \tag{24.5}$$

Note that a load impedance has been placed across port two, the response port for $Z_{21}(s)$, in Figure 24.7a. Also, a load has been connected across port one, the response port for $G_{12}(s) = V_1(s)/V_2(s)$ in Figure 24.7b. It is assumed that all independent sources have been properly removed in both networks

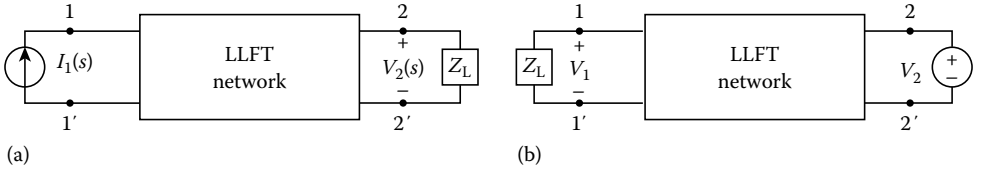


FIGURE 24.7 (a) Network configured for finding $Z_{21}(s)$. (b) Network for determining $G_{12}(s)$.

TABLE 24.1 Network Functions of Two-Port Networks

	Excitation Port	
Response Port	1	2
1	Z_{11}, Y_{11}	Z_{12}, Y_{12}, G_{12}, α_{12}
2	Z_{21}, Y_{21}, G_{21}, α_{21}	Z_{22}, Y_{22}

in Figure 24.7, and this assumption also applies to the loads. Of course, if a load impedance is changed, usually the network function will change. Thus, the load, if any, must be specified.

Table 24.1 lists the network functions of a two-port network. "G" denotes a voltage ratio, and "α" denotes a current ratio. The functions can also be grouped into driving-point and transfer functions. Driving-point functions are ones in which the excitation and response occur at the same port, and transfer network functions are ones in which the excitation and response occur at different ports. For example, $Z_{11}(s) = V_1(s)/I_1(s)$ is a driving-point network function, and $G_{21}(s) = V_2(s)/V_1(s)$ is a transfer network function. Of the twelve network functions for a two-port network, four are driving-point functions and eight are transfer functions. The two network functions for a one-port network are, of necessity, driving-point network functions.

Network functions are related to loop and node equations. Consider the LLFT network in Figure 24.8. Independent sources within the network have been properly removed. Assume the network has n independent nodes plus the ground node, and assume for simplicity that the network has no mutual inductance or dependent sources. Let us determine $Z_{11} = V_1/I_1$ in terms of a quotient of determinants of the nodal admittance matrix. The node equations, all written with currents leaving a node as positive currents, are

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} I_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{24.6}$$

where V_i, $i = 1, 2, \ldots, n$, are the unknown node voltages. The elements y_{ij}, $i, j = 1, 2, \ldots, n$, of the nodal admittance matrix above have the form

$$y_{ij} = \pm\left\{ g_{ij} + \frac{\Gamma_{ij}}{s} + sC_{ij} \right\} \tag{24.7}$$

where the plus sign is taken if $i = j$ and the minus sign is taken if i and j are unequal. The quantity g_{ij} is the sum of the conductances connected to node i if $i = j$, and if i does not equal j, it is the sum of the

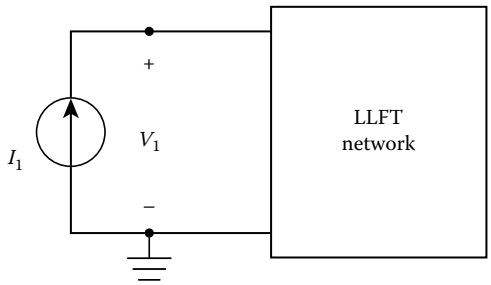


FIGURE 24.8 LLFT network with n independent nodes plus the ground node.

conductances connected between nodes i and j. A similar statement applies to C_{ij}. The quantity Γ_{ij} is the sum of the reciprocal inductances ($\Gamma = 1/L$) connected to node i if $i = j$, and it is the sum of the reciprocal inductances connected between nodes i and j if i does not equal j.

Solving for V_1 using Cramer's rule yields

$$V_1 = \frac{\begin{vmatrix} I_1 & y_{12} & \cdots & y_{1n} \\ 0 & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & y_{n2} & \cdots & y_{nn} \end{vmatrix}}{\Delta'} \tag{24.8}$$

where Δ' is the determinant of the nodal admittance matrix. Thus,

$$V_1 = I_1 \frac{\Delta'_{11}}{\Delta'} \tag{24.9}$$

where Δ'_{11} is the cofactor of element y_{11} of the nodal admittance matrix. Thus, we can write

$$\frac{V_1}{I_1} = Z_{11} = \frac{\Delta'_{11}}{\Delta'} \tag{24.10}$$

If mutual inductance and dependent sources exist in the LLFT network, the nodal admittance matrix elements are modified. Furthermore, there may be more than one entry in the column matrix containing excitations. However, Z_{11} can still be expressed as a quotient of determinants of the nodal admittance matrix.

Next, consider the network in Figure 24.9, which is assumed to have n independent nodes plus a ground node. The response port exists between terminals j and k. In this network, we are making the pair of terminals j and k serve as the second port. Denote the transimpedance V_{jk}/I_1 as Z_{j1}. Let us express this transfer function Z_{j1} as a quotient of determinants. All independent sources within the network have been properly removed. Note that the node voltages are measured with respect to the ground terminal indicated, but the output voltage is the difference of the node voltages V_j and V_k. Thus, we have to solve for these two node voltages. Writing node equations, again taking currents leaving a node as positive currents, and solving for V_j using Cramer's rule we have

$$V_j = \frac{\begin{vmatrix} y_{11} & y_{12} & \cdots & y_{1(j-1)} & I_1 & y_{1k} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2(j-1)} & 0 & y_{2k} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{n(j-1)} & 0 & y_{nk} & \cdots & y_{nn} \end{vmatrix}}{\Delta'} \tag{24.11}$$

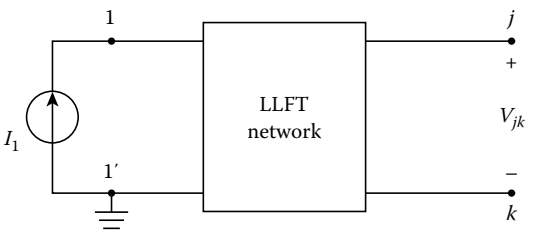


FIGURE 24.9 LLFT network with two ports indicated.

which can be written as

$$V_j = I_1 \frac{\Delta'_{1j}}{\Delta'} \tag{24.12}$$

where Δ'_{1j} is the cofactor of element y_{1j} of the nodal admittance matrix. Similarly, we can solve for V_k and obtain

$$V_k = I_1 \frac{\Delta'_{1k}}{\Delta'} \tag{24.13}$$

Then, the transimpedance Z_{j1} can be expressed as

$$Z_{j1} = \frac{V_j - V_k}{I_1} = \frac{\Delta'_{1j} - \Delta'_{1k}}{\Delta'} \tag{24.14}$$

If terminal k in Figure 24.9 is common with the ground node so that the network is a grounded two-port network, then V_k is zero, and Z_{j1} can be expressed as Δ'_{1j}/Δ'. This result can be extended so that if the output voltage is taken between any node h and ground, then the transimpedance can be expressed as $Z_{h1} = \Delta'_{1h}/\Delta'$.

These results can be used to obtain an expression for G_{21} in terms of the determinants of the nodal admittance matrix. Figure 24.10 shows an LLFT network with a voltage excitation applied at port 1 and with port 2 open. The current $I_1(s)$ is given by V_1/Z_{11}. Then, V_2 is given by $V_2(s) = I_1 Z_{21}$. Thus,

$$G_{21} = \frac{V_2}{V_1} = \frac{I_1 Z_{21}}{I_1 Z_{11}} = \frac{\Delta'_{12}}{\Delta'_{11}} \tag{24.15}$$

Note that the determinants in the quotient are of equal order so that G_{21} is dimensionless.

Of course, network functions can also be expressed in terms of determinants of the loop impedance matrix. Consider the two-port network in Figure 24.11, which is excited with a voltage source applied to

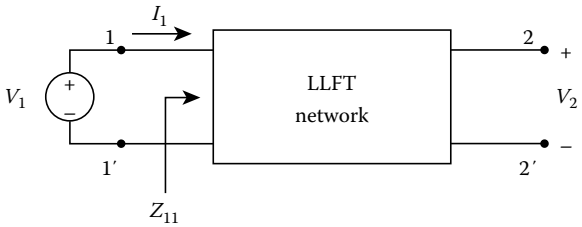


FIGURE 24.10 LLFT network with port 2 open.

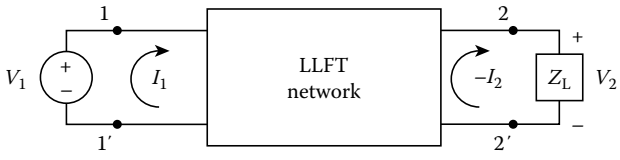


FIGURE 24.11 LLFT network with load Z_L connected across port 2.

port 1 and has a load Z_L connected across port 2. Let us find the voltage transfer function G_{21} using loop equations. Assume that n independent loops exist, of which two are illustrated explicitly in Figure 24.11, and assume that no independent sources are present within the network. Also, assume for simplicity that the network contains no dependent sources or mutual inductance and that the loops are chosen so that V_1 is in only one loop. The loop equations are

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ -I_2 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} V_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{24.16}$$

where $I_j, j = 1, 3, \ldots, n$, and $-I_2$ are the loop currents, and the elements z_{ij} of the loop impedance matrix are given by

$$z_{ij} = \pm\left(R_{ij} + sL_{ij} + \frac{D_{ij}}{s}\right), \quad i, j = 1, 2, \ldots, n \tag{24.17}$$

where we have assumed that all loop currents are taken in the same direction such as clockwise. The plus sign applies if $i = j$, and the minus sign is used if $i \neq j$. R_{ij} is the sum of the resistances in loop i if $i = j$, and R_{ij} is the sum of the resistances common to loops i and j if $i \neq j$. L_{ij} is the sum of the inductances in loop i if $i = j$, and it is the sum of the inductances common to loops i and j if $i \neq j$. A similar statement applies to the reciprocal capacitances $D_{ij}(D = 1/C)$. However, the element z_{22} includes the extra term Z_L which could be a quotient of determinants itself. Solving for $-I_2$ using Cramer's rule, we have Δ, which is the determinant of the $n \times n$ loop impedance matrix, and Δ_{12} is the cofactor of element z_{12} of the loop impedance matrix.

$$-I_2 = \frac{\begin{vmatrix} z_{11} & V_1 & z_{13} & \cdots & z_{1n} \\ z_{21} & 0 & z_{23} & \cdots & z_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{n1} & 0 & z_{n3} & \cdots & z_{nn} \end{vmatrix}}{\Delta} = V_1 \frac{\Delta_{12}}{\Delta} \tag{24.18}$$

The transform voltage V_2 is given by $-I_2(s)Z_L$, and the transfer function G_{21} can be expressed as

$$G_{21} = \frac{V_2}{V_1} = Z_L \frac{\Delta_{12}}{\Delta} \tag{24.19}$$

Thus, G_{21} can be represented as a quotient of determinants of the loop impedance matrix multiplied by Z_L.

In a similar manner, we can write

$$Y_{jk} = \frac{\Delta_{kj}}{\Delta}, \quad j, k = 1, 2 \tag{24.20}$$

Then, we can use this result to write

$$V_1 = I_1 \frac{1}{Y_{11}} \quad \text{and} \quad I_2 = V_1 Y_{21} \tag{24.21}$$

TABLE 24.2 Network Function in Terms of Quotients of Determinants

$Y_{jk} = \Delta_{kj}/\Delta$	$G_{jk} = \Delta'_{kj}/\Delta'_{kk}$
$Z_{jk} = \Delta'_{kj}/\Delta'$	$\alpha_{jk} = \Delta_{kj}/\Delta_{kk}$

Thus,

$$\frac{I_2}{I_1} = \alpha_{21} = \frac{\Delta_{12}}{\Delta_{11}} \tag{24.22}$$

Table 24.2 summarizes some of these results.

24.1.1 Properties of LLFT Network Functions

We now list properties of network functions of LLFT networks. These properties are useful as a check of an analysis of such networks.

1. Network functions of LLFT networks are real, rational functions of s, and therefore have the form

$$N(s) = \frac{P(s)}{Q(s)} = \frac{a_0 s^m + a_1 s^{m-1} + \cdots + a_{m-1}s + a_m}{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n} \tag{24.23}$$

   where the coefficients in both the numerator and denominator polynomials are real.

   A network function of an LLFT network is a rational function because network functions can be expressed as quotients of determinants of nodal admittance matrices or of loop impedance matrices. The elements of these determinants are at most simple rational functions, and when the determinants are expanded and the fractions cleared, the result is always a rational function. The coefficients a_i, $i = 0, 1, \ldots, m$, and b_j, $j = 0, 1, \ldots, n$, are functions of the real elements of the network R, L, C, M, and coefficients of dependent sources. The constants R, L, C, and M are real. In most networks, the coefficients of dependent sources are real constants. Thus, the coefficients of LLFT network functions are real, and therefore the network function is a real function. It is possible for the "coefficients" of dependent sources in LLFT networks to themselves be real, rational functions of s. But when all the fractions are cleared, the result is a real, rational function.
2. LLFT network function is completely defined by its self-poles, self-zeros, and the scale factor H.

   If the numerator and denominator polynomials are factored and there are no common factors, we have

$$N(s) = \frac{P(s)}{Q(s)} = \frac{a_0}{b_0}\frac{(s - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)} \tag{24.24}$$

   where $a_0/b_0 = \mathrm{H}$ is the scale factor. The values of $s = z_1, z_2, \ldots, z_m$ are zeros of the polynomial $P(s)$ and self-zeros of the network function. Also, $s = p_1, p_2, \ldots, p_{\mathrm{n}}$ are zeros of the polynomial $Q(s)$ and self-poles of the network function. In addition, $N(s)$ may have other poles or zeros at infinity. We call these poles and zeros mutual poles and mutual zeroes because they result from the difference in degrees of the numerator and denominator polynomials.
3. Counting both self and mutual poles and zeros, and counting a kth order pole or zero k times, $N(s)$ has the same number of poles as zeros, and this number is equal to the highest power of s in $N(s)$. If $m = n$, then there are n self-zeros and n self-poles and no mutual poles or zeros. If $n > m$, then there are m self-zeros and n self-poles. There are also $n - m$ mutual zeros. Thus, there are n poles and $m + (n - m) = n$ zeros. A similar statement can be constructed for $n < m$.

4. Complex roots of $P(s)$ and $Q(s)$ occur in conjugate pairs.

   This property follows from the fact that the coefficients of the numerator and denominator polynomials are real. Thus, complex factors of these polynomials have the form

$$(s + c + jd)(s + c - jd) = [(s + c)^2 + d^2] \tag{24.25}$$

   where c and d are real constants.
5. A driving point function of a network having no dependent sources can have neither poles nor zeros in the right-half s-plane (RHP), and poles and zeros on the imaginary axis must be simple. The same restrictions apply to the poles of transfer network functions of such networks but not to the zeros of transfer network functions.

   Elsewhere in this handbook it is shown that the denominator polynomials of LLFT networks having no dependent sources cannot have RHP roots, and roots on the imaginary axis, if any, must be simple. However, the reciprocal of a driving-point network function is also a network function. For example, $1/Y_{22} = Z_{22}$. Thus, restrictions on the locations of poles of driving-point network functions also apply to zeros of driving-point network functions.

   However, the reciprocal of a transfer network function is not a network function (see [5]). For example, $1/Y_{21} \neq Z_{21}$. Thus, restrictions on the poles of a transfer function do not apply to its zeros.

   We can make a classification of the factors corresponding to the allowed types of poles as follows:

   The Type A factor corresponds to a pole on the $-\sigma$-axis. If $a = 0$, then the factor corresponds to a pole on the imaginary axis, and so only one such factor is allowed. Type B factors correspond to poles in the left-half s-plane (LHP), and Type C factors correspond to poles on the imaginary axis.
6. The coefficients of the numerator and denominator polynomials of a driving-point network function of an LLFT network with no dependent sources are positive. The coefficients of the denominator polynomial of a transfer network function are all one sign. Without loss of generality, we take the sign to be positive. But some or all of the coefficients of the numerator polynomial of a transfer network function may be negative.

   A polynomial made up of the factors listed in Table 24.3 would have the form:

$$Q(s) = (s + a_1) \cdots \left[(s + b_1)^2 + c_1^2\right] \cdots \left(s^2 + d_1^2\right) \cdots$$

   Note that all the constants are positive in the expression for $Q(s)$, and so it is impossible for any of the coefficients of $Q(s)$ to be negative.
7. There are no missing powers of s in the numerator and denominator polynomials of a driving-point network function of an LLFT network with no dependent sources unless all even or all odd powers of s are missing or the constant term is missing. This statement holds for the denominator polynomials of transfer functions of such networks, but there may be missing powers of s in the numerator polynomials of transfer functions.

   Property 7 is easily illustrated by combining types of factors from Table 24.3. Thus, a polynomial consisting only of type A factors contains all powers of s between the highest power and the constant term unless one of the "a" constants is zero. Then, the constant term is missing. Two a

TABLE 24.3 A Classification of Factors of Network Functions of LLFT Networks Containing No Dependent Sources

Type	Factor(s)	Conditions
A	$(s + a)$	$a \geq 0$
B	$(s + b + jc)(s + b - jc)$	$b > 0$, $c > 0$
C	$(s + jd)(s - jd)$	$d > 0$

constants cannot be zero because then there would be two roots on the imaginary axis at the same location. The roots on the imaginary axis would not be simple.

A polynomial made up of only type B factors contains all powers of s, and a polynomial containing only type C factors contains only even powers of s. A polynomial constructed from type C factors except for one A type factor with $a = 0$ contains only odd powers of s. If a polynomial is constructed from type B and C factors, then it contains all power of s.

8. The orders of the numerator and denominator polynomials of a driving-point network function of an LLFT network, which contains no dependent sources can differ by no more than one.

   The limiting behavior at high frequency must be that of an inductor, a resistor, or a capacitor. That is, if $N_{dp}(s)$ is a driving-point network function, then

$$\lim_{s\to\infty} N_{dp}(s) = \begin{cases} K_1 s \\ K_2 \\ K_3/s \end{cases}$$

   where $K_i, = 1, 2, 3$, are real constants.

9. The terms of lowest order in the numerator and denominator polynomials of a driving-point network function of an LLFT network containing no dependent sources can differ in order by no more than one.

   The limiting behavior at low frequency must be that of an inductor, a resistor, or a capacitor. That is,

$$\lim_{s\to 0} N_{dp}(s) = \begin{cases} K_4 s \\ K_5 \\ K_6/s \end{cases}$$

   where the constants K_i, $i = 4, 5, 6$, are real constants.

10. The maximum order of the numerator polynomials of the dimensionless transfer functions G_{12}, G_{21}, α_{12}, and α_{21}, of an LLFT network containing no dependent sources is equal to the order of the denominator polynomials. The maximum order of the numerator polynomial of the transfer functions Y_{12}, Y_{21}, Z_{12}, and Z_{21} is equal to the order of the denominator polynomial plus 1. However, the minimum order of the numerator polynomial of any transfer function may be zero.

If dependent sources are included in an LLFT network, then it is possible for the network to have poles in the RHP or multiple poles at locations on the imaginary axis. However, an important application of stable networks containing dependent sources is to mimic (simulate) the behavior of LLFT networks that contain no dependent sources. For example, networks that contain resistors, capacitors, and dependent sources can mimic the behavior of networks containing only resistors, capacitors, and inductors. Thus, low-frequency filters can be constructed without the need for heavy, expensive inductors that would ordinarily be required in such applications.

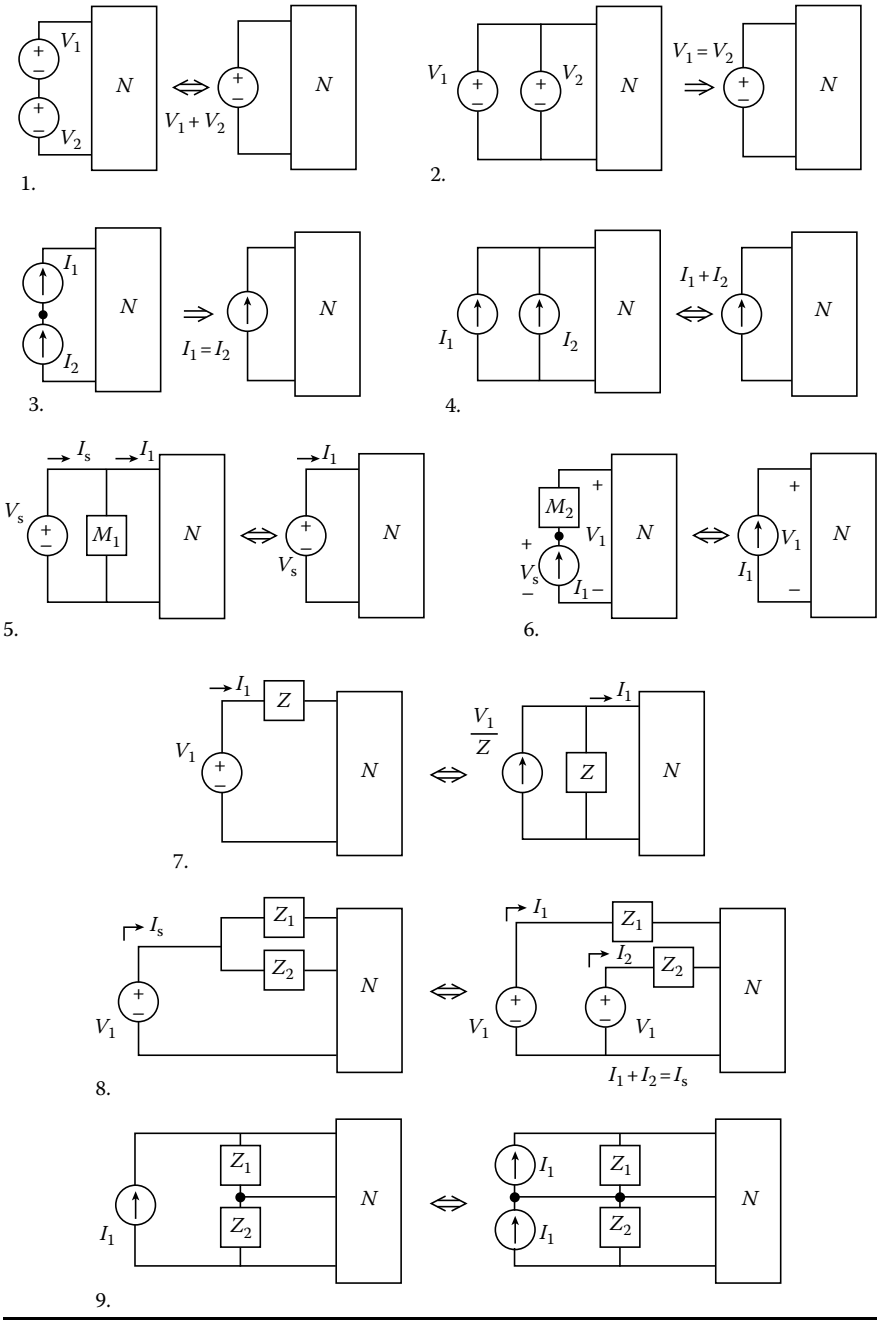
24.2 Network Theorems

In this section, we provide techniques, strategies, equivalences, and theorems for simplifying the analysis of LLFT networks or for checking the results of an analysis. They can save much work in the analysis of some networks if one remembers to apply them. Thus, it is convenient to have them listed in one place. To begin, we list nine equivalences that are often called source transformations.

24.2.1 Source Transformations

Table 24.4 is a collection of memory aids for the nine source transformations. Source transformations are simple ways the elements and sources externally connected to a network N can be combined or

TABLE 24.4 Source Transformations



N is an arbitrary network in which analysis for a voltage or current is to be performed. M_1 is an arbitrary one-port network or network element except a voltage source. M_2 is an arbitrary one-port network or network element except a current source. It is assumed there is no magnetic coupling between *N* and M_1 or M_2. There are no dependent sources in *N* in source transformation five that depend on I_s. Furthermore, there are no dependent sources in *N* in source transformation six that depend on V_s. However, M_1 and M_2 can have dependent sources that depend on voltages or currents in *N*. *Z*, Z_1 and Z_2 are one-port impedances.

eliminated without changing the voltages and currents within network N thereby simplifying the problem of finding a voltage or current within N.

Source transformation one in Table 24.4 shows the equivalence between two voltage sources connected in series and a single voltage source having a value that is the sum of the voltages of the two sources. A double-headed arrow is shown between the two network representations because it is sometimes advantageous to use this source transformation in reverse. For example, if a voltage source that has both DC and AC components is applied to a linear network N, it may be useful to represent that voltage source as two voltage sources in series—one a DC source and the other an AC source.

Source transformation two shows two voltage sources connected in parallel. Unless V_1 and V_2 are equal, the network would not obey Kirchhoff's law as evidenced by a loop equation written in the loop formed by the two voltage sources. A network that does not obey Kirchhoff's laws is termed a contradiction. Thus, a single-headed arrow is shown between the two network representations.

Source transformations three and four are duals, respectively, of source transformations two and one. The current sources must be equal in transformation three or else Kirchhoff's law would not be valid at the node indicated, and the circuit would be a contradiction.

Source transformation five shows that the circuit M_1 can be removed without altering any of the voltages and currents inside N. Whether M_1 is connected as shown or is removed, the voltage applied to N remains V_s. However, the current supplied by the source V_s changes from I_s to I_1.

Source transformation six shows that circuit M_2 can be replaced by a short circuit without affecting voltages and currents in N. Whether M_2 is in series with the current source I_1 as shown or removed, the current applied to N is the same. However, if network M_2 is removed, then the voltage across the current source changes from V_s to V_1.

Source transformation seven is sometimes termed a Thévenin circuit to Norton circuit transformation. This transformation, as shown by the double-headed arrow, can be used in the reverse direction. Thévenin's theorem is discussed thoroughly later in this section.

Source transformation eight is sometimes described as ''pushing a voltage source through a node,'' but we will term it as ''splitting a voltage source.'' Loop equations remain the same with this transformation, and the current leaving network N through the lowest wire continues to be I_s.

Source transformation nine shows that if a current source is not in parallel with one element, then it can be ''split'' as shown. Now, each one of the current sources I_1 has an impedance in parallel. Thus, analysis of network N may be simplified because source transformation seven can be applied.

Source transformations cannot be applied to all networks, but when they can be employed, they usually yield useful simplifications of the network.

Example 24.2

Use source transformations to find V_0 for the network shown in Figure 24.12. Initial current through the inductor in the network is zero.

Solution

The network can be readily simplified by employing source transformation five from Table 24.4 to eliminate R_1 and also I_2. Then, source transformation six can be used to eliminate V_1 because it is an element in series with a current source. The results to this point are illustrated in Figure 24.13a. If we then apply transformation seven, we obtain the network in Figure 24.13b. Next, we can apply transformation

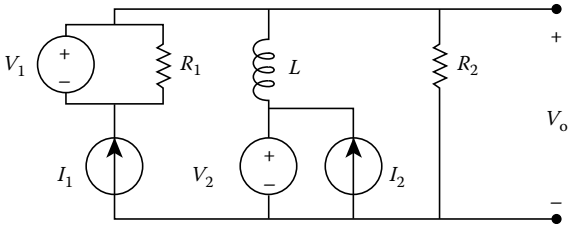


FIGURE 24.12 Network for Example 24.2.

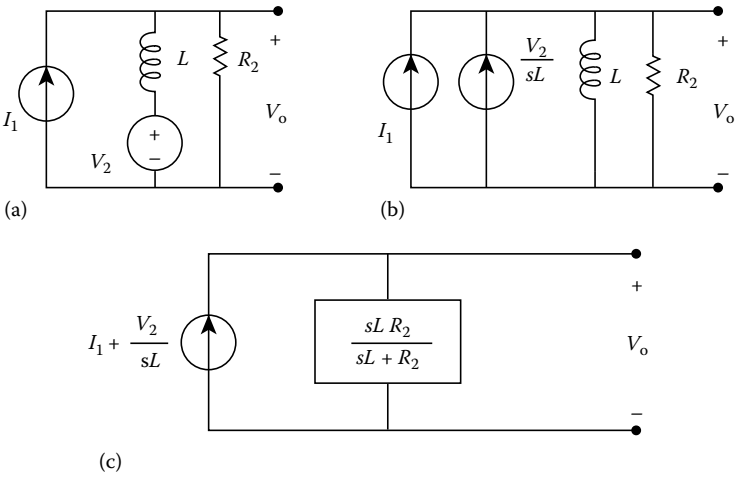


FIGURE 24.13 Application of source transformation six to eliminate V_1 as illustrated in (a), then applying transformation seven to obtain the network shown in (b), and finally applying transformation four to obtain the single loop network shown in (c).

four to obtain the single loop network in Figure 24.13c. The output voltage can be written in the frequency domain as

$$V_o = \left(I_1 + \frac{V_2}{sL}\right)\left(\frac{sLR_2}{sL + R_2}\right)$$

Source transformations can often be used advantageously with the following theorems.

24.2.2 Dividers

Current dividers and voltage dividers are circuits that are employed frequently, especially in the design of electronic circuits. Thus, dividers must be analyzed quickly. The relationships derived next satisfy this need.

Figure 24.14 is a current divider circuit. The source current I_s divides between the two impedances, and we wish to determine the current through Z_2. Writing a loop equation for the loop indicated, we have

$$I_2Z_2 - (I_s - I_2)Z_1 = 0 \tag{24.26}$$

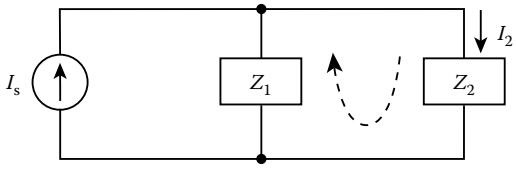


FIGURE 24.14 Current divider.

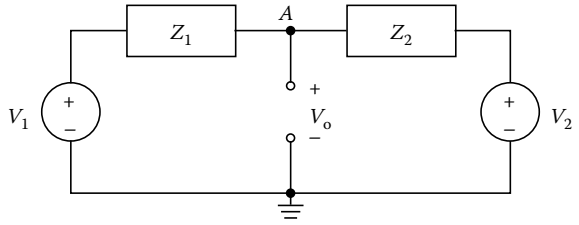


FIGURE 24.15 Enhanced voltage divider.

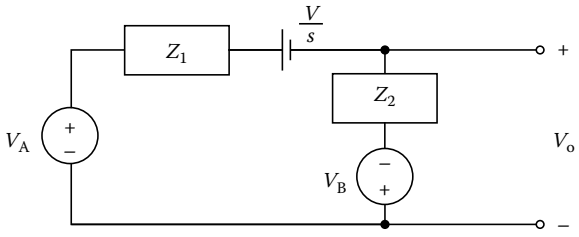


FIGURE 24.16 Circuit for Example 24.3.

from which we obtain

$$I_2 = I_s \frac{Z_1}{Z_1 + Z_2} \tag{24.27}$$

A circuit that we term an enhanced voltage divider is depicted in Figure 24.15. This circuit contains two voltage sources instead of the usual single source, but the enhanced voltage divider occurs more often in electronic circuits. Writing a node equation at node A and solving for V_o, we obtain

$$V_o = \frac{V_1 Z_2 + V_2 Z_1}{Z_1 + Z_2} \tag{24.28}$$

If V_2, for example, is zero, then the results from the enhanced voltage divider reduce to those of the single source voltage divider.

Example 24.3

Use Equation 24.28 to find V_o for the network in Figure 24.16.

Solution

The network in Figure 24.16 matches with the network used to derive Equation 24.28 even though it is drawn somewhat differently and has three voltage sources instead of two. However, we can use Equation 24.28 to write the answer for V_o by inspection.

$$V_o = \frac{(V_A - (V/s))Z_z - V_B Z_1}{Z_1 + Z_2}$$

The following example illustrates the use of source transformations together with the voltage divider.

Example 24.4

Find V_o for the network shown in Figure 24.17. The units of K, the coefficient of the dependent source, are ohms, and the capacitor is initially uncharged.

Solution

We note that the dependent voltage source is not in series with any one particular element and that the independent current source is not in parallel with any one particular element. However, we can split both the voltage source and the current source using source transformations eight and nine, respectively, from Table 24.4. Then, employing transformations five and seven, we obtain the network configuration depicted in Figure 24.18, for which we can use the voltage divider to write

$$V_o = \frac{I(K + R_1)R_2 + KI(R_1 + (1/sC))}{R_1 + R_2 + (1/sC)}$$

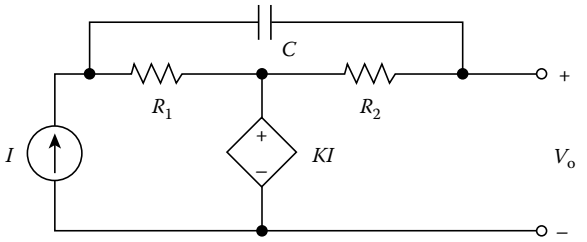


FIGURE 24.17 Network for Example 24.4.

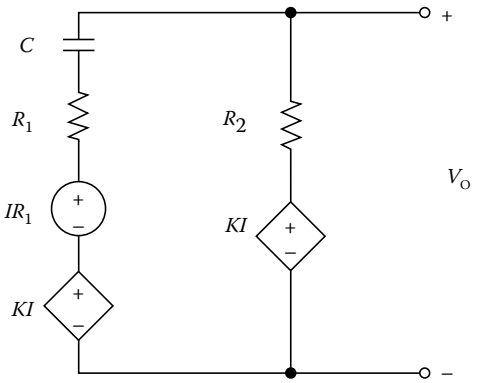


FIGURE 24.18 Results after employing source transformations on the network in Figure 24.17.

It should be mentioned that the method used to find V_0 in this example is not the most efficient one. For example, loops can be chosen for the network in Figure 24.17, so only one unknown loop is current. However, source transformations and dividers become more powerful analysis tools as they are coupled with additional network theorems.

24.2.3 Superposition

Superposition is a property of all linear networks, and whether it is used directly in the analysis of a network or not, it is a concept that is valuable in thinking about LLFT networks. Consider the LLFT network shown in Figure 24.19 in which, say, we wish to solve for I_1. Assume the network has n independent loops, and, for simplicity, assume no sources are within the box in the figure and that initial voltages across capacitors and initial currents through inductors are zero or are represented by independent sources external to the box. Note that one dependent source is shown in Figure 24.19 that depends on a voltage V_x in the network and that two independent sources, V_1 and V_2, are applied to the network. If loops are chosen so that each source has only one loop current flowing through it as indicated in Figure 24.19, then the loop equations can be written as

$$\begin{bmatrix} V_1 \\ V_2 \\ KV_x \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \tag{24.29}$$

where the elements of the loop impedance matrix are defined in the section describing network functions. Solving for I_1 using Cramer's rule, we have

$$I_1 = V_1 \frac{\Delta_{11}}{\Delta} + V_2 \frac{\Delta_{21}}{\Delta} + KV_x \frac{\Delta_{31}}{\Delta} \tag{24.30}$$

where

Δ is the determinant of the loop impedance matrix
Δ_{j1}, $j = 1, 2, 3$, are the cofactors

The expression for I_1 given in Equation 24.30 is an intermediate and not a finished solution. The finished solution would express I_1 in terms of the independent sources and the parameters (*Rs*, *Ls*, *Cs*, *Ms*, and *Ks*) of the network and not in terms of an unknown V_x. Thus, one normally has to eliminate V_x from the

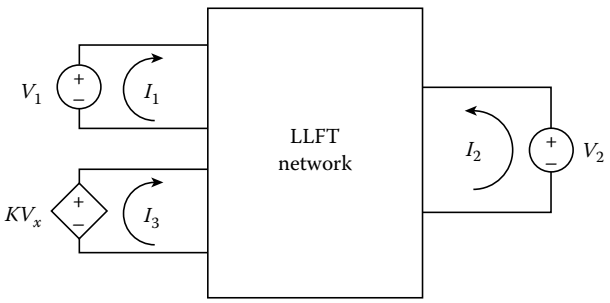


FIGURE 24.19 LLFT network with three voltage sources, of which one is dependent.

expression for I_1; but the intermediate expression for I_1 illustrates superposition. There are three components that add up to I_1 in Equation 24.30—one for each source including one for the dependent source. Furthermore, we see that each source is multiplied by a transadmittance (or a driving-point admittance in the case of V_1). Thus, we can write

$$I_1 = V_1 Y_{11} + V_2 Y_{12} + K V_x Y_{13} \tag{24.31}$$

where each admittance is found from the port at which a voltage source (whether independent or dependent) is applied. The response variable for each of these admittances is I_1 at port 1.

The simple derivation that led to Equation 24.30 is easily extended to both types of independent excitations (voltage sources and current sources) and to all four types of dependent sources. The generalization of Equation 24.30 leads to the conclusion:

> To apply superposition in the analysis of a network containing at least one independent source and a variety of other sources, dependent or independent, one finds the contribution to the response from each source in turn with all other source, dependent or independent, properly removed and then adds the individual contributions to obtain the total response. No distinction is made between independent and dependent sources in the application of superposition other than requiring the network to have at least one independent source.

However, if dependent sources are present in the network, the quantities (call them V_x and I_x) on which the dependent sources depend must often be eliminated from the answer by additional analysis if the answer is to be useful unless V_x or I_x are themselves the variables of independent sources or the quantities sought in the analysis.

Some examples will illustrate the procedure.

Example 24.5

Find V_0 for the circuit shown using superposition. In this circuit, only independent sources are present.

Solution

Two sources in the network, therefore, we abstract two fictitious networks from Figure 24.20. The first is shown in Figure 24.21a and is obtained by properly removing the current source I_1 from the original network. The impedance of the capacitor can then be combined in parallel with $R_1 + R_2$, and the contribution to V_o from V_1 can be found using a voltage divider. The result is

$$V_o \text{ due to } V_1 = V_1 \frac{s + \frac{1}{C(R_1+R_2)}}{s + \frac{R_1+R_2+R_3}{C(R_1+R_2)R_3}}$$

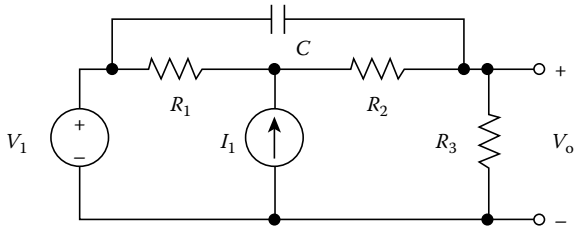


FIGURE 24.20 Network for Example 24.5.

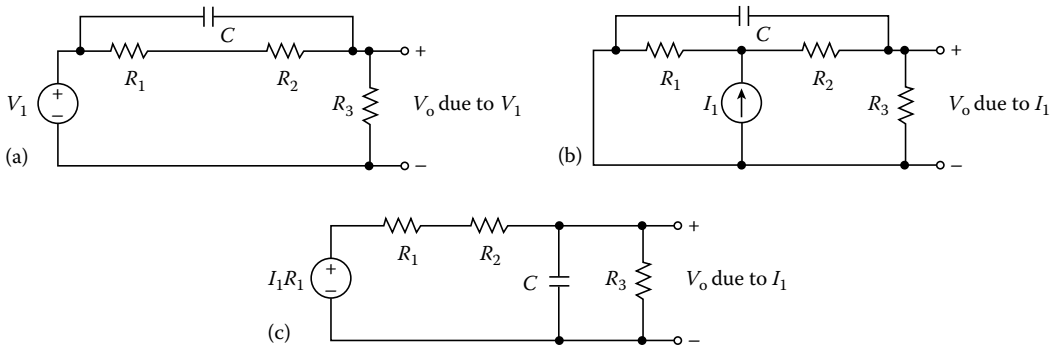


FIGURE 24.21 (a) Network obtained by properly removing the current source I_1 from the original network. (b) Network obtained from the original network by properly removing the voltage source V_1. (c) Final network obtained by redrawing the circuit and employing source transformation seven (in reverse).

The second fictitious network, shown in Figure 24.21b, is obtained from the original network by properly removing the voltage source V_1. Redrawing the circuit and employing source transformation seven (in reverse) yields the circuit in Figure 24.21c. Again, employing a voltage divider, we have

$$V_o \text{ due to } I_1 = I_1 \frac{\frac{R_1}{C(R_1+R_2)}}{s+\frac{R_1+R_2+R_3}{C(R_1+R_2)R_3}}$$

Then, adding the two contributions, we obtain

$$V_o = \frac{V_1\left[s+\frac{1}{C(R_1+R_2)}\right]+I_1\frac{R_1}{C(R_1+R_2)}}{s+\frac{R_1+R_2+R_3}{C(R_1+R_2)R_3}}$$

The next example includes a dependent source.

Example 24.6

Find i in the network shown in Figure 24.22 using superposition.

Solution

Since there are two sources, we abstract two fictitious networks from Figure 24.22. The first one is shown in Figure 24.23a and is obtained by properly removing the dependent current source. Thus,

$$i \text{ due to } v_1 = \frac{v_1}{R_1+R_2}$$

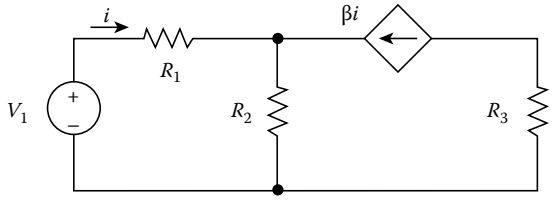


FIGURE 24.22 Network for Example 24.6.

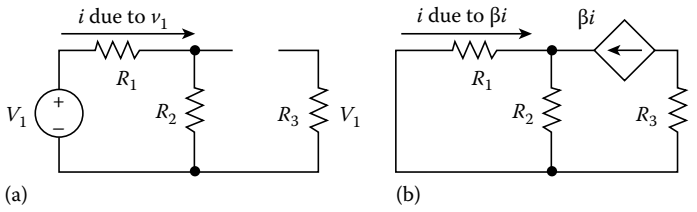


FIGURE 24.23 (a) Network obtained by properly removing the dependent current source. (b) Network obtained by properly removing the voltage source V_1.

Next, voltage source v_1 is properly removed yielding the fictitious network in Figure 24.23b. An important question immediately arises about this network. Namely, why is not i in this network zero? The reason i is not zero is that the network in Figure 24.23b is merely an abstracted network that concerns a step in the analysis of the original circuit. It is an artifice in the application of superposition, and the dependent source is considered to be independent for this step. Thus,

$$i \text{ due to } \beta i = -\frac{\beta i R_2}{R_1 + R_2}$$

Adding the two contributions, we obtain the intermediate result

$$i = \frac{v_1}{R_1 + R_2} - \frac{\beta i R_2}{R_1 + R_2}$$

Collecting the terms containing i, we obtain the finished solution for i:

$$i = \frac{v_1}{(\beta + 1)R_2 + R_1}$$

We note that the finished solution depends only on the independent source v_1 and parameters of the network, which are R_1, R_2, and β.

The following example involves a network in which a dependent source depends on a voltage that is neither the voltage of an independent source nor the voltage being sought in the analysis.

Example 24.7

Find V_o using superposition for the network shown in Figure 24.24. Note that K, the coefficient of the VCCS, has the units of siemens.

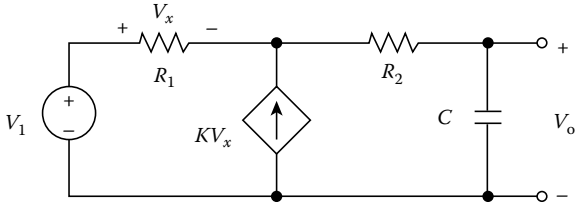


FIGURE 24.24 Network for Example 24.7.

Solution

When the dependent current source is properly removed, the network reduces to a simple voltage divider, and the contribution to V_o due to V_1 can be written as

$$V_o \text{ due to } V_1 = V_1 \frac{1}{sC(R_1 + R_2) + 1}$$

Then, reinserting the current source and properly removing the voltage source, we obtain the fictitious network shown in Figure 24.25. Using the current divider to obtain the current flowing through the capacitor and then multiplying this current by the impedance of the capacitor, we have

$$V_o \text{ due to } KV_x = \frac{KV_xR_1}{sC(R_1 + R_2) + 1}$$

Adding the individual contributions to form V_o provides the equation

$$V_o = \frac{V_1 + KV_xR_1}{sC(R_1 + R_2) + 1}$$

This is a valid expression for V_o. It is not a finished expression however, because it includes V_x, an unknown voltage. Superposition has taken us to this point in the analysis, but more work must be done to eliminate V_x. However, superposition can be applied again to solve for V_x, or other analysis tools can be used. The results for V_x are

$$V_x = \frac{V_1 sCR_1}{sC[R_1 + R_2 + R_1R_2K] + R_1K + 1}$$

Then, eliminating V_x from the equation for V_o, we obtain the finished solution as

$$V_o = V_1 \frac{R_1K + 1}{sC[R_1 + R_2 + R_1R_2K] + R_1K + 1}$$

Clearly, superposition is not the most efficient technique to use to analyze the network in Figure 24.24. Analysis based on a node equation written at the top end of the current source would yield a finished result for V_o with less algebra. However, this example does illustrate the application of superposition when a dependent source depends on a rather arbitrary voltage in the network.

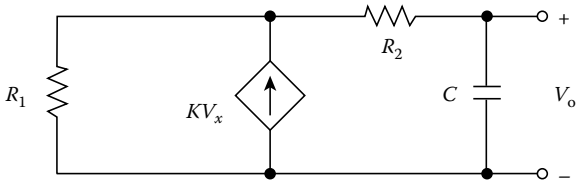


FIGURE 24.25 Fictitious network obtained when the voltage source is properly removed in Figure 24.24.

If the dependent current source in the previous example depended on V_1 instead of on the voltage across R_1, the network would be a different network. This is illustrated by the next example.

Example 24.8

Use superposition to determine V_o for the circuit in Figure 24.26.

Solution

If the current source is properly removed, the results are the same as for the previous example. Thus,

$$V_o \text{ due to } V_1 = \frac{V_1}{sC(R_1 + R_2) + 1}$$

Then, if the current source is reinserted, and the voltage source is properly removed, we have the circuit depicted in Figure 24.27. A question that can be asked for this circuit is why include the dependent source KV_1 if the voltage on which it depends, namely V_1, has been set to zero? However, the network shown in Figure 24.27 is merely a fictitious network that serves as an aid in the application of superposition, and superposition deals with all sources, whether they are dependent or independent, as if they were independent. Thus, we can write

$$V_o \text{ due to } KV_1 = \frac{KV_1R_1}{sC(R_1 + R_2) + 1}$$

Adding the contributions to form V_o, we obtain

$$V_o = V_1 \frac{KR_1 + 1}{sC(R_1 + R_2) + 1}$$

and this is the finished solution.

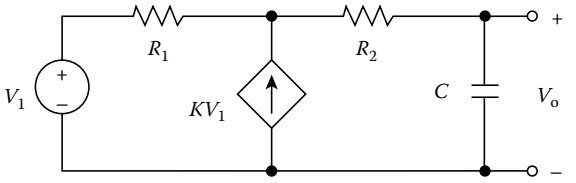


FIGURE 24.26 Network for Example 24.8.

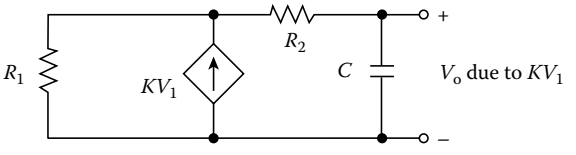


FIGURE 24.27 Step in the application of superposition to the network in Figure 24.26.

In this example, we did not have the task of eliminating an unknown quantity from an intermediate result for V_o because the dependent source depended on an independent source V_1, which is assumed to be known.

Superposition is often useful in the analysis of circuits having only independent sources, but it is especially useful in the analysis of some circuits having both independent and dependent sources because it deals with all sources as if they were independent.

24.2.4 Thévenin's Theorem

Thévenin's theorem is useful in reducing the complexity of a network so that analysis of the network for a particular voltage or current can be performed more easily. For example, consider Figure 24.28a, which is composed of two subnetworks A and B that have only two nodes in common. In order to facilitate analysis in subnetwork B, it is convenient to reduce subnetwork A to the network in Figure 24.28b which is termed the Thévenin equivalent of subnetwork A. The requirement on the Thévenin's equivalent network is that, when it replaces subnetwork A, the voltages and currents in subnetwork B remain unchanged. We assume that no inductive coupling occurs between the subnetworks, and that dependent sources in B are not dependent on voltages or currents in A. We also assume that subnetwork A is an LLFT network, but subnetwork B does not have to meet this assumption.

To find the Thévenin equivalent network, we need only determine V_{TH} and Z_{TH}. V_{TH} is found by unhooking B from A and finding the voltage that appears across the terminals of A. In other words, we abstract a fictitious network from the complete network as depicted in Figure 24.29a, and find the voltage that appears between the terminals that were common to B. This voltage is V_{TH}.

Z_{TH} is also obtained from a fictitious network that is created from the fictitious network used for finding V_{TH} by properly removing all independent sources. The effects that dependent sources have on the procedure are discussed later in this section. The fictitious network used for finding Z_{TH} is depicted in Figure 24.29b. Oftentimes, the expression for Z_{TH} cannot be found by mere inspection of this network, and, therefore, we must excite the network in Figure 24.29b by a voltage source or a current source and find an expression for the other variable at the port in order to find Z_{TH}.

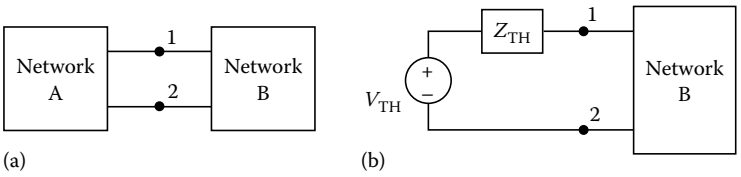


FIGURE 24.28 (a) Two subnetworks having a common pair of terminals. (b) The Thévenin equivalent for subnetwork *A*.

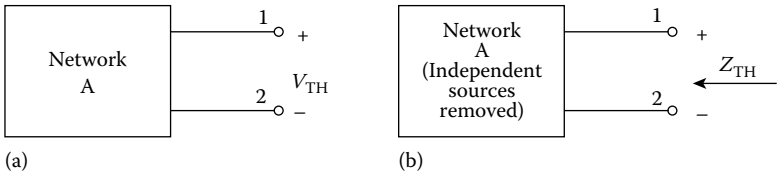


FIGURE 24.29 (a) Network used for finding V_{TH}. (b) Network used for obtaining Z_{TH}.

Example 24.9

Find the Thévenin equivalent of subnetwork *A* in Figure 24.30.

Solution

No dependent sources exist in subnetwork A, but the capacitor has an initial voltage *V* across it. However, the charged capacitor can be represented by an uncharged capacitor in series with a transformed voltage source *V*/*s*. The fictitious network used for finding V_{TH} is given in Figure 24.31a.

It should be noted that although subnetwork B has been removed and the two terminals that were connected to B are now "open circuited" in Figure 24.31a, current is still flowing in network A. V_{TH} is easily obtained using a voltage divider:

$$V_{TH} = \frac{V_1(s) + VCR}{sCR + 1}$$

Z_{TH} is obtained from the fictitious network in Figure 24.31b, which is obtained by properly removing the independent source and the voltage representing the initial voltage across the capacitor in Figure 24.31a. We see by inspection that $Z_{TH} = R/(sCR + 1)$. Thus, if subnetwork *A* is removed from Figure 24.30 and replaced by the Thévenin equivalent network, the voltages and currents in subnetwork B remain unchanged.

It is assumed that B in Figure 24.28 has no dependent sources that depend on voltages or currents in A, although dependent sources in B can depend on voltages and currents in B. However, A can have dependent sources, and these dependent sources create a modification in the procedure for finding the Thévenin equivalent network. There may be dependent sources in A that depend on voltages and currents that also exist in A. We call these dependent sources Case I-dependent sources. There may

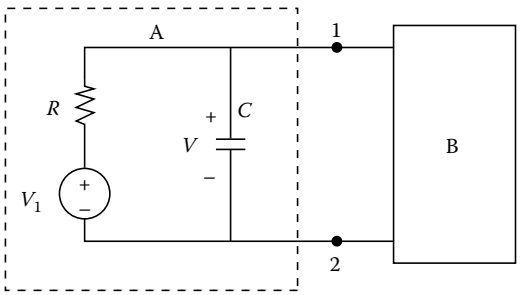


FIGURE 24.30 Network for Example 24.9.

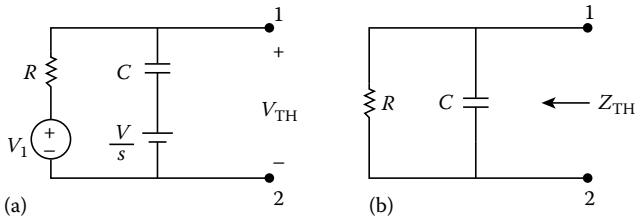


FIGURE 24.31 (a) Network of finding V_{TH}. (b) Network that yields Z_{TH}.

also be dependent sources in *A* that depend on voltages and currents in *B*, and we label these sources as Case II-dependent sources. Then, the procedure for finding the Thévenin equivalent network is:

V_{TH} is the voltage across the terminals of Figure 24.29a. The voltages and currents that Case I-dependent sources depend on must be eliminated from the expression for V_{TH} unless they happen to be the voltages of independent voltage sources or the currents of independent current sources in A. Otherwise, the expression for V_{TH} would not be a finished solution. However, Case II-dependent sources are handled as if they were independent sources. That is, Case II-dependent sources are included in the results for V_{TH} just as independent sources would be.

Z_{TH} is the impedance looking into the terminals in Figure 24.29b. In this fictitious network, independent sources are properly removed and Case II-dependent sources are properly removed. Case I-dependent sources remain in the network and influence the result for the Thévenin impedance. The finished solution for Z_{TH} is a function only of the parameters of the network in Figure 24.29b which are *Rs*, *Ls*, *Cs*, *Ms* (there may be inductive coupling between coils in this network), and the coefficients of the Case I-dependent sources.

Thus, Case II-dependent sources, sources that depend on voltages or currents in subnetwork B, are uniformly treated as if they were independent sources in finding the Thévenin equivalent network. Some examples will clarify the issue.

Example 24.10

Find the Thévenin equivalent network for subnetwork A in Figure 24.32. Assume the initial current through the inductor is zero.

Solution

There is one independent source and one Case I-dependent source. Figure 24.33a depicts the fictitious network to be analyzed to obtain V_{TH}. No current is flowing through R_2 in this figure, therefore, we can write $V_{TH} = V_1 - V_x$. To eliminate V_x from our intermediate expression for V_{TH}, we can use the results of the enhanced voltage divider to write

$$V_{TH} = \frac{V_1 R_1 + sKL(V_1 - V_{TH})}{R_1 + sL}$$

The finished solution for V_{TH} is

$$V_{TH} = V_1 \frac{sKL + R_1}{(K+1)sL + R_1}$$

Z_{TH} is obtained from Figure 24.33b where a current source excitation is shown already applied to the fictitious network. Two node equations, with unknown node voltages V and $-V_x$, enable us to obtain I in

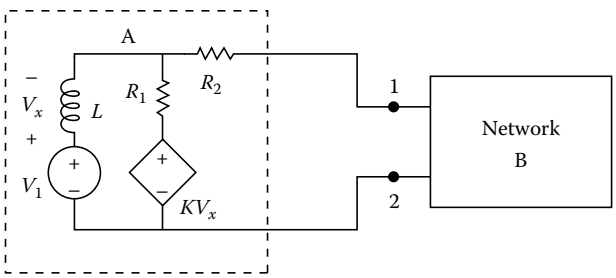


FIGURE 24.32 Network for Example 24.10.

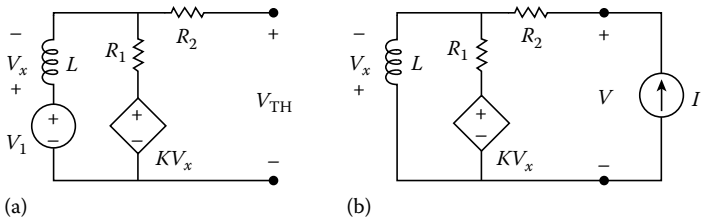


FIGURE 24.33 (a) Abstracted network for finding V_{TH}. (b) Abstracted network for finding Z_{TH}.

terms of V while eliminating V_x. We also note that Z_{TH} consists of resistor R_2 in series with some unknown impedance, so we could remove R_2 (replaced it by a short) if we remember to add it back later. The finished result for Z_{TH} is

$$Z_{TH} = R_2 + \frac{sLR_1}{(K+1)sL + R_1}$$

The following example involves a network having a Case II-dependent source.

Example 24.11

Find the Thévenin equivalent network for subnetwork *A* in the network illustrated in Figure 24.34. In this instance, subnetwork B is outlined explicitly.

Solution

Subnetwork A contains one independent source and one Case II-dependent source. Figure 24.35a is the abstracted network for finding V_{TH}. Thus,

$$V_{TH} = V_1 + KIR_1$$

Then, both V_1 and the dependent source KI are deleted from Figure 24.35a to obtain Figure 24.35b, the network used for finding Z_{TH}. Thus, $Z_{TH} = R_1$.

Of course, the subnetwork for which the Thévenin equivalent is being determined may have both Case I- and Case II-dependent sources, but these sources can be handled concurrently using the procedures given previously.

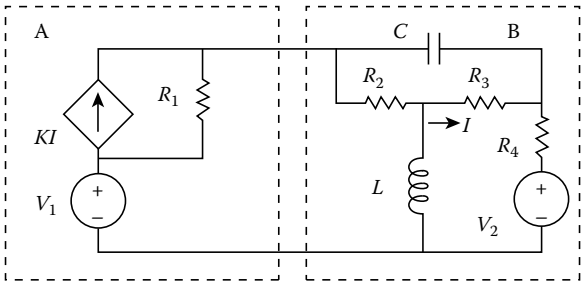


FIGURE 24.34 Network for Example 24.11.

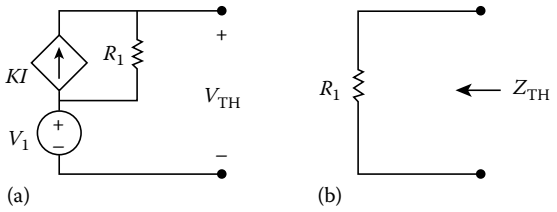


FIGURE 24.35 (a) Network used to find V_{TH}. (b) Network for finding Z_{TH}.

Special conditions can arise in the application of Thévenin's theorem. One condition is $Z_{TH} = 0$ and the other is $V_{TH} = 0$. The conditions for which Z_{TH} is zero are

1. If the circuit (subnetwork A) for which the Thévenin equivalent is being determined has an independent voltage source connected between terminals 1 and 2, then $Z_{TH} = 0$. Figure 24.36a illustrates this case.
2. If subnetwork A has a dependent voltage source connected between terminals 1 and 2, then Z_{TH} is zero provided neither of the port variables associated with the port formed by terminals 1 and 2 is coupled back into the network. Figure 24.36b is a subnetwork A for which Z_{TH} is zero. However, Figure 24.36c depicts a subnetwork A in which the port variable I is coupled back into A by the dependent source K_1I. If I is considered to be a variable of subnetwork A so that K_1I is a Case I-dependent source, then Z_{TH} is not zero.

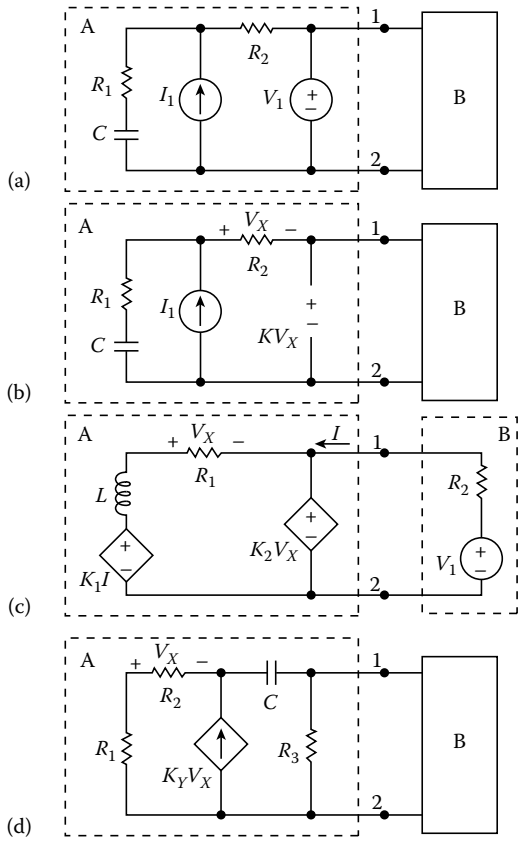


FIGURE 24.36 Special cases of Thévenin's theorem. (a, b) Z_{TH} equals zero. (c) A port variable is coupled back into A. (d) V_{TH} is zero.

The other special condition, $V_{TH} = 0$, occurs if subnetwork A contains only Case I-dependent sources, no independent sources, and no Case II-dependent sources. An example of such a network is given in Figure 24.36d. With subnetwork B disconnected, subnetwork A is a dead network, and its Thévenin voltage is zero.

The network in Figure 24.36c is of interest because the dependent source $K_1 I$ can be considered as a Case I or a Case II-dependent source hinging on whether I is considered a variable of subnetwork A or B.

Example 24.12

Solve for I in Figure 24.36c using two versions of the Thévenin equivalent for subnetwork A. For the first version, consider I to be associated with A, and therefore both dependent sources are Case I-dependent sources. In the second version, consider/to be associated with B.

Solution

If I is considered as associated with A, then V_{TH} is zero by inspection because A contains only Case I-dependent sources. Figure 24.37a depicts subnetwork A with a current excitation applied in order to determine Z_{TH}. Clearly, $V = K_2 V_x$. Also, writing a loop equation in the loop encompassed by the two dependent sources, we obtain

$$K_1 I - \frac{V_x}{R_1}(sL + R_1) = V$$

Eliminating V_x, we have

$$\frac{V}{I} = Z_{TH} = \frac{K_1 K_2 R_1}{sL + R_1(K_2 + 1)}$$

Once Z_{TH} is obtained, it is an easy matter to write from Figure 24.36c:

$$I = \frac{V_1}{R_2 + Z_{TH}} = \frac{V_1[sL + R_1(K_2 + 1)]}{sLR_2 + R_1 R_2(K_2 + 1) + K_1 K_2 R_1}$$

If I is associated with B, then V_{TH} is found from the network in Figure 24.37b with the source $K_1 I$ treated as if it were independent. The equation for V_{TH} may contain the variable I, but V_x must be eliminated from the finished expression for V_{TH}. We obtain

$$V_{TH} = I\frac{K_1 K_2 R_1}{sL + R_1(K_2 + 1)}$$

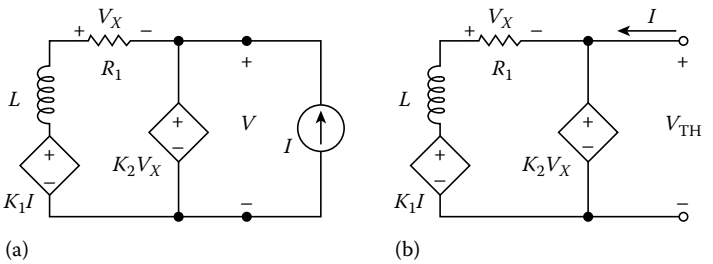


FIGURE 24.37 (a) Network for finding Z_{TH} when both sources are Case I-dependent sources. (b) Network for finding V_{TH} when a Case II-dependent source exists in the network.

Also, Z_{TH} is zero because if $K_1 I$ is removed, subnetwork A reduces to a network with only a Case I-dependent source and a port variable is not coupled back into the network. Finally, I can be written as

$$I = \frac{V_1 - V_{TH}}{R_2}$$

which yields the same result for I as was found previously.

The following example illustrates the interplay that can be achieved among these theorems and source transformations.

Example 24.13

Find V_0/V_1 for the bridged T network shown in Figure 24.38.

Solution

The application of source transformation eight to the network yields the ladder network in Figure 24.39a. Thévenin's theorem is particularly useful in analyzing ladder networks. If it is applied to the left and right sides of the network, taking care not to obscure the nodes between which V_0 exists, we obtain the single loop network in Figure 24.39b. Then, using a voltage divider, we obtain

$$\frac{V_o}{V_1} = \frac{Z_4[Z_1(Z_2 + Z_3) + Z_2Z_3 + Z_2Z_5]}{(Z_1 + Z_2)(Z_3Z_4 + Z_3Z_5 + Z_4Z_5) + Z_1Z_2(Z_4 + Z_5)}$$

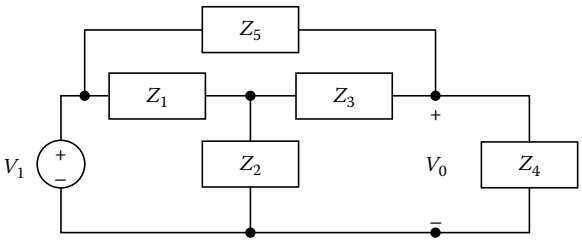


FIGURE 24.38 Network for Example 24.13.

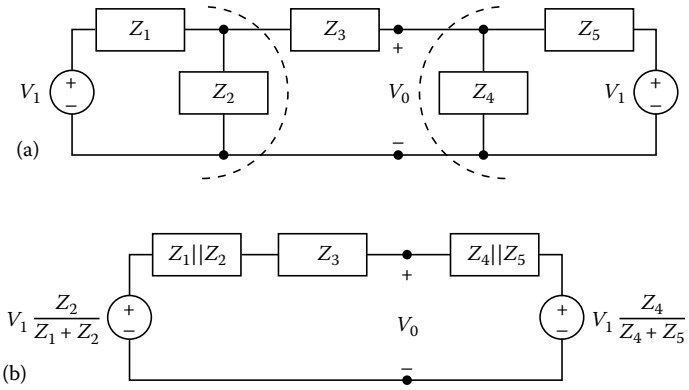


FIGURE 24.39 (a) Results after applying source transformation eight to the network shown in Figure 24.38. (b) Results of two applications of Thévenin's theorem.

24.2.5 Norton's Theorem

If a source transformation is applied to the Thévenin equivalent network consisting of V_{TH} and Z_{TH} in Figure 24.28b, then a Norton equivalent network, illustrated in Figure 24.40a, is obtained. The current source $I_{sc} = V_{TH}/Z_{TH}$, $Z_{TH} \neq 0$. If Z_{TH} equals zero in Figure 24.28b, then the Norton equivalent network does not exist. The subscripts sc on the current source stand for short circuit and indicate a procedure for finding the value of this current source. To find I_{sc} for subnetwork A in Figure 24.28a, we disconnet subnetwork B and place a short circuit between nodes 1 and 2 of subnetwork *A*. Then, I_{sc} is the current flowing through the short circuit in the direction indicated in Figure 24.40b. I_{sc} is zero if subnetwork A has only Case I-dependent sources and no other sources. Z_{TH} is found in the same manner as for Thévenin's theorem.

It is sometimes more convenient to find I_{sc} and V_{TH} instead of Z_{TH}.

Example 24.14

Find the Norton equivalent for the network "seen" by Z_L in Figure 24.41. That is, Z_L is subnetwork B and the rest of the network is A, and we wish to find the Norton equivalent network for A.

Solution

Figure 24.42a is the network with Z_L replaced by a short circuit. An equation for I_{sc} can be obtained quickly using superposition. This yields

$$I_{sc} = I_1 + \frac{KI_1}{R_2}$$

but I_1 must be eliminated from this equation. I_1 is obtained as: $I_1 = V_1/(sL + R_1)$. Thus,

$$I_{sc} = \frac{V_1\left(1 + \frac{K}{R_2}\right)}{sL + R_1}$$

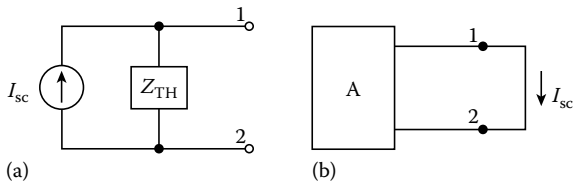


FIGURE 24.40 (a) Norton equivalent network. (b) Reference direction for I_{sc}.

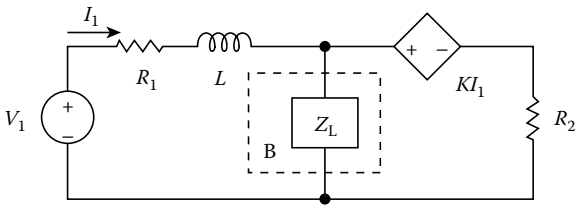


FIGURE 24.41 Network for Example 24.14.

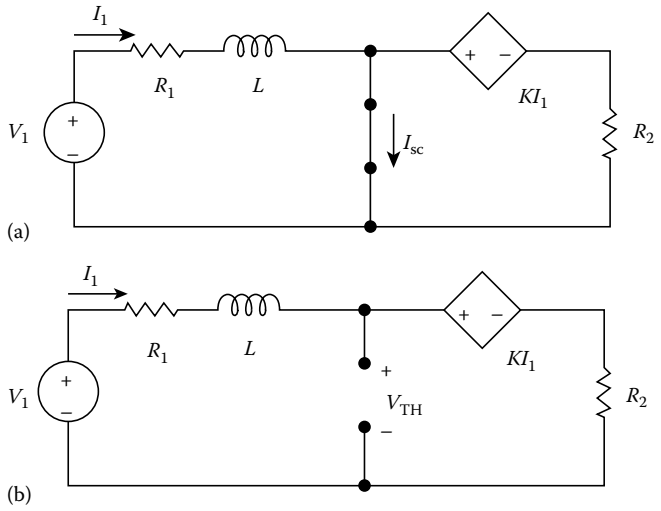


FIGURE 24.42 (a) Network for finding $I_{sc.}$ (b) Network for V_{TH}.

V_{TH} is found from the network shown in Figure 24.42b. The results are:

$$V_{TH} = \frac{V_1(K + R_2)}{sL + R_1 + R_2 + K}$$

Z_{TH} can be found as V_{TH}/I_{sc}.

24.2.6 Thévenin's and Norton's Theorems and Network Equations

Thévenin's and Norton's theorems can be related to loop and node equations. Here, we examine the relationship to loop equations by means of the LLFT network in Figure 24.43. Assume that the network N in Figure 24.43 has n independent loops with all the loop currents chosen in the same direction. Without loss of generality, assume that only one loop current, say I_1, flows through Z_L as shown so that Z_L appears in only one loop equation. For simplicity, assume that no dependent sources or inductive couplings in N exist, and that all current sources have been source-transformed so that only voltage source excitations remain. Then the loop equations are

$$\begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \tag{24.32}$$

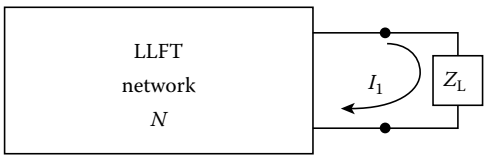


FIGURE 24.43 LLFT network N with n independent loops.

where V_i, $i = 1, 2, \ldots, n$, is the sum of all voltage sources in the ith loop. Thus, V_i may consist of several terms, some of which may be negative depending on whether a voltage source is a voltage rise or a voltage drop. Also, the impedances z_{ij} are given by

$$z_{ij} = \pm\left[R_{ij} + sL_{ij} + \frac{D_{ij}}{s}\right] \tag{24.33}$$

where $i, j = 1, 2, \ldots, n$, and where the plus sign is taken if $i = j$, and the minus sign is used if $i \neq j$. R_{ij} is the sum of the resistances in loop i if $i = j$, and R_{ij} is the sum of the resistances common to loops i and j if $i \neq j$. Similar statements apply to the inductances L_{ij} and to the reciprocal capacitances D_{ij}. The currents I_i, $i = 1, 2, \ldots, n$, are the unknown loop currents.

Note that Z_L is included only in z_{11} so that z_{11} can be written as $z_{11} = z_{11A} + Z_L$, where z_{11A} is the sum of all the impedances around loop one except Z_L. Solving for I_1 using Cramer's rule, we have

$$I_1 = \frac{\begin{vmatrix} V_1 & z_{12} & \cdots & z_{1n} \\ V_2 & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ V_n & z_{n2} & \cdots & z_{nn} \end{vmatrix}}{\Delta} \tag{24.34}$$

where Δ is the determinant of the loop impedance matrix. Thus, we can write

$$I_1 = \frac{V_1\Delta_{11} + V_2\Delta_{21} + \cdots + V_n\Delta_{n1}}{z_{11}\Delta_{11} + z_{21}\Delta_{21} + \cdots + z_{n1}\Delta_{n1}} \tag{24.35}$$

or

$$I_1 = \frac{V_1 + V_2\frac{\Delta_{21}}{\Delta_{11}} + \cdots + V_n\frac{\Delta_{n1}}{\Delta_{11}}}{z_{11} + z_{21}\frac{\Delta_{21}}{\Delta_{11}} + \cdots + z_{n1}\frac{\Delta_{n1}}{\Delta_{11}}} \tag{24.36}$$

where Δ_{ij} are cofactors of the loop impedance matrix. Then, forming the product of I_1 and Z_L, we have

$$I_1 Z_L = \frac{Z_L\left(V_1 + V_2\frac{\Delta_{21}}{\Delta_{11}} + \cdots + V_n\frac{\Delta_{n1}}{\Delta_{11}}\right)}{Z_L + z_{11A} + z_{21}\frac{\Delta_{21}}{\Delta_{11}} + \cdots + z_{n1}\frac{\Delta_{n1}}{\Delta_{11}}} \tag{24.37}$$

If we take the limit of $I_1 Z_L$ as Z_L approaches infinity, we obtain the "open circuit" voltage V_{TH}. That is,

$$\lim_{Z_L \to \infty} I_1 Z_L = V_{TH} = \left(V_1 + V_2\frac{\Delta_{21}}{\Delta_{11}} + \cdots + V_n\frac{\Delta_{n1}}{\Delta_{11}}\right) \tag{24.38}$$

and if we take the limit of I_1 as Z_L approaches zero, we obtain the "short circuit" current I_{sc}:

$$\lim_{Z_L \to 0} I_1 = I_{sc} = \frac{V_{TH}}{z_{11A} + z_{21}\frac{\Delta_{21}}{\Delta_{11}} + \cdots + z_{n1}\frac{\Delta_{n1}}{\Delta_{11}}} \tag{24.39}$$

Finally, the quotient of V_{TH} and I_{sc} yields

$$\frac{V_{TH}}{I_{sc}} = Z_{TH} = z_{11A} + z_{21}\frac{\Delta_{21}}{\Delta_{11}} + \cdots + z_{n1}\frac{\Delta_{n1}}{\Delta_{11}} \tag{24.40}$$

If network N contains coupled inductors (but not coupled to Z_L), then some elements of the loop impedance matrix may be modified in value and sign. If network N contains dependent sources, then auxiliary equations can be written to express the quantities on which the dependent sources depend in terms of the independent excitations and/or the unknown loop currents. Thus, dependent sources may modify the elements of the loop impedance matrix in value and sign, and they may modify the elements of the excitation column matrix $[V_i]$. Nevertheless, we can obtain expressions similar to those obtained previously for V_{TH} and I_{sc}. Of course, we must exclude from this illustration dependent sources that depend on the voltage across Z_L because they violate the assumption that Z_L appears in only one loop equation and are beyond the scope of this discussion.

24.2.7 π–T Conversion

The π–T conversion is employed for the simplification of circuits, especially in power systems analysis. The "π" refers to a circuit having the topology shown in Figure 24.44. In this figure, the left-most and right-most loop currents have been chosen to coincide with the port currents for convenience of notation only.

A circuit having the topology shown in Figure 24.45 is referred to as a "T" or as a "Y." We wish to determine equations for Z_1, Z_2, and Z_3 in terms of Z_A, Z_B, and Z_C so that the π can be replaced by a T without affecting any of the port variables. In other words, if an overall circuit contains a π subcircuit,

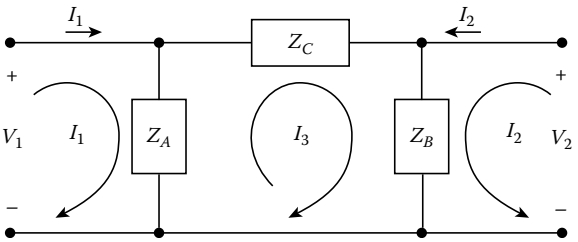


FIGURE 24.44 π network shown with loop currents.

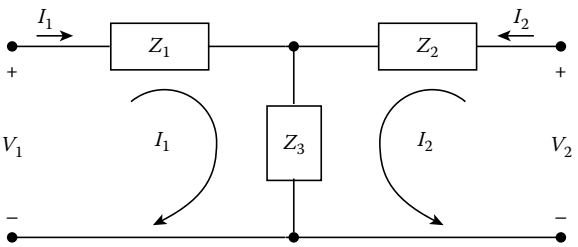


FIGURE 24.45 T network.

we wish to replace the π subscript with a T subscript without disturbing any of the other voltages and currents within the overall circuit. To determine what Z_1, Z_2, and Z_3 should be, we first write loop equations for the π network. The results are

$$V_1 = I_1 Z_A - I_3 Z_A \tag{24.41}$$

$$V_2 = I_2 Z_B + I_3 Z_B \tag{24.42}$$

$$0 = I_3(Z_A + Z_B + Z_C) - I_1 Z_A + I_2 Z_B \tag{24.43}$$

But the T circuit has only two loop equations given by

$$V_1 = I_1(Z_1 + Z_3) + I_2 Z_3 \tag{24.44}$$

$$V_2 = I_1 Z_3 + I_2(Z_2 + Z_3) \tag{24.45}$$

We must eliminate one of the loop equations for the π circuit, and so we solve for I_3 in Equation 24.43 and substitute the result into Equations 24.41 and 24.42 to obtain

$$V_1 = I_1\left[\frac{Z_A(Z_B + Z_C)}{Z_A + Z_B + Z_C}\right] + I_2\left[\frac{Z_A Z_B}{Z_A + Z_B + Z_C}\right] \tag{24.46}$$

$$V_2 = I_1\left[\frac{Z_A Z_B}{Z_A + Z_B + Z_C}\right] + I_2\left[\frac{Z_B(Z_A + Z_C)}{Z_A + Z_B + Z_C}\right] \tag{24.47}$$

From a comparison of the coefficients of the currents in Equations 24.46 and 24.47 with those in Equations 24.44 and 24.45, we obtain the following relationships.

24.2.7.1 Replacing π with T

$$Z_1 = \frac{Z_A Z_C}{S_Z}; \quad Z_2 = \frac{Z_B Z_C}{S_Z}; \quad Z_3 = \frac{Z_A Z_B}{S_Z} \tag{24.48}$$

where

$$S_Z = Z_A + Z_B + Z_C$$

We can also replace a T network by a π network. To do this we need equations for Z_A, Z_B, and Z_C in terms of Z_1, Z_2, and Z_3. The required equations can be obtained algebraically from Equation 24.48.

24.2.7.2 From T to π

$$Z_A = Z_1 + Z_3 + \frac{Z_1 Z_3}{Z_2}; \quad Z_B = Z_2 + Z_3 + \frac{Z_2 Z_3}{Z_1}; \quad Z_C = Z_1 + Z_2 + \frac{Z_1 Z_2}{Z_3} \tag{24.49}$$

24.2.8 Reciprocity

If an LLFT network contains only *Rs*, *Ls*, *Cs*, and transformers but contains no dependent sources, then its loop impedance matrix is symmetrical with respect to the main diagonal. That is, if z_{ij} is an element of

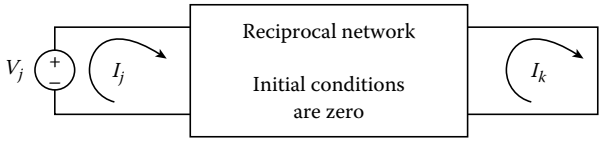


FIGURE 24.46 Reciprocal network with m independent loops.

the loop impedance matrix (see Equation 24.17), occupying the position at row i and column j, then $z_{ji} = z_{ij}$, where z_{ji} occupies the position at row j and column i. Such a network has the property of reciprocity and is termed a reciprocal network.

Assume that a reciprocal network, depicted in Figure 24.46, has m loops and is in the zero state. It has only one excitation—a voltage source in loop j. To solve for the loop current in loop k, we write the loop equations:

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ z_{j1} & z_{j2} & \cdots & z_{jm} \\ z_{k1} & z_{k2} & \cdots & z_{km} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \cdots & z_{mm} \end{bmatrix} \begin{bmatrix} I_2 \\ I_2 \\ \vdots \\ I_j \\ I_k \\ \vdots \\ I_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ V_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{24.50}$$

The column excitation matrix has only one nonzero entry. To determine I_k using Cramer's rule, we replace column k by the excitation column and then expand along this column. Only one nonzero term is in the column, therefore, we obtain a single term for I_k:

$$I_k = V_j \frac{\Delta_{jk}}{\Delta} \tag{24.51}$$

where

Δ_{jk} is the cofactor

Δ is the determinant of the loop impedance matrix

Next, we replace the voltage source by a short circuit in loop j, cut the wire in loop k, and insert a voltage source V_k. Figure 24.47 outlines the modifications to the circuit. Then, we solve for I_j obtaining

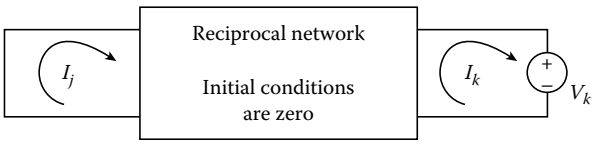


FIGURE 24.47 Interchange of the ports of excitation in the network in Figure 24.46.

$$I_j = V_k \frac{\Delta_{kj}}{\Delta} \tag{24.52}$$

Because the network is reciprocal, $\Delta_{jk} = \Delta_{kj}$ so that

$$\frac{I_k}{V_j} = \frac{I_j}{V_k} \tag{24.53}$$

Equation 24.53 is the statement of reciprocity for the network in Figures 24.46 and 24.47 with the excitations shown.

Figure 24.48a is a reciprocal network with a current excitation applied to node j and a voltage response, labeled V_k, taken between nodes k and m. We assume the network has n independent nodes plus the ground node indicated and is not a grounded network (does not have a common connection between the input and output ports shown). If we write node equations to solve for V_k in Figure 24.48a and use Cramer's rule, we have

$$V_k = I_j \frac{\Delta'_{jk} - \Delta'_{jm}}{\Delta'} \tag{24.54}$$

where the primes indicate node-basis determinants. Then, we interchange the ports of excitation and response as depicted in Figure 24.48b. If we solve for V_j in Figure 24.48b, we obtain

$$V_j = I_k \frac{\Delta'_{kj} - \Delta'_{mj}}{\Delta'} \tag{24.55}$$

Because the corresponding determinants in Equations 24.54 and 24.55 are equal because of reciprocity, we have

$$\frac{V_k}{I_j} = \frac{V_j}{I_k} \tag{24.56}$$

Note that the excitations and responses are of the opposite type in Figures 24.46 and 24.48. The results obtained in Equations 24.53 and 24.56 do not apply if the excitation and response are both voltages or both currents because when the ports of excitation and response are interchanged, the impedance levels of the network are changed [2].

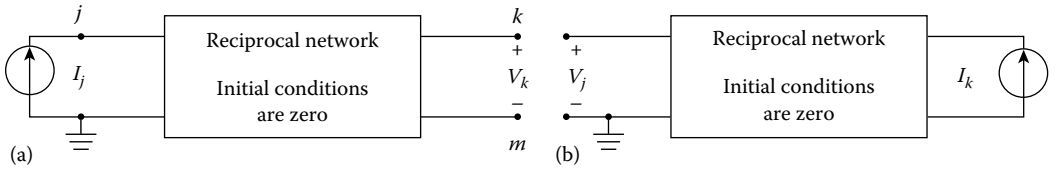


FIGURE 24.48 (a) Reciprocal ungrounded network with a current source excitation. (b) Interchange of the ports of excitation and response.

24.2.9 Middlebrook's Extra Element Theorem

Middlebrook's extra element theorem is useful in developing tests for analog circuits and for predicting the effects that parasitic elements may have on a circuit. This theorem has two versions: the parallel version and the series version. Both versions present the results of connecting an extra network element in the circuit as the product of the network function obtained without the extra element times a correction factor. This is a particularly convenient form for the results because it shows exactly the effects of the extra element on the network function.

Parallel version. Consider an arbitrary LLFT network having a transfer function $A_1(s)$. In the parallel version of the theorem, an impedance is added between any two independent nodes of the network. The modified transfer function is then obtained as $A_1(s)$ multiplied by a correction factor. Figure 24.49 is an arbitrary network in which the quantities U_i and U_o represent a general input and a general output, respectively, whether they are voltages or currents. The extra element is to be connected between terminals 1 and 1′ in Figure 24.49, and the port variables for this port are V_2 and I_2.

We can write

$$\begin{aligned} U_o &= A_1 U_i + A_2 I_2 \\ V_2 &= B_1 U_i + B_2 I_2 \end{aligned} \tag{24.57}$$

where

$$\begin{aligned} A_1 &= \left.\frac{U_o}{U_i}\right|_{I_2=0} \quad A_2 = \left.\frac{U_o}{I_2}\right|_{U_i=0} \\ B_1 &= \left.\frac{V_2}{U_i}\right|_{I_2=0} \quad B_2 = \left.\frac{V_2}{I_2}\right|_{U_i=0} \end{aligned} \tag{24.58}$$

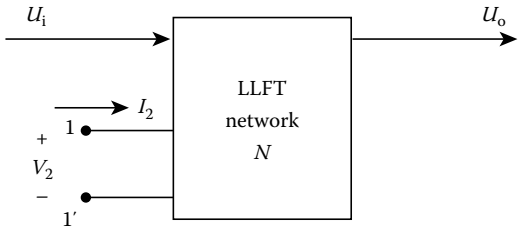


FIGURE 24.49 An arbitrary LLFT network.

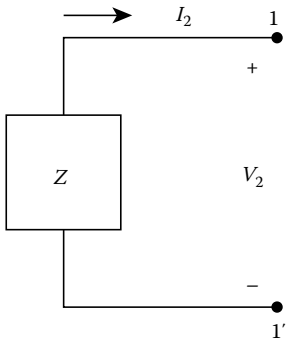


FIGURE 24.50 Extra element Z.

Note that A_1 is assumed to be known.

The extra element Z to be added across terminals 1 and 1′ is depicted in Figure 24.50. It can be described as $Z = V_2/(-I_2)$ which yields $I_2 = V_2/(-Z)$. Substituting this expression for I_2 into Equation 24.57 results in

$$\begin{aligned} U_o &= A_1 U_i + A_2\left(\frac{-V_2}{Z}\right) \\ V_2\left(1 + \frac{B_2}{Z}\right) &= B_1 U_i \end{aligned} \tag{24.59}$$

After eliminating V_2 and solving for U_o/U_i, we obtain

$$\frac{U_o}{U_i} = A_1\left[\frac{1 + \frac{1}{Z}\left(\frac{A_1B_2 - A_2B_1}{A_1}\right)}{1 + \frac{B_2}{Z}}\right] \tag{24.60}$$

Next, we provide physical interpretations for the terms in Equation 24.60. Clearly, B_2 is the

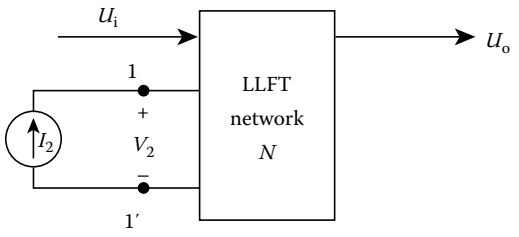


FIGURE 24.51 Network of Figure 24.49 with two excitations applied.

impedance seen looking into the network between terminals 1 and 1′ with $U_i = 0$. Thus, rename $B_2 = Z_d$ where d stands for "dead network."

To find a physical interpretation of $(A_1B_2 - A_2B_1)/A_1$, examine the network in Figure 24.51. Here, two excitations are applied to the network, namely U_i and I_2. Simultaneously adjust both inputs so as to null output U_o. Thus, with $U_o = 0$, we have from Equation 24.57,

$$U_i = \frac{-A_2}{A_1} I_2 \tag{24.61}$$

Substituting this result into the equation for V_2 in Equation 24.57, we have

$$V_2 = B_1 \left(\frac{-A_2}{A_1} \right) I_2 + B_2 I_2 \tag{24.62}$$

or

$$\left. \frac{V_2}{I_2} \right|_{U_o = 0} = \frac{A_1B_2 - A_2B_1}{A_1}$$

Because the quantity $(A_1B_2 - A_2B_1)/A_1$ is the ratio of V_2 to I_2 with the output nulled, we rename this quantity as Z_N. Then rewriting Equation 24.60 with Z_d and Z_N, we have

$$\frac{U_o}{U_i} = A_1 \left[\frac{1 + \frac{Z_N}{Z}}{1 + \frac{Z_d}{Z}} \right] \tag{24.63}$$

Equation 24.63 demonstrates that the results of connecting the extra element Z into the circuit can be expressed as the product of A_1, which is the network function with Z set to infinity, times a correction factor given in the brackets in Equation 24.63.

Example 24.15

Use the parallel version of Middlebrook's extra element theorem to find the voltage transfer function of the ideal op-amp circuit in Figure 24.52 when a capacitor C is connected between terminals 1 and 1′.

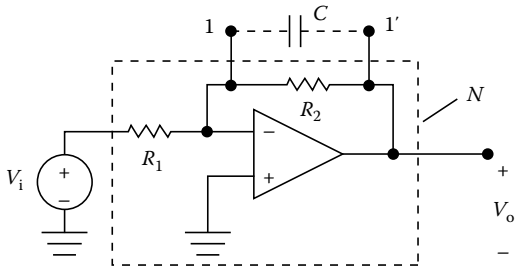


FIGURE 24.52 Ideal op-amp circuit for Example 24.15.

Solution

With the capacitor not connected, the voltage transfer function is

$$\left.\frac{V_o}{V_i}\right|_{Z=\infty} = -\frac{R_2}{R_1} = A_1$$

Next, we determine Z_d from the circuit illustrated in Figure 24.53a, where a model has been included for the ideal op-amp, the excitation V_i has been properly removed, and a current excitation I_2 has been applied to the port formed by terminals 1 and 1′. Because no voltage flows across R_1 in Figure 24.53a, no current flows through it, and all the current I_2 flows through R_2. Thus, $V_2 = I_2R_2$, and $Z_d = R_2$. We next find Z_N from Figure 24.53b. We observe in this figure that the right end of R_2 is zero volts above ground because V_i and I_2 have been adjusted so that V_o is zero. Furthermore, the left end of R_2 is zero volts above ground because of the virtual ground of the op-amp. Thus, zero is current flowing through R_2, and so V_2 is zero. Consequently, $Z_N = V_2/I_2 = 0$. Following the format of Equation 24.63, we have

$$\frac{V_o}{V_i} = -\frac{R_2}{R_1}\left(\frac{1}{1 + sCR_2}\right)$$

Note that for V_o to be zero in Figure 24.53b, V_i and I_2 must be adjusted so that $V_i/R_1 = -I_2$, although this information was not needed to work the example.

Series version. The series version of the theorem allows us to cut a loop of the network, add an impedance Z in series, and obtain the modified network function as $A_1(s)$ multiplied by a correction factor. A_1 is the

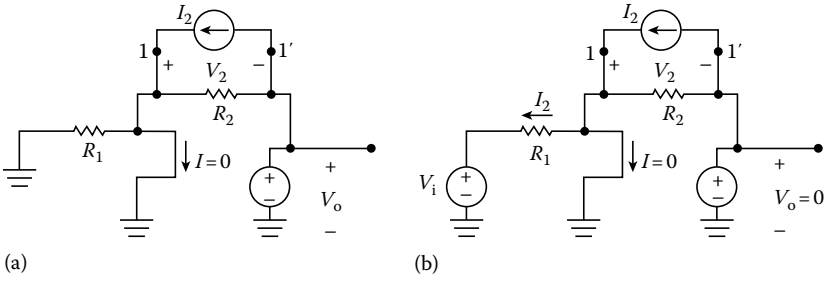


FIGURE 24.53 (a) Network for finding Z_d. (b) Network used to determine Z_N.

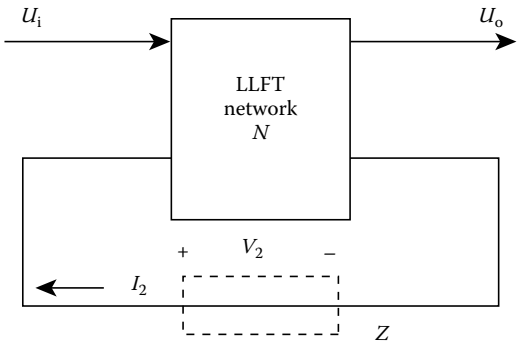


FIGURE 24.54 LLFT network used for the series version of Middlebrook's extra element theorem.

network function when $Z = 0$. Figure 24.54 is an LLFT network with part of a loop illustrated explicitly. The quantities U_i and U_o represent a general input and a general output, respectively, whether they be a voltage or a current. Define

$$\begin{aligned} A_1 &= \left.\frac{U_o}{U_i}\right|_{V_2=0} \quad A_2 = \left.\frac{U_o}{V_2}\right|_{U_i=0} \\ B_1 &= \left.\frac{I_2}{U_i}\right|_{V_2=0} \quad B_2 = \left.\frac{I_2}{V_2}\right|_{U_i=0} \end{aligned} \tag{24.64}$$

where V_2 and I_2 are depicted in Figure 24.54, and A_1 is assumed to be known. Then using superposition, we have

$$\begin{aligned} U_o &= A_1 U_i + A_2 V_2 \\ I_2 &= B_1 U_i + B_2 V_2 \end{aligned} \tag{24.65}$$

The impedance of the extra element Z can be described by $Z = V_2/(-I_2)$ so that $V_2 = -I_2 Z$. Substituting this relation for V_2 into Equation 24.65 and eliminating I_2, we have

$$\frac{U_o}{U_i} = A_1 \left[\frac{1 + Z\left(B_2 - B_1 \frac{A_2}{A_1}\right)}{1 + B_2 Z}\right] \tag{24.66}$$

Again, as we did for the parallel version of the theorem, we look for physical interpretations of the quantities in the square bracket in Equation 24.66. From Equation 24.65 we see that B_2 is the admittance looking into the port formed by cutting the loop in Figure 24.54 with $U_i = 0$. This is depicted in Figure 24.55a. Thus, B_2 is the admittance looking into a dead network, and so let $B_2 = 1/Z_d$.

To find a physical interpretation of the quantity $(A_1B_2 - A_2B_1)/A_1$, we examine Figure 24.55b in which both inputs, V_2 and U_i, are adjusted to null the output U_o. From Equation 24.65 with $U_o = 0$, we have

$$U_i = -\frac{A_2}{A_1} V_2 \tag{24.67}$$

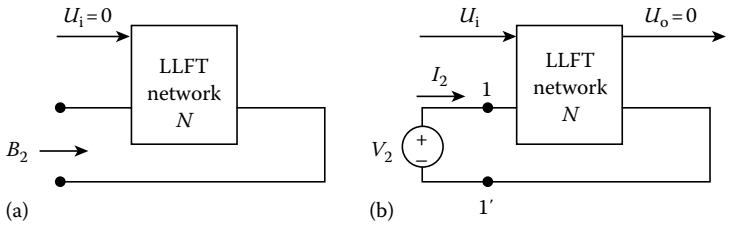


FIGURE 24.55 (a) Looking into the network with U_i equal zero. (b) U_i and V_2 are simultaneously adjusted to null the output U_o.

Then, eliminating U_i in Equation 24.65 we obtain

$$\frac{A_1B_2 - A_2B_1}{A_1} = \left.\frac{I_2}{V_2}\right|_{U_o=0} \tag{24.68}$$

Because this quantity is the admittance looking into the port formed by terminals 1 and 1′ in Figure 24.55b with U_o nulled, rename it as $1/Z_N$. Thus, from Equation 24.66 we can write

$$\frac{U_o}{U_i} = A_1\left[\frac{1+\frac{Z}{Z_N}}{1+\frac{Z}{Z_d}}\right] \tag{24.69}$$

Equation 24.69 is particularly convenient for determining the effects of adding an impedance Z into a loop of a network.

Example 24.16

Use the series version of Middlebrook's extra element theorem to determine the effects of inserting a capacitor C in the location indicated in Figure 24.56.

Solution

The voltage transfer function for the network without the capacitor is found to be

$$A_1 = \left.\frac{V_o}{V_i}\right|_{Z=0} = \frac{-\beta R_L}{R_s + (\beta+1)R_e\left(1+\frac{R_s}{R_b}\right)}$$

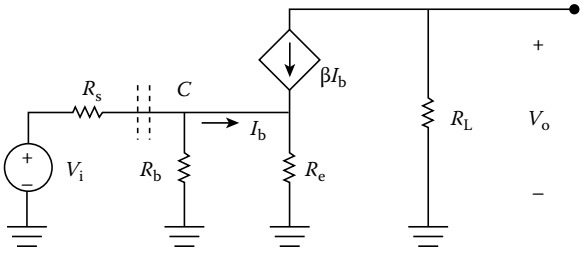


FIGURE 24.56 Network for Example 24.16.

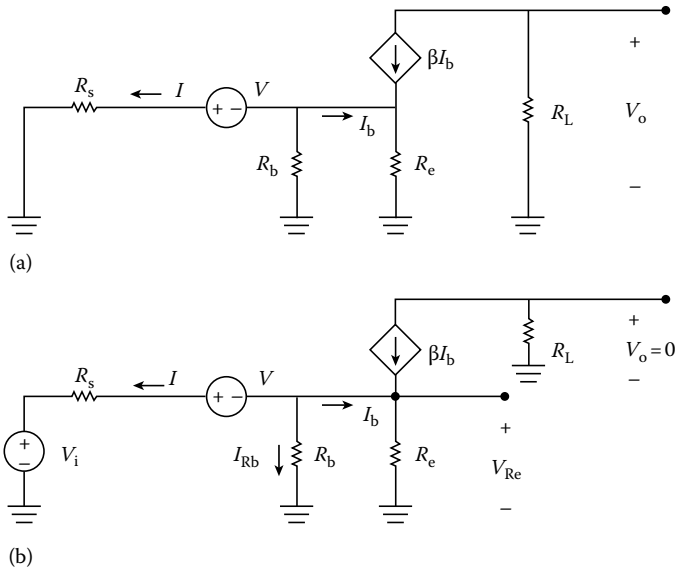


FIGURE 24.57 (a) Network used to obtain Z_d. (b) Network that yields Z_N.

Next, we find Z_d from Figure 24.57a. This yields

$$Z_d = \frac{V}{I} = R_s + [R_b \parallel (\beta + 1)R_e]$$

The impedance Z_N is found from Figure 24.57b where the two input sources, V_i and V, are adjusted so that V_o equals zero. If V_o equals zero, then βI_b equals zero because no current flows through R_L. Thus, I_b equals zero, which implies that V_{Re}, the voltage across R_e as indicated in Figure 24.57b, is also zero. We see that the null is propagating through the circuit. Continuing to analyze Figure 24.57b, we see that I_{Rb} is zero so that we conclude that I is zero. Because $Z_N = V/I$, we conclude that Z_N is infinite. Using the format given by Equation 24.69 with $Z = 1/(sC)$, we obtain the result as

$$\frac{V_o}{V_i} = A_1 \left\{ \frac{1}{1 + \frac{1/(sC)}{R_s + [R_b \parallel (\beta+1)R_e]}} \right\}$$

It is interesting to note that to null the output so that Z_N could be found in Example 24.16, V_i is set to V, although this fact is not needed in the analysis.

24.2.10 Substitution Theorem

Figure 24.58a is an LLFT network consisting of two subnetworks A and B, which are connected to each other by two wires. If the voltage $v(t)$ is known, the voltages and currents in subnetwork A remain unchanged if a voltage source of value $v(t)$ is substituted for subnetwork B as illustrated in Figure 24.58b.

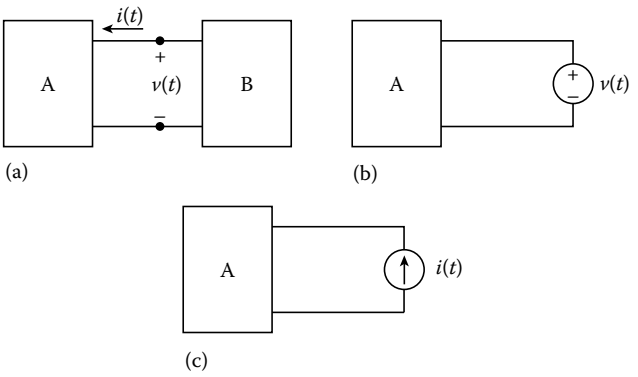


FIGURE 24.58 (a) An LLFT network consisting of two subnetworks A and B connected by two wires. (b) A voltage source can be substituted for subnetwork B if $v(t)$ is known in (a). (c) A current source can be substituted for B if i is a known current.

Example 24.17

Determine $i_1(t)$ in the circuit in Figure 24.59. The voltage $v_1(t)$ is known from a previous analysis.

Solution

Because $v_1(t)$ is known, the substitution theorem can be applied to obtain the circuit in Figure 24.60. Analysis of this simplified circuit yields

$$i_1 = i_{\text{in}} \frac{R_3}{R_2 + R_3} + v_1 \frac{1}{R_2 + R_3}$$

If the current $i(t)$ is known in Figure 24.58a, then the substitution in Figure 24.58c can be employed.

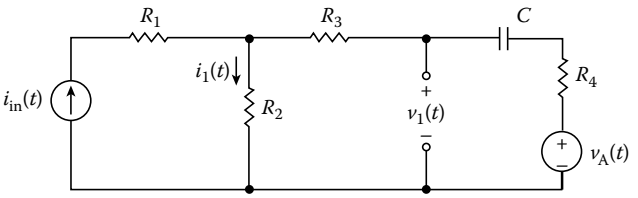


FIGURE 24.59 Circuit for Example 24.17.

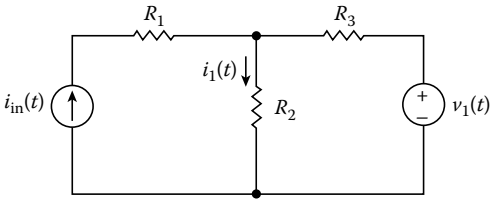


FIGURE 24.60 Circuit that results when the substitution theorem is applied to the circuit in Figure 24.59.

24.3 Sinusoidal Steady-State Analysis and Phasor Transforms

24.3.1 Sinusoidal Steady-State Analysis

In this section, we develop techniques for analyzing LLFT networks in the sinusoidal steady state. These techniques are important for analyzing and designing networks ranging from AC power generation systems to electronic filters.

To put the development of sinusoidal steady-state analysis in its context, we list the following definitions of responses of circuits:

1. Zero-input response is the response of a circuit to its initial conditions when the input excitations are set to zero.
2. Zero-state response is the response of a circuit to a given input excitation or set of input excitations when the initial conditions are all set to zero.

   The sum of the zero-input response and the zero-state response yields the total response of the system being analyzed. However, the total response can also be decomposed into the forced response and the natural response if the input excitations are DC, real exponentials, sinusoids, and/or sinusoids multiplied by real exponentials and if the exponent(s) in the input excitation differs from the exponents appearing in the zero-input response. These excitations are very common in engineering applications, and the decomposition of the response into forced and natural components corresponds to the particular and complementary (homogeneous) solutions, respectively, of the linear, constant coefficient, ordinary differential equations that characterize LLFT networks in the time domain. Therefore, we define
3. Forced response is the portion of the total response that has the same exponents as the input excitations.
4. Natural response is the portion of the total response that has the same exponents as the zero-input response.

   The sum of the forced and natural responses is the total response of the system.

   For a strictly stable LLFT network, meaning that the poles of the system transfer function $T(s)$ are confined to the open LHP, the natural response must decay to zero eventually. The forced response may or may not decay to zero depending on the excitation and the network to which it is applied, and so it is convenient to define the terms steady-state response and transient response:
5. Transient response is the portion of the response that dies away or decays to zero with time.
6. Steady-state response is the portion of the response that does not decay with time.

The sum of the transient and steady-state responses is the total response, but a specific circuit with a specific excitation may not have a transient response or it may not have a steady-state response. The following example illustrates aspects of these six definitions.

Example 24.18

Find the total response of the network shown in Figure 24.61, and identify the zero-state, zero-input, forced, natural, transient, and steady-state portions of the response.

Solution

Note that a nonzero initial condition is represented by V_c. Using superposition and the simple voltage divider concept, we can write

$$V_o(s) = \left[\frac{V_1}{s+1} + \frac{V_2}{s^2+1}\right]\frac{2}{s+2} + \frac{V_c}{s}\left(\frac{s}{s+2}\right)$$

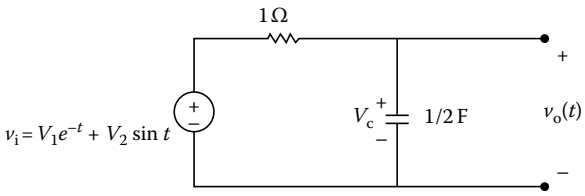


FIGURE 24.61 Circuit for Example 24.18.

The partial fraction expansion for $V_o(s)$ is

$$V_o(s) = \frac{A}{s+1} + \frac{B}{s+2} + \frac{Cs+D}{s^2+1}$$

where

$$\begin{aligned} A &= 2V_1 \\ B &= -2V_1 + 0.4V_2 + V_c \\ C &= -0.4V_2 \\ D &= 0.8V_2 \end{aligned}$$

Thus, for $t \geq 0$, $v_o(t)$ can be written as

$$v_o(t) = 2V_1e^{-t} - 2V_1e^{-2t} + 0.4V_2e^{-2t} + V_ce^{-2t} - 0.4V_2 \cos t + 0.8V_2 \sin t$$

With the aid of the angle sum and difference formula

$$\sin(\alpha \pm \beta) = \sin\alpha \ \cos\beta \ \pm \ \cos\alpha \ \sin\beta$$

and the sketch in Figure 24.62, we can combine the last two terms in the expression for $v_o(t)$ to obtain

$$v_o(t) = 2V_1e^{-t} - 2V_1e^{-2t} + 0.4V_2e^{-2t} + V_ce^{-2t} + 0.4\sqrt{5}V_2 \sin(t + \theta)$$

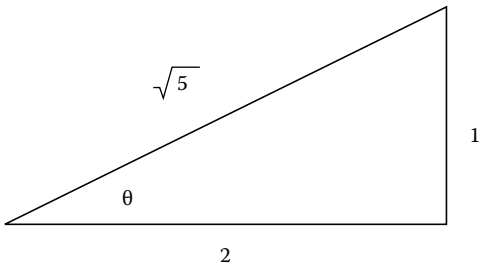


FIGURE 24.62 Sketch for determining the phase angle θ.

where

$$\theta = -\tan^{-1}\left(\frac{1}{2}\right)$$

The terms of $v_o(t)$ are characterized by our definitions as follows:

$$\begin{aligned}
\text{Zero-input response} &= V_c e^{-2t} \\
\text{Zero-state response} &= 2V_1 e^{-t} - 2V_1 e^{-2t} + 0.4V_2 e^{-2t} + 0.4\sqrt{5}V_2 \sin(t+\theta) \\
\text{Natural response} &= [-2V_1 + 0.4V_2 + V_c]e^{-2t} \\
\text{Forced response} &= 2V_1 e^{-t} + 0.4\sqrt{5}V_2 \sin(t+\theta) \\
\text{Transient response} &= 2V_1 e^{-t} + [-2V_1 + 0.4V_2 + V_c]e^{-2t} \\
\text{Steady-state response} &= 0.4\sqrt{5}V_2 \sin(t+\theta)
\end{aligned}$$

As can be observed by comparing the preceding terms above with the total response, part of the forced response is the steady-state response, and the rest of the forced response is included in the transient response in this example.

If the generator voltage in the previous example had been $v_i = V_1 + V_2 \sin(t)$, then there would have been two terms in the steady-state response—a DC term and a sinusoidal term. On the other hand, if the transfer function from input to output had a pole at the origin and the excitation were purely sinusoidal, there would also have been a DC term and a sinusoidal term in the steady-state response. The DC term would have arisen from the pole at the origin in the transfer function, and therefore would also be classed as a term in the natural response.

Oftentimes, it is desirable to obtain only the sinusoidal steady-state response, without having to solve for other portions of the total response. The ability to solve for just the sinusoidal steady-state response is the goal of sinusoidal steady-state analysis.

The sinusoidal steady-state response can be obtained based on analysis of the network using Laplace transforms. Figure 24.63 illustrates an LLFT network that is excited by the voltage sine wave $v_i(t) = V \sin(\omega t)$, where V is the peak amplitude of the sine wave and ω is the frequency of the sine wave in radians/second. Assume that the poles of the network transfer function $V_o(s)/V_i(s) = T(s)$ are confined to the open LHP except possibly for a single pole at the origin. Then, the forced response of the network is

$$v_{oss}(t) = V|T(j\omega)| \sin(\omega t + \theta) \tag{24.70}$$

where the extra subscripts "ss" on $v_o(t)$ indicate sinusoidal steady state, and where

$$\theta = \tan^{-1}\left(\frac{\Im T(j\omega)}{\Re T(j\omega)}\right) \tag{24.71}$$

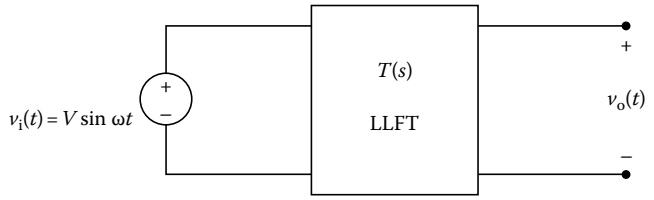


FIGURE 24.63 LLFT network with transfer function $T(s)$.

The symbols $\mathcal{I}$ and $\mathcal{R}$ are read as "imaginary part of" and "real part of," respectively. In other words, the LLFT network modifies the sinusoidal input signal in only two ways at steady state. The network multiplies the amplitude of the signal by $|T(j\omega)|$ and shifts the phase by θ. If the transfer function of the network is known beforehand, then the sinusoidal steady-state portion of the total response can be easily obtained by means of Equations 24.70 and 24.71.

To prove Equations 24.70 and 24.71, we assume that $T(s) = V_o(s)/V_i(s)$ in Figure 24.63 is real for s real and that the poles of $T(s)$ are confined to the open LHP except possibly for a single pole at the origin. Without loss of generality, assume the order of the numerator of $T(s)$ is at most one greater than the order of the denominator. Then the transform of the output voltage is

$$V_o(s) = V_i(s)T(s) = \frac{V\omega}{s^2 + \omega^2} T(s)$$

If $V_o(s)$ is expanded into partial fractions, we have

$$V_o(s) = \frac{A}{s - j\omega} + \frac{B}{s + j\omega} + \text{other terms due to the poles of } T(s)$$

The residue A is

$$\begin{aligned} A &= [(s - j\omega)V_o(s)]_{s=j\omega} = \left[\frac{V\omega}{s + j\omega} T(s)\right]_{s=j\omega} \\ &= \frac{V}{2j} T(j\omega) \end{aligned}$$

But

$$T(j\omega) = |T(j\omega)|e^{j\theta} \quad \text{where } \theta = \tan^{-1} \frac{\mathcal{I}T(j\omega)}{\mathcal{R}T(j\omega)}$$

Thus, we can write the residue A as

$$A = \frac{V}{2j} |T(j\omega)|e^{j\theta}$$

Also, $B = A^*$ where "*" denotes "conjugate," and so

$$\begin{aligned} B &= -\frac{V}{2j} T(-j\omega) = -\frac{V}{2j} T((j\omega)^*) \\ &= -\frac{V}{2j} T^*(j\omega) = -\frac{V}{2j} |T(j\omega)|e^{-j\theta} \end{aligned}$$

In the equation for the residue B, we can write $T((j\omega)^*) = T^*(j\omega)$ because of the assumption that $T(s)$ is real for s real (see property 1 in Section 24.1.1).

All other terms in the partial fraction of $V_o(s)$ will yield, when inverse transformed, functions of time that decay to zero except for a term arising from a pole at the origin of $T(s)$. A pole at the origin yields, when its partial fraction is inverse transformed, a DC term that is part of the steady-state solution in the time domain. However, only the first two terms in $V_o(s)$ will ultimately yield a sinusoidal function. We can rewrite these two terms as

$$V_{oss}(s) = \frac{V}{2j} |T(j\omega)| \left[\frac{e^{j\theta}}{s - j\omega} - \frac{e^{-j\theta}}{s + j\omega}\right]$$

The extra subscripts "*ss*" denote sinusoidal steady state. The time-domain equation for the sinusoidal steady-state output voltage is

$$\begin{aligned} v_{oss}(t) &= \frac{V}{2j}|T(j\omega)|\left[e^{j\theta}e^{j\omega t} - e^{-j\theta}e^{-j\omega t}\right] \\ &= V|T(j\omega)|\sin(\omega t + \theta) \end{aligned}$$

where θ is given by Equation 24.71. This completes the proof.

Example 24.19

Verify the expression for the sinusoidal steady-state response found in the previous example.

Solution

The transfer function for the network in Figure 24.61 is $T(s) = 2/(s+2)$, and the frequency of the sinusoidal portion of $v_i(t)$ is $\omega = 1$ rad/s. Thus,

$$T(j\omega) = \frac{2}{j+2} = \frac{2}{\sqrt{4+1}}e^{j\theta}$$

where

$$\theta = \tan^{-1}\left(\frac{-2}{4}\right) = -\tan^{-1}\left(\frac{1}{2}\right)$$

If the excitation in Figure 24.63 were $v_i(t) = V\sin(\omega t + \Phi)$, then the sinusoidal steady-state response of the network would be

$$v_{oss}(t) = V|T(j\omega)|\sin(\omega t + \Phi + \theta) \tag{24.72}$$

where θ is given by Equation 24.71. Similarly, if the excitation were $v_i(t) = V[\cos(\omega t + \Phi)]$, then the sinusoidal steady-state response would be expressed as

$$v_{oss}(t) = V|T(j\omega)|\cos(\omega t + \Phi + \theta) \tag{24.73}$$

with θ again given by Equation 24.71.

24.3.2 Phasor Transforms

In the sinusoidal steady-state analysis of stable LLFT networks, we find that both the inputs and outputs are sine waves of the same frequency. The network only modifies the amplitudes and the phases of the sinusoidal input signals; it does not change their nature. Thus, we need only keep track of the amplitudes and phases, and we do this by using phasor transforms. Phasor transforms are closely linked to Euler's identity:

$$e^{\pm j\omega t} = \cos(\omega t) \pm j\sin(\omega t) \tag{24.74}$$

If, for example, $v_i(t) = V \sin(\omega t + \Phi)$, then we can write $v_i(t)$ as

$$v_i(t) = \mathcal{I}[Ve^{j(\omega t+\Phi)}] = \mathcal{I}[Ve^{j\Phi}e^{j\omega t}] \tag{24.75}$$

Similarly, if $v_i(t) = V \cos(\omega t + \Phi)$, then we can write

$$v_i(t) = \mathcal{R}[Ve^{j(\omega t+\Phi)}] = \mathcal{R}[Ve^{j\Phi}e^{j\omega t}] \tag{24.76}$$

If we confine our analysis to single-frequency sine waves, then we can drop the imaginary sign and the term $e^{j\omega t}$ in Equation 24.75 to obtain the phasor transform. That is,

$$\wp[v_i(t)] = \wp[V \sin(\omega t + \Phi)] = \wp\{\mathcal{I}[Ve^{j\Phi}e^{j\omega t}]\} = Ve^{j\Phi} \tag{24.77}$$

The first and last terms in Equation 24.77 are read as "the phasor transform of $v_i(t)$ equals $Ve^{j\Phi}$." Note that $v_i(t)$ is not equal to $Ve^{j\Phi}$ as can be seen from the fact that $v_i(t)$ is a function of time while $Ve^{j\Phi}$ is not. Phasor transforms will be denoted with bold letters that are underlined as in $\wp[v_i(t)] = \underline{\boldsymbol{V}}_i$.

If our analysis is confined to single-frequency cosine waves, we perform the phasor transform in the following manner:

$$\wp[V \cos(\omega t + \Phi)] = \wp\{\mathcal{R}[Ve^{j\Phi}e^{j\omega t}]\} = Ve^{j\Phi} = \underline{V} \tag{24.78}$$

In other words, to perform the phasor transform of a cosine function, we drop both the real sign and the term $e^{j\omega t}$. Both sines and cosines are sinusoidal functions, but when we transform them, they lose their identities. Thus, before starting an analysis, we must decide whether to perform the analysis all in sines or all in cosines. The two functions must not be mixed when using phasor transforms. Furthermore, we cannot simultaneously employ the phasor transforms of sinusoids at two different frequencies. However, if a linear network has two excitations which have different frequencies, we can use superposition in an analysis for a voltage or current, and add the solutions in the time domain.

Three equivalent representations are used for a phasor $\underline{\boldsymbol{V}}$:

$$\underline{\boldsymbol{V}} = \begin{cases} Ve^{j\Phi} & \text{exponential form} \\ V(\cos\Phi + j\sin\Phi) & \text{rectangular form} \\ V\angle\Phi & \text{polar form} \end{cases} \tag{24.79}$$

If phasors are to be multiplied or divided by a complex number, the exponential or polar forms are the most convenient. If phasors are to be added or subtracted, the rectangular form is the most convenient. The relationships among the equivalent representations are illustrated in Figure 24.64. In this figure, the

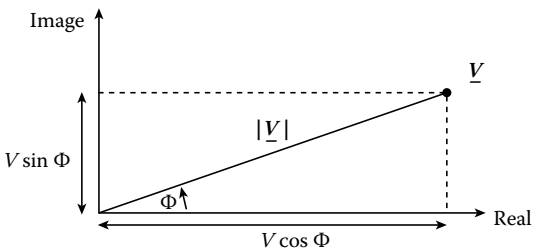


FIGURE 24.64 Relationships among phasor representations.

phasor $\underline{\boldsymbol{V}}$ is denoted by a point in the complex plane. The magnitude of the phasor, $|\underline{\boldsymbol{V}}| = V$, is illustrated by the length of the line drawn from the origin to the point. The phase of the phasor, Φ, is shown measured counterclockwise from the horizontal axis. The real part of $\underline{\boldsymbol{V}}$ is $V \cos \Phi$ and the imaginary part of $\underline{\boldsymbol{V}}$ is $V \sin \Phi$.

Phasors can be developed in a way that parallels, to some extent, the usual development of Laplace transforms. In the following theorems, we assume that the constants V_1, V_2, Φ_1, and Φ_2 are real.

THEOREM 24.1

For sinusoids of the same type (*either sines or cosines*) *and of the same frequency* ω, $\wp[V_1 \sin(\omega t + \Phi_1) + V_2 \sin(\omega t + \Phi_2)] = V_1 \wp[\sin(\omega t + \Phi_1)] + V_2 \wp[\sin(\omega t + \Phi_2)]$. *A similar relation can be written for cosines.*

This theorem demonstrates that the phasor transform is a linear transform.

THEOREM 24.2

If $\wp[V_1 \sin(\omega t + \Phi)] = V_1 e^{j\Phi}$, *then*

$$\wp\left[\frac{d}{dt} V_1 \sin(\omega t + \Phi)\right] = j\omega V_1 e^{j\Phi} \tag{24.80}$$

To prove Theorem 24.2, we can write

$$\begin{aligned}\wp\left[\frac{d}{dt} V_1 \mathfrak{I}(e^{j\Phi} e^{j\omega t})\right] &= \wp\left[V_1 \mathfrak{I}\left(e^{j\Phi} \frac{d}{dt} e^{j\omega t}\right)\right] \\ &= \wp\left[V_1 \mathfrak{I} e^{j\Phi} j\omega e^{j\omega t}\right] = V_1 j\omega e^{j\Phi}\end{aligned}$$

Note the interchange of the derivative and the imaginary sign in the proof of the theorem. Also, Theorem 24.2 can be generalized to

$$\wp\left[\frac{d^n}{dt^n} V_1 \sin(\omega t + \Phi)\right] = (j\omega)^n V_1 e^{j\Phi} \tag{24.81}$$

These results are useful for finding the sinusoidal steady-state solutions of linear, constant-coefficient, ordinary differential equations assuming the roots of the characteristic polynomials lie in the open LHP with possibly one at the origin.

THEOREM 24.3

If $\wp[V_1 \sin(\omega t + \Phi)] = V_1 e^{j\Phi}$, *then*

$$\wp\left[\int V_1 \sin(\omega t + \Phi) dt\right] = \frac{1}{j\omega} V_1 e^{j\Phi} \tag{24.82}$$

The proof of Theorem 24.3 is easily obtained by writing

$$\begin{aligned}\wp\left[\int V_1 \sin(\omega t + \Phi)\mathrm{d}t\right] &= \wp\left[\int \Im\left[V_1 e^{j(\omega t+\Phi)}\right]\mathrm{d}t\right] \\ &= \wp\left[\Im \int V_1 e^{j(\omega t+\Phi)}\mathrm{d}t\right] = \frac{V_1}{j\omega}e^{j\Phi}\end{aligned}$$

It should be noted that no constant of integration is employed because a constant is not a sinusoidal function and is therefore not permitted when using phasors. A constant of integration arises in LLFT network analysis because of initial conditions, and we are interested only in the sinusoidal steady-state response and not in a zero-input response. No limits are used with the integral either, because the (constant) lower limit would also yield a constant, which would imply that we are not at sinusoidal steady state.

Theorem 24.3 is easily extended to the case of n integrals:

$$\wp\left[\int \ldots \int V_1 \sin(\omega t + \Phi)(\mathrm{d}t)^n\right] = \frac{V_1}{(j\omega)^n}e^{j\Phi} \tag{24.83}$$

This result is useful for finding the sinusoidal steady-state solution of integro-differential equations.

24.3.3 Inverse Phasor Transforms

To obtain time domain results, we must be able to inverse transform phasors. The inverse transform operation is denoted by $\wp^{-1}$. This is an easy operation that consists of restoring the term $e^{j\omega t}$, restoring the imaginary sign (the real sign if cosines are used), and dropping the inverse transform sign. That is,

$$\begin{aligned}\wp^{-1}[V_1 e^{j\Phi}] &= \Im\left[V_1 e^{j\Phi}e^{j\omega t}\right] \\ &= V_1 \sin(\omega t + \Phi)\end{aligned} \tag{24.84}$$

The following example illustrates both the use of Theorem 24.2 and the inverse transform procedure.

Example 24.20

Determine the sinusoidal steady-state solution for the differential equation:

$$\frac{d^2 f(t)}{\mathrm{d}t^2} + 4\frac{df(t)}{\mathrm{d}t} + 3f(t) = V\sin(\omega t + \Phi)$$

Solution

We note that the characteristic polynomial, $D^2 + 4D + 3$, has all its roots in the open LHP. The next step is to phasor transform each term of the equation to obtain

$$-\omega^2 \underline{\boldsymbol{F}} + 4j\omega\underline{\boldsymbol{F}} + 3\underline{\boldsymbol{F}} = Ve^{j\Phi}$$

where $\underline{\boldsymbol{F}}(j\omega) = \wp[f(t)]$. Therefore, when we solve for $\underline{\boldsymbol{F}}$, we obtain

$$\begin{aligned}\underline{\boldsymbol{F}} &= \frac{Ve^{j\Phi}}{(3-\omega^2)+j4\omega} \\ &= \frac{Ve^{j\Phi}e^{j\theta}}{\sqrt{(3-\omega^2)^2+16\omega^2}}\end{aligned}$$

where

$$\theta = \tan^{-1}\frac{-4\omega}{3-\omega^2} = \tan^{-1}\frac{4\omega}{\omega^2-3}$$

Thus,

$$\underline{\boldsymbol{F}} = \frac{V}{\sqrt{\omega^4+10\omega^2+9}}e^{j(\phi+\theta)}$$

To obtain a time-domain function, we inverse transform $\underline{\boldsymbol{F}}$ to obtain

$$\mathcal{F}^{-1}[\underline{\boldsymbol{F}}(j\omega)] = f(t) = \frac{V}{\sqrt{\omega^4+10\omega^2+9}}\sin(\omega t+\Phi+\theta)$$

In this example, we see that the sinusoidal steady-state solution consists of the sinusoidal forcing term, $V\sin(\omega t+\Phi)$, modified in amplitude and shifted in phase.

24.3.4 Phasors and Networks

Phasors are time-independent representations of sinusoids. Thus, we can define impedances in the phasor transform domain and obtain Ohm's law-like expressions relating currents through network elements with the voltages across those elements. In addition, the impedance concept allows us to combine dissimilar elements, such as resistors with inductors, in the transform domain.

The time-domain expressions relating the voltages and currents for *Rs*, *Ls*, and *Cs*, repeated here for convenience, are

$$v_R(t) = i_R(t)R, \quad v_L(t) = \frac{L\mathrm{d}i_L}{\mathrm{d}t}, \quad v_C(t) = \frac{1}{C}\int i_C\mathrm{d}t$$

Note that initial conditions are set to zero. Then, performing the phasor transform of the time-domain variables, we have

$$Z_R = R, \quad Z_L = j\omega L, \quad Z_C = \frac{1}{j\omega C}$$

We can also write the admittances of these elements as $Y_R = 1/Z_R$, $Y_L = 1/Z_L$, and $Y_C = 1/Z_C$. Then, we can extend the impedance and admittance concepts for two-terminal elements to multiport networks in the same manner as was done in the development of Laplace transform techniques for network analysis. For example, the transfer function of the circuit shown in Figure 24.65 can be written as

$$\frac{\underline{\boldsymbol{V}}_o(j\omega)}{\underline{\boldsymbol{V}}_i(j\omega)} = G_{21}(j\omega)$$

where the $j\omega$ indicates that the analysis is being performed at sinusoidal steady state [1]. It is also assumed that no other excitations exist in *N* in Figure 24.65. With impedances and transfer functions defined, then all the theorems developed for Laplace transform analysis, including source transformations, have a phasor transform counterpart.

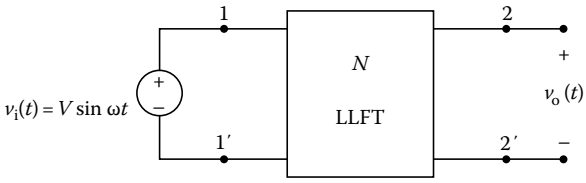


FIGURE 24.65 LLFT network excited by a sinusoidal voltage source.

Example 24.21

Use phasor analysis to find the transfer function $G_{21}(j\omega)$ and $v_{oss}(t)$ for the circuit in Figure 24.66.

Solution

The phasor transform of the output voltage can be obtained easily by means of the simple voltage divider. Thus,

$$\underline{\boldsymbol{V}}_o = \underline{\boldsymbol{V}}_i \frac{j\omega L + R_2}{j\omega L + R_2 + \frac{R_1}{1+j\omega CR_1}}$$

To obtain $G_{21}(j\omega)$, we form $\underline{\boldsymbol{V}}_o/\underline{\boldsymbol{V}}_i$, which yields

$$\frac{\underline{\boldsymbol{V}}_o}{\underline{\boldsymbol{V}}_i} = G_{21}(j\omega) = \frac{(R_2 + j\omega L)(1 + j\omega CR_1)}{(R_2 + j\omega L)(1 + j\omega CR_1) + R_1}$$

Expressing the numerator and denominator of G_{21} in exponential form produces

$$G_{21} = \frac{\sqrt{(R_2 - \omega^2 LCR_1)^2 + (\omega L + \omega CR_1R_2)^2}\, e^{j\alpha}}{\sqrt{(R_1 + R_2 - \omega^2 LCR_1)^2 + (\omega L + \omega CR_1R_2)^2}\, e^{j\beta}}$$

where

$$\alpha = \tan^{-1} \frac{(\omega L + \omega CR_1R_2)}{R_2 - \omega^2 LCR_1}$$
$$\beta = \tan^{-1} \frac{(\omega L + \omega CR_1R_2)}{R_1 + R_2 - \omega^2 LCR_1}$$

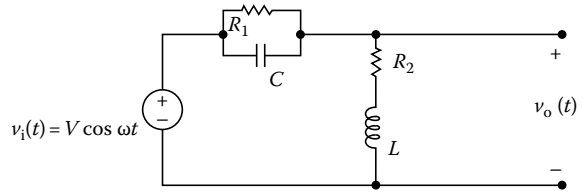


FIGURE 24.66 Circuit for Example 24.21.

Thus,

$$G_{21}(j\omega) = Me^{j\theta}$$

where

$$M = \sqrt{\frac{(R_2 - \omega^2 LCR_1)^2 + (\omega L + \omega CR_1R_2)^2}{(R_1 + R_2 - \omega^2 LCR_1)^2 + (\omega L + \omega CR_1R_2)^2}}$$

and

$$\theta = \alpha - \beta$$

The phasor transform of $v_i(t)$ is

$$\underline{\boldsymbol{V}}_i = \wp[V\Re e^{j\omega t}] = Ve^{j\theta} = V$$

and, therefore, the time-domain expression for the sinusoidal steady-state output voltage is

$$v_{oss}(t) = VM\cos(\omega t + \theta)$$

Driving point impedances and admittances as well as transfer functions are not phasors because they do not represent sinusoidal waveforms. However, an impedance or transfer function is a complex number at a particular real frequency, and the product of a complex number times a phasor is a new phasor.

The product of two arbitrary phasors is not ordinarily defined because $\sin^2(\omega t)$ or $\cos^2(\omega t)$ are not sinusoidal and have no phasor transforms. However, as we will see later, power relations for AC circuits can be expressed in efficient ways as functions of products of phasors. Because such products have physical interpretations, we permit them in the context of power calculations.

Division of one phasor by another is permitted only if the two phasors are related by a driving point or transfer network function such as $\underline{\boldsymbol{V}}_o/\underline{\boldsymbol{V}}_i = G_{21}(j\omega)$.

24.3.5 Phase Lead and Phase Lag

The terms ''phase lead'' and ''phase lag'' are used to describe the phase shift between two or more sinusoids of the same frequency. This phase shift can be expressed as an angle in degrees or radians, or it can be expressed in time as seconds. For example, suppose we have three sinusoids given by

$$v_1(t) = V_1\sin(\omega t), \quad v_2(t) = V_2\sin(\omega t + \Phi), \quad v_3 = V_3\sin(\omega t - \Phi)$$

where V_1, V_2, V_3, and Φ are all positive. Then, we say that v_2 leads v_1 and that v_3 lags v_1. To see this more clearly, we rewrite v_2 and v_3 as

$$v_2 = V_2\sin[\omega(t + t_0)] \quad v_3 = V_3\sin[\omega(t - t_0)]$$

where the constant $t_0 = \Phi/\omega$. Figure 24.67 plots the three sinusoids sketched on the same axis, and from this graph we see that the zero crossings of $v_2(t)$ occur t_0 seconds before the zero crossing of $v_1(t)$. Thus, $v_2(t)$ leads $v_1(t)$ by t_0 seconds. Similarly, we see that the zero crossings of $v_3(t)$ occur t_0 seconds after the zero crossings of $v_1(t)$. Thus, $v_3(t)$ lags $v_1(t)$. We can also say that $v_3(t)$ lags $v_2(t)$. When comparing the phases of sine waves with $V\sin(\omega t)$, the key thing to look for in the arguments of the sines are the signs of the angles following ωt. A positive sign means lead and a negative sign means lag. If two sines or two cosines have the same phase angle, then they are called ''in phase.''

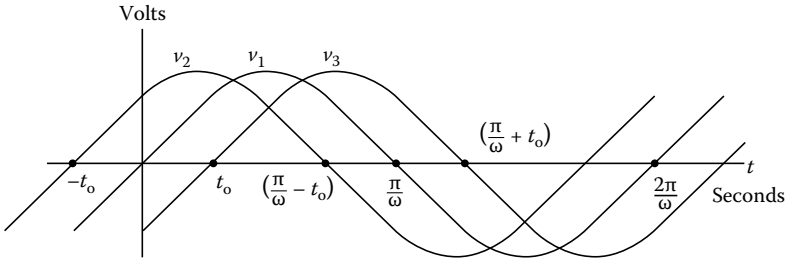


FIGURE 24.67 Three sinusoids sketched on a time axis.

If we have $i_1(t) = I_1\ [\cos(\omega t - \pi/4)]$ and $i_2(t) = I_2\ [\cos(\omega t - \pi/3)]$, then i_2 lags i_1 by $\pi/12$ rad or 15° because even though the phases of both cosines are negative, the phase of $i_1(t)$ is less negative than the phase of $i_2(t)$. We can also say that i_1 leads i_2 by 15°.

Example 24.22

Suppose we have five signals with equal peak amplitudes and equal frequencies but with differing phases. The signals are: $i_1 = I[\sin\ (\omega t)]$, $i_2 = I\ [\cos(\omega t)]$, $i_3 = I[\cos(\omega t + \theta)]$, $i_4 = -I[\sin(\omega t + \psi)]$, and $i_5 = -I[\cos(\omega t - \Phi)]$. Assume I, θ, ψ, and Φ are positive.

1. How much do the signals i_2 through i_5 lead i_1?
2. How much do the signals i_1 and i_3 through i_5 lead i_2?

Solution

For part (1), we express i_2 through i_5 as sines with lead angles. That is,

$$
\begin{aligned}
i_2 &= I\cos(\omega t) = I\sin\left(\omega t + \frac{\pi}{2}\right)\\
i_3 &= I\cos(\omega t + \theta) = I\sin\left(\omega t + \theta + \frac{\pi}{2}\right)\\
i_4 &= -I\sin(\omega t + \psi) = I\sin(\omega t + \psi \pm \pi)\\
i_5 &= -I\cos(\omega t - \Phi) = I\cos(\omega t - \Phi \pm \pi)\\
&= I\sin\left(\omega t - \Phi \pm \pi + \frac{\pi}{2}\right)
\end{aligned}
$$

Thus, i_2 leads i_1 by $\pi/2$ rad, and i_3 leads i_1 by $\theta + \pi/2$. For i_4, we can take the plus sign in the argument of the sign to obtain $\psi + \pi$, or we can take the minus sign to obtain $\psi - \pi$. The current i_5 leads i_1 by $(3\pi/2 - \Phi)$ or by $(-\pi/2 - \Phi)$. An angle of $\pm 2\pi$ can be added to the argument without affecting lead or lag relationships.

For part (2), we express i_1 and i_3 through i_5 as cosines with lead angles yielding

$$
\begin{aligned}
i_1 &= I\sin(\omega t) = I\cos\left(\omega t - \frac{\pi}{2}\right)\\
i_3 &= I\cos(\omega t + \theta)\\
i_4 &= -I\sin(\omega t + \psi) = I\sin(\omega t + \psi \pm \pi)\\
&= I\cos\left(\omega t + \psi \pm \pi - \frac{\pi}{2}\right)\\
i_5 &= -I\cos(\omega t - \Phi) = I\cos(\omega t - \Phi \pm \pi)
\end{aligned}
$$

We conclude that i_1 leads i_2 by $(-\pi/2)$ rad. (We could also say that i_1 lags i_2 by $(\pi/2)$ rad.) Also, i_3 leads i_2 by θ. The current i_4 leads i_2 by $(\psi + \pi/2)$ where we have chosen the plus sign in the argument of the cosine. Finally, i_5 leads i_2 by $(\pi - \Phi)$, where we have chosen the plus sign in the argument.

In the previous example, we have made use of the identities:

$$\cos(\alpha) = \sin\left(\alpha + \frac{\pi}{2}\right); \quad -\sin(\alpha) = \sin(\alpha \pm \pi)$$
$$-\cos(\alpha) = \cos(\alpha \pm \pi); \quad \sin(\alpha) = \cos\left(\alpha - \frac{\pi}{2}\right)$$

The concepts of phase lead and phase lag are clearly illustrated by means of phasor diagrams, which are described in Section 23.4.6.

24.3.6 Phasor Diagrams

Phasors are complex numbers that represent sinusoids, so phasors can be depicted graphically on a complex plane. Such graphical illustrations are called phasor diagrams. Phasor diagrams are valuable because they present a clear picture of the relationships among the currents and voltages in a network. Furthermore, addition and subtraction of phasors can be performed graphically on a phasor diagram. The construction of phasor diagrams is demonstrated in the next example.

Example 24.23

For the network in Figure 24.68a, find $\underline{\boldsymbol{I}}_1$, $\underline{\boldsymbol{V}}_{R1}$, and $\underline{\boldsymbol{V}}_C$. For Figure 24.68b, find $\underline{\boldsymbol{I}}_2$, $\underline{\boldsymbol{V}}_{R2}$, and $\underline{\boldsymbol{V}}_L$. Construct phasor diagrams that illustrate the relations of the currents to the voltage excitation and the other voltages of the networks.

Solution

For Figure 24.68a, we have

$$\wp[v(t)] = V\angle 0^\circ \quad \text{and} \quad \wp[i_1(t)] = \underline{\boldsymbol{I}}_1 = \frac{V}{R_1 + \frac{1}{j\omega C}}$$

Rewriting $\underline{\boldsymbol{I}}_1$, we have

$$\underline{\boldsymbol{I}}_1 = \frac{Vj\omega C}{1 + j\omega CR_1} = \frac{Vj\omega C}{1 + j\omega CR_1}\left[\frac{1 - j\omega CR_1}{1 - j\omega CR_1}\right] = V\left[\frac{\omega^2 C^2 R_1 + j\omega C}{\omega^2 C^2 R_1^2 + 1}\right] = \frac{V\omega C}{\sqrt{\omega^2 C^2 R_1^2 + 1}} e^{j\theta_1}$$

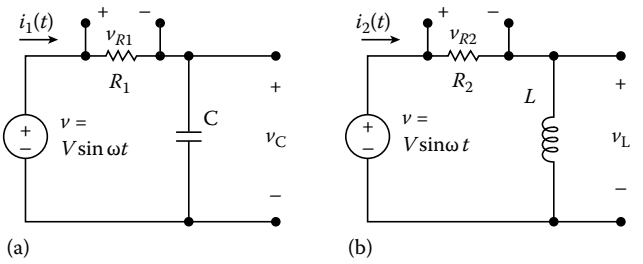


FIGURE 24.68 (a) An RC network. (b) An RL network.

where

$$\theta_1 = \tan^{-1} \frac{\omega C}{\omega^2 C^2 R_1} = \tan^{-1} \frac{1}{\omega C R_1}$$

Note that we have multiplied the numerator and denominator of $\underline{\boldsymbol{I}}_1$ by the conjugate of the denominator. The resulting denominator of $\underline{\boldsymbol{I}}_1$ is purely real, and so we need only consider the terms in the numerator of $\underline{\boldsymbol{I}}_1$ to obtain an expression for the phase. Thus, the resulting expression for the phase contains only one term which has the form:

$$\theta_1 = \tan^{-1} \frac{\mathcal{I}(\text{numerator})}{\mathcal{R}(\text{numerator})}$$

We could have obtained the same results without application of this artifice. In this case, we would have obtained

$$\theta_1 = \frac{\pi}{2} - \tan^{-1} \omega C R_1$$

For $\omega C R_1 \geq 0$, it is easy to show that the two expressions for θ_1 are equivalent.

Because the same current flows through both network elements, we have

$$\underline{\boldsymbol{V}_{R1}} = \frac{V \omega C R_1}{\sqrt{\omega^2 C^2 R_1^2 + 1}} e^{j\theta_1}$$

and

$$\underline{\boldsymbol{V}_C} = \underline{\boldsymbol{I}_1}\left(\frac{1}{j\omega C}\right) = \frac{-jV}{\sqrt{\omega^2 C^2 R_1^2 + 1}} e^{j\theta_1} = \frac{V}{\sqrt{\omega^2 C^2 R_1^2 + 1}} e^{j\psi}$$

where

$$\psi = -\frac{\pi}{2} + \theta_1 = -\tan^{-1} \omega C R_1$$

For $\underline{\boldsymbol{I}}_2$ in Figure 24.68b, we obtain

$$\underline{\boldsymbol{I}_2} = \frac{V \angle 0^\circ}{R_2 + j\omega L} = \frac{V}{\sqrt{R_2^2 + w^2 L^2}} e^{j\theta_2}$$

where θ_2 is given by

$$\theta_2 = -\tan^{-1} \frac{\omega L}{R_2}$$

The phasor current $\underline{\boldsymbol{I}}_2$ flows through both R_2 and L. So we have

$$\underline{\boldsymbol{V}_{R2}} = \underline{\boldsymbol{I}_2} R_2$$

and

$$\underline{\boldsymbol{V}}_{\mathrm{L}} = j\omega L\underline{\boldsymbol{I}}_2 = \frac{V\omega L}{\sqrt{\omega^2 L^2 + R_2^2}} e^{j\Phi}$$

where

$$\Phi = \frac{\pi}{2} + \theta_2$$

To construct the phasor diagram in Figure 24.69a for the RC network in Figure 24.68a, we first draw a vector corresponding to the phasor transform $\underline{\boldsymbol{V}} = V\angle 0°$ of the excitation. Because the phase of this phasor is zero, it is represented as a vector along the positive real axis. The length of this vector is $|\underline{\boldsymbol{V}}|$. Then we construct the vector representing $\underline{\boldsymbol{I}}_1 = |\underline{\boldsymbol{I}}_1|\ e^{j\theta 1}$. Again, the length of the vector is $|\underline{\boldsymbol{I}}_1|$, and it is drawn at the angle θ_1. The vector representing $\underline{\boldsymbol{V}}_{R1}$ lies along $\underline{\boldsymbol{I}}_1$ because the voltage across a resistor is always in phase or 180° out of phase with the current flowing through the resistor. The vector representing the current leads $\underline{\boldsymbol{V}}_{\mathrm{C}}$ by exactly 90°. It should be noted from the phasor diagram that $\underline{\boldsymbol{V}}_{R1}$ and $\underline{\boldsymbol{V}}_{\mathrm{C}}$ add to produce $\underline{\boldsymbol{V}}$ as required by Kirchhoff's law.

Figure 24.69b presents the phasor diagram for the RL network in Figure 24.68b. For this network, $\underline{\boldsymbol{I}}_2$ lags $\underline{\boldsymbol{V}}_{\mathrm{L}}$ by exactly 90°. Also, the vector sum of the voltages $\underline{\boldsymbol{V}}_{\mathrm{L}}$ and $\underline{\boldsymbol{V}}_{R2}$ must be the excitation voltage $\underline{\boldsymbol{V}}$ as indicated by the dotted lines in Figure 24.69b.

If the excitation $V\sin(\omega t)$ had been $V\sin(\omega t + \Phi)$ in Figure 24.68 in the previous example, then the vectors in the phasor diagrams in Figure 24.69 would have just been rotated around the origin by Φ. Thus, for example, $\underline{\boldsymbol{I}}_1$ in Figure 24.69a would have an angle equal to $\theta_1 + \Phi$. The lengths of the vectors and the relative phase shifts between the vectors would remain the same.

If R_1 in Figure 24.68a is decreased, then from the expression for $\theta_1 = \tan^{-1}(1/(\omega C R_1))$, we see that the phase of $\underline{\boldsymbol{I}}_1$ is increased. As R_1 is reduced further, θ_1 approaches 90°, and the circuit becomes more nearly like a pure capacitor. However, as long as $\underline{\boldsymbol{I}}_1$ leads $\underline{\boldsymbol{V}}$, we label the circuit as capacitive.

As R_2 in Figure 24.68b is decreased, then θ_2 in Figure 24.69b decreases (becomes more negative) and approaches −90°. Nevertheless, as long as $\underline{\boldsymbol{I}}_2$ lags $\underline{\boldsymbol{V}}$, we refer to the circuit as inductive.

If both inductors and capacitors are in a circuit, then it is possible for the circuit to appear capacitive at some frequencies and inductive at others. An example of such a circuit is provided in Section 24.3.7.

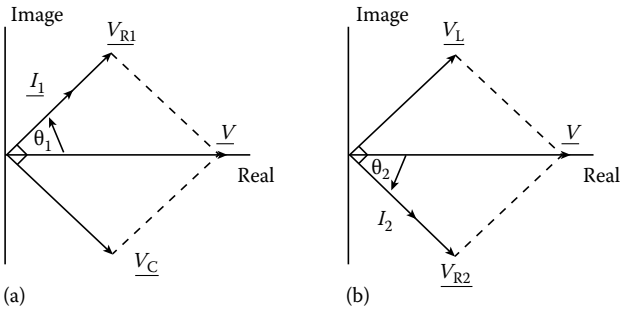


FIGURE 24.69 (a) Phasor diagram for the voltages and currents in Figure 24.68a. (b) Phasor diagram for Figure 24.68b.

24.3.7 Resonance

Resonant networks come in two basic types: the parallel resonant network and the series resonant (sometimes called antiresonant) network. More complicated networks may contain a variety of both types of resonant circuits. To see what happens at resonance, we examine a parallel resonant network at sinusoidal steady state [1]. Figure 24.70 is a network consisting of a capacitor and inductor connected in parallel, often called a tank circuit or tank, and an additional resistor R_1 connected in parallel with the tank. The phasor transforms of the excitation and the currents through the elements in Figure 24.70 are

$$\bar{\boldsymbol{V}} = V\angle 0^\circ; \quad \underline{\boldsymbol{I}_{R1}} = \frac{V}{R_1}; \quad \underline{\boldsymbol{I}_C} = j\omega CV; \quad \underline{\boldsymbol{I}_L} = \frac{V}{j\omega L} \tag{24.85}$$

where V is the peak value of the excitation. The transform of the current supplied by the source is

$$\underline{\boldsymbol{I}_1} = \boldsymbol{I}_1\angle\theta_1 = \underline{\boldsymbol{I}_{R1}} + \underline{\boldsymbol{I}_C} + \underline{\boldsymbol{I}_L} = V\left[\frac{1}{R_1} + j\omega C\left(1 - \frac{1}{\omega^2 LC}\right)\right] \tag{24.86}$$

The peak value of the current $i_1(t)$ at steady state is

$$I_1 = V\sqrt{\left(\frac{1}{R_1}\right)^2 + \omega^2 C^2\left(1 - \frac{1}{\omega^2 LC}\right)^2} \tag{24.87}$$

It is not difficult to determine that the minimum value of I_1 occurs at

$$\omega = \frac{1}{\sqrt{LC}} \tag{24.88}$$

which is the condition for resonance, and $I_{1\,\min}$ is given by

$$I_{1\,\min} = \frac{V}{R_1} \tag{24.89}$$

This result is somewhat surprising since it means that at resonance, the source in Figure 24.70 delivers no current to the tank at steady state. However, this result does not mean that the currents through the capacitor and inductor are zero. In fact, for $\omega^2 = 1/(LC)$ we have

$$\underline{\boldsymbol{I}_C} = jV\sqrt{\frac{C}{L}} \quad \text{and} \quad \underline{\boldsymbol{I}_L} = -jV\sqrt{\frac{C}{L}}$$

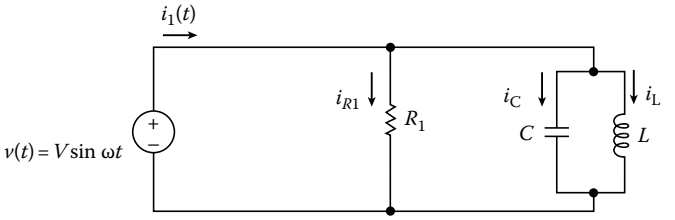


FIGURE 24.70 Parallel resonant circuit.

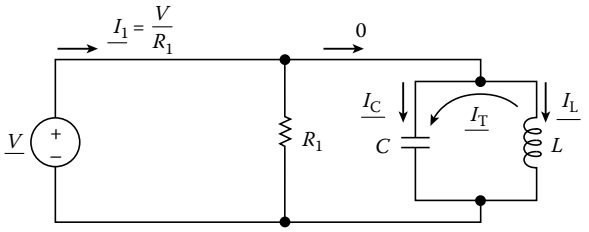


FIGURE 24.71 Circuit of Figure 24.70 at resonance. No current is supplied to the tank by the source, but a circulating current occurs in the tank.

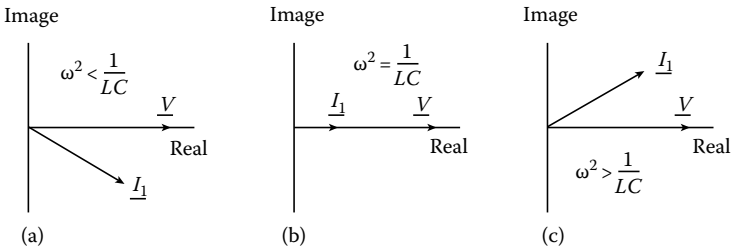


FIGURE 24.72 Phasor diagrams for the circuit in Figure 24.70. (a) $\omega^2 < 1/LC$. (b) Diagram at resonance. (c) $\omega^2 > 1/(LC)$.

That is, the current through the inductor is 180° out of phase with the current through the capacitor, and, because their magnitudes are equal, their sum is zero. Thus, at steady state and at the frequency given by Equation 24.88, the tank circuit looks like an open circuit to the voltage source. Yet, a circulating current occurs in the tank, labeled $\underline{\boldsymbol{I}}_T$ in Figure 24.71, which can be quite large depending on the values of C and L. That is, at resonance,

$$\underline{\boldsymbol{I}}_T = jV\sqrt{\frac{C}{L}} = \underline{\boldsymbol{I}}_C = -\underline{\boldsymbol{I}}_L \tag{24.90}$$

Therefore, energy is being transferred back and forth between the inductor and the capacitor. If the inductor and capacitor are ideal, the energy transferred would never decrease. In practice, parasitic resistances, especially in a physical inductor, would eventually dissipate this energy. Of course, parasitic resistances can be modeled as additional elements in the network.

Another interesting aspect of the network in Figure 24.70 is that, at low frequencies ($\omega^2 < 1/(LC)$), $\underline{\boldsymbol{I}}_1$ lags $\underline{\boldsymbol{V}}$, and so the network appears inductive to the voltage source. At high frequencies ($\omega^2 > 1/(LC)$), $\underline{\boldsymbol{I}}_1$ leads $\underline{\boldsymbol{V}}$, and the network looks capacitive to the voltage source. At resonance, the network appears as only a resistor R_1 to the source. Figure 24.72 depicts phasor diagrams of $\underline{\boldsymbol{V}}$ and $\underline{\boldsymbol{I}}_1$ at low frequency, at resonance, and at high frequency.

Figure 24.73 is the second basic type of resonant circuit—a series resonant circuit which is excited by a sinusoidal current source with phasor transform $\underline{\boldsymbol{I}} = I\angle 0°$. This circuit is dual to the circuit in Figure 24.70. The voltages across the network elements can be expressed as

$$\underline{\boldsymbol{V}}_R = IR; \quad \underline{\boldsymbol{V}}_C = -j\left(\frac{1}{\omega C}\right)I; \quad \underline{\boldsymbol{V}}_L = j\omega LI \tag{24.91}$$

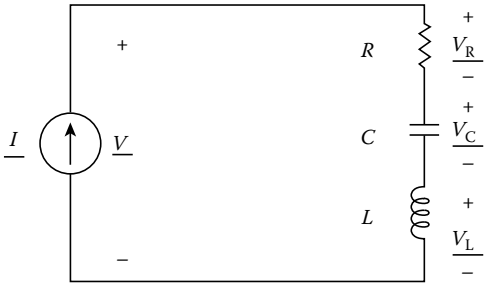


FIGURE 24.73 Series resonant circuit.

Then, the voltage $\underline{\boldsymbol{V}}$ is

$$\underline{\boldsymbol{V}} = I\left[R + j\left(\omega L - \frac{1}{\omega C}\right)\right] \tag{24.92}$$

The peak value of $\underline{\boldsymbol{V}}$ is

$$V = I\sqrt{R^2 + \left(\omega L - \frac{1}{\omega C}\right)^2} \tag{24.93}$$

where I is the peak value of $\underline{\boldsymbol{I}}$. The minimum value of V is

$$V_{\min} = IR \tag{24.94}$$

and this occurs at the frequency $\omega = 1/\sqrt{(LC)}$, which is the same resonance condition as for the circuit in Figure 24.70.

Equation 24.94 demonstrates that at resonance, the voltage across the LC subcircuit in Figure 24.73 is zero. However, the individual voltages across L and across C are not zero and can be quite large in magnitude depending on the values of the capacitor and inductor. These voltages are given by

$$\underline{\boldsymbol{V}}_C = -jI\sqrt{\frac{L}{C}} \quad \text{and} \quad \underline{\boldsymbol{V}}_L = jI\sqrt{\frac{L}{C}} \tag{24.95}$$

and therefore the voltage across the capacitor is exactly 180° out of phase with the voltage across the inductor.

At frequencies below resonance, $\underline{\boldsymbol{V}}$ lags $\underline{\boldsymbol{I}}$ in Figure 24.73, and therefore the circuit looks capacitive to the source. Above resonance, $\underline{\boldsymbol{V}}$ leads $\underline{\boldsymbol{I}}$, and the circuit looks inductive to the source. If the frequency of the source is $\omega = 1/\sqrt{(LC)}$ the circuit looks like a resistor of value R to the source.

24.3.8 Power in AC Circuits

If a sinusoidal voltage $v(t) = V\sin(\omega t + \theta_V)$ is applied to an LLFT network that possibly contains other sinusoidal sources having the same frequency ω, then a sinusoidal current $i(t) = I\sin(\omega t + \theta_I)$ flows at steady state as depicted in Figure 24.74. The instantaneous power delivered to the circuit by the voltage source is

$$p(t) = v(t)i(t) = VI\sin(\omega t + \theta_V)\sin(\omega t + \theta_I) \tag{24.96}$$

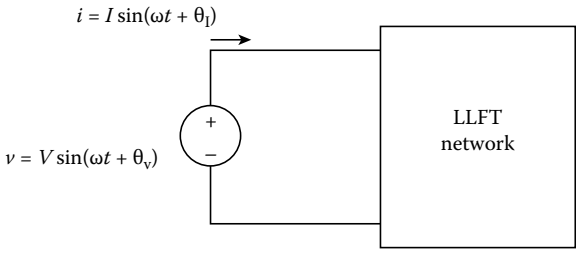


FIGURE 24.74 LLFT network that may contain other sinusoidal sources at the same frequency as the external generator.

where the units of $p(t)$ are watts (W). With the aid of the trigonometric identity

$$\sin\alpha\sin\beta = \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)]$$

we rewrite Equation 24.96 as

$$p(t) = \frac{1}{2}VI[\cos(\theta_v - \theta_I) - \cos(2\omega t + \theta_V + \theta_I)] \tag{24.97}$$

The instantaneous power delivered to the network in Figure 24.74 has a component that is constant and another component that has a frequency twice that of the excitation. At different instances of time, $p(t)$ can be positive or negative, meaning that the voltage source is delivering power to the network or receiving power from the network, respectively.

In AC circuits, however, it is usually the average power P that is of more interest than the instantaneous power $p(t)$ because average power generates the heat or performs the work.

The average over a period of a periodic function $f(t)$ with period T is

$$[f(t)]_{avg} = F = \frac{1}{T}\int_0^T f(t)\mathrm{d}t \tag{24.98}$$

The period of $p(t)$ in Equation 24.97 is $T = \pi/\omega$, and so

$$[p(t)]_{avg} = P = \frac{\omega}{\pi}\int_0^{\frac{\pi}{\omega}} p(t)\mathrm{d}t = \frac{1}{2}VI\cos(\theta_V - \theta_I) \tag{24.99}$$

The cosine term in Equation 24.99 plays an important role in power calculations and so is designated as the power factor (PF). Thus,

$$\text{Power factor} = \text{PF} = \cos(\theta_V - \theta_I) \tag{24.100}$$

If $|\theta_V - \theta_I| = \pi/2$, then PF = 0, and the average power delivered to the network in Figure 24.74 is zero; but if PF = 1, then P delivered to the network by the source is $VI/2$. If $0 < |\theta_V - \theta_I| < \pi/2$, then P is positive, and the source is delivering average power to the network. However, the network delivers average power to the source when P is negative, and this occurs if $\pi/2 < |\theta_V - \theta_I| < 3\pi/2$.

If the current leads the voltage in Figure 24.74, the convention is to consider PF as leading, and if current lags the voltage, the PF is regarded as lagging. However, it is not possible from PF alone to determine whether a current leads or lags voltage.

Example 24.24

Determine the average power delivered to the network shown in Figure 24.68a.

Solution

The phasor transform of the applied voltage is $\underline{\boldsymbol{V}} = V/0°$, and we determined in Example 24.24 that the current supplied was

$$\underline{\boldsymbol{I}}_1 = \frac{V\omega C e^{j\theta_1}}{\sqrt{\omega^2 C^2 R_1^2 + 1}}, \quad \theta_1 = \tan^{-1}\frac{1}{\omega C R_1}$$

The power factor is

$$\text{PF} = \cos(0 - \theta_1) = \cos(\theta_1)$$

which, with the aid of the triangle in Figure 24.75, can be rewritten as

$$\text{PF} = \frac{\omega C R_1}{\sqrt{(\omega C R_1)^2 + 1}}$$

Thus, the average power delivered to the circuit is

$$P = \frac{1}{2}\frac{V^2\omega C}{\sqrt{(\omega C R_1)^2 + 1}}\left[\frac{\omega C R_1}{\sqrt{(\omega C R_1)^2 + 1}}\right] = \frac{V^2\omega^2 C^2 R_1}{2(\omega^2 C^2 R_1^2 + 1)} = \frac{I_1^2 R_1}{2}$$

We note that if R_1 were zero in the previous example, then $P = 0$ because the circuit would be purely capacitive, and *PF* would be zero.

If no sources are in the network in Figure 24.74, then the network terminal variables are related by

$$\underline{\boldsymbol{V}} = \underline{\boldsymbol{I}} Z(j\omega) \tag{24.101}$$

where $Z(j\omega)$ is the input impedance of the network. Because Z is, general and complex, we can write it as

$$Z(j\omega) = R(\omega) + jX(\omega) = |Z|e^{j\theta_Z} \tag{24.102}$$

where

$$R(\omega) = \Re Z(j\omega); \quad X(\omega) = \Im Z(j\omega) \quad \text{and} \quad \theta_Z = \tan^{-1}\left(\frac{X(\omega)}{R(\omega)}\right) \tag{24.103}$$

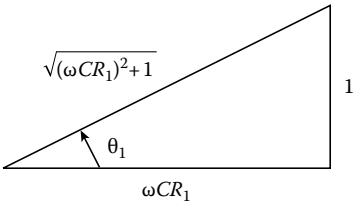


FIGURE 24.75 Triangle for determining *PF*.

In Equation 24.103, the (real) function $X(\omega)$ is termed the reactance. Employing the polar form of the phasors, we can rewrite Equation 24.101 as

$$V\angle\theta_V = I\angle\theta_I|Z|\angle\theta_Z = I|Z|\angle(\theta_I + \theta_Z) \tag{24.104}$$

Equating magnitudes and angles, we obtain

$$V = I|Z| \quad \text{and} \quad \theta_V = \theta_I + \theta_Z \tag{24.105}$$

Thus, we can express P delivered to the network as

$$P = \frac{1}{2}VI\cos(\theta_V - \theta_I) = \frac{1}{2}I^2|Z|\cos\theta_Z \tag{24.106}$$

But $|Z|\cos(\theta_Z) = R(\omega)$ so that

$$P = \frac{1}{2}I^2R(\omega) \tag{24.107}$$

Equation 24.107 indicates that the real part of the impedance absorbs the power. The imaginary part of the impedance, $X(\omega)$, does not absorb average power. Example 24.24 in this section provides an illustration of Equation 24.107.

An expression for average power in terms of the input admittance $Y(j\omega) = 1/Z(j\omega)$ can also be obtained. Again, if no sources are within the network, then the terminal variables in Figure 24.74 are related by

$$\underline{\boldsymbol{I}} = \underline{\boldsymbol{V}}Y(j\omega) \tag{24.108}$$

The admittance $Y(j\omega)$ can be written as

$$Y(j\omega) = |Y(j\omega)|e^{j\theta_Y} = G(\omega) + jB(\omega) \tag{24.109}$$

where $G(\omega)$ is conductance and $B(\omega)$ is susceptance, and where

$$G(\omega) = \Re Y(j\omega); \quad B(\omega) = \Im Y(j\omega) \quad \text{and} \quad \theta_Y = \tan^{-1}\left[\frac{B(\omega)}{G(\omega)}\right] \tag{24.110}$$

Then, average power delivered to the network can be expressed as

$$P = \frac{1}{2}V^2|Y|\cos\theta_Y = \frac{1}{2}V^2G(\omega) \tag{24.111}$$

If the network contains sinusoidal sources, then Equation 24.99 should be employed to obtain P instead of Equation 24.107 or 24.111.

Consider a resistor R with a voltage $v(t) = V\sin(\omega t)$ across it and therefore a current $i(t) = I\sin(\omega t) = v(t)/R$ through it. The instantaneous power dissipated by the resistor is

$$p(t) = v(t)i(t) = \frac{v^2(t)}{R} = i^2(t)R \tag{24.112}$$

The average power dissipated in R is

$$P = \frac{1}{T}\int_0^T i^2(t)R\mathrm{d}t = I_{\mathrm{eff}}^2 R \tag{24.113}$$

where we have introduced the new constant I_{eff}. From Equation 24.113, we can express I_{eff} as

$$I_{\mathrm{eff}} = \sqrt{\frac{1}{T}\int_0^T i^2(t)\mathrm{d}t} \tag{24.114}$$

This expression for I_{eff} can be read as "the square root of the mean (average) of the square of $i(t)$" or, more simply, as "the root mean square value of $i(t)$," or, even more succinctly, as "the RMS value of $i(t)$." Another designation for this constant is I_{rms}. Equation 24.114 can be extended to any periodic voltage or current.

The RMS value of a pure sine wave such as $i(t) = I\sin(\omega t + \theta_1)$ or $v(t) = V\sin(\omega t + \theta_v)$ is

$$I_{\mathrm{rms}} = \frac{I}{\sqrt{2}} \quad \text{or} \quad V_{\mathrm{rms}} = \frac{V}{\sqrt{2}} \tag{24.115}$$

where I and V are the peak values of the sine waves. Normally, the voltages and currents listed on the nameplates of power equipment and household appliances are given in terms of RMS values instead of peak values. For example, a 120 V, 100 W lightbulb is expected to dissipate 100 W when a voltage $120(\sqrt{2})[\sin(\omega t)]$ is impressed across it. The peak value of this voltage is 170 V.

If we employ *RMS* values, Equation 24.99 can be rewritten as

$$P = V_{\mathrm{rms}} I_{\mathrm{rms}} PF \tag{24.116}$$

Equation 24.116 emphasizes the fact that the concept of *RMS* values of voltages and currents was developed in order to simplify the calculation of average power.

Because $PF = \cos(\theta_v - \theta_1)$, we can rewrite Equation 24.116 as

$$\begin{aligned} P &= V_{\mathrm{rms}} I_{\mathrm{rms}} \cos(\theta_{\mathrm{V}} - \theta_{\mathrm{I}}) = \Re[V_{\mathrm{rms}} e^{j\theta_{\mathrm{V}}} I_{\mathrm{rms}} e^{-j\theta_{\mathrm{I}}}] \\ &= \Re[\underline{\boldsymbol{V}}\,\underline{\boldsymbol{I}}^{\star}] \end{aligned} \tag{24.117}$$

where $\underline{\boldsymbol{I}}^{\star}$ is the conjugate of $\boldsymbol{I}$. If $P = \Re[\underline{\boldsymbol{V}}\,\underline{\boldsymbol{I}}^{\star}]$, the question arises as to what the imaginary part of $\underline{\boldsymbol{V}}\,\underline{\boldsymbol{I}}^{\star}$ represents. This question leads naturally to the concept of complex power, denoted by the bold letter $\boldsymbol{S}$, which has the units of volt-amperes (VA). If P represents real power, then we can write

$$\boldsymbol{S} = P + jQ \tag{24.118}$$

where

$$\boldsymbol{S} = \underline{\boldsymbol{V}}\,\underline{\boldsymbol{I}}^{\star} \tag{24.119}$$

and where

$$Q = \Im[\underline{\boldsymbol{V}}\,\underline{\boldsymbol{I}}^{\star}] = V_{\mathrm{rms}} I_{\mathrm{rms}} \sin(\theta_{\mathrm{V}} - \theta_{\mathrm{I}}) \tag{24.120}$$

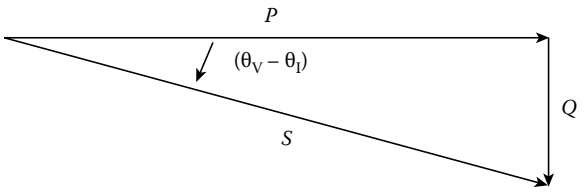


FIGURE 24.76 Power triangle for a capacitive circuit.

Thus, Q represents imaginary or reactive power. The units of Q are VARs, which stands for volt-amperes reactive. Reactive power is not available for conversion into useful work. It is needed to establish and maintain the electric and magnetic fields associated with capacitors and inductors [4]. It is an overhead required for delivering P to loads, such as electric motors, that have a reactive part in their input impedances.

The components of complex power can be represented on a power triangle. Figure 24.76 is a power triangle for a capacitive circuit. Real and imaginary power are added as shown to yield the complex power $\boldsymbol{S}$. Note that $(\theta_v - \theta_1)$ and Q are both negative for capacitive circuits. The following example illustrates the construction of a power triangle for an RL circuit.

Example 24.25

Determine the components of power delivered to the RL circuit in Figure 24.77. Provide a phasor diagram for the current and the voltages, construct a power triangle for the circuit, and show how the power diagram is related to the impedances of the circuit.

Solution

We have

$$\underline{\boldsymbol{V}} = Ve^{j\theta} \quad \text{and} \quad \underline{\boldsymbol{I}} = \frac{Ve^{j\theta_I}}{\sqrt{R^2 + (\omega L)^2}}$$

where

$$\theta_I = -\tan^{-1}\frac{\omega L}{R}$$

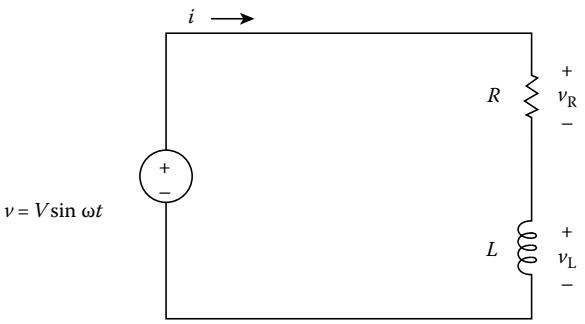


FIGURE 24.77 Network for Example 24.25.

Because $\theta_v = 0$, *PF* is

$$PF = \cos(\theta_V - \theta_I) = \frac{R}{\sqrt{R^2 + (\omega L)^2}}$$

and is lagging. The voltages across *R* and *L* are given by

$$\underline{\boldsymbol{V}}_R = \underline{\boldsymbol{I}}R = \frac{VRe^{j\theta_I}}{\sqrt{R^2 + (\omega L)^2}}$$
$$\underline{\boldsymbol{V}}_L = j\omega L\underline{\boldsymbol{I}} = \frac{V\omega L}{\sqrt{R^2 + (\omega L)^2}} e^{j\left(\frac{\pi}{2} + \theta_I\right)}$$

and *Z* is

$$Z = R + j\omega L = \sqrt{R^2 + (\omega L)^2} e^{-j\theta_I}$$

The real and imaginary components of the complex power are simply calculated as

$$P = I_{rms}^2 R(\omega) = \frac{V_{rms}^2 R}{R^2 + (\omega L)^2}$$
$$Q = \frac{V_{rms}^2 \omega L}{R^2 + (\omega L)^2}$$

Figure 24.78 presents the phasor diagram for this circuit in which we have taken the reference phasor as $\underline{\boldsymbol{I}}$ and therefore have shown $\underline{\boldsymbol{V}}$ leading $\underline{\boldsymbol{I}}$ by $(\theta_v - \theta_1)$. Also, we have moved $\underline{\boldsymbol{V}}_L$ parallel to itself to form a triangle. These operations cause the phasor diagram to be similar to the power triangle. Figure 24.79a shows a representation for the impedance in Figure 24.77. If each side of the triangle in Figure 24.79a is multiplied by I_{rms}, then we obtain voltage triangle in Figure 24.79b. Next, we multiply the sides of the voltage triangle by I_{rms} again to obtain the power triangle in Figure 24.79c. The horizontal side is the average power *P*, the vertical side is *Q*, and the hypotenuse has a length that represents the magnitude of the complex power $\boldsymbol{S}$. All three triangles in Figure 24.79 are similar. The angles between sides are preserved.

If *P* remains constant in Figure 24.76, but the magnitude of the angle becomes larger so that the magnitude of *Q* increases, then [$\boldsymbol{S}$] increases. If the magnitude of the voltage is fixed, then the magnitude of the current supplied must increase. But then, either power would be lost in the form of heat in the wires supplying the load or larger diameter, more expensive wires, would be needed. For this reason,

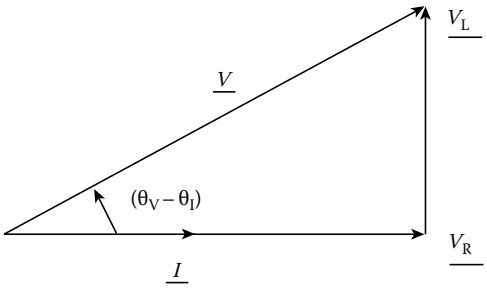


FIGURE 24.78 Phasor diagram for Example 24.25.

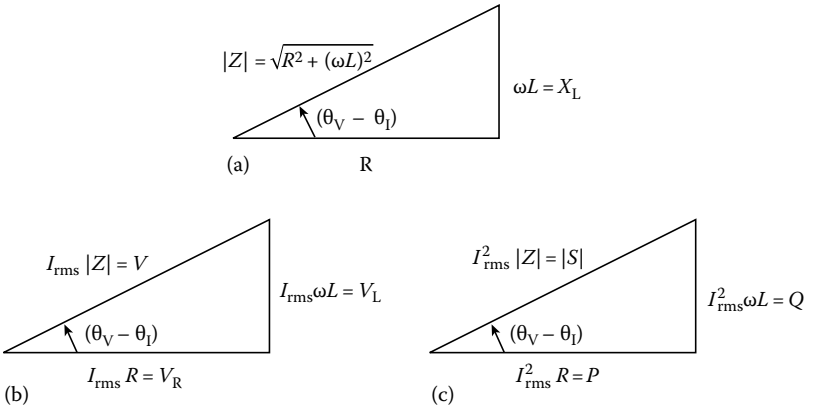


FIGURE 24.79 (a) Impedance triangle for circuit in Example 24.25. (b) Corresponding voltage triangle. (c) Power triangle.

power companies that supply power to large manufacturing firms that have many large motors impose unfavorable rates. However, the manufacturing firm can improve its rates if it improves its power factor. The following example illustrates how improving (correcting) *PF* is done.

Example 24.26

Determine the value of the capacitor to be connected in parallel with the RL circuit in Figure 24.80 to improve the PF of the overall circuit to one. The excitation is a voltage source having an amplitude of 120 V RMS and frequency $2\pi(60\text{ Hz}) = 377$ rad/s. What are the RMS values of the current supplied by the source at steady state before and after the capacitor is connected?

Solution

The current through the RL branch in Figure 24.80 is

$$\underline{\boldsymbol{I}}_{RL} = \frac{Ve^{j\theta}}{\sqrt{R^2 + (\omega L)^2}}; \quad \theta = -\tan^{-1}\frac{\omega L}{R}$$

and the current through the capacitor is

$$\underline{\boldsymbol{I}}_C = j\omega CV = V\omega Ce^{j(\pi/2)}$$

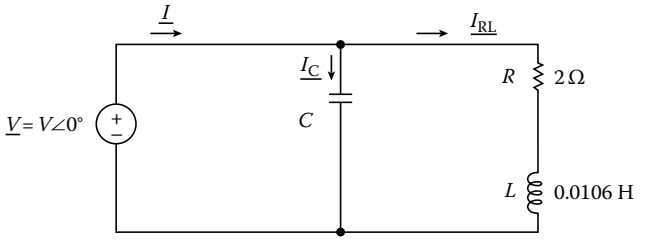


FIGURE 24.80 Circuit for Example 24.26.

Thus, the current supplied by the source to the *RLC* network is

$$\begin{aligned}\underline{\boldsymbol{I}} &= \underline{\boldsymbol{I}}_{\mathrm{RL}} + \underline{\boldsymbol{I}}_{\mathrm{C}} \\ &= \frac{V\cos\theta}{\sqrt{R^2 + (\omega L)^2}} + jV\left[\frac{-\omega L}{R^2 + (\omega L)^2} + \omega C\right]\end{aligned}$$

To improve the *PF* to one, the current $\underline{\boldsymbol{I}}$ should be in phase with $\underline{\boldsymbol{V}}$. Thus, we set the imaginary term in the equation for $\underline{\boldsymbol{I}}$ equal to zero, yielding:

$$C = \frac{L}{R^2 + (\omega L)^2} = 530\ \mu\mathrm{F}$$

a rather large capacitor. Before this capacitor is connected, the *RMS* value of the current supplied by the voltage source is $I_{\mathrm{rms}} = 26.833$ A. After the capacitor is connected, the source has to supply only 12 A RMS, a considerable reduction. In both cases, *P* delivered to the load is the same.

The following example also illustrates PF improvement.

Example 24.27

A load with PF = 0.7 lagging, depicted in Figure 24.81, consumes 12 kW of power. The line voltage supplied is 220 V RMS at 60 Hz. Find the size of the capacitor needed to correct the PF to 0.9 lagging, and determine the values of the currents supplied by the source both before and after the *PF* is corrected.

Solution

We will take the phase of the line voltage to be 0°. From $P = V_{\mathrm{rms}} I_{\mathrm{rms}} \mathrm{PF} = 12$ kW, we obtain $I_{\mathrm{rms}} = 77.922$ A. Because PF is 0.7 lagging, the phase of the current through the load relative to the phase of the line voltage is $-\cos^{-1}(0.7) = -45.57°$. Therefore, $\underline{\boldsymbol{I}}_{\mathrm{load}} = 77.922/(-45.57°°)$ A RMS. When *C* is connected in parallel with the load,

$$\begin{aligned}\underline{\boldsymbol{I}} &= \underline{\boldsymbol{I}}_{\mathrm{C}} + \underline{\boldsymbol{I}}_{\mathrm{load}} = 220(377)jC + 77.922e^{-j0.7954} \\ &= 54.54 - j[55.64 - 82{,}940C]\end{aligned}$$

If the PF were to be corrected to unity, we would set the imaginary part of the previous expression for current to zero; but this would require a larger capacitor (671 μF), which may be uneconomical. Instead, to retain a lagging but improved PF = 0.9, and corresponding to the current lagging the voltage by 25.84°, we write

$$0.9 = \frac{54.54}{\sqrt{54.54^2 + (55.64 - 82{,}940C)^2}}$$

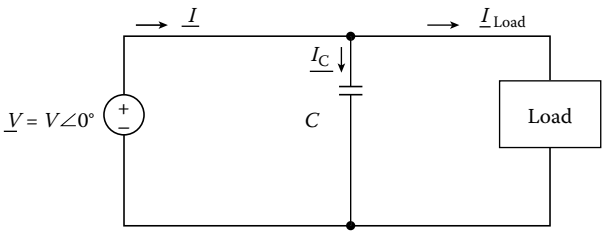


FIGURE 24.81 Circuit for Example 24.27 showing the load and the capacitor to be connected in parallel with the load to improve the power factor.

Therefore, $C = 352\ \mu$F. The line current is now

$$\underline{\boldsymbol{I}} = \underline{\boldsymbol{I}_C} + \underline{\boldsymbol{I}_{load}} = 60.615\angle(-25.87°)\ \text{A RMS}$$

Previous examples have employed ideal voltage sources to supply power to networks. However, in many electronic applications, the source has a fixed impedance associated with it, and the problem is to obtain the maximum average power transferred to the load [2]. Here, we assume that the resistance and reactance of the load can be independently adjusted. Let the source impedance be

$$Z_S(j\omega) = R_S(\omega) + jX_S(\omega)$$

The load impedance is denoted as

$$Z(j\omega) = R(\omega) + jX(\omega)$$

Figure 24.82 depicts these impedances. We assume that all the elements, including the voltage source, within the box formed by the dotted lines are fixed. The voltage source is $v(t) = V\sin(\omega t)$, and thus $i(t) = I\sin(\omega t + \theta)$, where V and I are peak values and

$$\theta = -\tan^{-1}\left[\frac{X_S(\omega) + X(\omega)}{R_S(\omega) + R(\omega)}\right]$$

The average power delivered to Z is

$$P = I_{rms}^2 R(\omega)$$

where $I_{rms} = I/\sqrt{2}$ and

$$I = \frac{V}{\sqrt{[R_S(\omega) + R(\omega)]^2 + [X_S(\omega) + X(\omega)]^2}} \tag{24.121}$$

Thus, the average power delivered to Z can be written as

$$P = \frac{V_{rms}^2 R(\omega)}{[R_S(\omega) + R(\omega)]^2 + [X_S(\omega) + X(\omega)]^2} \tag{24.122}$$

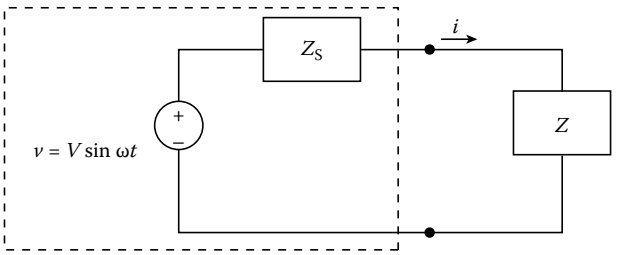


FIGURE 24.82 Z_S is fixed, and Z is to be chosen so that maximum average power is transferred to Z.

To maximize P, we first note that the term $[X_S(\omega) + X(\omega)]^2$ is always positive, and so this term always contributes to a larger denominator unless it is zero. Thus, we set

$$X(\omega) = -X_S(\omega) \tag{24.123}$$

and Equation 24.122 becomes

$$P = \frac{V_{rms}^2 R(\omega)}{[R_S(\omega) + R(\omega)]^2} \tag{24.124}$$

Second, we set the partial derivative with respect to $R(\omega)$ of the expression in Equation 24.124 to zero to obtain

$$\frac{\partial P}{\partial R} = V_{rms}^2 \frac{(R_S + R)^2 - 2R(R_S + R)}{(R_S + R)^4} = 0 \tag{24.125}$$

Equation 24.125 is satisfied for

$$R(\omega) = R_S(\omega) \tag{24.126}$$

and this value of $R(\omega)$ together with $X(\omega) = -X_S(\omega)$, yields maximum average power transferred to Z. Thus, we should adjust Z to

$$Z(j\omega) = Z_S^*(j\omega) \tag{24.127}$$

and we obtain

$$P_{max} = \frac{V_{rms}^2}{4R(\omega)} \tag{24.128}$$

Example 24.28

Find Z for the network in Figure 24.83 so that maximum average power is transferred to Z. Determine the value of P_{max}.

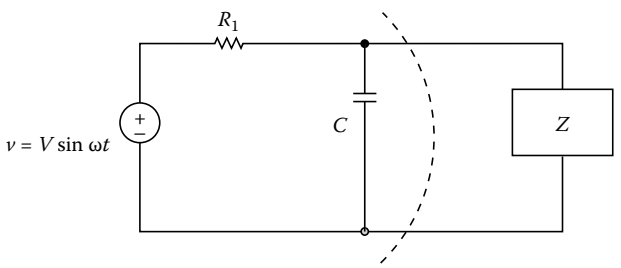


FIGURE 24.83 Circuit for Example 24.28.

Solution

We first obtain the Thévenin equivalent of the circuit to the left of the dotted arc in Figure 24.83 in order to reduce the circuit to the form of Figure 24.82.

$$\underline{\boldsymbol{V}}_{\mathrm{TH}} = \frac{\underline{\boldsymbol{V}}}{1 + j\omega R_1 C}$$
$$Z_{\mathrm{TH}} = \frac{R_1}{1 + j\omega R_1 C}$$

Thus,

$$Z = Z_{\mathrm{TH}}^{*} = \frac{R_1}{1 - j\omega R_1 C} = \frac{R_1}{1 + \frac{\omega R_1 C}{j}}$$
$$= \frac{jR_1}{j + \omega R_1 C} = \frac{j\frac{1}{\omega C}R_1}{R_1 + j\frac{1}{\omega C}}$$

The term $j(\omega C)$ appears inductive (at a single frequency), and so we equate it to $j\omega L$ to obtain

$$L = \frac{1}{\omega^2 C}$$

The impedance Z is therefore formed by the parallel connection of a resistor R_1 with the inductor L. Figure 24.84 depicts the resulting circuit. To determine $P_{\max}$, we note that the capacitor and inductor constitute a parallel circuit which is resonant at the frequency of excitation. It therefore appears as an open circuit to the source. Thus, $P_{\max}$ is easily obtained as

$$P_{\max} = I_{\mathrm{rms}}^2 R_1 = \frac{V^2}{8R_1}$$

where V is the peak value of $v(t)$.

Suppose Z is fixed and Z_S is adjustable in Figure 24.82. What should Z_S be so that maximum average power is delivered to Z? This is a problem that is applicable in the design of electronic amplifiers. The average power delivered to Z is given by Equation 24.122, and to maximize P, we set $X_S(\omega) = -X(\omega)$ as before. We therefore obtain Equation 24.124 again; but if R_S is adjustable instead of R, we see from Equation 24.124 that $P_{\max}$ is obtained when R_S equals zero.

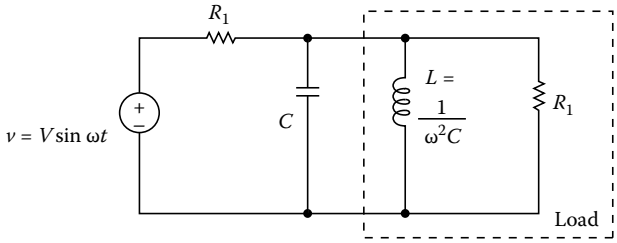


FIGURE 24.84 Circuit with load chosen to obtain maximum average power.

Acknowledgments

The author conveys his gratitude to Dr. Jacek Zurada, Mr. Tongfeng Qian, and to Dr. K. Wang for their help in proofreading this manuscript, and to Dr. Zbigniew J. Lata and Mr. Peichu (Peter) Sheng for producing the drawings.

References

1. A. Budak, *Circuit Theory Fundamentals and Applications*, 2nd edn., Englewood Cliffs, NJ: Prentice Hall, 1987.
2. L. P. Huelsman, *Basic Circuit Theory with Digital Computations*, Englewood Cliffs, NJ: Prentice Hall, 1972.
3. L. P. Huelsman, *Basic Circuit Theory*, 3rd edn., Englewood Cliffs, NJ: Prentice Hall, 1991.
4. S. Karni, *Applied Circuit Analysis*, New York: John Wiley & Sons, 1988.
5. L. Weinberg, *Network Analysis and Synthesis*, New York: McGraw-Hill, 1962.

25

Symbolic Analysis

25.1 Introduction and Definition **25**-1
25.2 Frequency Domain Analysis **25**-3
25.3 Traditional Methods (Single Expressions) **25**-4
Tree Enumeration Methods • Signal Flowgraph Method • Parameter Extraction Method • Interpolation Method
25.4 Hierarchical Methods (Sequence of Expressions) **25**-17
25.5 Approximate Symbolic Analysis **25**-20
25.6 Methods Based on Determinant Decision Diagrams **25**-23
25.7 Time Domain Analysis **25**-25
Fully Symbolic • Semisymbolic
References **25**-26

Benedykt S. Rodanski
University of Technology, Sydney

Marwan M. Hassoun
Iowa State University

25.1 Introduction and Definition

Symbolic circuit analysis, simply stated, is a term that describes the process of studying the behavior of electrical circuits using symbols instead of, or in conjunction with, numerical values. As an example to illustrate the concept, consider the input resistance of the simple circuit in Figure 25.1.

Analyzing the circuit using the unique symbols for each resistor without assigning any numerical values to them yields the input resistance of the circuit in the form:

$$\frac{V_{\text{in}}}{I_{\text{in}}} = \frac{R_1R_2 + R_1R_3 + R_1R_4 + R_2R_3 + R_2R_4}{R_2 + R_3 + R_4} \tag{25.1}$$

Equation 25.1 is the symbolic expression for the input resistance of the circuit in Figure 25.1. The formal definition of symbolic circuit analysis can be written as

Definition 25.1: Symbolic circuit analysis is the process of producing an expression that describes a certain behavioral aspect of the circuit with one, some, or all the circuit elements represented as symbols.

The idea of symbolic circuit analysis is not new; engineers and scientists have been using the process to study circuits since the inception of the concept of circuits. Engineers have used symbolic circuit analysis during their education process. Most still use it in their every day job functions. As an example, all electrical engineers have symbolically analyzed the circuit in Figure 25.2.

The equivalent resistance between nodes i and j is known to be

$$\frac{1}{R_{ij}} = \frac{1}{R_1} + \frac{1}{R_2}$$

or

$$R_{ij} = \frac{R_1 R_2}{R_1 + R_2}$$

This is the most primitive form of symbolic circuit analysis.

The basic justification for performing symbolic analysis rather than numerical analysis on a circuit can be illustrated by considering the circuit in Figure 25.1 again. Assume that the values of all the resistances R_1 through R_4 are given as 1Ω and that the input resistance was analyzed numerically. The result obtained would be

$$\frac{V_{\text{in}}}{I_{\text{in}}} = \frac{5}{3} \approx 1.667\,\Omega \tag{25.2}$$

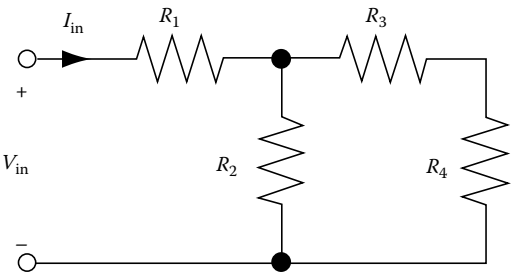


FIGURE 25.1 Symbolic circuit analysis example.

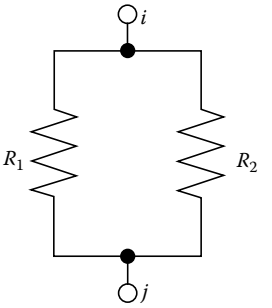


FIGURE 25.2 Common symbolic analysis problem.

Now consider the problem of increasing the input resistance of the circuit by adjusting only one of the resistor values. Equation 25.2 provides no insight into which resistor has the greatest impact on the input resistance. However, Equation 25.1 clearly shows that changing R_2, R_3, or R_4 would have very little impact on the input resistance since the terms appear in both the numerator and the denominator of the symbolic expression. It can also be seen that R_1 should be the resistor to change since it only appears in the numerator of the expression. Symbolic analysis has provided an insight into the problem.

From a circuit design perspective, numerical results from the simulation of a circuit can be obtained by evaluating the results of the symbolic analysis at a specific numerical point for each symbol. So ideally, only one simulation run is needed in order to analyze the circuit, and successive evaluations of the results replace the need for any extra iterations through the simulator. Other applications include sensitivity analysis [Lin92], [Bal04], circuit stability analysis, symbolic poles and zeros calculations [Con91], [Dro96], device modeling [Gie91], and circuit optimization [Kon92].

While the above ''hand calculations'' and somewhat trivial examples are used to illustrate symbolic circuit analysis, the thrust of the methods developed for symbolic analysis are aimed at computer implementations that are capable of symbolically analyzing circuits that cannot be analyzed ''by hand.'' Several such implementations have been developed over the years [Fer91], [Fid73], [Gie89], [Has89], [Lin70], [Mcn68], [Mie78], [Pot68], [San80], [Did91], [Sin77], [Som91], [Sta86], [Man91], [Wie89], [Kon88], [Hue89], [Has90], [Arn91], [Hsu93], [Lee92], [Li92], [Pie01], [Lib95], [Hen00].

Symbolic circuit analysis, referred to as simply symbolic analysis for the rest of this section, in its current form is limited to linear,* lumped, and time-invariant† networks. The scope of the analysis is

* Some references are made to the ability to analyze ''weakly nonlinear'' circuits [Gie91], [Top98]; however, the actual symbolic analysis is performed on a linearized model of the weakly nonlinear circuit. Other techniques are applicable to circuits with only a single strongly nonlinear variable [Wen98].

† One method, reported in [Lib93] and briefly discussed in Section 25.7, does deal with a limited class of time-variant networks.

primarily concentrated in the frequency domain, both *s*-domain [Fer91], [Fid73], [Gie89], [Has89], [Lin70], [Mcn68], [Mie78], [Pot68], [San80], [Did91], [Sin77], [Som91], [Sta86], [Man91], [Wie89], [Hue89], [Has93], [Hsu93], [Lee92] and *z*-domain [Mar93], [Li92], [Arn91], [Kon88]; however, the predominant development has been in the *s*-domain. Also, recent work has expanded symbolic analysis into the time domain [Als93], [Gre93], [Lib93], [Has91]. Sections 25.2 through 25.6 will discuss the basic methods used in symbolic analysis for mainly *s*-domain frequency analysis. However, Section 25.7 highlights the currently known time domain techniques.

25.2 Frequency Domain Analysis

Traditional symbolic circuit analysis is performed in the frequency domain where the results are in terms of the frequency variable *s*. The main goal of performing symbolic analysis on a circuit in the frequency domain is to obtain a symbolic transfer function of the form

$$H(s,\mathbf{x}) = \frac{N(s,\mathbf{x})}{D(s,\mathbf{x})}, \quad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix}, p \le p_{\text{all}} \tag{25.3}$$

The expression is a rational function of the complex frequency variable *s*, and the variables x_1 through x_p representing the variable circuit elements, where p is the number of variable circuit elements and p_{all} is the total number of circuit elements. Both the numerator and the denominator of $H(s,\mathbf{x})$ are polynomials in *s* with real coefficients. Therefore, we can write

$$H(s,\mathbf{x}) = \frac{\sum_{i=0}^{m} a_i(\mathbf{x}) s^i}{\sum_{i=0}^{n} b_i(\mathbf{x}) s^i} = \frac{\prod_{i=1}^{m} [s - z_i(\mathbf{x})]}{\prod_{i=1}^{n} [s - p_i(\mathbf{x})]} \tag{25.4}$$

Most symbolic methods to date concentrate on the first form of $H(s,\mathbf{x})$ and several algorithms exist to obtain coefficients $a_i(\mathbf{x})$ and $b_i(\mathbf{x})$ in fully symbolic, partially symbolic (semisymbolic), or numerical form. The zero/pole representation of $H(s,\mathbf{x})$, although more useful in gaining insight into circuit behavior, proved to be very difficult to obtain in symbolic form for anything but very simple circuits. For large circuits, various approximation techniques must be employed [Hen00], [Con96].

A more recent approach to representing the above network function emerged in the 1980s and is based on a decomposed hierarchical form of Equation 25.3 [Sta86], [Has89], [Has93], [Sta96], [Pie96]. This hierarchical representation is referred to as a sequence of expressions (SoE) representation to distinguish it from the single expression representation of Equation 25.3 and is addressed in Section 25.4.

Several methodologies exist to perform symbolic analysis in the frequency domain. The early work was to produce a transfer function $H(s)$ with the frequency variable *s* being the only symbolic variable. Computer programs with these capabilities include CORNAP [Pot68] and NASAP [Mcn68]. The interest in symbolic analysis today is in the more general case when some or all of the circuit elements are represented by symbolic variables. The methods developed for this type of analysis fall under one of the following categories:

Traditional methods (single expression):

1. Tree enumeration methods
   a. Single graph methods
   b. Two graph methods
2. Signal flowgraph (SFG) methods
3. Parameter extraction methods
   a. Modified nodal analysis-based (MNA) methods
   b. Tableau formulation-based methods
4. Interpolation method

Hierarchical methods (sequence of expressions):

1. SFG methods
2. MNA-based methods

The classification above includes the exact methods only. For large circuits the traditional methods suffer from exponential growth of the number of terms in the formula with circuit size. If a certain degree of error is allowed, it may be possible to simplify the expression considerably, by including only the most significant terms. Several *approximate symbolic methods* have been investigated [Hsu93], [Yu96], [Hen00].

Sections 25.3 through 25.5 will discuss the basic theory for the above methods. Circuit examples are shown for all major methods except for the interpolation method due to its limited current usage* and its inability to analyze fully symbolic circuits.

25.3 Traditional Methods (Single Expressions)

This class of methods attempts to produce a single transfer function in the form of Equation 25.3. The major advantage of having a symbolic expression in that form is the insight that can be gained by observing the terms in both the numerator and the denominator. The effects of the different terms can, perhaps, be determined by inspection. This process is valid for the cases where there are relatively few symbolic terms in the expression.

Before indulging in the explanation of the different methods covered by this class, some definition of terms is in order.

Definition 25.2: *$RLCg_m$ circuit* is one that may contain only resistors, inductors, capacitors, and voltage-controlled current sources with the gain (transconductance) designated as g_m.

Definition 25.3: *Term cancellations* is the process in which two equal symbolic terms cancel out each other in the symbolic expression. This can happen in one of two ways: by having two equal terms with opposite signs added together, or by having two equal terms (regardless of their signs) divided by each other. For example, the equation

$$\frac{ab(ab+cd)-ab(cd-ef)}{ab(cd-gh)} \tag{25.5}$$

where a, b, c, d, e, f, g, and h are symbolic terms and can be reduced by observing that the terms ab in the numerator and denominator cancel each other and the terms $+cd$ and $-cd$ cancel each other in the numerator. The result is

$$\frac{ab+ef}{cd-gh} \tag{25.6}$$

Definition 25.4: *Cancellation-free*: Equation 25.6 is said to be a cancellation-free equation (i.e., no possible cancellations exist in the expression) while Equation 25.5 is not.

---

* The main applications of the polynomial interpolation method in symbolic analysis are currently in numerical reference generation for symbolic approximation [Fer98] and calculation of numerical coefficients in semisymbolic analysis [Pie93].

Definition 25.5: *Cancellation-free algorithm*: The process of term cancellation can occur during the execution of an algorithm where a cancellation-free equation is generated directly rather than generating an expression with possible term cancellations in it. Cancellation-free algorithms are more desirable because, otherwise, an overhead is needed to generate and keep the terms that are to be canceled later. The different methods that fall under the traditional class are explained next.

25.3.1 Tree Enumeration Methods

Several programs have been produced based on this method [Cal65], [Mcc72], [Gat70], [Man72]. Practical implementations of the method can only handle small circuits in the range of 15 nodes and 30 branches [Chu75]. The main reason is the exponential growth in the number of symbolic terms generated. The method can only handle one type of controlled sources, namely voltage-controlled current sources. So only $RLCg_m$ circuits can be analyzed. Also, the method does not produce any symbolic term cancellations for RLC circuits, and produces only a few for $RLCg_m$ circuits.

The basic idea of the tree enumeration method is to construct an augmented circuit (a slightly modified version of the original circuit), its associated directed graph, and then enumerating all the directed trees of the graph. The admittance products of these trees are then used to find the node admittance matrix (NAM) determinant and cofactors (the matrix itself is never constructed) to produce the required symbolic transfer functions. For a circuit with n nodes (with node n designated as the reference node) where the input is an excitation between nodes 1 and n and the output is taken between nodes 2 and n, the transfer functions of the circuit can be written as

$$Z_{\text{in}} = \frac{V_1}{I_1} = \frac{\Delta_{11}}{\Delta} \tag{25.7}$$

$$\frac{V_{\text{o}}}{I_{\text{in}}} = \frac{V_2}{I_1} = \frac{\Delta_{12}}{\Delta} \tag{25.8}$$

$$\frac{V_{\text{o}}}{V_{\text{in}}} = \frac{V_2}{V_1} = \frac{\Delta_{12}}{\Delta_{11}} \tag{25.9}$$

where

Δ is the determinant of the NAM $\mathbf{Y}_n$ (dimension $n-1 \times n-1$)
Δ_{ij} is the ijth cofactor of $\mathbf{Y}_n$

It can be shown that a simple method for obtaining Δ, Δ_{11}, and Δ_{12} is to construct another circuit comprised of the original circuit with an extra admittance $\hat{y}_{\text{S}}$ in parallel with a voltage-controlled current source, $\hat{g}_{\text{m}}V_2$, connected across the input terminals (nodes 1 and n). The determinant of $\hat{\mathbf{Y}}_n$ (the NAM for the new, slightly modified, circuit) can be written as

$$\hat{\Delta} = \Delta + \hat{y}_{\text{S}}\Delta_{11} + \hat{g}_{\text{m}}\Delta_{12} \tag{25.10}$$

This simple trick allows the construction of the determinant expression of the original circuit and its two needed cofactors by simply formulating the expression for the new augmented circuit. Example 25.1 illustrates this process.

The basic steps of the tree enumeration algorithm are (condensed from [Chu75])

1. Construct the augmented circuit from the original circuit by adding an admittance $\hat{y}_{\text{S}}$ and a transconductance $\hat{g}_{\text{m}}V_2$, in parallel between the input node and the reference node.
2. Construct a directed graph $\mathbf{G}_{\text{ind}}$ associated with the augmented circuit. The stamps used to generate $\mathbf{G}_{\text{ind}}$ are shown in Figure 25.3.
3. Find all directed trees for $\mathbf{G}_{\text{ind}}$. A directed tree rooted at node i is a subgraph of $\mathbf{G}_{\text{ind}}$ with node i having no incoming branches and each other node having exactly one incoming branch.

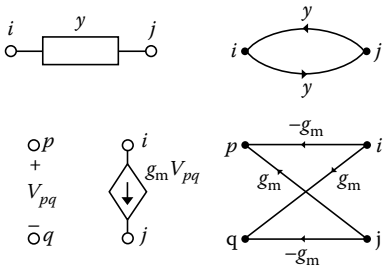


FIGURE 25.3 Element stamps for generating $\mathbf{G}_{ind}$.

4. Find the admittance product for each directed tree. An admittance product of a directed tree is simply a term that is the product of all the weights of the branches in that tree.
5. Apply Theorem 25.1.

THEOREM 25.1: [CHU75]

For any RLCg_m circuit, the determinant of the NAM (with any node as the reference node) is equal to the sum of all directed tree admittance products of $\mathbf{G}_{ind}$ (with any node as the root).

In other words

$$\hat{\Delta} = \sum \text{tree admittance products} \tag{25.11}$$

Arranging Equation 25.11 in the form of Equation 25.10 results in the necessary determinant and cofactors of the original circuit and the required transfer functions are generated from Equations 25.7 through 25.9.

Example 25.1

A circuit and its augmented counterpart are shown in Figure 25.4. The circuit is the small-signal model of a simple inverting CMOS amplifier, shown with the coupling capacitance C_C taken into account. Figure 25.5 shows the directed graph associated with the augmented circuit constructed using the rules shown in Figure 25.3. The figure also shows all the directed trees rooted at node 3 of the graph. Parallel branches heading in the same direction are combined into one branch with a weight equal to the sum of the weights of the individual parallel branches.

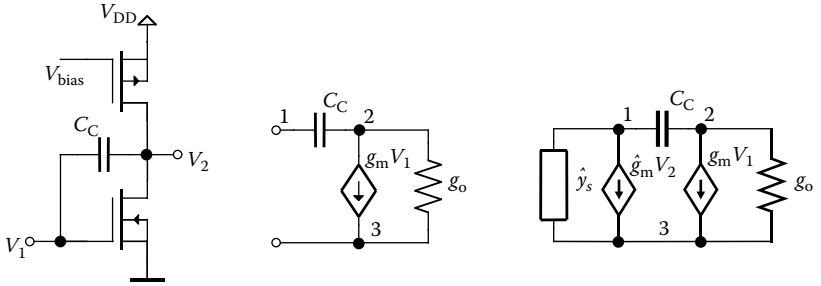


FIGURE 25.4 Circuit of Example 25.1 and its augmented equivalent diagram.

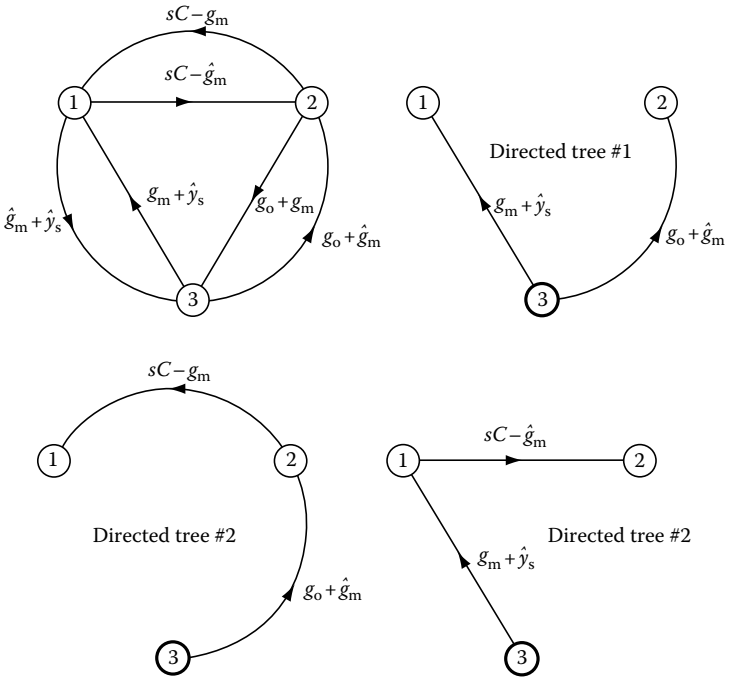


FIGURE 25.5 Graph and its directed trees of Example 25.1.

Applying Equation 25.11 and rearranging the terms results in

$$\begin{aligned}\hat{\Delta} &= (g_m+\hat{y}_S)(g_o+\hat{g}_m)+(sC_C-g_m)(g_o+\hat{g}_m)+(g_m+\hat{y}_S)(sC_C-\hat{g}_m)\\ &= \underbrace{sC_C(g_m+g_o)}_{\Delta}+\hat{y}_S\underbrace{(sC_C-g_o)}_{\Delta_{11}}+\hat{g}_m\underbrace{(sC_C-g_m)}_{\Delta_{12}}\end{aligned} \tag{25.12}$$

Note the fact that Equation 25.12, which is the direct result of the algorithm, is not cancellation-free. Some terms cancel out to result in the determinant of the original circuit and its two cofactors of interest. The final transfer functions can be readily obtained by substituting the above results into Equations 25.7 through 25.9.

25.3.2 Signal Flowgraph Method

Two types of flowgraphs are used in symbolic analysis. The first is referred to as a Mason's SFG and the second as Coates graph. Mason's SFG is by far a more popular and well-known SFG which has been used extensively in symbolic analysis among other control applications. Both the Mason's SFG and the Coates graph are used as basis for hierarchical symbolic analysis. However, the Coates graph was introduced to symbolic analysis by [Sta86] solely for the purpose of performing hierarchical symbolic analysis. This section will cover the Mason's SFG only.

The symbolic methods developed here are based on the idea formalized by Mason [Mas56] in the 1950s. Formulation of the SFG and then the evaluation of the gain formula associated with it (Mason's formula) is the basis for symbolic analysis using this method. This method is used in the publicly available programs NASAP [Mcn68], [Okr70], and SNAP [Lin70]. The method has the same circuit size limitations as the tree enumeration method due to the exponential growth in the number of symbolic terms. However, the SFG method allows all four types of controlled sources to be analyzed which made it a more popular method for symbolic analysis. The method is not cancellation-free which contributes to

the circuit size limitation mentioned earlier. An improved SFG method that avoids term cancellations was described in [Mie78].

The analysis process of a circuit consists of two parts, the first is constructing the SFG for the given circuit and the second is to perform the analysis on the SFG. Some definitions are needed before proceeding to the details of these two parts.

Definition 25.6: *Signal flowgraph*: An SFG is a weighted directed graph representing a system of simultaneous linear equations. Each node (x_i) in the SFG represents a circuit variable (node voltage, branch voltage, branch current, capacitor charge, or inductor flux) and each branch weight (w_{ij}) represents a coefficient relating x_i to x_j.

Every node in the SFG can be looked at as a summer. For a node x_k with m incoming branches

$$x_k = \sum_i w_{ik} x_i \tag{25.13}$$

where i spans the indices of all incoming branches from x_i to x_k.

Definition 25.7: *Path weight*: The weight of a path from x_i to x_j (P_{ij}) is the product of all the branch weights in the path.

Definition 25.8: *Loop weight*: The weight of a loop is the product of all the branch weights in that loop. This also holds for a loop with only one branch in it (self-loop).

Definition 25.9: *nth order loop*: An nth order loop is a set of n loops that have no common nodes between any two of them. The weight of an nth order loop is the product of the weights of all n loops.

Any transfer function x_j/x_i, where x_i is a source node, can be found by the application of Mason's formula:

$$\frac{x_j}{x_i} = \frac{1}{\Delta} \sum_k P_k \Delta_k \tag{25.14}$$

where

$$\begin{aligned} \Delta = 1 &- \sum_{\text{all}} \text{directed loop weights} \\ &+ \sum_{\text{all}} \text{2nd order loop weights} \\ &- \sum_{\text{all}} \text{3rd order loop weights} \\ &+ \cdots \end{aligned} \tag{25.15}$$

$$\begin{aligned} P_k &= \text{weight of the } k\text{th path from the source node } x_i \text{ to } x_j \\ \Delta_k &= \Delta \text{ with all loop contributions that are touching } P_k \text{ eliminated} \end{aligned} \tag{25.16}$$

The use of the above equations can be illustrated via Example 25.2.

Example 25.2

Consider the circuit in Figure 25.6.

The formulation of the SFG for this circuit takes on the following steps:

1. Find a tree and a cotree of the circuit such that all current sources are in the cotree and all voltage sources are in the tree.
2. Use Kirchhoff's current law (KCL), branch admittances, and tree branch voltages to find an expression for every cotree link current. In the case of a controlled source, simply use the branch relationship. For the previous example, this yields:

$$I_C = sC_C(V_1 - V_2) = sC_C V_1 - sC_C V_2$$
$$I = g_m V_1$$

3. Use Kirchhoff's voltage law (KVL), branch impedances, and cotree link currents to find an expression for every tree branch voltage. In the case of a controlled source, simply use the branch relationship. For the previous example, this yields:

$$V_{g_o} = V_2 = \frac{1}{g_o}(-I + I_C)$$

4. Create the SFG by drawing a node for each current source, voltage source, tree branch voltage, and cotree link current.
5. Use Equation 25.13 to draw the branches between the nodes that realize the linear equations developed in the previous steps.

Figure 25.7 shows the result of executing the above five steps on the example circuit. This formulation is referred to as the compact SFG. Any other variables that are linear combinations of the variables in the SFG (e.g., node voltages) can be added to the SFG by simply adding the extra node and implementing the linear relationship using SFG branches. A more detailed discussion of SFGs can be found in [Chu75] and [Lin91].

Now applying Equations 25.15 and 25.16 yields:

$$P_1 = -\frac{g_m}{g_o}, \quad P_2 = \frac{sC_C}{g_o}, \quad L_1 = -\frac{sC_C}{g_o}, \quad \Delta = 1 - \left(-\frac{sC_C}{g_o}\right), \quad \Delta_1 = 1, \quad \Delta_2 = 1$$

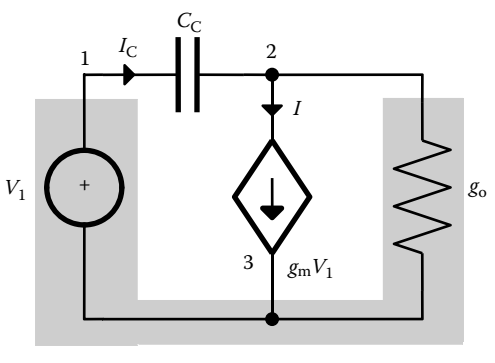


FIGURE 25.6 Circuit for Example 25.2 with its tree highlighted.

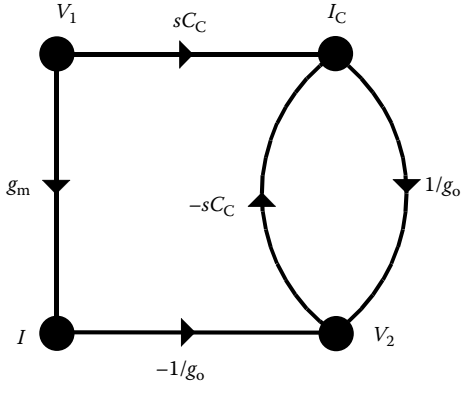


FIGURE 25.7 SFG for Example 25.2.

Equation 25.14 then produces the final transfer function

$$\frac{V_2}{V_1} = \frac{1}{1 + \frac{sC_C}{g_o}} \left(-\frac{g_m}{g_o} + \frac{sC_C}{g_o} \right) = \frac{sC_C - g_m}{sC_C + g_o}$$

25.3.3 Parameter Extraction Method

This method is best suited when few parameters in a circuit are symbolic while the rest of the parameters are in numeric form (s being one of the symbolic variables). The method was introduced in 1973 [Ald73]. Other variations on the method were proposed later in [Sin77], [San80], and [Pie93]. The advantage of the method is that it is directly related to the basic determinant properties of widely used equation formulation methods like the modified nodal method [Ho75] and the tableau method [Hac71]. As the name of the method implies, it provides a mechanism for extracting the symbolic parameters out of the matrix formulation breaking the matrix solution problem into a numeric part and a symbolic part. The numeric part can then be solved using any number of standard techniques and recombined with the extracted symbolic part. The method has the advantage of being able to handle larger circuits than the previously discussed fully symbolic methods if only a few parameters are represented symbolically. If the number of symbolic parameters in a circuit is high, the method will exhibit the same exponential growth in the number of symbolic terms generated and will have the same circuit size limitations as the other algorithms previously discussed.

The method does not limit the type of matrix formulation used to analyze the circuit. However, the extraction rules depend on the pattern of the symbolic parameters in the matrix. [Ald70] use the indefinite admittance matrix (IAM) as the basis of the analysis and the rules depend on the appearance of a symbolic parameter in four locations in the matrix: (i, i), (i, j), (j, i), and (j, j). Singhal and Vlach [Sin77] use the tableau equations and can handle a symbolic parameter that only appears once in the matrix. Sannuti and Puri [San80] force the symbolic parameters to appear only on the diagonal using a two-graph method [Chu75] to write the tableau equations. The parameter extraction method was further simplified in [Pie93] where the formula is given to calculate a coefficient (generally a polynomial in s) at every symbol combination. Some invalid symbol combinations (i.e., the ones that do not appear in the final formula) can be eliminated before calculations by topological considerations. To illustrate both approaches to parameter extraction, this section will present the IAM formulation and the most recent two-graph method. Details of other formulations can be found in [Lin91], [Sin77], [San80], and [Mie78].

25.3.3.1 Indefinite Admittance Matrix Approach

One of the basic properties of the IAM is the symmetric nature of the entries sometimes referred to as *quadrantal* entries [Chu75], [Lin91]. A symbolic variable α will always appear in four places in the IAM, $+\alpha$ in entries (i, k) and (j, m), and $-\alpha$ in entries (i, m) and (j, k) as shown in the following equation:

$$\begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} k & & m \end{array} \\ \begin{bmatrix} \vdots & & \vdots \\ \alpha & \cdots & -\alpha \\ \vdots & & \vdots \\ -\alpha & \cdots & \alpha \\ \vdots & & \vdots \end{bmatrix} \end{array}$$

where $i \neq j$ and $k \neq m$. For the case of an admittance y between nodes i and j, we have $k = i$ and $j = m$. The basic process of extracting the parameter (the symbol) α can be performed by applying the following equation [Chu75], [Ald73]:

$$\text{Cofactor of } \mathbf{Y}_{\text{ind}} = \text{cofactor of } \mathbf{Y}_{\text{ind},\alpha=0} + (-1)^{j+m}\alpha \text{ (cofactor of } \mathbf{Y}_{\alpha}) \tag{25.17}$$

where $\mathbf{Y}_\alpha$ is a matrix that does not contain α and is obtained by

1. Adding row j to row i
2. Adding column m to column k
3. Deleting row j and column m

For the case where several symbols exist, the above extraction process can be repeated and would result in

$$\text{cof}(\mathbf{Y}_{\text{ind}}) = \sum_j P_j \,\text{cof}(\mathbf{Y}_j)$$

where

P_j is some product of symbolic parameters including the sign
$\mathbf{Y}_j$ is a matrix with the frequency variable s, possibly, being the only symbolic variable

The cofactor of $\mathbf{Y}_j$ may be evaluated using any of the usual evaluation methods [Chu75], [Vla94]. Programs implementing this technique include NAPPE2 [Lin91] and SAPWIN [Lib95].

Example 25.3: [Chu75]

Consider the resistive circuit in Figure 25.8. The goal is to find the input impedance Z_{14} using the parameter extraction method where g_m is the only symbolic variable in the circuit. In order to use Equations 25.7 and 25.10 an admittance $\hat{y}_S$ is added across the input terminals of the circuit to create the augmented circuit.

The IAM is then written as (conductances in siemens [S])

$$\hat{\mathbf{Y}}_{\text{ind}} = \begin{bmatrix} 6+\hat{y}_S & -5 & -1 & -\hat{y}_S \\ g_m - 5 & 15.1 & -g_m - 10 & -0.1 \\ -g_m - 1 & -10 & g_m + 13 & -2 \\ -\hat{y}_S & -0.1 & -2 & \hat{y}_S + 2.1 \end{bmatrix}$$

Applying Equation 25.17 to extract $\hat{y}_S$ results in

$$\text{cof}(\hat{\mathbf{Y}}_{\text{ind}}) = \text{cof}\begin{bmatrix} 6 & -5 & -1 & 0 \\ g_m - 5 & 15.1 & -g_m - 10 & -0.1 \\ -g_m - 1 & -10 & g_m + 13 & -2 \\ 0 & -0.1 & -2 & 2.1 \end{bmatrix} + \hat{y}_S \text{cof}\begin{bmatrix} 8.1 & -5.1 & -3 \\ g_m - 5.1 & 15.1 & -g_m - 10 \\ -g_m - 3 & -10 & g_m + 13 \end{bmatrix}$$

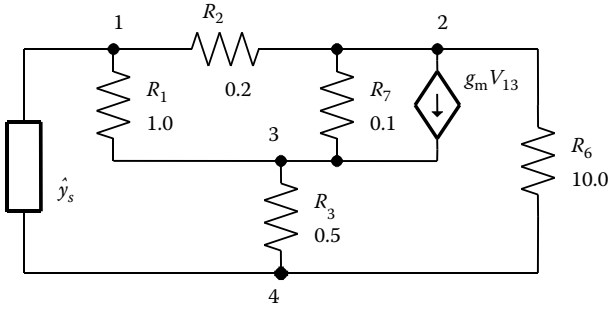


FIGURE 25.8 Circuit for the parameter extraction method (resistances in Ω).

Applying Equation 25.17 again to extract g_m yields

$$\text{cof}(\hat{\mathbf{Y}}_{\text{ind}}) = \text{cof}\begin{bmatrix} 6 & -5 & -1 & 0 \\ -5 & 15.1 & -10 & -0.1 \\ -1 & -10 & +13 & -2 \\ 0 & -0.1 & -2 & 2.1 \end{bmatrix} + g_m\ \text{cof}\begin{bmatrix} 5 & -5 & 0 \\ -3 & 5.1 & -2.1 \\ -2 & -0.1 & 2.1 \end{bmatrix}$$
$$+ \hat{y}_S\text{cof}\begin{bmatrix} 8.1 & -5.1 & -3 \\ -5.1 & 15.1 & -10 \\ -3 & -10 & 13 \end{bmatrix} + \hat{y}_S g_m\ \text{cof}\begin{bmatrix} 5.1 & -5.1 \\ -5.1 & 5.1 \end{bmatrix}$$

After evaluating the cofactors numerically, the equation reduces to

$$\text{cof}(\hat{\mathbf{Y}}_{\text{ind}}) = 137.7 + 10.5g_m + 96.3\hat{y}_S + 5.1\hat{y}_S g_m$$

From Equation 25.10 this results in

$$Z_{14} = \frac{\Delta_{11}}{\Delta} = \frac{96.3 + 5.1g_m}{137.7 + 10.5g_m}$$

25.3.3.2 Two Graph-Based Tableau Approach [Pie93]

This approach also employs the circuit augmentation by $\hat{y}_S$ and $\hat{g}_m V_o$, as in the tree enumeration method. It calls for a construction of two graphs: the voltage graph (G_V or V-graph) and the current graph (G_I or I-graph). For the purpose of parameter extraction (as well as generation of approximate symbolic expressions, Section 25.5), it is required that both graphs have the same number of nodes (n). This means that the method can be directly applied only to RLCg_m circuits. (All basic circuit components, including ideal op-amps, can be handled by this approach after some circuit transformations [Lin91], [Vla94]. For the sake of simplicity, however, only RLCg_m circuits will be considered in this presentation.) The two graphs are constructed based on the element stamps shown in Figure 25.9. Once the two graphs are constructed, a *common spanning tree* (i.e., a set of $n-1$ branches that form a spanning tree in both voltage and current graphs) is chosen. Choosing the common spanning tree (referred to just as ''tree'' in the remainder of this section) uniquely determines the cotree in each graph.

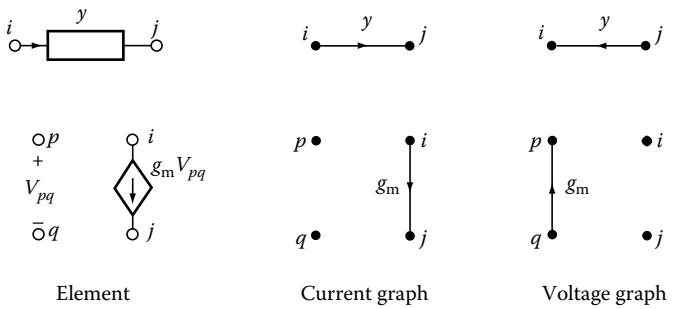


FIGURE 25.9 Element stamps for generating G_V and G_I.

The tableau equation for such a network can be written as

$$\mathbf{Hx} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & -\mathbf{Z}_T & \mathbf{0} \\ \mathbf{B}_T & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{Q}_C \\ \mathbf{0} & -\mathbf{Y}_C & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_T \\ \mathbf{V}_C \\ \mathbf{I}_T \\ \mathbf{I}_C \end{bmatrix} = \mathbf{0} \tag{25.18}$$

The first and last rows of the system matrix **H** in Equation 25.18 consists of tree ($\bullet_T$) and cotree ($\bullet_C$) branch voltage–current relationships and the second and third rows consist of fundamental loop and fundamental cut-set equations for G_V and G_I, respectively.

Let the circuit have n nodes and b branches and contain k symbolic components ($Y_{S_1}, \ldots, Y_{S_k}$) in the cotree branches (links) and l symbolic components ($Z_{S_1}, \ldots, Z_{S_l}$) in the tree branches; we define $w = b - n - k + 1$, $t = n - l - 1$. Diagonal matrices $\mathbf{Y}_C$ and $\mathbf{Z}_T$ can be partitioned as follows:

$$\mathbf{Y}_C = \begin{bmatrix} \mathbf{Y}_{Cs} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{Cn} \end{bmatrix}, \quad \mathbf{Z}_T = \begin{bmatrix} \mathbf{Z}_{Ts} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_{Tn} \end{bmatrix}$$

where subscript "s" denotes immitances of symbolic components and subscript "n" denotes immitances of components given numerically.

Matrices $\mathbf{B}_T$ (fundamental loop matrix in G_V) and $\mathbf{Q}_C$ (fundamental cut-set matrix in G_I) can also be partitioned as follows:

$$\mathbf{B}_T = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}, \quad \mathbf{Q}_C = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}$$

Rows of $\mathbf{B}_{11}$ and $\mathbf{B}_{12}$ correspond to symbolic cotree branches (in G_V) and their columns correspond to symbolic and numeric tree branches, respectively. Rows of $\mathbf{B}_{21}$ and $\mathbf{B}_{22}$ correspond to numeric cotree branches. Rows of $\mathbf{Q}_{11}$ and $\mathbf{Q}_{12}$ correspond to symbolic tree branches and their columns correspond to symbolic and numeric cotree branches, respectively. Rows of $\mathbf{Q}_{21}$ and $\mathbf{Q}_{22}$ correspond to numeric tree branches. The submatrices are therefore of the following order: $\mathbf{B}_{11}$: $k \times l$, $\mathbf{B}_{22}$: $w \times t$, $\mathbf{Q}_{11}$: $l \times k$, $\mathbf{Q}_{22}$: $t \times w$.

Let $S_x = \{1, 2, \ldots, x\}$. For a given matrix **F** of order $a \times b$, let $\mathbf{F}(I_u, J_v)$ be the submatrix of **F** consisting of the rows and columns corresponding to the integers in the sets I_u, J_v, respectively. The sets $I_u = \{i_1, i_2, \ldots, i_u\}$ and $J_v = \{j_1, j_2, \ldots, j_v\}$ are subsets of S_a and S_b, respectively. Let us also introduce the following notation:

$$\mathbf{1}_d^c = \text{diag}[\, e_1 \quad e_2 \quad \cdots \quad e_d \,]; \quad c < d$$

$$e_x = \begin{cases} 0 & \text{for } x \in \{1, 2, \ldots, c\} \\ 1 & \text{for } x \in \{c+1, c+2, \ldots, d\} \end{cases}$$

The determinant of the system matrix **H** in Equation 25.18, when some parameters take fixed numerical values, is

$$\begin{aligned} \det \mathbf{H} = a &+ \sum_{J_v} b(\alpha_v) Z_{S_{j1}} Z_{S_{j2}} \cdots Z_{S_{jv}} + \sum_{I_u} c(\beta_u) Y_{S_{i1}} Y_{S_{i2}} \cdots Y_{S_{iu}} \\ &+ \sum_{I_u} \sum_{J_v} d(\alpha_v \beta_u) Z_{S_{j1}} Z_{S_{j2}} \cdots Z_{S_{jv}} Y_{S_{i1}} Y_{S_{i2}} \cdots Y_{S_{iu}} \end{aligned} \tag{25.19}$$

where the summations are taken over all possible symbol combinations α_v (symbolic tree elements) and β_u (symbolic cotree elements), and the numerical coefficients are given by

$$
\begin{aligned}
\det H &= a + \sum_{J_v} b(\alpha_v) Z_{s_{j1}} Z_{s_{j2}} \cdots Z_{s_{jv}} + \sum_{I_u} c(\beta_u) Y_{s_{i1}} Y_{s_{i2}} \cdots Y_{s_{iu}} \\
&\quad + \sum_{I_u} \sum_{J_v} d(\alpha_v \beta_u) Z_{s_{j1}} Z_{s_{j2}} \cdots Z_{s_{jv}} Y_{s_{i1}} Y_{s_{i2}} \cdots Y_{s_{iu}} \\
a &= \det\left[\mathbf{1}_w + \mathbf{B}'_{22}\left(-\mathbf{Q}'_{22}\right)\right] = \det\left[\mathbf{1}_t + \left(-\mathbf{Q}'_{22}\right)\mathbf{B}'_{22}\right] \\
b(\alpha_v) &= \det\left(\mathbf{1}^v_{t+v} + \begin{bmatrix} -\mathbf{Q}_{12}(J_v, I_w) \\ -\mathbf{Q}'_{22} \end{bmatrix} \begin{bmatrix} \mathbf{B}'_{21}(I_w, J_v) & \mathbf{B}'_{22} \end{bmatrix}\right) \\
c(\beta_u) &= \det\left(\mathbf{1}^u_{w+u} + \begin{bmatrix} \mathbf{B}_{12}(I_u, J_t) \\ \mathbf{B}'_{22} \end{bmatrix} \begin{bmatrix} -\mathbf{Q}'_{21}(J_t, I_u) & -\mathbf{Q}'_{22} \end{bmatrix}\right) \\
d(\alpha_v \beta_u) &= \det\left(\mathbf{1}^v_{v+t+u} + \begin{bmatrix} -\mathbf{Q}_{11}(J_v, I_u) & -\mathbf{Q}_{12}(J_v, I_w) \\ -\mathbf{Q}'_{21}(J_t, I_u) & -\mathbf{Q}'_{22} \\ \mathbf{1}_u & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11}(I_u, J_v) & \mathbf{B}_{12}(I_u, J_t) & -\mathbf{1}_u \\ \mathbf{B}'_{21}(I_w, J_v) & \mathbf{B}'_{22} & \mathbf{0} \end{bmatrix}\right)
\end{aligned}
\tag{25.20}
$$

In the above equations, $\mathbf{0}$ represents a zero matrix of appropriate order, and the submatrices $\mathbf{B}'_{ij}$ and $\mathbf{Q}'_{ij}$ are defined as

$$
\begin{aligned}
\mathbf{B}'_{21}(I_w, J_v) &= \mathbf{Y}_{Cn}\mathbf{B}_{21}(I_w, J_v), \quad \mathbf{B}'_{22} = \mathbf{Y}_{Cn}\mathbf{B}_{22} \\
\mathbf{Q}'_{21}(J_t, I_u) &= \mathbf{Z}_{Tn}\mathbf{Q}_{21}(J_t, I_u), \quad \mathbf{Q}'_{22} = \mathbf{Z}_{Tn}\mathbf{Q}_{22}
\end{aligned}
\tag{25.21}
$$

where the submatrix $\mathbf{B}_{21}(I_w, J_v)$ is obtained from the submatrix $\mathbf{B}_{21}$ by including all of its rows and only columns corresponding to a particular combination (α_v) of symbolic tree elements; submatrix $\mathbf{Q}_{21}(J_t, I_u)$ is obtained from the submatrix $\mathbf{Q}_{21}$ by including all of its rows and only columns corresponding to a particular combination (β_u) of symbolic cotree elements.

Application of Equations 25.19 and 25.20 for a circuit with m symbolic parameters requires, theoretically, the calculation of 2^m determinants. Not all of these determinants may need to be calculated due to the following property of the determinants in Equation 25.20. If a set of symbolic tree elements (α_v) forms a cut-set in G_I (*symbolic tree cut-set*), then the corresponding coefficients $b(\alpha_v)$ and $d(\alpha_v\beta_u)$ in Equation 25.19 equal to zero. Likewise, if the set of symbolic cotree elements (β_u) forms a loop in G_V (*symbolic cotree loop*), the corresponding coefficients $c(\beta_u)$ and $d(\alpha_v\beta_u)$ in Equation 25.19 equal to zero.

Once the determinant det($\mathbf{H}$) is obtained from Equation 25.19, the sorting scheme, identical to that expressed in Equation 25.10, is applied and the required network functions can be calculated using Equations 25.7 through 25.9.

The main feature of this approach is the fact that each coefficient at a valid symbol combination is obtained directly by calculating a single, easily formulated determinant (a polynomial in s, in general case). The method was implemented in a computer program UTSSNAP [Pie98]. Example 25.4 illustrates this technique of parameter extraction.

Example 25.4

Consider again the circuit in Figure 25.8. Assume this time that two components, R_1 and g_m, are given symbolically. The goal is again to find the input impedance Z_{41} in a semisymbolic form using the parameter extraction method based on the two-graph tableau formulation.

The voltage and current graphs of the circuit are shown in Figure 25.10. The common spanning tree chosen is $T = \{R_1, R_2, R_3\}$ with one symbolic element. For this circuit we have $n = 4$, $b = 7$, $k = 2$, $l = 1$, $w = 2$, and $t = 2$.

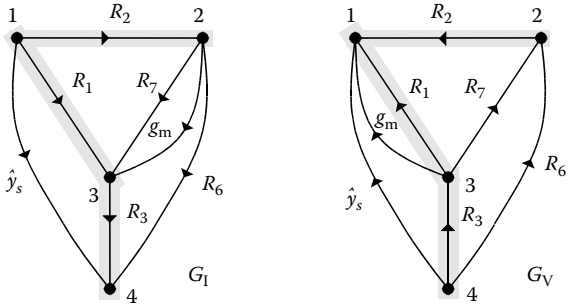


FIGURE 25.10 The current and voltage graphs for the circuit in Figure 25.8 with the common spanning tree highlighted.

The matrices, $\mathbf{Y}_C$, $\mathbf{Z}_T$, $\mathbf{Q}_C$, and $\mathbf{B}_T$, can now be determined as

$$\mathbf{Y}_C = \left[\begin{array}{cc|c} \hat{y}_S & & \\ & g_m & \\ \hline & & \begin{matrix} 0.1 & \\ & 10 \end{matrix} \end{array}\right] \quad \mathbf{Z}_T = \left[\begin{array}{c|cc} R_1 & & \\ \hline & 0.2 & \\ & & 0.5 \end{array}\right]$$

$$\mathbf{Q}_C = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 \\ \hline 1 & 0 & 1 & 0 \end{array}\right] \quad \mathbf{B}_T = \left[\begin{array}{c|cc} -1 & 0 & -1 \\ -1 & 0 & 0 \\ \hline -1 & 1 & -1 \\ -1 & 1 & 0 \end{array}\right]$$

Using Equation 25.21 we can calculate matrices $\mathbf{B}'_{22}$ and $\mathbf{Q}'_{22}$:

$$\mathbf{B}'_{22} = \begin{bmatrix} 0.1 & 0 \\ 0 & 10 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.1 & -0.1 \\ 10 & 0 \end{bmatrix}, \quad \mathbf{Q}'_{22} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.5 \end{bmatrix}\begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} -0.2 & -0.2 \\ 0.5 & 0 \end{bmatrix}$$

Now, applying Equation 25.20, the coefficient a in Equation 25.19 is calculated as

$$a = \det\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.1 & -0.1 \\ 10 & 0 \end{bmatrix}\begin{bmatrix} 0.2 & 0.2 \\ -0.5 & 0 \end{bmatrix}\right) = \det\begin{bmatrix} 1.07 & 0.02 \\ 2 & 3 \end{bmatrix} = 3.17$$

Since there is only one symbolic tree element, namely R_1, we have: $\boldsymbol{\alpha}_v = \{R_1\}$ and the associated sets: $J_v = \{1\}$, $I_w = \{1, 2\}$. Using Equation 25.21, we calculate

$$\mathbf{B}'_{21}(I_w, J_v) = \mathbf{Y}_{Cn}\mathbf{B}_{21}(I_w, J_v) = \begin{bmatrix} 0.1 & 0 \\ 0 & 10 \end{bmatrix}\begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.1 \\ -10 \end{bmatrix}$$

The coefficient $b(R_1)$ can be now obtained from

$$b(R_1) = \det\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \left[\begin{array}{cc} -1 & -1 \\ \hline 0.2 & 0.2 \\ -0.5 & 0 \end{array}\right]\begin{bmatrix} -0.1 & 0.1 & -0.1 \\ -10 & 10 & 0 \end{bmatrix}\right)$$

$$= \det\begin{bmatrix} 10.1 & -10.1 & 0.1 \\ -2.02 & 3.02 & -0.02 \\ 0.05 & -0.05 & 1.05 \end{bmatrix} = 10.6$$

Other numerical coefficients in Equation 25.19 are calculated in similar way:

$c(\hat{y}_S) = 1.51$, $c(g_m) = 0$, $c(\hat{y}_S g_m) = 0$, $d(R_1\hat{y}_S) = 8.12$, $d(R_1 g_m) = 1.05$, $d(R_1\hat{y}_S g_m) = 0.51$

Adding all terms, sorting according to Equation 25.10, and applying Equation 25.7 finally results in

$$Z_{41} = \frac{1.51 + 8.12R_1 + 0.51R_1 g_m}{3.17 + 10.6R_1 + 1.05R_1 g_m}$$

Matrices in Equation 25.20 may contain terms dependent on the complex frequency s. Determinants of such matrices are polynomials in s as long as all matrix elements are of the form: $a = \alpha + s\beta$. An interpolation method may be used to calculate the coefficients of those polynomials. One such method is briefly described in Section 25.3.4.

25.3.4 Interpolation Method

This method is best suited when s is the only symbolic variable. In such case, a transfer function has the rational form

$$H(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{m} a_i s^i}{\sum_{i=0}^{n} b_i s^i}$$

where $N(s)$ and $D(s)$ are polynomials in s with real coefficients and $m \leq n$.

Coefficients of an nth order polynomial

$$P(s) = \sum_{k=0}^{n} p_k s^k$$

can be obtained by calculating the value of $P(s)$ at $n+1$ distinct points s_i and then solving the following set of equations:

$$\begin{bmatrix} 1 & s_0 & s_0^2 & \cdots & s_0^n \\ 1 & s_1 & s_1^2 & \cdots & s_1^n \\ & \vdots & & & \\ 1 & s_n & s_n^2 & \cdots & s_n^n \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} P(s_0) \\ P(s_1) \\ \vdots \\ P(s_n) \end{bmatrix} \tag{25.22}$$

Since the matrix in Equation 25.22 is nonsingular, the unique solution exists. It is well known [Sin74], [Vla94] that for numerical accuracy and stability the best choice of the interpolation points is a set of $q \geq n+1$ points s_i uniformly spaced on the unit circle in the complex plane. Once all the values of $P(s_i)$ are known, the polynomial coefficients can be calculated through the discrete Fourier transform (DFT).

To apply this technique to the problem of finding a transfer function, let us assume that a circuit behavior is described by a linear equation

$$\mathbf{Ax} = \mathbf{b} \tag{25.23}$$

in which the coefficient matrix has entries of the form $a = \alpha + s\beta$ (both the modified nodal and the tableau methods have this property). Then, each transfer function of such circuit has the same denominator $D(s) = |\mathbf{A}|$. If the circuit Equation 25.23 is solved by LU factorization at $s = s_i$, both the transfer function $H(s_i)$ and its denominator $D(s_i)$ are obtained simultaneously. The value of the numerator is then

calculated simply as $N(s_i) = H(s_i)D(s_i)$. Repeating this process for all points s_i ($i = 0, 1, \ldots, q$) and then applying the DFT to both sets of values, $D(s_i)$ and $N(s_i)$, give the required coefficients of the numerator and denominator polynomials.

If the number of interpolation points is an integer power of 2 ($q = 2^k$), the method has the advantage that the fast Fourier transform (FFT) can be used to find the coefficients. This greatly enhances the execution time [Lin91]. The method has been extended to handle several symbolic variables in addition to s [Sin74]. The program implementation [Vla94] allows a maximum of five symbolic parameters in a circuit.

With the emergence of approximate symbolic analysis, the polynomial interpolation method has attracted new interest. (It is desirable to know the accurate numerical value of polynomial coefficients before one attempts an approximation.) Recently a new adaptive scaling mechanism was proposed [Fer98] that significantly increases the circuit size that can be handled accurately and efficiently.

There are other classifications of symbolic methods that have been reported [Gie91]. These methods can be considered as variations on the above basic four methods. The reported methods include elimination algorithms, recursive determinant-expansion algorithms, nonrecursive nested-minors method. All three are based on the use of Cramer's rule to find the determinant and the cofactors of a matrix. Another reported class of algorithms uses MNA [Ho75] as the basis of the analysis, sometime referred to as a direct network approach [Has89], [Lib93]. This class of methods is covered in Section 25.4.

The first generation of computer programs available for symbolic circuit simulation based on these methods includes NASAP [Mcn68] and SNAP [Lin70]. Research in the late 1980s and early 1990s has produced newer symbolic analysis programs. These programs include ISSAC [Gie89], SCAPP [Has89], ASAP [Fer91], EASY [Som91], SYNAP [Sed92], SAPEC [Man91], SAPWIN [Lib95], SCYMBAL [Kon88], GASCAP [Hue89], SSPICE [Wie89], and STAINS [Pie01].

25.4 Hierarchical Methods (Sequence of Expressions)

All of the methods presented in Section 25.3 have circuit size limitations. The main problem is the exponential growth of the number of symbolic terms involved in the expression for the transfer function in Equation 25.3 as the circuit gets larger. The solution to analyze large-scale circuits lies in a total departure from the traditional procedure of trying to state the transfer function as a single expression and using a SoE procedure instead. The idea is to produce a succession of small expressions with a backward hierarchical dependency on each other. The growth of the number of expressions in this case will be, at worse case, quadratic [Has89].

The advantage of having the transfer function stated in a single expression lies in the ability to gain insight to the relationship between the transfer function and the network elements by inspection [Lin73]. For large expressions, though, this is not possible and the single expression loses that advantage. ISSAC [Wam92], ASAP [Fer92], SYNAP [Sed92], and Analog Insydes [Hen00] attempt to handle larger circuits by maintaining the single expression method and using circuit-dependent approximation techniques. The trade-off is accuracy for insight. Therefore, the SoE approach is more suitable for accurately handling large-scale circuits. Example 25.5 illustrates the features of the SoE.

Example 25.5

Consider the resistance ladder network in Figure 25.11.

The goal is to obtain the input impedance function of the network, $Z_{in} = V_{in}/I_{in}$. The single expression transfer function Z_4 is

$$Z_4 = \frac{R_1R_3 + R_1R_4 + R_2R_3 + R_2R_4 + R_3R_4}{R_1 + R_2 + R_3}$$

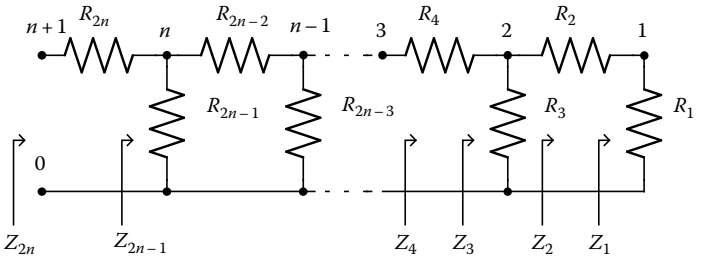


FIGURE 25.11 Resistance ladder network.

The number of terms in the numerator and denominator is given by the Fibonacci numbers satisfying the following difference equation:

$$y_{k+2} = y_{k+1} + y_k,\ k = 0, 1, 2, \ldots; \quad y_0 = 0; \quad y_1 = 1$$

An explicit solution to the above equation is

$$y_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n \approx 0.168 \cdot 1.618^n \text{ for large } n$$

The solution shows that the number of terms in Z_n increases exponentially with n. Any single expression transfer function has this inherent limitation.

Now using the SoE procedure the input impedance can be obtained from the following expressions:

$$Z_1 = R_1; \quad Z_2 = Z_1 + R_2; \quad Z_3 = \frac{Z_2 R_3}{Z_2 + R_3}; \quad Z_4 = Z_3 + R_4$$

It is obvious for each additional resistance added, the SoE will grow by one expression, either of the form $Z_{i-1} + R_i$ or $Z_{i-1}R_i/(Z_{i-1} + R_i)$. The number of terms in the SoE can be calculated from the formula:

$$y_n = \begin{cases} 2.5n - 2 & \text{for } n \text{ even} \\ 2.5n - 1.5 & \text{for } n \text{ odd} \end{cases}$$

which exhibits a linear growth with respect to n. Therefore, to find the input impedance of a 100 resistor ladder network the single expression methods would produce 7.9×10^{20} terms which requires unrealistically huge computer storage capabilities. On the other hand, the SoE method would produce only 248 terms, which is even within the scope of some desk calculators.

Another advantage of the SoE is the number of arithmetic operations needed to evaluate the transfer function. To evaluate Z_9, for example, the single expression methods would require 302 multiplications and 87 additions. The SoE method would only require eight multiplications and eight additions, a large reduction in computer evaluation time. All this makes the concept of symbolic circuit simulation of large-scale networks very possible.

Two topological analysis methods for symbolic simulation of large-scale circuits have been proposed in [Sta86] and [Has93]. The first method utilizes the SoE idea to obtain the transfer functions. The method operates on the Coates graph [Coa59] representing the circuit. A partitioning is proposed onto the flowgraph and not the physical network. The second method also utilizes the SoE and a Mason's signal flowgraph [Mas56] representation of the circuit. The method makes use of partitioning on the physical level rather than on the graph level. Therefore, for a hierarchical circuit, the method can operate on the subcircuits in a hierarchical fashion in order to produce a final solution. The fundamentals of both signal flowgraph methods were described in the Section 25.3.2.

Another hierarchical approach is one that is based on MNA [Ho75]. This method [Has89] exhibits a linear growth (for practical circuits) in the number of terms in the symbolic solutions. The analysis methodology introduces the concept of the reduced modified nodal analysis (RMNA) matrix. This allows the characterization of symbolic circuits in terms of only a small subset of the network variables (external variables) rather than the complete set of variables. The method was made even more effective by introducing a locally optimal pivot selection scheme during the reduction process [Pie01]. For a circuit containing several identical* subcircuits, the analysis algorithm is most efficient when network partitioning is used. For other circuits, the best results (the most compact SoE) are obtained when the entire circuit is analyzed without partitioning.

The SoE generation process starts with the formulation of a symbolic modified NAM (MNAM) for a circuit [Lin91], [Vla94]. Then all internal variables are suppressed one by one using Gaussian elimination with locally optimal pivot selection. Each elimination step produces a series of expressions and modifies some entries in the remaining portion of the MNAM. When all internal variables are suppressed, the resulting matrix is known as the reduced MNAM (RMNAM). Usually it will be a 2×2 matrix of a two-port.† Most transfer functions of interest to circuit designer can be represented by formulas involving the elements of RMNAM and the terminating admittances. A detailed discussion of the method can be found in [Pie01]. Based on this approach a personal computer program STAINS has been developed [Hue02].

For a circuit with several identical subcircuits the reduction process is first applied to all internal variables‡ of the subcircuit, resulting in an intermediate RMNAM describing the subcircuit. Those RMNAMs are then recombined with the MNAM of the remaining circuit and the reduction process is repeated on the resulting matrix. To further illustrate the SoE approach, we present Example 25.6.

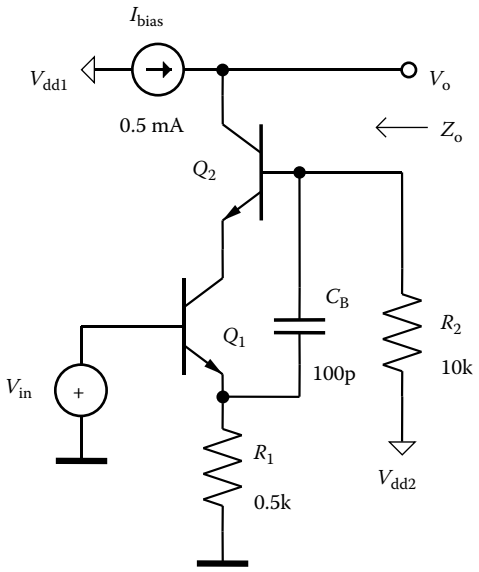


FIGURE 25.12 Bipolar cascode stage.

Example 25.6

Consider a bipolar cascode stage with bootstrap capacitor C_B, shown in Figure 25.12 [Gie91]. With the BJTs replaced by their low-frequency hybrid-π models (with r_π, g_m, and r_o only), the full symbolic analysis yields the output admittance formula shown in Figure 25.13. The formula requires 48 additions and 117 multiplication/division operations. Program STAINS can generate several different sequences of expressions. One of them is shown in Figure 25.14. It requires only 24 additions and 17 multiplications/divisions.§

Although SoE techniques can produce very compact exact expressions for network functions, they are of limited use if one wishes to obtain symbolic expressions for coefficients at individual powers of s in Equation 25.4. Sections 25.5 and 25.6 describe two possible approaches to this problem.

* The subcircuits have to be truly identical, that is, they must have the same topology and component symbols. A typical example would be a large active filter containing a number of identical, nonideal op-amps.

† In sensitivity calculations using SoE [Bal04] the final RMNAM may need to be larger than 2×2.

‡ The internal variables are the variables not directly associated with the subcircuit's connections to the rest of the circuit.

§ Counting of visible arithmetic operations gives only a rough estimate of the SoE complexity, especially when complex numbers are involved. Issues related to SoE computational efficiency are discussed in [Rod00].

```
!
Zo=(G2*Gm1*Gm2+G1*G2*Gm1+G2*Gm2*Gp1+G2*Gm1*Gp2+Gm2*Go1*Gp1+G1*G2*Gp1+...
    G2*Gm2*Go1+G2*Gm1*Go2+G1*Go2*Gp1+G2*Gp1*Gp2+G1*Go1*Gp1+G1*G2*Go2+...
    G1*G2*Go1+Go2*Gp1*Gp2+Go1*Gp1*Gp2+G2*Go2*Gp1+G2*Go1*Gp2+G2*Go2*Gp2+...
    Go1*Go2*Gp1+G2*Go1*Go2+s*(Cb*Gm1*Gm2+Cb*G1*Gm1+Cb*Gm1*Gp2+Cb*G2*Gm1+...
    Cb*G1*Gp1+Cb*Gm2*Go1+Cb*Gp1*Gp2+Cb*G1*Go2+Cb*G2*Gp1+Cb*G1*Go1+...
    Cb*Go1*Gp2+Cb*Go2*Gp2+Cb*Go1*Gp1+Cb*G2*Go1+Cb*G2*Go2+Cb*Go1*Go2))/...
    (Go1*G2*Gm2*Gp1+Go1*G1*G2*Gp1+Go1*G2*Gp1*Gp2+Go1*G1*Go2*Gp1+...
    Go1*G1*G2*Go2+Go1*Go2*Gp1*Gp2+Go1*G2*Go2*Gp2+Go1*G2*Gp2*Gp1+...
    s*(Cb*Go1*G1*Gp1+Cb*Go1*Gp1*Gp2+Cb*Go1*G2*Gp1+Cb*Go1*G1*Go2+...
    Cb*Go1*Go2*Gp2+Cb*Go1*G2*Go2));
!
```

FIGURE 25.13 Full symbolic expression for Z_o of the cascode in Figure 25.12.

25.5 Approximate Symbolic Analysis

```
d1 = -(G2+Gp2+s*Cb)/(s*Cb);
x1 =  (Go1+Gm1)*d1-Gp2-Gm2;
x2 = -s*Cb-(G1+Gp1+Go1+Gm1+s*Cb)*d1;
d2 = Gp2/(s*Cb);
x3 = Go1+Gp2+Go2+Gm2+(Go1+Gm1)*d2;
x4 = -Go1-(G1+Gp1+Go1+Gm1+s*Cb)*d2;
d3 = x2/(x4);
x5 = Gm2+(Go2+Gm2)*d3;
x6 = x1-x3*d3;
Yo = Go2+x5*Go2/(x6);
Zo = 1/Yo;
```

FIGURE 25.14 The SoE generated by STAINS for the cascode in Figure 25.12.

The SoE approach offers a solution for the exact symbolic analysis of large circuits. For some applications, it may be more important to obtain a simpler inexact expression, but the one that would clearly identify the dominant circuit components and their role in determining circuit behavior. Approximate symbolic analysis provides the answer. Of course, manual approximation (simplification) techniques have been known and practiced by engineers for decades. To obtain compact and meaningful expressions by computer, symbolic analysis software must be capable of performing those approximations that are applied in manual circuit analysis in an automatic fashion. In addition to that, computer algorithms should be able to employ simplification strategies not available (or impractical) in manual approximation.

In the last decade, a number of symbolic approximation algorithms have been developed and implemented in symbolic circuit analysis programs. Depending on the stage in the circuit analysis process in which they are applied, these algorithms can be categorized as *Simplification before Generation* (SBG), *Simplification during Generation* (SDG), and *Simplification after Generation* (SAG). Figure 25.15, adapted from [Hen00], presents an overview of the three types of approximation algorithms.

SBG involves removing circuit components and/or individual entries in the circuit matrix (the *sifting* approach [Hsu93]) or eliminating some graph branches (the *sensitivity-based two-graph simplification* [Yu96]) that do not contribute significantly to the final formula.

SDG is based on generation of symbolic terms in a decreasing order of magnitude. The generation process is stopped when the error reaches the specified level. The most successful approach to date is based on the two graph formulation [Wam98]. It employs an algorithm to generate the common spanning trees in strictly decreasing order of magnitude [Kat81]. In the case of frequency dependent circuit, this procedure is applied separately to different powers of s. Mathematical formalism of *matroids* is well suited to describe problems of SDG [Yu96]. Matroid-based algorithms are inherently slow. More efficient methods for generating s-expanded network functions, based on determinant decision diagrams (DDDs), have been recently developed [Shi00], [Shi01], [Ver02]. They will be briefly described in Section 25.6.

When applied alone, SAG is a very ineffective technique, as it requires generation and storage of a large number of unnecessary terms. When combined with SBG and SDG methods, however, it can

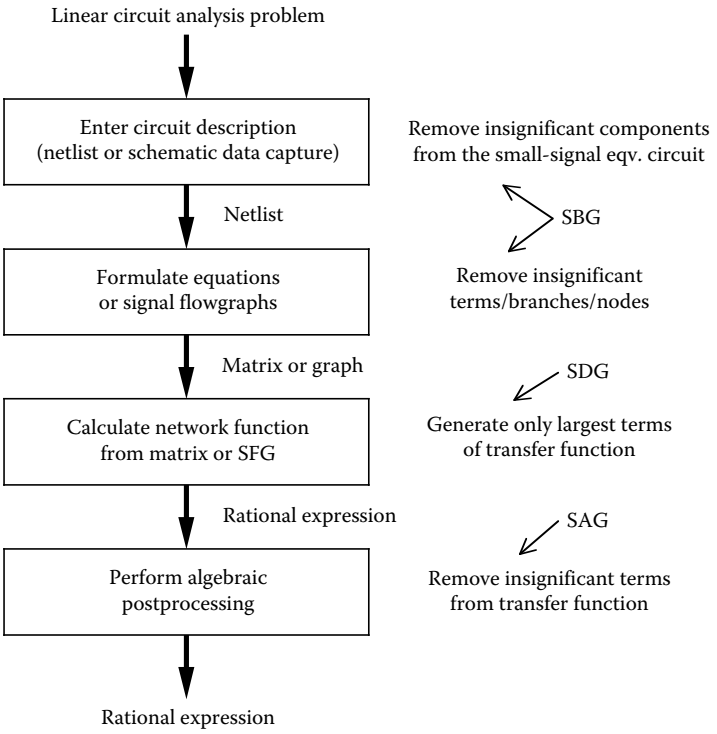


FIGURE 25.15 Classification of symbolic approximation techniques (From Henning, E., Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits, Doctoral dissertation, University of Kaiserslautern, Shaker Verlag, Aachen, Germany, 2000. With permission.)

produce most compact expressions by pruning redundant terms not detected earlier in the simplification process.

All simplification techniques require careful monitoring of the approximation amplitude and phase errors (ε_A and ε_p). The error criteria can be expressed as follows:

$$\left|\frac{|H(s,\mathbf{x})| - |H^{\star}(s,\mathbf{x})|}{|H(s,\mathbf{x})|}\right| \leq \varepsilon_A$$
$$|\angle H(s,\mathbf{x}) - \angle H^{\star}(s,\mathbf{x})| \leq \varepsilon_p$$

for $s = j\omega$, $\omega \in (\omega_1, \omega_2)$, $\mathbf{x} \in (\mathbf{x}_1, \mathbf{x}_2)$, where $H(s,\mathbf{x})$ is the exact transfer function, defined by Equation 25.3, and $H^{\star}(s,\mathbf{x})$ is the approximating function. The majority of the approximation methods developed to date use the simplified criteria, where the errors are measured only for a given set of circuit parameters $\mathbf{x}^0$ (the nominal design point) [Kon99].

Example 25.7, although quite simple, illustrates very well the advantages of approximate symbolic analysis.

Example 25.7: [Gie91]

Consider again the bipolar cascode stage, shown in Figure 25.12 and its fully symbolic expression for the output impedance, shown in Figure 25.13. Even for such a simple circuit, the full symbolic result is very hard to interpret and therefore not able to provide insight into the circuit behavior. Sequential form of the output impedance formula, presented in Figure 25.14, is more compact than the full expression but also cannot be utilized for interpretation.

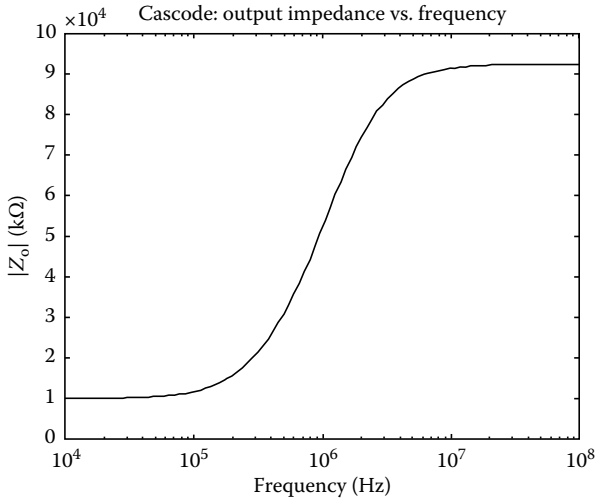


FIGURE 25.16 Plot of $|Z_o|$ of the cascode, obtained numerically from the exact formula.

A graph of $|Z_o|$ for a nominal set of component values ($r_\pi = 5$ kΩ, $g_m = 20$ mS, $r_o = 100$ kΩ for both BJTs), obtained numerically from the SoE in Figure 25.14, is plotted in Figure 25.16.

By examining the graph, one can appreciate the general behavior of the function but it is difficult to predict the influence of various circuit components on the output impedance.

Applying symbolic approximation techniques, we can obtain less accurate but still more revealing formulas. If a 10% maximum amplitude error is accepted, the simplified function takes the following form:

$$Z_{o(10\%)} = \frac{g_{m1}(g_{m2} + G_1)(G_2 + sC_B)}{g_{o1}g_{\pi 1}[G_2(g_{m2} + G_1) + sC_B(G_1 + g_{\pi 2})]}$$

If we allow a 25% magnitude error,* the output impedance formula can be simplified further:

$$Z_{o(25\%)} = \frac{g_{m1}g_{m2}(G_2 + sC_B)}{g_{o1}g_{\pi 1}(G_2 g_{m2} + sC_B G_1)} \tag{25.24}$$

The impedance levels as well as pole and zero estimates can be easily obtained from Equation 25.24:

$$\begin{aligned} Z_o(\text{low } f) &\cong \frac{g_{m1}}{g_{\pi 1}g_{o1}} = \frac{\beta_1}{g_{o1}} \\ Z_o(\text{high } f) &\cong \frac{g_{m1}g_{m2}}{g_{\pi 1}g_{o1}G_1} = \frac{g_{m2}}{G_1} Z_o(\text{low } f) \\ z &\cong -\frac{G_2}{C_B} \\ p &\cong -\frac{g_{m2}G_2}{G_1 C_B} \end{aligned} \tag{25.25}$$

An asymptotic graph of $|Z_o|$, based on Equation 25.25, is plotted in Figure 25.17.

* It is important to note that the approximate expressions were developed taking into account variations of BJT parameters; the fact that both simplified formulas give identical results at the nominal design point is purely coincidental.

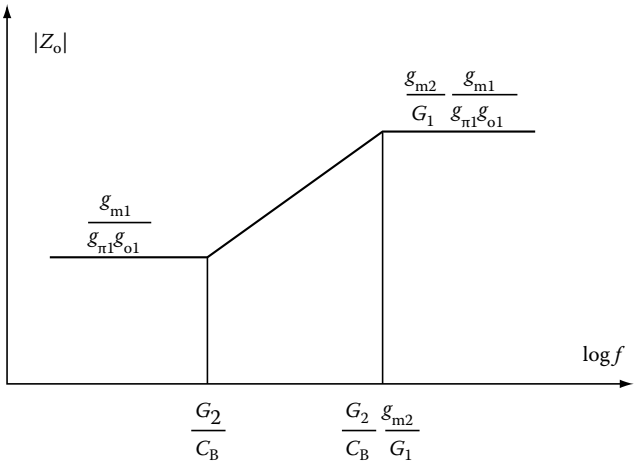
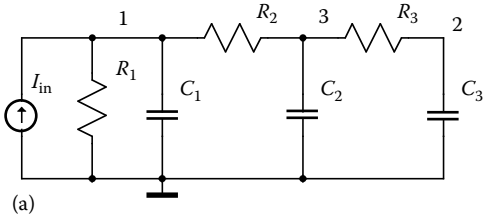


FIGURE 25.17 Asymptotic plot of $|Z_o|$ of the cascode based on Equation 25.25.

25.6 Methods Based on Determinant Decision Diagrams [Shi00], [Shi01], [Ver02]

To fully exploit the power of symbolic circuit analysis in helping designers gain insight into the circuit behavior, network functions in the *s*-expanded symbolic form are often required, as illustrated in Section 25.5. However, the approximate expressions may not be adequate for complete circuit characterization such as symbolic pole-zero derivation and sensitivity computation [Gie94]. Although there exists a mathematically elegant method for generating all terms at a given power of *s* in Equation 25.4 [Yu96], its practical applications are severely limited due to exponential dependence of time and storage complexity on circuit size.

Recently, a method was developed that employs DDDs to compactly represent the determinant of a circuit matrix and its cofactors [Shi00]. In order to represent individual coefficients of the powers of *s*, a notion of *multiroot DDDs* was introduced in [Shi01]. In order to introduce the DDD method we will resort again to a simple circuit example.



(a)

$$\begin{bmatrix} G_1+G_2+sC_1 & 0 & -G_2 \\ 0 & G_3+sC_3 & -G_3 \\ -G_2 & -G_3 & G_2+G_3+sC_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I_{in} \\ 0 \\ 0 \end{bmatrix}$$

(b)

FIGURE 25.18 An example circuit (a) and its node equation (b) (Note: $G_i = 1/R_i$).

Example 25.8

Consider a circuit in Figure 25.18a. Using nodal analysis (e.g., [Rod08]), we can formulate the system of node equations for this circuit, $\mathbf{Y}\cdot\mathbf{V} = \mathbf{J}$, as shown in Figure 25.18b.

Let us denote each entry of the nodal admittance matrix $\mathbf{Y}$ in Figure 25.18b by a distinct symbol as follows: $A = G_1 + G_2 + sC_1$, $B = -G_2$, $C = G_3 + sC_3$, $D = -G_3$, $E = -G_2$, $F = -G_3$, $G = G_2 + G_3 + sC_2$. The input impedance (as seen by the current source, I_{in}) is given by the expression (cf. Equation 25.6):

$$Z_{in} = \frac{V_1}{I_{in}} = \frac{\Delta_{11}}{\Delta} = \frac{GC - FD}{AGC - AFD - BEC} \tag{25.26}$$

A DDD can be used to represent the determinant of a matrix and its cofactors. Formally, a DDD is a signed, directed, acyclic graph with two terminal vertices, namely, the *zero-terminal* vertex and the *one-terminal* vertex. Each nonterminal vertex D is labeled by a symbol *D.label* and a positive or negative sign *D.sign*. It originates two outgoing edges, called *one-edge* (represented by a solid line in Figure 25.19) and *zero-edge* (represented by a broken line in Figure 25.19) pointing to its two children *D.child*1 and *D.child*0, respectively. Each vertex D represents a symbolic expression *D.expr*, defined recursively as follows:

1. If D is the one-terminal vertex, then $D.expr = 1$.
2. If D is the zero-terminal vertex, then $D.expr = 0$.
3. If D is a nonterminal vertex, then $D.expr = D.sign^{*}D.label^{*}(D.child1)expr + (D.child0)expr$, where $(D.child1)expr$ and $(D.child0)expr$ represent symbolic expressions corresponding to the vertices *D.child*1 and *D.child*0, respectively.

For example, it can be verified that Figure 25.19 is a DDD representation of $\Delta = \det(\mathbf{Y})$.

Given a vertex, a *one-path* is a path from that vertex to the one-terminal. A one-path represents a product of symbols that are labels of those vertices that *originate* all the one-edges along the one-path. For example, in Figure 25.19 there exist three one-paths from the root vertex labeled by A. They represent three product terms: AGC, $A(-F)D$, and $(-B)EC$. The root vertex A represents the sum of these product terms, being the determinant of $\mathbf{Y}$.

In order to achieve certain desired properties of a DDD, each vertex D is assigned an index *D.index*. Indices are chosen to be consecutive integers, starting from one and satisfying the following inequalities:

$$D.index > (D.child1).index$$

$$D.index > (D.child0).index$$

The process of assigning the indices is called *vertex ordering*. For example, the DDD in Figure 25.19 is formulated with a vertex order A, B, E, G, F, D, C (a number next to each vertex is its corresponding index).

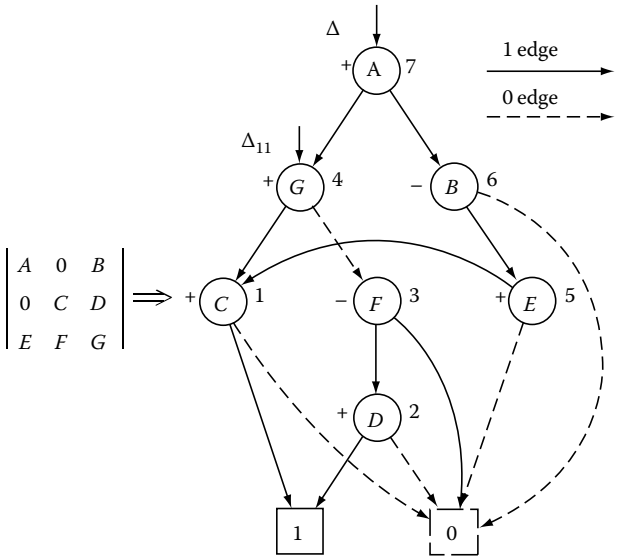


FIGURE 25.19 Example of matrix determinant and its DDD representation.

A network function is always a ratio of some cofactors (or the determinant and a cofactor) of a circuit matrix. To represent a network function using a DDD approach, the DDDs for the determinant and cofactors are constructed using the same vertex ordering. This results in sharing all common subgraphs (thus sharing all common subexpressions). This leads to one *shared DDD with multiple roots*, where each root represents either the determinant or a cofactor of the circuit matrix. For example, Figure 25.19 is a shared representation of Δ and Δ_{11} required in Equation 25.26. A *root* can be a vertex with no incoming edges or any vertex representing the cofactor of interest. The roots are marked by an additional single-ended incoming edge on the diagram.

The power of DDD-based approach lies in the fact that multiple roots share their common subgraphs. Thus, DDDs are capable of representing the determinants and cofactors of circuit matrices with the number of vertices of many orders of magnitude smaller than that of the product terms. Moreover, most symbolic analysis algorithms can be performed on a DDD with time complexity linear with the size of a DDD (i.e., the number of its vertices).

The DDD technique was originally developed for matrices with unique entries. When it is applied to the NAM of a circuit, the term cancellation occurs, dramatically slowing down the algorithm (by generating a large number of terms unnecessarily). Considerable improvement was achieved by introduction of the *just-in-time decancellation* [Ver02]. Other modifications to the DDD method were also reported in [Ver02], making the DDD-based approach very effective in both exact and approximate symbolic analyses.

The data structures and algorithms for creating DDDs and using them to obtain transfer functions are too complex to be presented in this short section. Interested readers should consult the literature, referenced here.

25.7 Time Domain Analysis

Sections 25.2 through 25.6 have discussed the different frequency domain techniques for symbolic analysis. Symbolic analysis methods in the transient domain have not appeared until the beginning of the 1990s [Has91], [Als93], [Gre93], [Lib93]. The main limitation to symbolic time domain analysis is the difficulty in handling the symbolic integration and differentiation needed to handle the energy storage elements (mainly capacitors and inductors). This problem, of course, does not exist in the frequency domain because of the use of Laplace transforms to represent these elements. While there exist symbolic algebra software packages that can be used to perform integration and differentiations like MATHMATICA, MAXIMA, and MAPLE, they have not been applied to transient symbolic analysis due to the execution time complexity of these programs. All but one of the approaches in the time domain is actually semisymbolic. The semisymbolic algorithms use a mixture of symbolic and numeric techniques to perform the analysis. The work here is still in its infancy. This section will lightly discuss the three contributions published in the literature so far.

All symbolic time domain techniques deal with linear circuits and can be classified under one of the two categories.

25.7.1 Fully Symbolic

Only one method has been reported in the literature that is fully symbolic [Gre94]. This method utilizes a direct and hierarchical symbolic transient analysis approach similar to the one reported in [Has89]. The formulation is based on the well-known discrete models for numerical integration of linear differential equations. Three of these integration methods are implemented symbolically, the Backward Euler, the Trapezoidal, and Gear's 2nd order backward differentiation [Gre94]. The inherent accuracy problems due to the approximations in these methods show up when the symbolic expressions are evaluated numerically. A detailed discussion of this method can be found in [Gre94].

25.7.2 Semisymbolic

Three such algorithms have been reported in the literature so far. Two of them [Lib93], [Has91] simply take the symbolic expressions in the frequency domain, evaluate them numerically for a range of frequencies and then perform a numeric inverse Laplace transformation or a FFT on the results. The approach reported in [Lib93] uses an MNA then a state-variable symbolic formulation to get the frequency domain response and can handle time-varying circuits, namely switch power converters. The approach in [Has91] uses a hierarchical network approach [Has89] to generate the symbolic frequency domain response. The third algorithm reported in [Als93] is a hierarchical approach that uses an MNA and a state-variable symbolic formulation and then uses the eigenvalues of the system to find a closed-form numerical transient solution.

References

[Ald70] G.E. Alderson, P.M. Lin, Integrating topological and numerical methods for semi-symbolic network analysis, *Proceedings of the 13th Midwest Symposium on Circuit Theory*, Minneapolis, MN, May 1970.

[Ald73] G.E. Alderson, P.M. Lin, Computer generation of symbolic network functions—A new theory and implementation, *IEEE Transactions on Circuit Theory*, 20(1), 48–56, Jan. 1973.

[Als93] B. Alspaugh, M. Hassoun, A mixed symbolic and numeric method for closed-form transient analysis, *Proceedings of ECCTD*, Davos, Switzerland, pp. 1687–1692, Sep. 1993.

[Arn91] Z. Arnautovic, P.M. Lin, Symbolic analysis of mixed continuous and sampled data systems, *Proceedings of IEEE ISCAS*, Singapore, pp. 798–801, June 1991.

[Bal04] F. Balik, B. Rodanski, Calculation of symbolic sensitivities for large-scale circuits in the sequence of expressions form via the transimpedance method, *Analog Integrated Circuits and Signal Processing*, 40(3), 265–276, Sep. 2004.

[Cal65] D.A. Calahan, Linear network analysis and realization—Digital computer programs and instruction manual, *University of Illinois Bulletin*, 62, Feb. 1965.

[Chu75] L.O. Chua, P.M. Lin, *Computer Aided Analysis of Electronic Circuits—Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[Coa59] C.L. Coates, Flow graph solutions of linear algebraic equations, *IRE Transactions on Circuit Theory*, CT-6, 170–187, June 1959.

[Con96] F. Constantinescu, M. Nitescu, Computation of symbolic pole/zero expressions for analog circuit design, *Proceedings of SMACD*, Haverlee, Belgium, Oct. 1996.

[Did91] G. DiDomenico et al., BRAINS: A symbolic solver for electronic circuits, *Proceedings of SMACD*, Paris, Oct. 1991.

[Dro96] G. Dröge, E.H. Horneber, Symbolic calculation of poles and zeros, *Proceedings of SMACD*, Haverlee, Belgium, Oct. 1996.

[Fer91] F.V. Fernandez, A. Rodriguez-Vazquez, J.L. Huertas, An advanced symbolic analyzer for the automatic generation of analog circuit design equations, *Proceedings of IEEE ISCAS*, Singapore, pp. 810–813, June 1991.

[Fer92] F.V. Fernandez et al., On simplification techniques for symbolic analysis of analog integrated circuits, *Proceedings of IEEE ISCAS*, San Diego, CA, pp. 1149–1152, May 1992.

[Fer98] F.V. Fernandez et al., Symbolic analysis of large analog integrated circuits: The numerical reference generation problem, *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 45(10), 1351–1361, Oct. 1998.

[Fid73] J.K. Fidler, J.I. Sewell, Symbolic analysis for computer-aided circuit design—The interpolative approach, *IEEE Transaction on Circuit Theory*, 20(6), 738–741, Nov. 1973.

[Gat70] T.F. Gatts, N.R. Malik, Topoloigical Analysis Program for Linear Active Networks (TAPLAN), *Proceedings of the 13th Midwest Symposium on Circuit Theory*, Minneapolis, MN, May 1970.

[Gie89] G. Gielen, H. Walscharts, W. Sansen, ISSAC: A symbolic simulator for analog integrated circuits, *IEEE Journal of Solid-State Circuits*, 25(6), 1587–1597, Dec. 1989.

[Gie91] G. Gielen, W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Boston, MA: Kluwer Academic, 1991.

[Gie94] G. Gielen, P. Wambacq, W. Sansen, Symbolic analysis methods and applications for analog circuits: A tutorial overview, *Proceedings of IEEE*, 82(2), 287–304, Feb. 1994.

[Gre93] S. Greenfield, *Transient Analysis for Symbolic Simulation*, MS thesis, Iowa State University, Dec. 1993.

[Gre94] S. Greenfield, M. Hassoun, Direct hierarchical symbolic transient analysis of linear circuits, *Proceedings of IEEE ISCAS*, London, England, May/June 1994.

[Hac71] G.D. Hachtel et al., The sparse tableau approach to network and design, *IEEE Transactions on Circuit Theory*, 18(1), 101–113, Jan. 1971.

[Has89] M.M. Hassoun, P.M. Lin, A new network approach to symbolic simulation of large-scale networks, *Proceedings of IEEE ISCAS*, Portland, OR, pp. 806–809, May 1989.

[Has90] M.M. Hassoun, P.M. Lin, An efficient partitioning algorithm for large-scale circuits, *Proceedings of IEEE ISCAS*, New Orleans, LA, pp. 2405–2408, May 1990.

[Has91] M.M. Hassoun, J.E. Ackerman, Symbolic simulation of large scale circuits in both frequency and time domains, *Proceedings of IEEE MWSCAS*, Calgary, Canada, pp. 707–710, Aug. 1990.

[Has93] M. Hassoun, K. McCarville, Symbolic analysis of large-scale networks using a hierarchical signal flowgraph approach, *Journal of Analog VLSI and Signal Processing*, 3(1), 31–42, Jan. 1993.

[Hen00] E. Henning, *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits*. Doctoral dissertation, University of Kaiserslautern. Aachen: Shaker Verlag, 2000.

[Ho75] C. Ho, A.E. Ruehli, P.A. Brennan, The modified nodal approach to network analysis, *IEEE Transactions on Circuits and Systems*, 22(6), 504–509, June 1975.

[Hsu93] J.J. Hsu, C. Sechen, Low-frequency symbolic analysis of large analog integrated circuits, *Proceedings of CICC*, San Diego, CA, pp. 14.7.1–14.7.4, May 1993.

[Hue89] L. Huelsman, Personal computer symbolic analysis programs for undergraduate engineering courses, *Proceedings of IEEE ISCAS*, Portland, OR, pp. 798–801, May 1989.

[Hue02] L. Huelsman, STAINS-Symbolic two-port analysis via internal node suppression, *IEEE Circuits and Devices Magazine*, 18(2), 3–6, Mar. 2002.

[Kat81] N. Katoh, T. Ibaraki, H. Mine, An algorithm for finding k minimum spanning trees, *SIAM Journal of Computer*, 10(2), 247–255, May 1981.

[Kon88] A. Konczykowska, M. Bon, Automated design software for switched capacitor ICs with symbolic simulator SCYMBAL, *Proceedings of DAC*, Anaheim, CA, pp. 363–368, June 1988.

[Kon92] A. Konczykowska et al., Symbolic analysis as a tool for circuit optimization, *Proceedings of IEEE ISCAS*, San Diego, CA, pp. 1161–1164, May 1992.

[Kon99] A. Konczykowska, Symbolic circuit analysis, *Wiley Encyclopedia of Electrical and Electronics Engineering*, J.G. Webster, Ed. New York: Wiley, 1999.

[Lee92] J. Lee, R. Rohrer, AWEsymbolic: Compiled analysis of linear(ized) circuits using asymptotic waveform evaluation, *Proceedings of DAC*, Anaheim, CA, pp. 213–218, June 1992.

[Li92] B. Li, D. Gu, SSCNAP: A program for symbolic analysis of switched capacitor circuits, *IEEE Transactions on CAD*, 11(3), 334–340, Mar. 1992.

[Lib93] A. Liberatore et al., Simulation of switching power converters using symbolic techniques, *Alta Frequenza*, 5(6), 16–23, Nov. 1993.

[Lib95] A. Liberatore et al., A new symbolic program package for the interactive design of analog circuits, *Proceedings of IEEE ISCAS*, Seattle, WA, pp. 2209–2212, May 1995.

[Lin70] P.M. Lin, G.E. Alderson, SNAP—A computer program for generating symbolic network functions, School of EE, Purdue University, West Lafayette, IN., Rep. TR-EE 70–16, Aug. 1970.

[Lin73] P.M. Lin, A survey of applications of symbolic network functions, *IEEE Transactions on Circuit Theory*, 20(6), 732–737, Nov. 1973.

[Lin91] P.M. Lin, *Symbolic Network Analysis*. Amsterdam: Elsevier Science, 1991.

[Lin92] P.M. Lin, Sensitivity analysis of large linear networks using symbolic programs, *Proceedings of IEEE ISCAS*, San Diego, CA, pp. 1145–1148, May 1992.

[Man72] V.K. Manaktala, G.L. Kelly, On the symbolic analysis of electrical networks, *Proceedings of the 15th Midwest Symposium on Circuit Theory*, Rolla, MO, pp. V4.1–V4.8, May 1972.

[Man91] S. Manetti, New approaches to automatic symbolic analysis of electric circuits, *Proceedings of IEE*, 138, 22–28, Feb. 1991.

[Mar93] M. Martins et al., A computer-assisted tool for the analysis of multirate SC networks by symbolic signal flow graphs, *Alta Frequenza*, 5(6), 6–10, Nov. 1993.

[Mas56] S.J. Mason, Feedback theory—Further properties of signal flow graphs, *Proceedings of IRE*, 44, 920–926, July 1956.

[Mcc72] J.O. McClanahan, S.P. Chan, Computer analysis of general linear networks using digraphs, *International Journal of Electronics*, 33(2), 153–191, Aug. 1972.

[Mcn68] L.P. McNamee, H. Potash, *A User's and Programmer's Manual for NASAP*, University of California at Los Angeles, Rep. pp. 63–38, Aug. 1968.

[Mie78] R.R. Mielke, A new signal flowgraph formulation of symbolic network functions, *IEEE Transactions on Circuits and Systems*, 25(6), 334–340, June 1978.

[Okr70] H. Okrent, L.P. McNamee, *NASAP-70 User's and Programmer's Manual*, UCLA, Technical Report ENG-7044, 1970.

[Pie93] M. Pierzchala, B. Rodanski, A new method of semi-symbolic network analysis, *Proceedings of IEEE ISCAS*, Chicago, IL, pp. 2240–2243, May 1993.

[Pie96] M. Pierzchala, B. Rodanski, Efficient generation of symbolic network functions for large-scale circuits, *Proceedings of IEEE MWSCAS*, Ames, IO, pp. 425–428, Aug. 1996.

[Pie98] M. Pierzchala, B. Rodanski, Direct calculation of numerical coefficients in semi-symbolic circuit analysis, *Proceedings of SMACD*, Kaiserslautern, Germany, pp. 173–176, Oct. 1998.

[Pie01] M. Pierzchala, B. Rodanski, Generation of sequential symbolic network functions for large-scale networks by circuit reduction to a two-port, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 48(7), 906–909, July 2001.

[Pot68] C. Pottle, *CORNAP User Manual*, School of Electrical Engineering, Cornell University, Ithaca, NY, 1968.

[Rod00] B. Rodanski, Computational efficiency of symbolic sequential formulae, *Proceedings of SMACD*, Lisbon, Portugal, pp. 45–50, Oct. 2000.

[Rod08] B. Rodanski, M. Hassoun, Symbolic analysis methods, *Computer Aided Design and Design Automation*, W.-K. Chen, Ed., Boca Raton: CRC Press, 2008.

[San80] P. Sannuti, N.N. Puri, Symbolic network analysis—An algebraic formulation, *IEEE Transactions on Circuits and Systems*, 27(8), 679–687, Aug. 1980.

[Sed92] S. Seda, M. Degrauwe, W. Fichtner, Lazy-expansion symbolic expression approximation in SYNAP, *International Conference on Computer-Aided Design*, Santa Clara, CA, pp. 310–317, 1992.

[Shi00] C.-J.R. Shi, X.-D. Tan, Canonical symbolic analysis of large analog circuits with determinant decision diagrams, *IEEE Transactions on Computer-Aided Design*, 19(1), 1–18, Jan. 2000.

[Shi01] C.-J.R. Shi, X.-D. Tan, Compact representation and efficient generation of *s*-expanded symbolic network functions for computer-aided analog circuit design, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(7), 813–827, July 2001.

[Sin74] K. Singhal, J. Vlach, Generation of immittance functions in symbolic form for lumped distributed active networks, *IEEE Transactions on Circuits and Systems*, 21(1), 57–67, Jan. 1974.

[Sin77] K. Singhal, J. Vlach, Symbolic analysis of analog and digital circuits, *IEEE Transactions on Circuits and Systems*, 24(11), 598–609, Nov. 1977.

[Som91] R. Sommer, EASY—An experimental analog design system framework, *Proceedings of SMACD*, Paris, Oct. 1991.

[Sta86] J.A. Starzyk, A. Konczykowska, Flowgraph analysis of large electronic networks, *IEEE Transactions on Circuits and Systems*, 33(3), 302–315, Mar. 1986.

[Sta96] J.A. Starzyk, J. Zou, Direct symbolic analysis of large analog networks, *Proceedings of IEEE MWSCAS*, Ames, IO, pp. 421–424, Aug. 1996.

[Top98] M.D. Topa, On symbolic analysis of weakly-nonlinear circuits, *Proceedings of SMACD*, Kaiserslautern, Germany, pp. 207–210, Oct. 1998.

[Ver02] W. Verhaegen, G. Gielen efficient DDD-based symbolic analysis of linear analog circuits, *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 49(7), 474–487, July 2002.

[Vla94] J. Vlach, K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd edn., New York: Van Nostrand Reinhold, 1994.

[Wen98] C. Wen, H. Floberg, Q. Shui-sheng, A unified symbolic method for steady-state analysis of nonlinear circuits and systems, *Proceedings of SMACD*, Kaiserslautern, Germany, pp. 218–222, Oct. 1998.

[Wie89] G. Wierzba et al., SSPICE—A symbolic SPICE program for linear active circuits, *Proceedings of IEEE MWSCAS*, Urbana, IL, pp. 1197–1201, Aug. 1989.

[Wam92] P. Wambacq, G. Gielen, W. Sansen, A cancellation free algorithm for the symbolic simulation of large analog circuits, *Proceedings of IEEE ISCAS*, San Diego, CA, pp. 1157–1160, May 1992.

[Wam98] P. Wambacq, G.E. Gielen, W. Sansen, Symbolic network analysis methods for practical analog integrated circuits: A Survey, *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 45(10), 1331–1341, Oct. 1998.

[Yu96] Q. Yu, C. Sechen, A unified approach to the approximate symbolic analysis of large analog integrated circuits, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 43(8), 656–669, Aug. 1996.

26

Analysis in the Time Domain

26.1 Signal Types **26**-1
Introduction • Step, Impulse, and Ramp • Sinusoids • Eternal • Causal • Periodic and Aperiodic Waveforms
References **26**-6
26.2 First-Order Circuits **26**-6
Introduction • Zero-Input and Zero-State Response • Transient and Steady-State Responses • Network Time Constant
References **26**-10
26.3 Second-Order Circuits **26**-11
Introduction • Zero-Input and Zero-State Response • Transient and Steady-State Responses • Network Characterization
References **26**-18

Robert W. Newcomb
University of Maryland

26.1 Signal Types

26.1.1 Introduction

Because information into and out of a circuit is carried via time domain signals we look first at some of the basic signals used in continuous time circuits. All signals are taken to depend on continuous time t over the full range $-\infty < t < \infty$. It is important to realize that not all signals of interest are functions in the strict mathematical sense; we must go beyond them to generalized functions (e.g., the impulse), which play a very important part in the signal processing theory of circuits.

26.1.2 Step, Impulse, and Ramp

The unit step function, denoted $1(\cdot)$, characterizes sudden jumps, such as when a signal is turned on or a switch is thrown; it can be used to form pulses, to select portions of other functions, and to define the ramp and impulse as its integral and derivative. The unit step function is discontinuous and jumps between two values, 0 and 1, with the time of jump between the two taken as $t = 0$. Precisely,

$$1(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{26.1}$$

which is illustrated in Figure 26.1 along with some of the functions to follow.

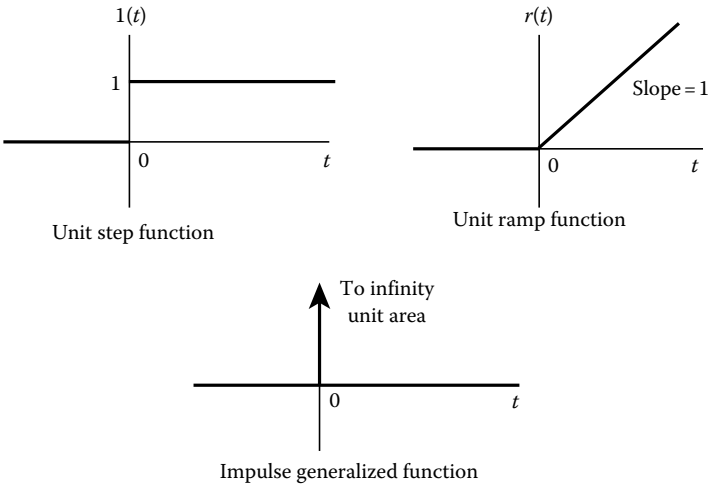


FIGURE 26.1 Step, ramp, and impulse functions.

Here, the value at the jump point, $t = 0$, purposely has been left free because normally it is immaterial and specifying it can lead to paradoxical results. Physical step functions used in the laboratory are actually continuous functions that have a continuous rise between 0 and 1, which occurs over a very short time. Nevertheless, instances occur in which one may wish to set 1(0) equal to 0 or to 1 or to 1/2 (the latter, for example, when calculating the values of a Fourier series at a discontinuity). By shifting the time argument the jump can be made to occur at any time, and by multiplying by a factor the height can be changed. For example, $1(t - t_0)$ has a jump at time t_0 and $a[1(t) - 1(t - t_0)]$ is a pulse of width t_0 and height a going up to a at $t = 0$ and down to 0 at time t_0. If $a = a(t)$ is a function of time, then that portion of $a(t)$ between 0 and t_0 is selected. The unit ramp, $r(\cdot)$ is the continuous function which ramps up linearly (with unit slope) from zero starting at $t = 0$; the ramp results from the unit step by integration

$$r(t) = \int_{-\infty}^{t} 1(\tau)\mathrm{d}\tau = t\ 1(t) = \begin{cases} t & \text{if } t > 0 \\ 0 & \text{if } t < 0 \end{cases} \tag{26.2}$$

As a consequence the unit step is the derivative of the unit ramp, while differentiating the unit step yields the unit impulse generalized function, $\delta(\cdot)$ that is

$$\delta(t) = \frac{d^1(t)}{\mathrm{d}t} = \frac{d^2 r(t)}{\mathrm{d}t^2} \tag{26.3}$$

In other words, the unit impulse is such that its integral is the unit step; that is, its area at the origin, $t = 0$, is 1. The impulse acts to sample continuous functions which multiply it, i.e.,

$$a(t)\delta(t - t_0) = a(t_0)\delta(t - t_0) \tag{26.4}$$

This sampling property yields an important integral representation of a signal $x(\cdot)$

$$\begin{aligned} x(t) &= \int_{-\infty}^{\infty} x(\tau)\delta(t - \tau)\mathrm{d}\tau \\ &= \int_{-\infty}^{\infty} x(\tau)\delta(t - \tau)\mathrm{d}\tau = x(t) \int_{-\infty}^{\infty} \delta(t - \tau)\mathrm{d}\tau \end{aligned} \tag{26.5}$$

where the validity of the first line is seen from the second line, and the fact that the integral of the impulse through its jump point is unity. Equation 26.5 is actually valid even when $x(\cdot)$ is discontinuous and, consequently, is a fundamental equation for linear circuit theory. Differentiating $\delta(t)$ yields an even more discontinuous object, the doublet $\delta'(\cdot)$. Strictly speaking, the impulse, all its derivatives, and signals of that class are not functions in the classical sense, but rather they are operators [1] or functionals [2], called generalized functions or, often, distributions. Their evaluations take place via test functions, just as voltages are evaluated on test meters.

The importance of the impulse lies in the fact that if a linear time-invariant system is excited by the unit impulse, then the response, naturally called the impulse response, is the inverse Laplace transform of the network function. In fact, if $h(t)$ is the impulse response of a linear time-invariant (continuous and continuous time) circuit, the forced response $y(t)$ to any input $u(t)$ can be obtained without leaving the time domain by use of the convolution integral, with the operation of convolution denoted by$\star$

$$y(t) = h \star u = \int_{-\infty}^{\infty} h(t-\tau)u(\tau)\mathrm{d}\tau \tag{26.6}$$

Equation 26.6 is mathematically rigorous, but justified on physical grounds through Equation 26.5 as follows. If we let $h(t)$ be the output when $\delta(t)$ is the input, then, by time invariance, $h(t-\tau)$ is the output when the input is shifted to $\delta(t-\tau)$. Scaling the latter by $u(\tau)$ and summing via the integral, as designated in Equation 26.5, we obtain a representation of the input $u(t)$. This must result in the output representation being in the form of Equation 26.6 by linearity of the system through similar scaling and summing of $h(t-\tau)$, as was performed on the input.

26.1.3 Sinusoids

Sinusoidal signals are important because they are self-reproducing functions (i.e., eigenfunctions) of linear time-invariant circuits. This is true basically because the derivatives of sinusoids are sinusoidal. As such, sinusoids are also the natural outputs of oscillators and are delivered in power sources, including laboratory signal generators and electricity for the home derived from the power company.

26.1.4 Eternal

Eternal signals are defined as being of the same nature for all time, $-\infty < t < \infty$, in which case an eternal cosine repeats itself eternally in both directions of time, with an origin of time, $t = 0$, being arbitrarily fixed. Because eternal sinusoids have been turned on forever, they are useful in describing the steady operation of circuits. In particular, the signal $A\cos(\omega t + \theta)$ over $-\infty < t < \infty$ defines an eternal cosine of amplitude A, radian frequency $\omega = 2\pi f$ (with f being real frequency, in hertz, which are cycles per second), at phase angle θ (in radians and with respect to the origin of time), with A, ω, and θ real numbers. When $\theta = \pi/2$ this cosine also represents a sine, so that all eternal sinusoidal signals are contained in the expression $A\cos(\omega t + \theta)$.

At times, it is important to work with sinusoids that have an exponential envelope, with the possibility that the envelope increases or decreases with time, that is, with positively or negatively damped sinusoids. These are described by $Ae^{st}\cos(\omega t + \theta)$, where the real number is the damping factor, giving signals that damp out in time when the damping factor is positive and signals that increase with time when the damping factor is negative. Of most importance when working with this class of signals is the identity

$$e^{\sigma t + j\omega t} = e^{st} = e^{\sigma t}[\cos(\omega t) + j\sin(\omega t)] \tag{26.7}$$

where $s = \sigma + j\omega$ with $j = \sqrt{-1}$. Here, s is called the complex frequency, with its imaginary part being the real (radian) frequency, ω. When no damping is present, $s = j\omega$, in which case the exponential form of

Equation 26.7 represents pure sinusoids. In fact, we see in this expression that the cosine is the real part of an exponential and the sine is its imaginary part. Because exponentials are usually easier than sinusoids to treat analytically, the consequence for real linear networks is that we can do most of the calculations with exponentials and convert back to sinusoids at the end. In other words, if a real linear system has a cosine or a damped cosine as a true input, it can be analyzed by using instead the exponential of which it is the real part as its (fictitious) input, finding the resulting (fictitious) exponential output, and then taking the real part at the end of the calculations to obtain the true output for the true input. Because exponentials are probably the easiest signals to work with in theory, the use of exponentials rather than sinusoids usually greatly simplifies the theory and calculations for circuits operating under steady-state conditions.

26.1.5 Causal

Because practical circuits have not existed since $t = -\infty$ they usually begin to be considered at a suitable starting time, taken to be $t = 0$, in which case the associated signals can be considered to be zero for $t < 0$. Mathematically, these functions are said to have support bounded on the left. The support of a signal is (the closure of) that set of times for which the signal is nonzero, therefore, the support of these signals is bounded on the left by zero. When signals are discontinuous functions they have the important property that they can be represented by multiplying with unit step functions signals which are differentiable and have nonbounded support. For example, $g(t) = e^{st} \cdot 1(t)$ has a jump at $t = 0$ with support at the half line 0 to ∞ but has e^{st} infinitely differential of "eternal" support.

A causal circuit is one for which the response is only nonzero after the input becomes nonzero. Thus, if the inputs are zero for $t < 0$, the outputs of causal circuits are also zero for $t < 0$. In such cases the impulse response, $h(t)$, or the response to an input impulse of "infinite jump" at $t = 0$, satisfies $h(t) = 0$ for $t < 0$ and the convolution form of the output, Equation 26.4, takes the form

$$y(t) = \left[\int_0^t h(t - \tau) u(\tau) \mathrm{d}\tau \right] 1(t) \tag{26.8}$$

26.1.6 Periodic and Aperiodic Waveforms

The pure sinusoids, although not the sinusoids with nonzero damping, are special cases of periodic signals. In other words, ones which repeat themselves in time every T seconds, where T is the period. Precisely, a time-domain signal $g(\cdot)$ is periodic of period T if $g(t) = g(t + T)$, where normally T is taken to be the smallest nonzero T for which this is true. In the case of the sinusoids, $A \cos(\omega t + \theta)$ with $\omega = 2\pi f$, the period is given by $T = 1/f$ because $\{2\pi[f(t + T)] + \theta\} = \{2\pi ft + 2\pi(fT) + \theta\} = \{2\pi ft + (2\pi + \theta)\}$, and sinusoids are unchanged by a change of 2π in the phase angle. Periodic signals need to be specified over only one period of time, e.g., $0 \le t < T$, and then can be extended periodically for all time by using $t = t \bmod(T)$ where $\bmod(\cdot)$ is the modulus function; in other words, periodic signals can be looked upon as being defined on a circle, if we imagine the circle as being a clock face.

Periodic signals represent rhythms of a system and, as such, contain recurring information. As many physical systems, especially biomedical systems, either possess directly or to a very good approximation such rhythms, the periodic signals are of considerable importance. Even though countless periodic signals are available besides the sinusoids, it is important to note that almost all can be represented by a Fourier series. Exponentials are eigenfunctions for linear circuits, thus, the Fourier series is most conveniently expressed for circuit considerations in terms of the exponential form. If $g(t) = g(t + T)$, then

$$g(t) \cong \sum_{n=-\infty}^{\infty} c_n e^{j(2\pi nt/T)} \tag{26.9}$$

where the coefficients are complex and are given by

$$c_n = \frac{1}{T}\int_0^T g(t)e^{-j(2\pi nt/T)}\mathrm{d}t = a_n + jb_n \tag{26.10}$$

Strictly speaking, the integral is over the half-open interval $[0,T)$ as seen by considering $g(\cdot)$ defined on the circle. In Equation 26.9, the symbol $>$ is used to designate the expression on the right as a representation that may not exactly agree numerically with the left side at every point when $g(\cdot)$ is a function; for example, at discontinuities the average is obtained on the right side. If $g(\cdot)$ is real, that is, $g(t) = g(t)$,* where the superscript * denotes complex conjugate, then the complex coefficients c_n satisfy $c_n = c_{-n}^*$. In this case the real coefficients a_n and b_n in Equation 26.10 are even and odd in the indices; n and the a_n combine to give a series in terms of cosines, and the b_n gives a series in terms of sines.

As an example the square wave, sqw(t), can be defined by

$$\mathrm{sqw(t)} = 1(t) - 1(t - [T/2])0 \le t < T \tag{26.11}$$

and then extended periodically to $-\infty < t < \infty$ by taking $t = t \bmod(T)$. The exponential Fourier series coefficients are readily found from Equation 26.10 to be

$$c_n = \begin{cases} 1/2 & \text{if } n = 0 \\ \frac{1}{j\pi n}\begin{cases} 0 & \text{if } n = 2k \neq 0 \ \ (\text{even} \neq 0) \\ 1 & \text{if } n = 2k+1 \ \ (\text{odd}) \end{cases} \end{cases} \tag{26.12}$$

for which the Fourier series is

$$\mathrm{sqw}(t) \cong \frac{1}{2} + \sum_{k=-\infty}^{\infty} \frac{1}{j\pi[2k+1]} e^{j2\pi[2k+1]t/T} \tag{26.13}$$

The derivative of sqw(t) is a periodic set of impulses

$$\frac{d[\mathrm{sqw}(t)]}{\mathrm{d}t} = \delta(t) - \delta(t - [T/2])\ 0 \le t < T \tag{26.14}$$

for which the exponential Fourier series is easily found by differentiating Equation 26.13, or by direct calculation from Equation 26.10, to be

$$\sum_{i=-\infty}^{\infty} (\delta(t - iT) - \delta(t - iT - [T/2]) \cong \sum_{k=-\infty}^{\infty} \frac{2}{T} e^{j(2\pi[2k+1]t/T)} \tag{26.15}$$

Combining the exponentials allows for a sine representation of the periodic generalized function signal. Further differentiation can take place, and by integrating Equation 26.15 we get the Fourier series for the square wave if the appropriate constant of integration is added to give the DC value of the signal. Likewise, a further integration will yield the Fourier series for the sawtooth periodic signal, and so on.

The importance of these Fourier series representations is that a circuit having periodic signals can always be considered to be processing these signals as exponential signals, which are usually self-reproducing signals for the system, making the design or analysis easy. The Fourier series also allows

visualization of which radian frequencies, $2\pi n/T$, may be important to filter out or emphasize. In many common cases, especially for periodically pulsed circuits, the series may be expressed in terms of impulses. Thus, the impulse response of the circuit can be used in conjunction with the Fourier series.

References

1. J. Mikusinski, *Operational Calculus*, 2nd edn., New York: Pergamon Press, 1983.
2. A. Zemanian, *Distribution Theory and Transform Analysis*, New York: McGraw-Hill, 1965.

26.2 First-Order Circuits

26.2.1 Introduction

First-order circuits are fundamental to the design of circuits because higher order circuits can be considered to be constructed of them. Here, we limit ourselves to single-input-output linear time-invariant circuits for which we take the definition of a first-order circuit to be one described by the differential equation

$$d_1 \cdot \frac{dy}{\mathrm{d}t} + d_0 \cdot y = n_1 \cdot \frac{\mathrm{d}u}{\mathrm{d}t} + n_0 \cdot u \tag{26.16}$$

where

d_0 and d_1 are the "denominator" constants and n_0 and n_1 are the "numerator" constants
$y = y(\cdot)$ is the output and $u = u(\cdot)$ is the input
both u and y are the generalized functions of time t

So that the circuit truly will be first order, we require that $d_1 n_0 - d_0 n_1 \neq 0$, which guarantees that at least one of the derivatives is actually present, but if both derivatives occur, the expressions in y and in u are not proportional, which would lead to cancellation, forcing y and u to be constant multiples of each other. Because a factorization of real higher-order systems may lead to complex first-order systems, we will allow the numerator and denominator constants to be complex numbers; thus, y and u may be complex-valued functions.

If the derivative is treated as an operator, $p = d[\cdot]/\mathrm{d}t$, then Equation 26.16 can be conveniently written as

$$y = \frac{n_1 p + n_0}{d_1 p + d_0} u = \begin{cases} \left[\dfrac{n_1}{d_0} p + \dfrac{n_0}{d_0}\right] u & \text{if } d_1 = 0 \\ \left[\dfrac{n_1}{d_1} + \dfrac{d_1 n_0 - d_0 n_1}{p + (d_0/d_1)}\right] u & \text{if } d_1 \neq 0 \end{cases} \tag{26.17}$$

where the two cases in terms of d_1 are of interest because they provide different forms of responses, each of which frequently occurs in first-order circuits. As indicated by Equation 26.17, the transfer function

$$H(p) = \frac{n_1 p + n_0}{d_1 p + d_0} \tag{26.18}$$

is an operator (as a function of the derivative operator p), which characterizes the circuit. Table 26.1 lists some of the more important types of different first-order circuits along with their transfer functions and causal impulse responses.

The following treatment somewhat follows that given in [1], although with a slightly different orientation in order to handle all linear time-invariant continuous time continuous circuits.

TABLE 26.1 Typical Transfer Functions of First-Order Circuits

Transfer Function	Description	Impulse Response
$\frac{n_1}{d_0}p$	Differentiator	$\frac{n_1}{d_0}\delta'(t)$
$\frac{n_0}{d_1 p}$	Integrator	$\frac{n_0}{d_1}1(t)$
$\frac{n_1 p + n_0}{d_1}$	Leaky differentiator	$\frac{n_0}{d_1}\delta(t) + \frac{n_1}{d_1}\delta'(t)$
$\frac{n_0}{d_1 p + d_0}$	Low-pass filter; lossy integrator	$\frac{n_0}{d_1}e^{-\frac{d_0}{d_1}t} \cdot 1(t)$
$\frac{n_1 p}{d_1 p + d_0}$	High-pass filter	$\frac{n_1}{d_1}\delta(t) + \frac{n_1 d_0}{d_1^2}e^{-\frac{d_0}{d_1}t} \cdot 1(t)$
$\frac{n_1}{d_1}\frac{p - (d_0/d_1)}{p + (d_0/d_1)}$	All-pass filter	$\frac{n_1}{d_1}\left[\delta(t) - 2\frac{d_0}{d_1}e^{-\frac{d_0}{d_1}t} \cdot 1(t)\right]$

26.2.2 Zero-Input and Zero-State Response

The response of a linear circuit is, via the linearity, the sum of two responses, one due to the input when the circuit is initially in the zero state, called the zero-state response, and the other due to the initial state when no input is present, the zero-input response. By the linearity the total response is the sum of the two separate responses, and thus we may proceed to find each separately. In order to investigate these two types of responses, we introduce the state vector $x(\cdot)$ and the state-space representation (as previously $p = \mathrm{d}[\cdot]/\mathrm{d}t$)

$$\begin{aligned} px &= Ax + Bu \\ y &= Cx + Du + Epu \end{aligned} \tag{26.19}$$

where A, B, C, D, E are constant matrices. For our first-order circuit two cases are exhibited, depending upon d_1 being zero or not. In the case of $d_1 = 0$,

$$y = (n_1/d_0)u + (n_1/d_0)pu \quad d_1 = 0 \tag{26.20a}$$

Here, $C = 0$ and A and B can be chosen anything, including empty. When $d_1 \neq 0$, our first-order circuit has the following set of (minimal size) state-variable equations

$$\begin{aligned} px &= \left[-\frac{d_0}{d_1}\right] \cdot x + [d_1 n_0 - d_0 n_1] \cdot u \\ & \qquad\qquad\qquad\qquad d_1 \neq 0 \\ y &= [1] \cdot x + \left[\frac{n_1}{d_1}\right] \cdot u \end{aligned} \tag{26.20b}$$

By choosing $u = 0$ in Equation 26.2, we obtain the equations that yield the zero input response. Specifically, the zero-input response is

$$y(t) = \begin{cases} 0 & \text{if } d_1 = 0 \\ e^{-\frac{d_0}{d_1}t} \cdot y(0) & \text{if } d_1 \neq 0 \end{cases} \tag{26.21}$$

which is also true by direct substitution into Equation 26.16. Here, we have set, in the $d_1 \neq 0$ case, the initial value of the state, $x(0)$, equal to the initial value of the output, $y(0)$, which is valid by our choice of state-space equations. Note that Equation 26.21 is valid for all time and y at $t = 0$ assumes the assigned initial value $y(0)$, which must be zero when the input is zero and no derivative occurs on the output.

The zero-state response is explained as the solution of Equation 26.21 when $x(0) = 0$. In the case that $d_1 = 0$, the zero-state response is

$$y = \frac{n_0}{d_0}u + \frac{n_1}{d_0}pu = \left\{\frac{n_0}{d_0}\delta(t) + \frac{n_1}{d_0}\delta'(t)\right\} \star u \quad d_1 = 0 \tag{26.22a}$$

where

$\star$ denotes convolution
$\delta(\cdot)$ is the unit impulse
$1(\cdot)$ is the unit step function

While in the case that $d_1 \neq 0$

$$y = \left\{\frac{n_1}{d_1}\delta(t) + \left[\frac{d_1 n_0 - d_0 n_1}{d_1}\right]e^{-\frac{d_0}{d_1}t}1(t)\right\} \star u \quad d_1 \neq 0 \tag{26.22b}$$

which is found by eliminating x from Equation 26.20b and can be checked by direct substitution into Equation 26.16. The terms in the braces are the causal impulse responses, $h(t)$, which are checked by letting $u = \delta$ with otherwise zero initial conditions, that is, with the circuit initially in the zero state. Actually, infinitely many noncausal impulse responses could be used in Equation 26.22b. One such response is found by replacing $1(t)$ by $-1(-t)$]. However, physically the causal responses are of most interest.

If $d_1 \neq 0$, the form of the responses is determined by the constant d_0/d_1, the reciprocal of which (when $d_0 \neq 0$) is called the time constant, t_c, of the circuit because the circuit impulse response decays to $1/e$ at time $t_c = d_1/d_0$. If the time constant is positive, the zero-input and the impulse responses asymptotically decay to zero as time approaches positive infinity, and the circuit is said to be asymptotically stable. On the other hand, if the time constant is negative, then these two responses grow without bounds as time approaches plus infinity, and the circuit is called unstable. It should be noted that as time goes in the reverse direction to minus infinity, the unstable zero-input response decays to zero. If $d_0/d_1 = 0$ the zero-input and impulse responses are still stable, but neither decay nor grow as time increases beyond zero.

By linearity of the circuit and its state-space equations, the total response is the sum of the zero-state response and the zero-input response; thus, even when $d_0 = 0$ or $d_1 = 0$

$$y(t) = e^{-\frac{d_0}{d_1}}y_0 + h(t) \star u(t) \tag{26.23}$$

Assuming that u and h are zero for $t < 0$ their convolution is also zero for $t < 0$, although not necessarily at $t = 0$, where it may even take on impulsive behavior. In such a case, we see that y_0 is the value of the output instantaneously before $t = 0$. If we are interested only in the circuit for $t > 0$, surprisingly, an input will yield the zero input response. That is, an equivalent input u_0 exists, which will yield the zero input response for $t > 0$, this being $u_0(t) = d_1 y_0 \exp(-t d_0/d_1)1(t)$. Thus, $y = h \star (u + u_0)$ gives the same result as Equation 26.23.

When $d_1 = 0$, the circuit acts as a differentiator and within the state-space framework it is treated as a special case. However, in practice it is not a special case because the current, i, versus voltage, v, for a

capacitor of capacitance C, in parallel with a resistor of conductance G is described by $i = Cpv + Gv$. Consequently, it is worth noting that all cases can be handled identically in the semistate description

$$\begin{bmatrix} d_1 & d_1 - 1 \\ 0 & 0 \end{bmatrix} px = \begin{bmatrix} -d_0 & -d_0 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} n_0 \\ n_1 \end{bmatrix} u$$
$$y = \begin{bmatrix} 1 & 1 \end{bmatrix} x \tag{26.24}$$

where $x(\cdot)$ is the semistate instead of the state, although the first components of the two vectors agree in many cases. In other words, the semistate description is more general than the state description, and handles all circuits in a more convenient fashion [2].

26.2.3 Transient and Steady-State Responses

This section considers stable circuits, although the techniques are developed so that they apply to other situations. In the asymptotically stable case, the zero input response decays eventually to zero; that is, transient responses due to initial conditions eventually will not be felt and concentration can be placed upon the zero-state response. Considering first eternal exponential inputs, $u(t) = U \exp(\text{st})$ for $-\infty < t < \infty$ at the complex frequency $s = \sigma + j\omega$, where s is chosen as different from the natural frequency $s_n = -d_0/d_1 = -1/t_c$ and U is a constant, we note that the response is $y(t) = Y(s) \exp(\text{st})$, as is observed by direct substitution into Equation 26.16; this substitution yields directly

$$Y(s) = \frac{n_1 s + n_0}{d_1 s + d_0} \cdot U \tag{26.25}$$

where $y(t) = Y(s) \exp(\text{st})$ for $u(t) = U \exp(\text{st})$ over $-\infty < t < \infty$. That is, an exponential excitation yields an exponential response at the same (complex) frequency $s = \sigma + j\omega$ as that for the input. When $\sigma = 0$, the excitation and response are both sinusoidal and the resulting response is called the sinusoidal steady state (SSS). Equation 26.25 shows that the SSS response is found by substituting the complex frequency $s = j\omega$ into the transfer function, now evaluated on complex numbers instead of differential operators as in Equation 26.18,

$$H(s) = \frac{n_1 s + n_0}{d_1 s + d_0} \tag{26.26}$$

This transfer function represents the impulse response, $h(t)$, of which it is actually the Laplace transform, and as we found earlier, the causal impulse response is

$$h(t) = \begin{cases} \dfrac{n_0}{d_0}\delta(t) + \dfrac{n_1}{d_0}\delta'(t), & \text{if } d_1 = 0 \\ \dfrac{n_1}{d_1}\delta(t) + \left[\dfrac{d_1 n_0 - d_0 n_1}{d_1}\right] e^{-\frac{d_0}{d_1}t} 1(t), & \text{if } d_1 \neq 0 \end{cases} \tag{26.27}$$

However, practical signals are started at some finite time, normalized here to $t = 0$, instead of at $t = -\infty$, as used for the preceding exponentials. Thus, consider an input of the same type but applied only for $t > 0$; i.e., let $u(t) = U \exp(\text{st}) 1(t)$. The output is found by using the convolution $y = h * u$; after a slight amount of calculation is evaluated to

$$\begin{aligned} y(t) &= h(t) * U e^{st} 1(t) \\ &= \begin{cases} H(s) U e^{st} 1(t) + \dfrac{n_1}{d_0} U \delta(t) & \text{for } d_1 = 0 \\ H(s) U e^{st} 1(t) - \dfrac{[d_1 n_0 - d_0 n_1]}{d_1 s + d_0} U e^{-\frac{d_0}{d_1}} 1(t) & \text{for } d_1 \neq 0 \end{cases} \end{aligned} \tag{26.28}$$

For $t > 0$, the SSS remains present, while there is another term of importance when $d_1 \neq 0$. This is a transient term, which disappears after a sufficient waiting time in the case of an asymptotically stable circuit. That is, the SSS is truly a steady state, although one may have to wait for it to dominate. If a nonzero zero-input response exists, it must be added to the right side of Equation 26.28, but for $t > 0$ this is of the same form as the transient already present, therefore, the conclusion is identical (the SSS eventually predominates over the transient terms for an asymptotically stable circuit).

Because a cosine is the real part of a complex exponential and the real part is obtained as the sum of two terms, we can use linearity of the circuit to quickly obtain the output to a cosine input when we know the output due to an exponential. We merely write the input as the sum of two complex conjugate exponentials and then take the complex conjugates of the outputs that are summed. In the case of real coefficients in the transfer function, this is equivalent to taking the real part of the output when we take the real part of the input; that is, $y = \Re(h \star u_3) = h \star u$, when $u = \Re(u_e)$, if y is real for all real u.

26.2.4 Network Time Constant

The time constant, t_c, was defined earlier as the time for which a transient decays to $1/e$ of the initial value. As such, the time constant shows up in signals throughout the circuit and is a very useful parameter when identifying a circuit from its responses. In an RC circuit, the time constant physically results from the interaction of the equivalent capacitor (of which only one exists in a first-order circuit) of capacitance C_{eq}, and the Thévenin's equivalent resistor, of resistance R_{eq}, that it sees. Thus, $t_c = R_{eq}C_{eq}$.

Closely related to the time constant is the rise time. Considering the low-pass case, the rise time, t_r is defined as the time for the unit step response to go between 10% and 90% of its final value from its initial value. This is easily calculated because the unit step response is given by

$$y_{1(\cdot)}(t) = h(t) \star 1(t) = \frac{n_0}{d_0}\left[1 - e^{-\frac{d_0}{d_1}t}\right] \cdot 1(t) \tag{26.29}$$

Assuming a stable circuit and setting this equal to 0.1 and 0.9 times the final value, n_0/d_0, it is readily found that

$$t_r = \frac{d_1}{d_0} \cdot \ln(9) = [\ln(9)] \cdot t_c \approx 2.2t_c \tag{26.30}$$

At this point, it is worth noting that for theoretical studies the time constant can be normalized to 1 by normalizing the time scale. Thus, assuming d_1 and $d_0 \neq 0$ the differential equation can be written as

$$d_0 \cdot \left[\frac{d_1}{d_0} \cdot \frac{dy}{d(d_1/d_0)(t(d_1/d_0))} + y\right] = d_0\left[\frac{dy}{dt_n} + y\right] \tag{26.31}$$

where $t_n = (d_0/d_1)t$ is the normalized time.

References

1. L. P. Huelsman, *Basic Circuit Theory with Digital Computations*, Englewood Cliffs, NJ: Prentice Hall, 1972.
2. R. W. Newcomb and B. Dziurla, Some circuits and systems applications of semistate theory, *Circuits, Systems, and Signal Processing*, 8(3), 235–260, 1989.

26.3 Second-Order Circuits

26.3.1 Introduction

Because real transfer functions can be factored into real second-order transfer functions, second-order circuits are probably the most important circuits available; most designs are based upon them. As with first-order circuits, this chapter is limited to single-input-single-output linear time-invariant circuits, and unless otherwise stated, here real-valued quantities are assumed. By definition a second-order circuit is described by the differential equation

$$d_2 \cdot \frac{d^2y}{dt^2} + d_1 \cdot \frac{dy}{dt} + d_0 \cdot y = n_2 \cdot \frac{d^2u}{dt^2} + n_1 \cdot \frac{du}{dt} + n_0 \cdot u \tag{26.32}$$

where d_i and n_i are "denominator" and "numerator" constants, $i = 0, 1, 2$, which, unless mentioned to the contrary, are taken to be real. Continuing the notation used for first-order circuits, $y = y(\cdot)$ is the output and $u = u(\cdot)$ is the input; both u and y are generalized functions of time t. Assume that $d_2 \neq 0$, which is the normal case because any of the other special cases can be considered as cascades of real degree one circuits.

Again, treating the derivative as an operator, $p = d[\cdot]/dt$, Equation 26.32 is written as

$$y = \frac{n_2p^2 + n_1p + n_0}{d_2p^2 + d_1p + d_0}u \tag{26.33}$$

with the transfer function

$$\begin{aligned} H(p) &= \frac{1}{d_2}\left[\frac{n_2p^2 + n_1p + n_0}{p^2 + (d_1/d_2)p + (d_0/d_2)}\right] \\ &= \frac{1}{d_2}\left[n_2 + \frac{(n_1 - (d_1/d_2)n_2)p + (n_0 - (d_0/d_2)n_2)}{p^2 + (d_1/d_2)p + (d_0/d_2)}\right] \end{aligned} \tag{26.34}$$

where the second form results by long division of the denominator into the numerator. Because they occur most frequently when second-order circuits are discussed, we rewrite the denominator in two equivalent customarily used forms:

$$p^2 + \frac{d_1}{d_2}p + \frac{d_0}{d_2} = p^2 + \frac{\omega_n}{Q}p + \omega_n^2 = p^2 + 2\zeta\omega_n p + \omega_n^2 \tag{26.35}$$

where

ω_n is the undamped natural frequency ≥ 0
Q is the quality factor
ζ is the damping factor $= 1/(2Q)$

The transfer function is accordingly

$$H(p) = \frac{1}{d_2}\left[\frac{n_2p^2 + n_1p + n_0}{p^2 + (\omega_n/Q)p + \omega_n^2}\right] = \frac{1}{d_2}\left[\frac{n_2p^2 + n_1p + n_0}{p^2 + 2\zeta\omega_n p + \omega_n^2}\right] \tag{26.36}$$

Table 26.2 lists several of the more important transfer functions, which, as in the first-order case, are operators as functions of the derivative operator p.

TABLE 26.2 Typical Second-Order Circuit Transfer Functions

Transfer Function	Description	Impulse Response
$\frac{n_0}{d_2}\frac{1}{p^2+2\zeta\omega_n p+\omega_n^2}$	Low-pass	$h_{\rm lp}(t)=\frac{n_0}{d_2}\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}\omega_n}\sin\left(\sqrt{1-\zeta^2}\omega_n t\right)1(t)$
$\frac{n_2}{d_2}\frac{p^2}{p^2+2\zeta\omega_n p+\omega_n^2}$	High-pass $0=\arctan2\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)$	$h_{\rm hp}(t)=\frac{n_2}{d_2}\left[\delta(t)-\frac{\omega_n e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin\left(\sqrt{1-\zeta^2}\omega_n t+2\theta\right)1(t)\right]$
$\frac{n_1}{d_2}\frac{p}{p^2+2\zeta\omega_n p+\omega_n^2}$	Bandpass $\theta=\arctan2\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right)$	$h_{\rm bp}(t)=\frac{n_1}{d_2}\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left(\sqrt{1-\zeta^2}\omega_n t+\theta\right)1(t)$
$\frac{n_2}{d_2}\frac{p^2+\omega_0^2}{p^2+2\zeta\omega_n p+\omega_n^2}$	Band-stop	$h_{\rm bs}(t)=h_{\rm hp}(t)+\frac{n_2\omega_0^2}{n_0}h_{\rm lp}(t)$
$\frac{n_2}{d_2}\frac{p^2-2\zeta\omega_n p+\omega_n^2}{p^2+2\zeta\omega_n p+\omega_n^2}$	All-pass	$h_{\rm ap}(t)=\frac{n_2}{d_2}\left[\delta(t)-\frac{4\zeta\omega_n e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left(\sqrt{1-\zeta^2}\omega_n t+\theta\right)1(t)\right]$
$\frac{n_0}{d_2}\frac{1}{p^2+\omega_n^2}$	Oscillator, when $u=0$	$h_{\rm osc}(t)=\frac{n_0}{d_2}\sin(\omega_n t)\cdot 1(t)$ $\lvert y(t)\rvert_{u=0}=y(0)\cdot\cos(\omega_n t)+\frac{y'(0)}{\omega_n}\cdot\sin(\omega_n t)$

26.3.2 Zero-Input and Zero-State Response

Again, as in the first-order case, a convenient tool for investigating the time-domain behavior of a second-order circuit is the state variable description. Letting the state vector be $x(\cdot)$, the state-space representation is

$$\begin{aligned} px &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{26.37}$$

where, as above, $p = d[\cdot]/dt$, and A, B, C, D are constant matrices. In the present case, these matrices are real and one convenient choice, among many, is

$$\begin{aligned} px &= \begin{bmatrix} 0 & 1 \\ -\frac{d_0}{d_2} & -\frac{d_1}{d_2} \end{bmatrix} x + \begin{bmatrix} n_1 - \frac{d_1}{d_2}n_2 \\ \left(n_0 - \frac{d_0}{d_2}n_2\right) - \left(n_1 - \frac{d_1}{d_2}n_2\right) \end{bmatrix} u \\ y &= \begin{bmatrix} \frac{1}{d_2} & 0 \end{bmatrix} x + \begin{bmatrix} \frac{n_1}{d_2} \end{bmatrix} u \end{aligned} \tag{26.38}$$

Here, the state is the two-vector $x = [x_1\ x_2]^{\rm T}$, with the superscript T denoting transpose. Normally, the state would consist of capacitor voltages and/or inductor currents, although at times one may wish to use linear combinations of these. From these state variable equations, a generic operational-amplifier (op-amp) RC circuit to realize any of this class of second-order circuits is readily designed and given in Figure 26.2. In the figure, all voltages are referenced to ground and normalized capacitor and resistor values are listed. Alternate designs in terms of only CMOS differential pairs and capacitors can also be given [3], while a number of alternate circuits exist in the catalog of Sallen and Key [4].

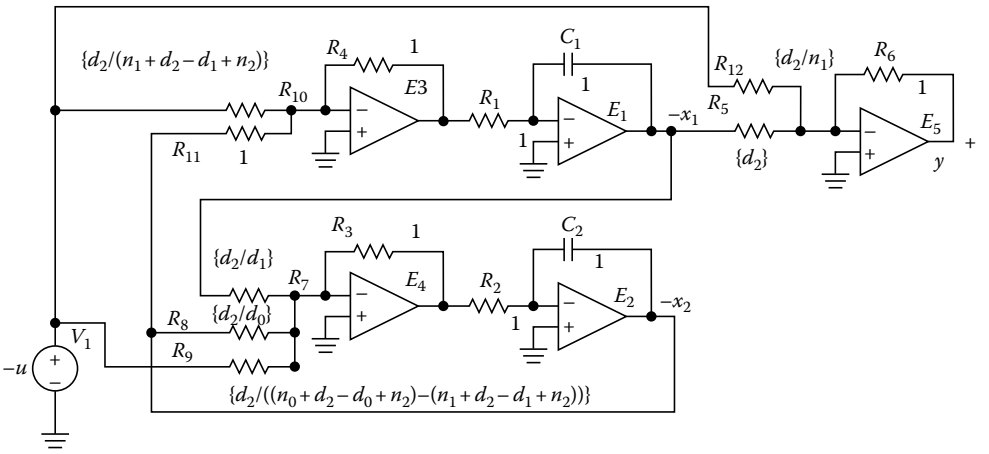


FIGURE 26.2 Generic, second-order op-amp RC circuit.

Because Equation 26.38 represents a set of linear constant coefficient differential equations, superposition applies and its solution can again be broken into two parts, the part due to initial conditions, $x(0)$, called the zero-input response, and the part due solely to the input u, the zero-state response.

The zero-input response is readily found by solving the state equations with $u = 0$ and initial conditions $x(0)$. The result is $y(t) = C\exp(At)\,x(0)$, which can be evaluated by several means, including the following. Using a prime to designate the time derivative, first note that when $u = 0$, $x_1(t) = d_2 y(t)$ and (from the first row of A) $x_1(t)' = x_2(t) = d_2 y(t)'$. Thus, $x_1(0) = d_2 y(0)$ and $x_2(0) = d_2\, y'(0)$, which allow the initial conditions to be expressed in terms of the measurable output quantities. To evaluate $\exp(At)$, note that its terms are linear combinations of terms with complex frequencies that are zeroes of the characteristic polynomial

$$\begin{aligned}\det(s1_2 - A) = \det\begin{pmatrix} s & -1 \\ \omega_n^2 & s + 2\zeta\omega_n \end{pmatrix} &= s^2 + 2\zeta\omega_n s + \omega_n^2 \\ &= (s - s_-)(s - s_+)\end{aligned} \tag{26.39}$$

For which the roots, called natural frequencies, are

$$s_\pm = \left(-\zeta \pm \sqrt{\zeta^2 - 1}\right)\omega_n = \left(-1 \pm \sqrt{1 - 4Q^2}\right)\frac{\omega_n}{2Q} \tag{26.40}$$

The case of equal roots will only occur when $\zeta^2 = 1$, which is the same as $Q^2 = 1/4$, for which the roots are real. Indeed, if the damping factor, ζ, is >1 in magnitude, or equivalently, if the quality factor, Q, is $<1/2$ in magnitude, the roots are real and the circuit can be considered a cascade of two first-order circuits. Thus, assume here and in the following that unless otherwise stated, $Q^2 > 0.25$, which is the same as $\zeta^2 < 1$, in which case the roots are complex conjugates, $s_- = s_+^*$

$$s_\pm = \left(-\zeta \pm j\sqrt{1 - \zeta^2}\right)\omega_n = \left(-1 \pm j\sqrt{4Q^2 - 1}\right)\frac{\omega_n}{2Q}, \quad j = \sqrt{-1} \tag{26.41}$$

By writing $y(t) = a \cdot \exp(s_+ t) + b \cdot \exp(s_- t)$, for unknown constants a and b, differentiating and setting $t = 0$ we can solve for a and b, and after some algebra and trigonometry obtain the zero-input response

$$y(t) = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \left\{ y(0) \cdot \cos\left(\sqrt{1-\zeta^2}\omega_n t - \theta\right) + \frac{y'(0)}{\omega_n} \cdot \sin\left(\sqrt{1-\zeta^2}\omega_n t\right) \right\} \tag{26.42}$$

where $\theta = \arctan2\left(\zeta/\sqrt{1-\zeta^2}\right)$ with $\arctan2(\cdot)$ being the arc tangent function that incorporates the sign of its argument.

The form given in Equation 26.42 allows for some useful observations. Remembering that this assumes $\zeta^2 < 1$, first note that if no damping occurs, that is, $\zeta = 0$, then the natural frequencies are purely imaginary, $s_+ = j\omega_n$ and $s_- = -s_+$, and the response is purely oscillatory, taking the form shown in the last line of Table 26.2. If the damping is positive, as it would be for a passive circuit having some loss, usually via positive resistors, then the natural frequencies lie in the left half s-plane, and y decays to zero at positive infinite time so that any transients in the circuit die out after a sufficient wait. The circuit is then called asymptotically stable. However, if the damping is negative, as it could be for some positive feedback circuits or those with negative resistance, then the response to nonzero initial conditions increases in amplitude without bound, although in an oscillatory manner, as time increases, and the circuit is said to be unstable. In the unstable case, as time decreases through negative time the amplitude also damps out to zero, but usually the responses backward in time are not of as much interest as those forward in time.

For the zero-state response, the impulse response, $h(t)$, is convoluted with the input, that is, $y = h * u$, for which we can use the fact that $h(t)$ is the inverse Laplace transform of $H(s) = C[sI_2 - A]^{-1}B$. The denominator of $H(s)$ is $\det(sI_2 - A) = s^2 + 2\zeta\omega_n s + \omega^2{}_n$, for which the causal inverse Laplace transform is

$$e^{s_+ t}1(t) * e^{s_- t}1(t) = \begin{cases} \dfrac{e^{s_+ t} - e^{s_- t}}{s_+ - s_-}1(t) & \text{if } s_- \neq s_+ \\ te^{s_+ t}1(t) & \text{if } s_- = s_+ \end{cases} \tag{26.43}$$

Here, the bottom case is ruled out when only complex natural frequencies are considered, following the assumption of handling real natural frequencies in first-order circuits, made previously. Consequently,

$$e^{s_+ t}1(t) * e^{s_- t}1(t) = \frac{e^{s_+ t} - e^{s_- t}}{s_+ - s_-}1(t) = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2\omega_n}} \sin\left(\sqrt{1-\zeta^2\omega_n t}\right) \cdot 1(t) \tag{26.44}$$

Again, assuming $\zeta^2 < 1$ using the preceding calculations give the zero-state response as

$$\begin{aligned} y(t) &= \frac{1}{d_2}\left\{ \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}\omega_n} \sin\left(\sqrt{1-\zeta^2}\omega_n t\right)1(t) * \right. \\ &\quad \left. \left[\left(n_1 - \frac{d_1}{d_2}n_2\right)\delta'(t) + \left(n_0 - \frac{d_0}{d_2}n_2\right)\delta(t)\right] + n_2\delta(t) \right\} * u(t) \\ &= \frac{1}{d_2}\left\{ \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}\omega_n} \sin\left(\sqrt{1-\zeta^2}\omega_n t\right)1(t) * \right. \\ &\quad \left. [n_2\delta''(t) + n_1\delta'(t) + n_0\delta(t)] \right\} * u(t) \end{aligned} \tag{26.45}$$

The bottom equivalent form is easily seen to result from writing the transfer function $H(p)$ as the product of two terms $1/[d_2(p^2 + 2\zeta\omega_n p + \omega_n^2)]$ and $[n_2 p^2 + n_1 p + n_0]$ convoluting the causal impulse response (the

inverse of the left half-plane converging Laplace transform), of each term. From Equation 26.45, we directly read the impulse response to be

$$h(t) = \frac{1}{d_2}\left\{ \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}\omega_n} \sin\left(\sqrt{1-\zeta^2}\omega_n t\right) 1(t) \right.$$
$$\left. *[n_2\delta''(t) + n_1\delta'(t) + n_0\delta(t)] \right\} \tag{26.46}$$

Equations 26.45 and 26.46 are readily evaluated further by noting that the convolution of a function with the second derivative of the impulse, the first derivative of the impulse, and the impulse itself is the second derivative of the function, the first derivative of the function, and the function itself, respectively. For example, in the low-pass case we find the impulse response to be, using Equation 26.46,

$$h_{\text{lp}}(t) = \frac{n_0}{d_2} \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}\omega_n} \sin\left(\sqrt{1-\zeta^2}\omega_n t\right) 1(t) \tag{26.47}$$

By differentiating we find the bandpass and then high-pass impulse responses to be, respectively,

$$h_{\text{bp}}(t) = \frac{n_1}{d_2} \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \cos\left(\sqrt{1-\zeta^2}\omega_n t + \theta\right) 1(t) \tag{26.48}$$

$$h_{\text{hp}}(t) = \frac{n_2}{d_2}\left[\delta(t) - \frac{\omega_n e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin\left(\sqrt{1-\zeta^2}\omega_n t + 2\theta\right) 1(t)\right] \tag{26.49}$$

In both cases, the added phase angle is given, as in the zero input response, via $\theta = \arctan2\left(\zeta/\sqrt{1-\zeta^2}\right)$. By adding these last three impulse responses suitably scaled the impulse responses of the more general second-order circuits are obtained.

Some comments on normalizations are worth mentioning in passing. Because $d_2 \neq 0$, one could assume d_2 to be 1 by absorbing its actual value in the transfer function numerator coefficients. If $\omega_n \neq 0$, time could also be scaled so that $\omega_n = 1$ could be taken, in which case a normalized time, t_n, is introduced.

Thus, $t = \omega_n t_n$ and, along with normalized time, comes a normalized differential operator $p_n = d[\cdot]/\mathrm{d}t_n = d[\cdot]/\mathrm{d}(t/\omega_n) = \omega_n p$. This, in turn, leads to a normalized transfer function by substituting $p = p_n/\omega_n$ into $H(p)$. Thus, much of the treatment could be carried out on the normalized transfer function x

$$H_n(p_n) = H(p) = \frac{n_{2n}p_n^2 + n_{1n}p_n + n_{0n}}{p_n^2 + 2\zeta p_n + 1} \quad p_n = \omega_n p \tag{26.50}$$

In this normalized form, it appears that the most important parameter in fixing the form of the response is the damping factor $\zeta = 1/(2Q)$.

26.3.3 Transient and Steady-State Responses

Let us now excite the circuit with an eternal exponential input, $u(t) = U\exp(\mathrm{st})$ for $-\infty < t < \infty$ at the complex frequency $s = \sigma + j\omega$, where s is chosen as different from either of the natural frequencies, $s_{\pm}$, and U is a constant. As with the first-order and, indeed, any higher-order, case the response is $y(t) = Y(s)\exp(\mathrm{st})$, as is observed by direct substitution into Equation 26.32. This substitution yields directly

$$Y(s) = \frac{1}{d_2}\left[\frac{n_2 s^2 + n_1 s + n_0}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] \cdot U \tag{26.51}$$

where $y(t) = Y(s)\exp(st)$ for $u(t) = U\exp(st)$ over $-\infty < t < \infty$. That is, an exponential excitation yields an exponential response at the same (complex) frequency $s = \sigma + j\omega$ as that for the input, as long as s is not one of the two natural frequencies. (s may have positive as well as negative real parts and is best considered as a frequency and not as the Laplace transform variable because the latter is limited to regions of convergence.) Because the denominator polynomial of $Y(\dot{s})$ has roots which are the natural frequencies, the magnitude of Y becomes infinite as the frequency of the excitation approaches s_+ or s_-. Thus, the natural frequencies s_+ and s_- are also called poles of the transfer function.

When $\sigma = 0$ the excitation and response are both sinusoidal and the resulting response is called the SSS. From Equation 26.51, the SSS response is found by substituting the complex frequency $s = j\omega$ into the transfer function, now evaluated on complex numbers rather than differential operators as above,

$$H(s) = \frac{1}{d_2}\left[\frac{n_2 s^2 + n_1 s + n_0}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] \tag{26.52}$$

Next, an exponential input is applied, which starts at $t = 0$ instead of at $t = -\infty$; i.e., $u(t) = U\exp(st)1(t)$. Then, the output is found by using the convolution $y = h * u$, which, from the discussion at Equation 26.45, is expressed as

$$\begin{aligned} y(t) = h * u &= \frac{1}{d_2} e^{s_+ t}1(t) * e^{s_- t}1(t) * [n_2\delta''(t) + n_1\delta'(t) + n_0\delta(t)] * e^{st}1(t) \\ &= H(s)Ue^{st}1(t) + \left\{\frac{1}{d_2(s_+ - s_-)}\left[\left(\frac{N(s)}{s_+ - s} + n_2(s + s_+) + n_1\right)e^{s_+ t}\right.\right. \\ &\quad \left.\left. -\left(\frac{N(s)}{s_+ - s} + n_2(s + s_-) + n_1\right)e^{s_- t}\right]1(t)\right\} \end{aligned} \tag{26.53}$$

in which $N(s)$ is the numerator of the transfer function and we have assumed that s is not equal to a natural frequency. The second term on the right within the braces varies at the natural frequencies and as such is called the transient response, while the first term is the term resulting directly from an eternal exponential, but now with the negative time portion of the response removed. If the system is stable, the transient response decays to zero as time increases and, thus, if we wait long enough the transient response of a stable system can be ignored if the complex frequency of the input exponential has a real part that is greater than that of the natural frequencies. Such is the case for exponentials that yield sinusoids; in that case $\sigma = 0$, or $s = j\omega$. In other words, for an asymptotically stable circuit the output approaches that of the SSS when the input frequency is purely imaginary. If we were to excite at a natural frequency then the first part of Equation 26.53 still could be evaluated using the time-multiplied exponential of Equation 26.43; however, the transient and the steady state are now mixed, both being at the same "frequency."

Because actual sinusoidal signals are real, we use superposition and the fact that the real part of a complex signal is given by adding complex conjugate terms:

$$\cos(\omega t) = \Re[e^{j\omega t}] = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \tag{26.54}$$

This leads to the SSS response for an asymptotically stable circuit excited by $u(t) = U\cos(\omega t)1(t)$ to be

$$\begin{aligned} y(t) &= \frac{H(j\omega)Ue^{j\omega t} + H(-j\omega)U^* e^{-j\omega t}}{2} \\ &= |H(j\omega)||U|\cos(\omega t + \angle H(j\omega) + \angle U) \end{aligned} \tag{26.55}$$

Here, we assumed that the circuit has real-valued components such that $H(-j\omega)$ is the complex conjugate of $H(j\omega)$. In which case, the second term in the middle expression is the complex conjugate of the first.

26.3.4 Network Characterization

Although the impulse response is useful for theoretical studies, it is difficult to observe it experimentally due to the impossibility of creating an impulse. However, the unit step response is readily measured, and from it the impulse response actually can be obtained by numerical differentiation if needed. However, it is more convenient to work directly with the unit step response and, consequently, practical characterizations can be based upon it. The treatment most conveniently proceeds from the normalized low-pass transfer function

$$H(p) = \frac{1}{p^2 + 2\zeta p + 1}, \quad 0 < \zeta < 1 \tag{26.56}$$

The unit step response follows by applying the input $u(t) = 1(t)$ and noting that the unit step is the special case of an exponential multiplied unit step, where the frequency of the exponential is zero. Conveniently, Equation 26.43 can be used to obtain

$$y_{\text{us}}(t) = 1(t) - \frac{e^{-\zeta t}}{\sqrt{1-\zeta^2}} \cos\left(\sqrt{1-\zeta^2}t - \theta\right) \cdot 1(t), \quad \theta = \arctan\left(\frac{\zeta}{\sqrt{1-\zeta^2}}\right) \tag{26.57}$$

Typical unit step responses are plotted in Figure 26.3 where, for a small damping factor, overshoot can be considerable, with oscillations around the final value and in addition, a long settling time before reaching the final value. In contrast, with a large damping factor, although no overshoot or oscillation occurs, the rise to the final value is long. A compromise for obtaining a quick rise to the final value with no oscillations is given by choosing a damping factor of 0.7, this being called the critical value; i.e., critical damping is $\zeta_{\text{crit}} = 0.7$.

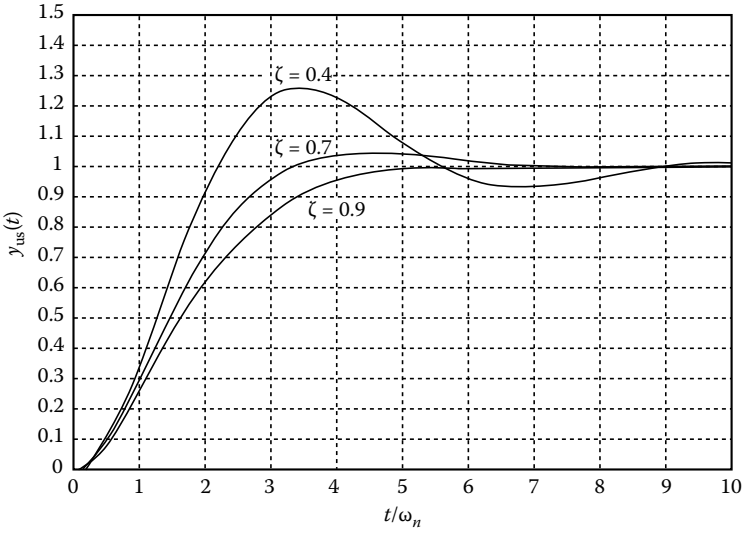


FIGURE 26.3 Unit step response for different damping factors.

References

1. L. P. Huelsman, *Basic Circuit Theory with Digital Computations*, Englewood Cliffs, NJ: Prentice Hall, 1972.
2. V. I. Arnold, *Ordinary Differential Equations*, Cambridge, MA: MIT Press, 1983.
3. J. E. Kardontchik, *Introduction to the Design of Transconductor-Capacitor Filters*, Boston: Kluwer Academic Publishers, 1992.
4. R. P. Sallen and E. L. Key, A practical method of designing RC active filters, *IRE Trans. Circuit Theory*, CT-2(1), 74–85, March 1955.

27

State-Variable Techniques

27.1 Concept of States **27**-1
27.2 State-Variable Formulation via Network Topology..... **27**-3
27.3 Natural Response and State Transition Matrix **27**-12
27.4 Complete Response **27**-20
References **27**-21

Kwong S. Chao
Texas Tech University

27.1 Concept of States

For resistive (or memoryless) circuits, given the circuit structure, the present output depends only on the present input. In order to analyze a dynamic circuit, however, in addition to the present input it is also necessary to know the state of the circuit at some time t_0. The state of the circuit at t_0 represents the condition of the circuit at $t = t_0$, and is related to the energy storage of the circuit, or the voltage (or electric charge) across the capacitor and the currents (or magnetic fluxes) through the inductors. These voltages and currents are considered as the state of the circuit at $t = t_0$. For $t > t_0$, the behavior of the circuit is completely characterized by these variables. In view of the preceding, a definition for the state of a circuit can now be given.

Definition: The state of a circuit at time t_0 is the minimum amount of information at t_0 that, along with the input to the circuit for $t \geq t_0$, uniquely determines the behavior of the circuit for $t \geq t_0$.

The concept of states is closely related to the order of complexity of the circuit. The order of complexity of a circuit is the minimum number of initial conditions which, along with the input, is sufficient to determine the future behavior of the circuit. Furthermore, if a circuit is described by an nth-order linear differential equation, it is well-known that the general solution for $t \geq t_0$ contains n arbitrary constants which are determined by n initial conditions. This set of n initial conditions contains information concerning the circuit prior to $t = t_0$ and constitutes the state of the circuit at $t = t_0$. Thus, the order of complexity or the order of a circuit is the same as the order of the differential equation that describes the circuit, and it is also the same as the number of state variables that can be defined in a circuit. For an nth-order circuit, the state of the circuit at $t = t_0$ consists of a set of n numbers that denotes a vector in an n-dimensional state space spanned by the n corresponding state variables. This key number n can simply be obtained by inspection of the circuit. Knowing the total number of energy storage elements, n_{LC}, the total number of independent capacitive loops, n_C, and the total number of independent inductive cutsets, n_L, the order of complexity n of a circuit is given by

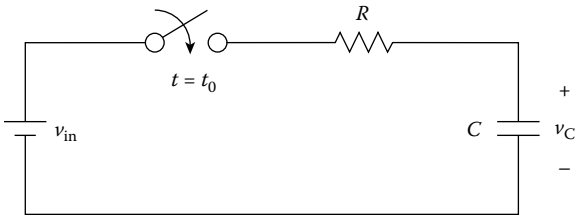


FIGURE 27.1 Simple RC circuit.

$$n = n_{LC} - n_L - n_C \tag{27.1}$$

A capacitive loop is defined as one that consists of only capacitors and possibly voltage sources while an inductive cutset represents a cutset that contains only inductors and possibly current sources. The following two examples illustrate the concept of states.

Example 27.1

Consider a simple *RC*-circuit in Figure 27.1. The circuit equation is

$$RC\frac{dv_c(t)}{dt} + v_c(t) = v_{in} \quad \text{for } t \geq t_0 \tag{27.2}$$

and the corresponding capacitor voltage is easily obtained as

$$v_c(t) = [v_c(t_0) - v_{in}]e^{-\frac{1}{RC}(t-t_0)} + v_{in} \quad \text{for } t \geq t_0 \tag{27.3}$$

For this first-order circuit, it is clear from Equation 27.3 the capacitor voltage for $t \geq t_0$ is uniquely determined by the initial condition $v_c(t_0)$ and the input voltage v_{in} for $t \geq t_0$. This is independent of the charging circuit for the capacitor prior to t_0. Hence, $v_c(t_0)$ is the state of the circuit at $t = t_0$ and $v_c(t)$ is regarded as the state variable of the circuit.

Example 27.2

As another illustration, consider the circuit of Figure 27.2, which is a slight modification of the circuit considered in the previous example. The circuit equation and its corresponding solution are readily obtained as

$$\frac{dv_{C_1}(t)}{dt} = -\frac{1}{R(C_1 + C_2)}v_{C_1}(t) + \frac{1}{R(C_1 + C_2)}v_{in} \tag{27.4}$$

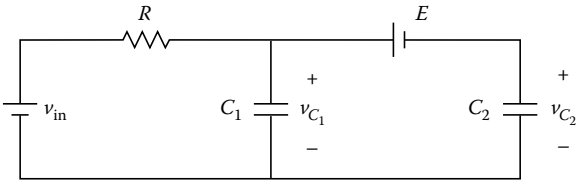


FIGURE 27.2 Circuit for Example 27.2.

and

$$v_{C_1}(t) = (v_{C_1}(t_0) - v_{\text{in}})e^{\frac{-1}{R(C_1+C_2)}(t-t_0)} + v_{\text{in}} \quad \text{for } t \geq t_0 \tag{27.5}$$

respectively. Even though two energy storage elements exist, one can only arbitrarily specify one independent initial condition. Once the initial condition on C_1, $v_{C_1}(t_0)$, is specified, the initial voltage on C_2 is automatically constrained by the loop equation $v_{C_2}(t) = v_{C_1}(t) - E$ at t_0. The circuit is thus still first order and only one state variable can be assigned for the circuit. It is clear from Equation 27.5 that with the input v_{in}, $v_{C_1}(t_0)$ is the minimum amount of information that is needed to uniquely determine the behavior of this circuit. Hence, $v_{C_1}(t)$ is the state variable of the circuit. One can just as well analyze the circuit by solving a first-order differential equation in terms of $v_{C_2}(t)$ with $v_{C_2}(t_0)$ defined as the state of the circuit at $t = t_0$. The selection of state variables is thus not unique. In this example, either $v_{C_1}(t)$ or $v_{C_2}(t)$ can be defined as the state variable of the circuit. In fact, it is easily shown that any linear combination of $v_{C_1}(t)$ and $v_{C_2}(t)$ can also be regarded as state variables.

27.2 State-Variable Formulation via Network Topology

Various mathematical descriptions of circuits are available. Depending on the type of analysis used, different formulations of circuit equations may result. In the state variable formulation, a system of n first-order differential equations is written in the form

$$\dot{\mathbf{x}} = \mathbf{f}\ (\mathbf{x}, t) \tag{27.6}$$

where

$\mathbf{x}$ is an $n \times 1$ vector consisting of n state variables for an nth-order circuit
t represents the time variable

This set of equations is usually referred to as the state equation in normal form.

When compared with other circuit descriptions, the state-variable representation is not necessarily the simplest. It does, however, simultaneously provide the solution of all state variables and hence yields the behavior of the entire circuit. The state equation is also particularly suitable for analysis by numerical techniques. Another distinct advantage of the state-variable approach is that it can be easily extended to nonlinear and/or time varying circuits.

Example 27.3

Consider the linear circuit of Figure 27.3. By inspection, the order of complexity of this circuit is three. Hence, three state variables are selected as $x_1 = v_{C_1}$, $x_2 = v_{C_2}$, and $x_3 = i_{\text{L}}$. Because the left-hand side of the normal form equation is the derivative of the state vector, it is necessary to express the voltage across

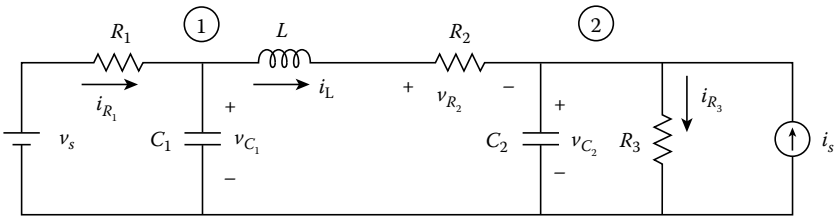


FIGURE 27.3 Circuit for Example 27.3.

the inductors and the currents through the capacitors in terms of the state variables and the input sources.

The current through C_1 can be obtained by writing a Kirchhoff's current law (KCL) equation at node 1 to yield

$$\begin{aligned} C_1 \frac{dv_{C_1}}{dt} &= i_{R_1} - i_L \\ &= \frac{1}{R_1}(v_s - v_{C_1}) - i_L \end{aligned}$$

or

$$\frac{dv_{C_1}(t)}{dt} = -\frac{1}{R_1 C_1} v_{C_1} - \frac{1}{C_1} i_L + \frac{1}{R_1 C_1} v_s \tag{27.7}$$

In a similar manner, applying KCL to node 2 gives

$$\begin{aligned} C_2 \frac{dv_{C_2}}{dt} &= i_L - i_{R_3} + i_s \\ &= i_L - \frac{1}{R_3} v_{C_2} + i_s \end{aligned}$$

or

$$\frac{dv_{C_2}(t)}{dt} = -\frac{1}{R_3 C_2} v_{C_2} + \frac{1}{C_2} i_L + \frac{1}{C_2} i_s \tag{27.8}$$

The expression for the inductor voltage is derived by applying KVL to the mesh containing L, R_2, C_2, and C_1 yielding

$$L \frac{di_L}{dt} = v_{C_1} - v_{C_2} - R_2 i_L$$

or

$$\frac{di_L}{dt} = \frac{1}{L} v_{C_1} - \frac{1}{L} v_{C_2} - \frac{R_2}{L} i_L \tag{27.9}$$

Equations 27.7 through 27.9 are the state equations that can be expressed in matrix form as

$$\begin{bmatrix} \frac{dv_{C_1}}{dt} \\ \frac{dv_{C_2}}{dt} \\ \frac{di_L}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C_1} & 0 & -\frac{1}{C_1} \\ 0 & -\frac{1}{R_3 C_2} & \frac{1}{C_2} \\ \frac{1}{L} & -\frac{1}{L} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1 C_1} & 0 \\ 0 & \frac{1}{C_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_s \\ i_s \end{bmatrix} \tag{27.10}$$

Any number of branch voltages and/or currents may be chosen as output variables. If i_{R_1} and v_{R_2} are considered as outputs for this example, then the output equations, written as a linear combination of state variables and input sources become

$$i_{R_1} = \frac{1}{R_1}(v_s - v_{C_1}) \tag{27.11}$$

$$v_{R_2} = R_2 i_L \tag{27.12}$$

or in matrix form

$$\begin{bmatrix} i_{R_1} \\ v_{R_2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1} & 0 & 0 \\ 0 & 0 & R_2 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_s \\ i_s \end{bmatrix} \tag{27.13}$$

In general, for an nth-order linear circuit with r input sources and m outputs, the state and output equations are represented by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{27.14}$$

and

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \tag{27.15}$$

where

$\mathbf{x}$ is an $n \times 1$ state vector
$\mathbf{u}$ is an $r \times 1$ vector representing the r input sources
$m \times 1$-vector $\mathbf{y}$ denotes the m output variables, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are of order $n \times n$, $n \times r$, $m \times n$, and $m \times r$, respectively

In the preceding example, the state equations are obtained by inspection for a simple circuit by writing voltage equations for inductors and current equations for capacitors and properly eliminating the nonstate variables. For more complicated circuits, a systematic procedure for eliminating the nonstate variables is desirable. Such a procedure can be generated with the aid of a proper tree. A proper tree is a tree obtained from the associated network graph that contains all capacitors, independent voltage sources, and possibly some resistive elements, but does not contain inductors and independent current sources. The selection of such a tree is always possible if the circuit contains no capacitive loops and no inductive cutsets. The reason for providing such a tree for writing state equations is obvious. With each tree branch, there is a unique cutset known as the fundamental cutset that contains only one tree branch and some links. Thus, if capacitors are in the tree, a fundamental cutset equation may be written for the corresponding currents through the capacitors. Similarly, every link (together with some tree branches) forms a unique loop called a fundamental loop. If inductors are selected as links, inductor voltages may be obtained by writing the corresponding fundamental loop equations. With the selection of a proper tree, state variables can be defined as the capacitor tree-branch voltages and inductive link currents. In view of the above observation, a systematic procedure for writing state equations can now be stated as follows:

Step 1: From the associated directed graph, pick a proper tree.

Step 2: Write fundamental cutset equations for the capacitive tree branches and express the capacitor currents in terms of link currents.

Step 3: Write fundamental loop equations for the inductive links and express the inductor voltages in terms of tree-branch voltages.

Step 4: Define the state variables. Capacitive tree-branch voltages and inductive link currents are selected as state variables. Other quantities such as capacitor charges and inductor fluxes may also be used.

Step 5: Group the branch relations and the remaining fundamental equations according to their element types into three sets: resistor, inductor, and capacitor equations. Solve for the nonstate variables that appeared in the equations obtained in Steps 2 and 3 from the corresponding set of equations in terms of the state variables and independent sources.

Step 6: Substitute the result of Step 5 into the equations obtained in Steps 2 and 3, and rearrange them in normal form.

Example 27.4

Consider again the same circuit in Figure 27.3. The various steps outlined previously are used to write the state equations.

Step 1: The associated graph and the proper tree of the circuit are shown in Figure 27.4. The tree branches include v_s, C_1, C_2, and R_2.

Step 2: The fundamental cutset associated with C_1 consists of tree branch C_1 and two links R_1 and L. By writing the current equation for this cutset, the capacitor current i_{c1} is expressed in terms of link currents as

$$i_{C_1} = i_{R_1} - i_L \tag{27.16}$$

Similarly, the fundamental cutset $\{L, C_2, R_3, i_s\}$ associated with C_2 leads to

$$i_{C_2} = i_L - i_{R_3} + i_s \tag{27.17}$$

Step 3: The fundamental loop associated with link L consists of L and tree branches R_2, C_2, and C_1. By writing the voltage equation around this loop, the inductor voltage can be written in terms of tree-branch voltages as

$$v_L = v_{C_1} - v_{C_2} - v_{R_2} \tag{27.18}$$

Step 4: The tree-branch capacitor voltages v_{C_1}, V_{C_1}, and inductive link current i_L are defined as the state variables of the circuit.

Step 5: The branch relation and the remaining two fundamental loops for R_1 and R_2, and the fundamental cutset equation for R_2 are grouped into three sets.

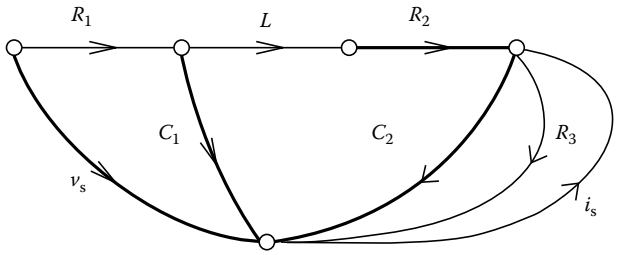


FIGURE 27.4 Directed graph associated with the circuit of Figure 27.3.

Resistor equations:

$$v_{R_1} + v_{C_1} - v_s = 0 \tag{27.19}$$

$$i_{R_1} = \frac{1}{R_1} v_{R_1} \tag{27.20}$$

$$i_{R_2} - i_L = 0 \tag{27.21}$$

$$v_{R_2} = R_2 i_{R_2} \tag{27.22}$$

$$v_{R_3} - v_{C_2} = 0 \tag{27.23}$$

$$v_{R_3} = R_3 i_{R_3} \tag{27.24}$$

Inductor equations:

$$\phi_L = L i_L \quad \text{or} \quad v_L = \frac{d\phi_L}{dt} = L \frac{di_L}{dt} \tag{27.25}$$

Capacitor equations:

$$q_1 = C_1 v_{C_1} \quad \text{or} \quad i_{C_1} = \frac{dq_1}{dt} = C_1 \frac{dv_{C_1}}{dt} \tag{27.26}$$

$$q_2 = C_2 v_{C_2} \quad \text{or} \quad i_{C_2} = \frac{dq_2}{dt} = C_2 \frac{dv_{C_2}}{dt} \tag{27.27}$$

The resistive link currents r_{R_1}, i_{R_3}, and resistive tree-branch voltage V_{R_2} are solved from Equations 27.19 through 27.24 in terms of the inductive link current i_L, the capacitive tree-branch voltages v_{C_1} and v_{C_2}, and sources as

$$i_{R_1} = \frac{1}{R_1}(v_s - v_{C_1}) \tag{27.28}$$

$$i_{R_3} = \frac{1}{R_3} v_{C_2} \tag{27.29}$$

and

$$v_{R_2} = R_2 i_L \tag{27.30}$$

For this example, i_L, v_{C_1}, and v_{C_2} have already been defined as state variables.

Step 6: Substituting Equations 27.28 through 27.30 into Equations 27.16 through 27.18 yields the desired state equation in matrix form:

$$\begin{bmatrix} \frac{dv_{C_1}}{dt} \\ \frac{dv_{C_2}}{dt} \\ \frac{di_L}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1 C_1} & 0 & -\frac{1}{C_1} \\ 0 & -\frac{1}{R_3 C_2} & \frac{1}{C_2} \\ \frac{1}{L} & -\frac{1}{L} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1 C_1} & 0 \\ 0 & \frac{1}{C_2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_s \\ i_s \end{bmatrix} \tag{27.31}$$

which, as expected, is the same as Equation 27.10 obtained previously by inspection.

As mentioned earlier, the selection of state variables is not unique. Instead of using capacitor voltages and inductor currents as state variables, basic quantities such as the capacitor charges and inductor fluxes may also be considered. If q_1, q_2, and ϕ_L are defined as state variables in Step 4, the inductive link current i_L and capacitive tree-branch voltages, v_{C_1} and v_{C_2}, can be solved from the inductor and capacitor equations in terms of state variables and possibly sources in Step 5 as

$$i_L = \frac{1}{L}\phi_L \tag{27.32}$$

$$v_{C_1} = \frac{1}{C_1}q_1 \tag{27.33}$$

$$v_{C_2} = \frac{1}{C_2}q_2 \tag{27.34}$$

Finally, state equations are obtained by substituting Equations 27.28 through 27.30 and Equations 27.32 through 27.34 into Equations 27.16 through 27.18 as

$$\begin{bmatrix} \frac{dq_1}{dt} \\ \frac{dq_2}{dt} \\ \frac{d\phi_L}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1C_1} & 0 & -\frac{1}{L} \\ 0 & -\frac{1}{R_3C_2} & \frac{1}{L} \\ \frac{1}{C_1} & -\frac{1}{C_2} & -\frac{R_2}{L} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \phi_L \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_s \\ i_s \end{bmatrix} \tag{27.35}$$

In the systematic procedure outlined previously, it is assumed that the network exists with neither inductive cutsets nor capacitive loops so that the selection of proper tree is always guaranteed. For networks that do have these constraints, it is not possible to include all the capacitors in a tree without forming a closed path. Also, in order for a tree to contain all the nodes, some inductors will have to be included in a tree. A tree that includes independent voltage sources, some resistors, and a maximum number of capacitors but no independent current sources is called a modified proper tree. In writing a state equation for such networks, the same systematic procedure can be applied with the selection of a modified proper tree. However, if capacitor tree-branch voltages and inductive link currents are defined as the state variables, the standard (**A**, **B**, **C**, **D**) description (Equations 27.14 and 27.15) may not exist. In fact, if inductive cutsets contain independent current sources and/or capacitive loops contain independent voltage sources, the derivative of these sources will appear in the state equation and the general equation is of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{u} + \mathbf{B}_2\dot{\mathbf{u}} \tag{27.36}$$

where $\mathbf{B}_1$ and $\mathbf{B}_2$ are $n \times r$ matrices and **A**, **x**, and **u** are defined as before. To recast Equation 27.36 into the standard form, it is necessary to redefine.

$$\mathbf{z} = \mathbf{x} - \mathbf{B}_2\mathbf{u} \tag{27.37}$$

as new state variables. Substituting Equation 27.37 into 27.36, yields

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u} \tag{27.38}$$

where

$$\mathbf{B} = \mathbf{B}_1 + \mathbf{A}\mathbf{B}_2 \tag{27.39}$$

It is noted from Equation 27.37, the new state variables represent a linear combination of sources and capacitor voltages or inductor currents which, except for the mathematical convenience, may not have sound physical significance. To avoid such state variables and transformation (Equation 27.37), Step 4 of the systematic procedure described earlier needs to be modified. By defining state variables as the algebraic sum of capacitor charges in the fundamental cutset associated with each of the capacitor tree branches, and the algebraic sum of inductor fluxes in the fundamental loop associated with each of the inductive links, the resulting state equation will be in the standard form. The preceding generalizations are illustrated by the following two examples.

Example 27.5

As a simple illustration, consider the same circuits given in Figure 27.2, where the constant DC voltage source E is replaced by a time-varying source $e(t)$. It can easily be demonstrated that the equation describing the circuit now becomes

$$\frac{dv_{C_1}(t)}{dt} = -\frac{1}{R(C_1 + C_2)}v_{C_1} + \frac{1}{R(C_1 + C_2)}v_{in}(t) + \frac{C_2}{R(C_1 + C_2)}\frac{de(t)}{dt} \tag{27.40}$$

The preceding equation is the same as the state Equation 27.4 with the exception of an additional term involving the first-order derivative of source $e(t)$. Equation 27.40 is clearly not the standard state equation described in Equation 27.41 with capacitor voltage v_{C_1} defined as the state variable.

Example 27.6

As another illustration, consider the circuit shown in Figure 27.5 which consists of an inductive cutset $\{L_1, L_2, i_s\}$ and a capacitive loop (C_1, v_{s_2}, C_2). The state equations are determined from the systematic procedure by first using the transformation (Equation 27.37) and then by defining the algebraic sum of charges and fluxes as state variables.

Step 1: The directed graph of the circuit is shown in Figure 27.6 where branches denoted by v_{s_1}, v_{s_2}, C_1, R_2, and L_2 are selected to form a modified proper tree.

Step 2: The fundamental cutset associated with C_1 consists of branches R_1, C_1, L_1, i_s, C_2, and R_3. Applying KCL to this cutset yields

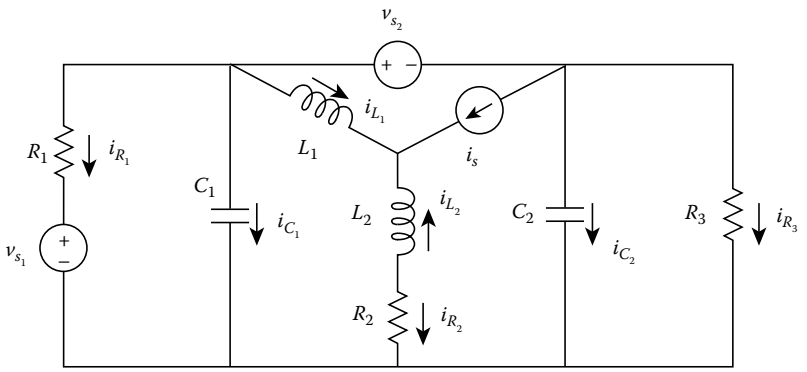


FIGURE 27.5 Circuit with a capacitive loop and an inductive cutset.

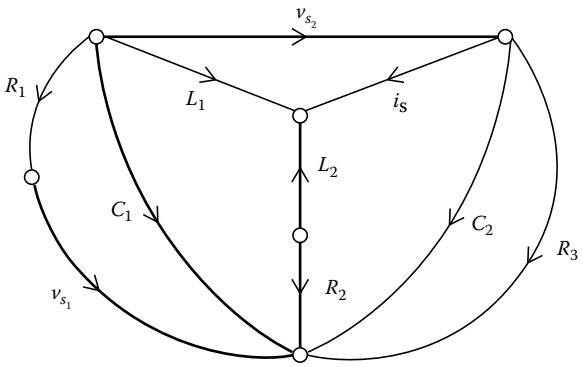


FIGURE 27.6 Directed graph associated with the circuit of Figure 27.5.

$$i_{C_1} = -i_{R_1} - i_{L_1} - i_s - i_{C_2} - i_{R_3} \tag{27.41}$$

Step 3: The fundamental loop equation associated with the inductive link L_1 is given by

$$v_{L_1} = v_{C_1} + v_{L_2} - v_{R_2} \tag{27.42}$$

where the link voltage v_{L_1} has been expressed in terms of tree-branch voltages.

Step 4: In the first illustration, the tree-branch capacitor voltage v_{C_1} and the inductive link current i_{L_1} are defined as the state variables.

Step 5: The branch relation and the remaining two fundamental equations are grouped into the following three sets:

Resistor equations:

$$v_{R_1} + v_{s_1} - v_{C_1} = 0 \tag{27.43}$$

$$v_{R_1} = R_1 i_{R_1} \tag{27.44}$$

$$i_{R_2} - i_{L_1} - i_s = 0 \tag{27.45}$$

$$v_{R_2} = R_2 i_{R_2} \tag{27.46}$$

$$v_{R_3} - v_{C_1} + v_{s_2} = 0 \tag{27.47}$$

$$v_{R_3} = R_3 i_{R_3} \tag{27.48}$$

Inductor equations:

$$i_{L_2} + i_{L_1} + i_s = 0 \tag{27.49}$$

$$\phi_{L_1} = L_1 i_{L_1} \quad \text{or} \quad v_{L_1} = L_1 \frac{\mathrm{d}i_{L_1}}{\mathrm{d}t} \tag{27.50}$$

$$\phi_{L_2} = L_2 i_{L_2} \quad \text{or} \quad v_{L_2} = L_2 \frac{\mathrm{d}i_{L_2}}{\mathrm{d}t} \tag{27.51}$$

Capacitor equations:

$$v_{C_2} - v_{C_1} + v_{s_2} = 0 \tag{27.52}$$

$$q_1 = C_1 v_{C_1} \quad \text{or} \quad i_{C_1} = C_1 \frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} \tag{27.53}$$

$$q_2 = C_2 v_{C_2} \quad \text{or} \quad i_{C_2} = C_2 \frac{\mathrm{d}v_{C_2}}{\mathrm{d}t} \tag{27.54}$$

For this example, the nonstate variables are identified as i_{R_1}, v_{R_2}, i_{R_3}, v_{L_2}, and i_{C_2}, from Equations 27.41 and 27.42. These variables are now solved from the corresponding group of equations in terms of state variables and independent sources:

$$i_{R_1} = \frac{1}{R_1}(v_{C_1} - v_{s_1}) \tag{27.55}$$

$$v_{R_2} = R_2(i_{L_1} + i_s) \tag{27.56}$$

$$i_{R_3} = \frac{1}{R_3}(v_{C_1} - v_{s_2}) \tag{27.57}$$

$$v_{L_2} = -L_2 \frac{\mathrm{d}i_{L_1}}{\mathrm{d}t} - L_2 \frac{\mathrm{d}i_s}{\mathrm{d}t} \tag{27.58}$$

$$i_{C_2} = C_2 \frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} - C_2 \frac{\mathrm{d}v_{s_2}}{\mathrm{d}t} \tag{27.59}$$

Step 6: Assuming the existence of the first-order derivatives of sources with respect to time and substituting Equations 27.50, 27.53, and 27.55 through 27.59 into Equations 27.41 and 27.42 yields

$$\begin{bmatrix} \frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} \\ \frac{\mathrm{d}i_{L_1}}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} -\frac{R_1+R_3}{R_1R_3(C_1+C_2)} & -\frac{1}{C_1+C_2} \\ \frac{1}{L_1+L_2} & -\frac{R_2}{L_1+L_2} \end{bmatrix} \begin{bmatrix} v_{C_1} \\ i_{L_1} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1(C_1+C_2)} & \frac{1}{R_3(C_1+C_2)} & -\frac{1}{(C_1+C_2)} \\ 0 & 0 & -\frac{R_2}{L_1+L_2} \end{bmatrix} \begin{bmatrix} v_{s_1} \\ v_{s_2} \\ i_s \end{bmatrix} + \begin{bmatrix} 0 & \frac{C_2}{(C_1+C_2)} & 0 \\ 0 & 0 & -\frac{L_2}{L_1+L_2} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_{s_1}}{\mathrm{d}t} \\ \frac{\mathrm{d}v_{s_2}}{\mathrm{d}t} \\ \frac{\mathrm{d}i_s}{\mathrm{d}t} \end{bmatrix} \tag{27.60}$$

Clearly, Equation 27.60 is not in the standard form. Applying transformation (Equation 27.37) with $x_1 = v_{C_1}$, $x_1 = i_{L_2}$, $u_1 = v_{C_1}$, $u_2 = v_{s_2}$, and $u_3 = i_s$ gives the state equation in normal form

$$\begin{bmatrix} \frac{\mathrm{d}z_1}{\mathrm{d}t} \\ \frac{\mathrm{d}z_2}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} -\frac{R_1+R_2}{R_1R_3(C_1+C_2)} & -\frac{1}{C_1+C_2} \\ \frac{1}{L_1+L_2} & -\frac{R_2}{L_1+L_2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1(C_1+C_2)} & \frac{R_1C_1-R_3C_2}{R_1R_3(C_1+C_2)^2} & -\frac{L_1}{(L_1+L_2)(C_1+C_2)} \\ 0 & \frac{C_2}{(L_1+L_2)(C_1+C_2)} & -\frac{R_2L_1}{(L_1+L_2)^2} \end{bmatrix} \times \begin{bmatrix} v_{s_1} \\ v_{s_2} \\ i_s \end{bmatrix} \tag{27.61}$$

where new state variables are defined as

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} v_{C_1} & -\frac{C_2}{C_1+C_2} v_{s_2} \\ i_{L_1} & +\frac{L_2}{L_1+L_2} i_s \end{bmatrix} \tag{27.62}$$

Alternatively, if the state variables are defined in Step 4 as

$$q_a = q_1 + q_2 \tag{27.63}$$

$$\phi_b = \phi_1 - \phi_2 \tag{27.64}$$

then Equations 27.41 and 27.42 become

$$\frac{dq_a}{dt} = \frac{dq_1}{dt} + \frac{dq_2}{dt} = -i_{R_1} - i_{L_1} - i_s - i_{R_3} \tag{27.65}$$

$$\frac{d\phi_b}{dt} = \frac{d\phi_1}{dt} - \frac{d\phi_2}{dt} = -v_{L_1} - v_{L_2} = v_{C_1} - v_{R_2} \tag{27.66}$$

respectively. In Step 5, the resistive link currents i_{R_1}, i_{R_3}, and the resistive tree-branch voltage V_{R_2} are solved from resistive Equations 27.43 through 27.48 in terms of inductive link currents, capacitive tree-branch voltages, and independent sources. The results are those given in Equations 27.55 through 27.57. By solving the inductor Equations 27.49, 27.50, and 27.64, inductive link current i_{L_1} is expressed as a function of state variables and independent sources:

$$i_{L_1} = \frac{1}{L_1 + L_2}(\phi_b - L_2 i_s) \tag{27.67}$$

Similarly, solving v_{C_1} from capacitor Equations 27.52 through 27.54, and 27.63, yields the capacitor tree-branch voltage

$$v_{C_1} = \frac{1}{C_1 + C_2}(q_a + C_2 v_{s_2}) \tag{27.68}$$

Finally, in Step 6, Equations 27.55 through 27.57, 27.67, and 27.68 are substituted into Equations 27.65 and 27.66 to form the state equation in normal form:

$$\begin{bmatrix} \frac{dq_a}{dt} \\ \frac{d\phi_b}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_1+R_3}{R_1R_3(C_1+C_2)} & -\frac{1}{L_1+L_2} \\ \frac{1}{C_1+C_2} & -\frac{R_2}{L_1+L_2} \end{bmatrix} \begin{bmatrix} q_a \\ \phi_b \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & \frac{R_1C_1-R_3C_2}{R_1R_3(C_1+C_2)} & -\frac{L_1}{(L_1+L_2)} \\ 0 & \frac{C_2}{(C_1+C_2)} & -\frac{R_2L_1}{(L_1+L_2)} \end{bmatrix} \begin{bmatrix} v_{s_1} \\ v_{s_2} \\ i_s \end{bmatrix} \tag{27.69}$$

27.3 Natural Response and State Transition Matrix

In Section 27.2, the state-variable description has been presented for linear time-invariant circuits. The response of the circuit depends on the solution of the state equation. The behavior of the circuit due to any arbitrary input sources can easily be obtained once the zero-input response or the natural response of the circuit is known. In order to find its natural response, the homogeneous state equation of the circuit.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{27.70}$$

is considered, where independent source term $u(t)$ has been set equal to zero. The preceding state equation is analogous to the scalar equation

$$\dot{x} = ax \tag{27.71}$$

where the solution is given by

$$x(t) = e^{at}x(0) \tag{27.72}$$

for any arbitrary initial condition $x(0)$ given at $t = 0$, or

$$x(t) = e^{a(t-t_0)}x(t_0) \tag{27.73}$$

if the initial time is specified at $t = t_0$.

It is thus reasonable to assume a solution for Equation 27.70 of the form

$$\mathbf{x}(t) = e^{(t-t_0)\lambda}\mathbf{p} \tag{27.74}$$

where λ is a scalar constant and $\mathbf{p}$ is a constant n-vector. Substituting Equation 27.74 into Equation 27.70 leads to

$$\mathbf{A}\mathbf{p} = \lambda\mathbf{p} \tag{27.75}$$

Therefore, Equation 27.74 is a solution of Equation 27.70 precisely when $\mathbf{p}$ is an eigenvector of $\mathbf{A}$ associated with the eigenvalue λ. For simplicity, it is assumed that $\mathbf{A}$ has n distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Because the corresponding eigenvectors denoted by $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$ are linearly independent, the general solution of Equation 27.70 can be uniquely written as a linear combination of n distinct normal modes of the form Equation 27.74:

$$\mathbf{x}(t) = c_1 e^{(t-t_0)\lambda_1}\mathbf{p}_1 + c_2 e^{(t-t_0)\lambda_2}\mathbf{p}_2 + \cdots + c_n e^{(t-t_0)\lambda_n}\mathbf{p}_n \tag{27.76}$$

where $c_1, c_2, \ldots, c_n$ are n arbitrary constants determined by the given initial conditions. Specifically,

$$\mathbf{x}(t_0) = c_1\mathbf{p}_1 + c_2\mathbf{p}_2 + \cdots + c_n\mathbf{p}_n \tag{27.77}$$

The general solution (Equation 27.76) can also be written in the form

$$\mathbf{x}(t) = e^{(t-t_0)\mathbf{A}}\mathbf{x}(t_0) \tag{27.78}$$

where the exponential function of a matrix is defined by a power series:

$$\begin{aligned} e^{(t-t_0)\mathbf{A}} &= \mathbf{I} + (t-t_0)\mathbf{A} + \frac{(t-t_0)^2}{2!}\mathbf{A}^2 + \cdots \\ &= \sum_{k=0}^{\infty} \frac{(t-t_0)^k}{k!}\mathbf{A}^k \end{aligned} \tag{27.79}$$

In fact, taking the derivative of Equation 27.78 with respect to t yields

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= \frac{d}{dt}\left[\mathbf{I} + (t-t_0)\mathbf{A} + \frac{(t-t_0)^2}{2!}\mathbf{A}^2 + \cdots\right]\mathbf{x}(t_0) \\ &= \left[\mathbf{A} + (t-t_0)\mathbf{A}^2 + \frac{(t-t_0)^2}{2!}\mathbf{A}^3 + \cdots\right]\mathbf{x}(t_0) \\ &= \mathbf{A}\left[\mathbf{I} + (t-t_0)\mathbf{A} + \frac{(t-t_0)^2}{2!}\mathbf{A}^2 + \cdots\right]\mathbf{x}(t_0) \\ &= \mathbf{A}e^{(t-t_0)\mathbf{A}}\mathbf{x}(t_0) = \mathbf{A}\mathbf{x}(t)\end{aligned} \tag{27.80}$$

Also, at $t = t_0$, Equation 27.78 gives

$$\mathbf{x}(t_0) = \mathbf{I}\mathbf{x}(t_0) = \mathbf{x}(t_0) \tag{27.81}$$

Thus, expression (Equation 27.78) satisfies both Equation 27.70 and the initial conditions and hence is the unique solution. The matrix $e^{(t-t_0)\mathbf{A}}$, usually denoted by $\mathbf{\Phi}(t-t_0)$, is called the state transition matrix or the fundamental matrix of the circuit described by Equation 27.70. The transition of the initial state $\mathbf{x}(t_0)$ to the state $\mathbf{x}(t)$ at any time t is thus governed by

$$\mathbf{x}(t) = \mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) \tag{27.82}$$

where

$$\mathbf{\Phi}(t-t_0) = e^{(t-t_0)\mathbf{A}} \tag{27.83}$$

is an $n \times n$ matrix with the following properties:

$$\mathbf{\Phi}(t_0 - t_0) = \mathbf{\Phi}(0) = \mathbf{I} \tag{27.84}$$

$$\mathbf{\Phi}(t+\tau) = \mathbf{\Phi}(t)\mathbf{\Phi}(\tau) \tag{27.85}$$

$$\mathbf{\Phi}(t_2 - t_1)\mathbf{\Phi}(t_1 - t_0) = \mathbf{\Phi}(t_2 - t_0) \tag{27.86}$$

$$\mathbf{\Phi}(t_2 - t_1) = \mathbf{\Phi}^{-1}(t_1 - t_2) \tag{27.87}$$

$$\mathbf{\Phi}^{-1}(t) = \mathbf{\Phi}(-t) \tag{27.88}$$

Once the state transition matrix is known, the solution of the state equation can be obtained from Equation 27.82. In general, it is rather difficult to obtain a closed-form solution from the infinite series representation of the state transition matrix. The formula given by Equation 27.79 is useful only if numerical solution by digital computer is desired. Several methods are available for finding a closed form expression for $\mathbf{\Phi}(t-t_0)$. The relationship between solution (Equation 27.76) and the state transition matrix is first established.

For simplicity, let $t_0 = 0$. According to Equation 27.82, the first column of $\mathbf{\Phi}(t)$ is the solution of the state equation generated by the initial condition

$$\mathbf{x}(0) = \mathbf{x}^{(1)}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{27.89}$$

Indeed, if Equation 27.89 is substituted into Equation 27.82, then

$$\mathbf{x}(t) \triangleq \mathbf{x}^{(1)}(t) = \mathbf{\Phi}(t)\mathbf{x}^{(1)}(0) = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1n} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \phi_{11} \\ \phi_{21} \\ \vdots \\ \phi_{n1} \end{bmatrix} \tag{27.90}$$

which can be computed from Equation 27.76 and the arbitrary constants $c_i = \Delta c_i^{(1)}$ for $i = 1, 2, \ldots, n$ are solved from Equation 27.77. The first column of the state transition matrix is thus given by

$$\begin{bmatrix} \phi_{11} \\ \phi_{21} \\ \vdots \\ \phi_{n1} \end{bmatrix} = c_1^{(1)} e^{\lambda_1 t}\mathbf{P}_1 + c_2^{(1)} e^{\lambda_2 t}\mathbf{P}_2 + \cdots + c_n^{(1)} e^{\lambda_n t}\mathbf{P}_n \tag{27.91}$$

Instead of Equation 27.89, if

$$\mathbf{x}(0) = \mathbf{x}^{(2)}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{27.92}$$

is used, the arbitrary constants $c_1, c_2, \ldots, c_n$ denoted by $c_1^{(2)}, c_2^{(2)}, \ldots c_n^{(2)}$ are solved. Then, the second column of $\mathbf{\Phi}(t)$ is given

$$\begin{bmatrix} \phi_{12} \\ \phi_{22} \\ \vdots \\ \phi_{n2} \end{bmatrix} = c_1^{(2)} e^{\lambda_1 t}\mathbf{P}_1 + c_2^{(2)} e^{\lambda_2 t}\mathbf{P}_2 + \cdots + c_n^{(2)} e^{\lambda_n t}\mathbf{P}_n \tag{27.93}$$

In a similar manner, the remaining columns of $\mathbf{\Phi}(t)$ are determined.

The closed form expression for state transition matrix can also be obtained by means of a similarity transformation of the form

$$\mathbf{AP} = \mathbf{PJ}$$

or

$$\mathbf{J} = \mathbf{P}^{-1}\mathbf{AP} \tag{27.94}$$

where $\mathbf{P}$ is a nonsingular matrix. If the eigenvalues of $\mathbf{A}$, $\lambda_1, \lambda_2, \ldots, \lambda_n$, are assumed to be distinct, $\mathbf{J}$ is a diagonal matrix with eigenvalues on its main diagonal:

$$\mathbf{J} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \tag{27.95}$$

and

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix} \tag{27.96}$$

where $\mathbf{p}_i$'s, for $i = 1, 2, \ldots n$, are the corresponding eigenvectors associated with the eigenvalue λ_i, for $i = 1, 2, \ldots, n$. Substituting Equation 27.94 into 27.83, the state transition matrix can now be written in the closed form

$$\begin{aligned} \mathbf{\Phi}(t - t_0) = e^{(t-t_0)\mathbf{A}} &= e^{(t-t_0)\mathbf{P}\mathbf{J}\mathbf{P}^{-1}} \\ &= \mathbf{P}e^{(t-t_0)J}\mathbf{P}^{-1} \end{aligned} \tag{27.97}$$

where

$$e^{(t-t_0)\mathbf{J}} = \begin{bmatrix} e^{(t-t_0)\lambda_1} & 0 & \cdots & 0 \\ 0 & e^{(t-t_0)\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{(t-t_0)\lambda_n} \end{bmatrix} \tag{27.98}$$

is a diagonal matrix.

In the more general case, where the **A** matrix has repeated eigenvalues, a diagonal matrix of the form Equation 27.95 may not exist. However, it can be shown that any square matrix **A** can be transformed by a similarity transformation to the Jordan canonical form

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{J}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{J}_l \end{bmatrix} \tag{27.99}$$

where $\mathbf{J}_i$'s, for $i = 1, 2, \ldots, l$ are known as Jordan blocks. Assuming that **A** has m distinct eigenvalues, λ_i, with multiplicity r_i, for $i = 1, 2, \ldots, m$, and $r_1 + r_2 + \cdots + r_m = n$. Associated with each λ_i there may exist several Jordan blocks. A Jordon block is a block diagonal matrix of order $k \times k (k \leq r_i)$ with λ_i on its main diagonal, all 1's on the superdiagonal, and zeros elsewhere. In the special case when $k = 1$, the Jordan block reduces to a 1×1 scalar block with only one element λ_i.

In fact, the number of Jordan blocks associated with the eigenvalue λ_i is equal to the dimension of the null space of $(\lambda_i \mathbf{I} - \mathbf{A})$. For each $k \times k$ Jordan block $\mathbf{J}(k)$ associated with the eigenvalue λ_i of the form

$$\mathbf{J}(k) = \begin{bmatrix} \lambda_i & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 0 & \cdots & \lambda_i \end{bmatrix} \tag{27.100}$$

the exponential function of $\mathbf{J}(k)$ takes the form

$$e^{(t-t_0)\mathbf{J}(k)} = \begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{k-1}}{(k-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{k-2}}{(k-2)!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} e^{(t-t_0)\lambda_i} \tag{27.101}$$

and the corresponding k columns of $\mathbf{P}$, known as the generalized eigenvectors, satisfy the equations

$$\begin{aligned}(\lambda_i\mathbf{I}-\mathbf{A})\mathbf{p}_i^{(1)} &= 0\\ (\lambda_i\mathbf{I}-\mathbf{A})\mathbf{p}_i^{(2)} &= -\mathbf{p}_i^{(1)}\\ &\vdots\\ (\lambda_i\mathbf{I}-\mathbf{A})\mathbf{p}_i^{(k)} &= -\mathbf{p}_i^{(k-1)}\end{aligned} \tag{27.102}$$

The closed form expression $\mathbf{\Phi}(t-t_0)$ for this general case now becomes

$$\mathbf{\Phi}(t-t_0) = \mathbf{P}e^{(t-t_0)\mathbf{J}}\mathbf{P}^{-1} \tag{27.103}$$

where

$$e^{(t-t_0)\mathbf{J}} = \begin{bmatrix} e^{(t-t_0)\mathbf{J}_1} & 0 & \cdots & 0\\ 0 & e^{(t-t_0)\mathbf{J}_2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & e^{(t-t_0)\mathbf{J}_l}\end{bmatrix} \tag{27.104}$$

and each of the $e^{(t-t_0)\mathbf{J}_i}$, for $i=1, 2, \ldots, l$, is of the form given in Equation 27.101.

The third approach for obtaining closed form expression for the state transition matrix involves the Laplace transform technique. Taking the Laplace transform of Equation 27.70 yields

$$s\mathbf{X}(s)-\mathbf{x}(0) = \mathbf{A}\mathbf{X}(s)$$

or

$$\mathbf{X}(s) = (s\mathbf{I}-\mathbf{A})^{-1}\mathbf{x}(0) \tag{27.105}$$

where $(s\mathbf{I}-\mathbf{A})^{-1}$ is known as the resolvent matrix. The time response

$$\mathbf{x}(t) = +^{-1}\left[(s\mathbf{I}-\mathbf{A})^{-1}\right]\mathbf{x}(0) \tag{27.106}$$

is obtained by taking the inverse Laplace transform of Equation 27.105. It is observed by comparing Equation 27.106 to Equations 27.82 and 27.83 with $t_0=0$ that

$$\mathbf{\Phi}(t) = e^{t\mathbf{A}} = \mathcal{L}^{-1}\left[(s\mathbf{I}-\mathbf{A})^{-1}\right] \tag{27.107}$$

By way of illustration, the following example is considered. The state transition matrix is obtained by using each of the three approaches presented previously.

Example 27.7

Consider the parallel *RLC* circuit in Figure 27.7. The state equation of the circuit is obtained as

$$\begin{bmatrix}\frac{di_L}{dt}\\ \frac{dv_C}{dt}\end{bmatrix} = \begin{bmatrix}0 & \frac{1}{L}\\ -\frac{1}{C} & -\frac{1}{RC}\end{bmatrix}\begin{bmatrix}i_L\\ v_C\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{C}\end{bmatrix}i_s \tag{27.108}$$

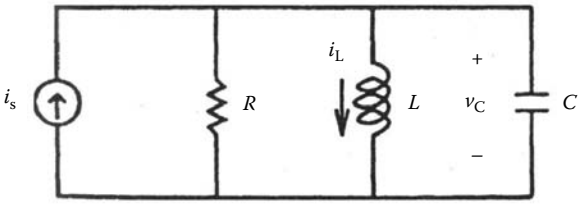


FIGURE 27.7 Parallel RLC circuit.

With $R = 2/3\ \Omega$, $L = 1\ H$, and $C = 1/2\ F$, the **A** matrix becomes

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \tag{27.109}$$

(a) *Normal mode approach:* The eigenvalues and the corresponding eigenvectors of the **A** are found to be

$$\lambda_1 = -1 \quad \lambda_2 = -2 \tag{27.110}$$

and

$$\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \tag{27.111}$$

Therefore, the natural response of the circuit is given as a linear combination of the two distinct normal modes as

$$\begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} = c_1 e^{-t} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + c_2 e^{-2t} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \tag{27.112}$$

When evaluated at $t = 0$, Equation 27.112 becomes

$$\begin{bmatrix} i_L(0) \\ v_C(0) \end{bmatrix} = \begin{bmatrix} c_1 + c_2 \\ -c_1 - 2c_2 \end{bmatrix} \tag{27.113}$$

In order to find the first column of $\mathbf{\Phi}(t)$, it is assumed that

$$\begin{bmatrix} i_L(0) \\ v_C(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{27.114}$$

With this initial condition, the solution of Equation 27.113 becomes

$$c_1 \stackrel{\Delta}{=} c_1^{(1)} = 2 \quad \text{and} \quad c_2 \stackrel{\Delta}{=} c_2^{(1)} = -1 \tag{27.115}$$

Substituting Equation 27.115 into 27.112 results in the first column of $\mathbf{\Phi}(t)$:

$$\begin{bmatrix} \phi_{11} \\ \phi_{21} \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} \end{bmatrix} \tag{27.116}$$

Similarly, for

$$\begin{bmatrix} i_L(0) \\ v_C(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{27.117}$$

constants c_1 and c_2 are solved from Equation 27.113 to give

$$c_1 \stackrel{\Delta}{=} c_2^{(2)} = 1 \quad \text{and} \quad c_2 \stackrel{\Delta}{=} c_2^{(2)} = -1 \tag{27.118}$$

The second column of $\mathbf{\Phi}(t)$:

$$\begin{bmatrix} \phi_{12} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} e^{-t} - e^{-2t} \\ -e^{-t} + 2e^{-2t} \end{bmatrix} \tag{27.119}$$

is obtained by substituting Equation 27.118 into 27.112. Combining Equations 27.116 and 27.119 yields the state transition matrix in closed form

$$\mathbf{\Phi}(t) = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \tag{27.120}$$

(b) *Similarity transformation method:* The eigenvalues are distinct, so the nonsingular transformation **P** is constructed from Equation 27.96 by the eigenvectors of **A**:

$$\mathbf{P} = [\,\mathbf{p}_1 \quad \mathbf{p}_2\,] = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \tag{27.121}$$

with

$$\mathbf{P}^{-1} = \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} \tag{27.122}$$

Substituting λ_1, λ_2, and **P** into Equations 27.97 and 27.98 yields the desired state transition matrix

$$\begin{aligned} \mathbf{\Phi}(t) = \mathbf{P}e^{\mathbf{J}t}\mathbf{P}^{-1} &= \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} e^{-t} & 0 \\ 0 & e^{-2t} \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \end{aligned} \tag{27.123}$$

which is in agreement with Equation 27.120.

(c) *Laplace transform technique:* The state transition matrix can also be computed in the frequency domain from Equation 27.107. The resolvent matrix is

$$\begin{aligned} (s\mathbf{I} - \mathbf{A})^{-1} &= \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{s+3}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ \frac{-2}{(s+1)(s+2)} & \frac{s}{(s+1)(s+2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{2}{s+1} - \frac{1}{s+2} & \frac{1}{s+1} - \frac{1}{s+2} \\ \frac{2}{s+1} + \frac{2}{s+2} & -\frac{1}{s+1} + \frac{2}{s+2} \end{bmatrix} \end{aligned} \tag{27.124}$$

where partial-fraction expansion has been applied. Taking the inverse Laplace transform of Equation 27.124 yields the same closed form expression as given previously in Equation 27.120 for $\mathbf{\Phi}(t)$.

27.4 Complete Response

When independent sources are present in the circuit, the complete response depends on the initial states of the circuits as well as the input sources. It is well-known that the complete response is the sum of the zero-input (or natural) response and the zero-state (or forced) response and satisfies the nonhomogeneous state equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{27.125}$$

subject to the given initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$. Equation 27.125 is again analogous to the scalar equation

$$\dot{x}(t) = ax(t) + bu(t) \tag{27.126}$$

which has the unique solution of the form

$$x(t) = e^{(t-t_0)a}x(t_0) + \int_{t_0}^{t} e^{(t-\tau)a}bu(\tau)\ \mathrm{d}\tau \tag{27.127}$$

It is thus assumed that the solution to the state equation is given by

$$\begin{aligned}\mathbf{x}(t) &= e^{(t-t_0)\mathbf{A}}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{(t-\tau)\mathbf{A}}\mathbf{B}\mathbf{u}(\tau)\ \mathrm{d}\tau \\ &= \mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} (t-\tau)\mathbf{B}\mathbf{u}(\tau)\ \mathrm{d}\tau\end{aligned} \tag{27.128}$$

Indeed, one can show by direct substitution that Equation 27.128 satisfies the state Equation 27.125. Differentiating both sides of Equation 27.128 with respect to t yields

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \frac{d}{\mathrm{d}t}\mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) + \frac{d}{\mathrm{d}t}\int_{t_0}^{t}\mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)\ \mathrm{d}\tau \\ &= \mathbf{A}\mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) + \int_{t_0}^{t}\frac{d}{\mathrm{d}t}\mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)\ \mathrm{d}\tau + \mathbf{\Phi}(t-t)\mathbf{B}\mathbf{u}(t) \\ &= \mathbf{A}\mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) + \int_{t_0}^{t}\mathbf{A}\mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)\mathrm{d}\tau + \mathbf{B}\mathbf{u}(t) \\ &= \mathbf{A}\left[\mathbf{\Phi}(t-t_0)\mathbf{x}(t_0) + \int_{t_0}^{t}\mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)\mathrm{d}\tau\right] + \mathbf{B}\mathbf{u}(t) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\end{aligned} \tag{27.129}$$

Also, at $t = t_0$, Equation 27.128 becomes

$$\begin{aligned}\mathbf{x}(t_0) &= \mathbf{\Phi}(t_0 - t_0)\mathbf{x}(t_0) + \int_{t_0}^{t_0} \mathbf{\Phi}(t_0 - \tau)\mathbf{B}\mathbf{u}(\tau)\mathrm{d}\tau \\ &= \mathbf{I}\mathbf{x}(t_0) + \mathbf{0} = \mathbf{x}(t_0)\end{aligned} \tag{27.130}$$

The assumed solution (Equation 27.128) thus satisfies both the state Equation 27.125 and the given initial condition. Hence, $\mathbf{x}(t)$ as given by Equation 27.128 is the unique solution.

It is observed from Equation 27.128 that if $\mathbf{u}(t)$ is set to zero, the solution reduces to the zero-input response or the natural response given in Equation 27.82. On the other hand, if the original circuit is relaxed, i.e., $\mathbf{x}(t_0) = 0$, the solution represented by the convolution integral, the second term on the right-hand side of Equation 27.128, is the forced response on the zero-state response. Thus, Equation 27.128 verifies the fact that the complete response is the sum of the zero-input response and the zero-state response. The previous result is illustrated by means of the following example.

Example 27.8

Consider again the same circuit given in Example 27.7, where the input current source is assumed to be a unit step function applied to the circuit at $t = 0$.

The state equation of the circuit is found from Equation 27.108 to be

$$\begin{bmatrix} \frac{\mathrm{d}i_L}{\mathrm{d}t} \\ \frac{\mathrm{d}v_C}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} i_s(t) \tag{27.131}$$

where the state transition matrix $\mathbf{\Phi}(t)$ is given in Equation 27.120.

The zero-state response for $t > 0$ is obtained by evaluating the convolution integral indicated in Equation 27.128:

$$\begin{aligned}\int_0^t \mathbf{\Phi}(t - \tau)\mathbf{B}\mathbf{u}(\tau)\mathrm{d}\tau &= \int_0^t \begin{bmatrix} 2e^{-(t-\tau)} - e^{-2(t-\tau)} & e^{-(t-\tau)} - e^{-2(t-\tau)} \\ -2e^{-(t-\tau)} + 2e^{-2(t-\tau)} & -e^{-(t-\tau)} + 2e^{-2(t-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \mathrm{d}\tau \\ &= 2\int_0^t \begin{bmatrix} e^{-(t-\tau)} - e^{-2(t-\tau)} \\ -e^{-(t-\tau)} + 2e^{-2(t-\tau)} \end{bmatrix} \mathrm{d}\tau = \begin{bmatrix} 1 - 2e^{-t} + e^{-2t} \\ 2e^{-t} - 2e^{-2t} \end{bmatrix}\end{aligned} \tag{27.132}$$

By adding the zero-input response represented by $\mathbf{\Phi}(t)\mathbf{x}(0)$ to Equation 27.132, the complete response for any given initial condition $\mathbf{x}(0)$ becomes

$$\begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \begin{bmatrix} i_L(0) \\ v_C(0) \end{bmatrix} + \begin{bmatrix} 1 - 2e^{-t} + 2e^{-2t} \\ 2e^{-t} - 2e^{-2t} \end{bmatrix} \tag{27.133}$$

for $t > 0$.

References

1. T. C. Chen, *Linear System Theory and Design*, New York: Holt, Rinehart & Winston, 1970.
2. W. K. Chen, *Linear Networks and Systems*, Monterey, CA: Brooks/Cole Engineering Division, 1983.

3. L. O. Chua and P. M. Lin, *Computer-Aided Analysis of Electronics Circuits: Algorithms and Computational Techniques*, Englewood Cliffs, NJ: Prentice Hall, 1969.
4. P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*, New York: John Wiley & Sons, 1965.
5. C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*, New York: McGraw-Hill, 1969.
6. B. C. Kuo, *Linear Networks and Systems*, New York: McGraw-Hill, 1967.
7. K. Ogata, *State Space Analysis of Control Systems*, Englewood Cliffs, NJ: Prentice Hall, 1967.
8. R. A. Rohrer, Circuit *Theroy*: *An Introduction to the State Variable Approach*, New York: McGraw-Hill, 1970.
9. D. G. Schultz and J. L. Melsa, *State Functions and Linear Control Systems*, New York: McGraw-Hill, 1967.
10. T. E. Stern, *Theory of Nonlinear Networks and Systems: An Introduction*, Reading MA: Addison-Wesley, 1965.
11. L. K. Timothy and B. E. Bona, *State Space Analysis: An Introduction*, New York: McGraw Hill, 1968.
12. L. A. Zadeh and C. A. Desoer, *Linear System Theory*, New York: McGraw-Hill, 1963.

Index

A

AC circuit, LLFT networks
  average power, 24-63–24-73
  input impedance, 24-64
  load impedance, 24-71
  maximum average power, 24-72–24-73
  power factor (PF), 24-63
  power triangle, 24-67
  RMS values, 24-66
  sinusoidal sources, 24-62–24-63
Active filter synthesis, SFG
  filter transfer function
    all-pole transfer function, 21-20
    differential-input op-amp circuit, 21-17
    dummy branch insertion, 21-13
    four op-amp circuit, 21-18–21-19
    gain function, 21-12–21-13
    lowpass filter specifications, 21-19
    Mason's gain formula, 21-12
    RC-op-amp circuits, 21-14–21-15
    state variable biquad, 21-19–21-20
    transfer voltage ratio, 21-11
    two op-amp circuit realizations, 21-15–21-16
  passive filter circuit
    bandpass passive filter, 21-9
    Butterworth low-pass filter, 21-7–21-9
    Chebyshev low-pass response, 21-10
    component op-amp circuits, 21-5–21-7
    doubly terminated passive filter, 21-4
    general ladder network, 21-4–21-5
    leapfrog realization, 21-5–21-9
    normalization, 21-10–21-11
Adaptive fault tolerance (AFT), 4-27–4-28
Adjustable capacitor, 11-27–11-28
Air inductor
  advantages, 11-48
  coil dimensions, 11-45–11-47
  design guidance, 11-47
  disadvantages, 11-49
  frequency ranges, 11-48
  quality factor, 11-47–11-48
Algebra
  ring concept, 2-2
  vector multiplication, 2-2–2-3
Analog 2-D filter
  quadrantal and octagonal symmetry, 10-28–10-29
  rotational and diagonal symmetry, 10-29
  transfer function, 10-27
Arc coloring theorem
  graph painting, 7-25–7-26
  induction hypothesis, 7-26
  resistances and voltages, 7-27

B

Bandpass filter (BPF)
  MATLAB optimization, 10-32
  specifications, 10-31–10-32
Bandpass sampling theorem, 6-5–6-6
Bandstop filter (BSF)
  coefficients and specification, 10-35
  contour and 3-D magnitude plots, 10-36
Bijective function, 1-4
Bilinear operators
  basis tensors
    finite-dimensional vector spaces, 2-5–2-8
    Kronecker product, 2-8
  determinants, 2-9–2-11
  matrix inversion, 2-14–2-16
  multiple products, 2-8–2-9
  property interchange, 2-16–2-18
  skew symmetric products
    k interchanges, 2-11
    p-linear operator, 2-12–2-14
  tensor product
    arbitrary bilinear operator, 2-4
    isomorphism, 2-5
Binet–Cauchy theorem, 7-15–7-16
Bipolar junction transistor (BJT) amplifier
  amplification, 15-1–15-2
  circuit schematic symbol, 18-1, 18-2
  discrete *vs*. IC amplifier component differences, 15-6–15-7

parasitic differences, 15-7–15-8
history, 15-2
IC amplifiers
active loads, 15-10–15-11
BJT differential pair, 15-20–15-21
cascode amplifier stage, 15-15–15-17
common-base stage, 15-13–15-14
common emitter stage, 15-11–15-13
differential stage, 15-18–15-20
emitter follower, 15-14–15-15
small-signal models, 15-8–15-10
IC power output stages
circuit efficiency, 15-31–15-32
class-B output stage, 15-32–15-33
class-D output stage, 15-33–15-36
limitations, 15-29–15-30
thermal resistance, 15-30–15-31
operational amplifiers
analog computer, 15-21–15-22
classical architecture, 15-22–15-23
features, 15-22
high-gain differential stage, 15-23–15-24
second amplifier stage, 15-24–15-26
specifications, 15-26–15-27
types and applications
distortion-free and nonlinear distortion, 15-4
frequency range and output power, 15-3
input and output variables, 15-2–15-3
intercept points, 15-5
operational amplifiers (op-amp), 15-5–15-6
total harmonic distortion (THD), 15-4
wideband amplifiers
composite stages, 15-27–15-28
feedback cascades, 15-28–15-29
BJT amplifier, *see* Bipolar junction transistor amplifier
Broad-sense Hurwitz polynomial (BHP), 9-5
Butterworth low-pass filter
fourth-order, 21-7
third-order, 21-8–21-9

C

Capacitive tree-branch voltages, 27-7, 27-8
Capacitor
dielectric constant, 11-24
geometry
air variable capacitor, 11-36
chip capacitor, 11-37
ESI and ESR, 11-34, 11-36
high-voltage capacitor, 11-29, 11-37
linear capacitor
adjustable capacitor, 11-27–11-28
energetic power capacitor, 11-28, 11-29
fixed capacitor-class 1, 11-26–11-27
high-voltage capacitor, 11-28–11-29
interference suppression capacitor, 11-30
leakage current and ionization threshold, 11-39
power factor and capacitance dependence, 11-37–11-38
nonlinear capacitor
ferrodielectric capacitor, 11-31–11-32
polar polymer dielectric capacitor, 11-32–11-33
power and quality factor, 11-25
properties
nonorganic capacitor, 11-33–11-34
organic capacitor, 11-34, 11-35
super capacitor, 11-34
υ' index, 11-25–11-26
Cauchy's residue theorem, 5-7
Chirp z-transform algorithm, 5-17
Circuit elements
characterization
dynamic model, 13-9–13-10
formal methods, 13-3–13-4
memristive element, 13-8
reactive element, 13-7–13-8
resistive element, 13-4–13-7
connecting multiport (CMP)
linear circuit, KCL and KVL, 13-10–13-11
nonlinear circuit and partition, 13-1–13-2
lumped circuit approximation, 13-1
tableau formulation, 13-11
CMOS, *see* Complementary metal oxide semiconductor
Coates' gain formula
coefficient matrix, 8-7–8-8
n×n matrix, 8-6–8-7
nonsingular matrix, 8-5
topological formula, 8-4
Common-mode rejection ratio (CMRR), 15-27
common-mode gain, 16-16–16-17
differential and noninverting configuration, 16-17
total output voltage, 16-17–16-18
Continuous-time Fourier transform
convergence, 4-10–4-11
CT and the DT sampled model, 4-5–4-6
exponential Fourier series, 4-7–4-8
Fourier spectrum, 4-6
periodic continuous-time signals, 4-11
properties, 4-2–4-5
trigonometric expansion, 4-8–4-10
Continuous-time signals
convergence, 4-10–4-11
CT and the DT sampled model, 4-5–4-6
exponential Fourier series, 4-7–4-8
Fourier spectrum, 4-6
periodic continuous-time signals, 4-11
properties, 4-2–4-5

trigonometric expansion, 4-8–4-10
Continuous wavelet transform (CWT), 6-15
Controlled circuit elements
controlled sources
CCCS and CCVS, 14-4–14-7
circuit representation, 14-1, 14-2
VCCS and VCVS, 14-1–14-4
signal converters
circulator, 14-12–14-15
current negative impedance converter, 14-11–14-12
definition, 14-15
gyrator, 14-8–14-10
network ports, 14-7–14-8
voltage negative impedance converter, 14-11
Cooley–Tukey FFT, 10-11
Cored inductor
advanced inductor, 11-56
closed magnetic circuit
air gap, 11-51–11-53
apparent permeability, 11-54
equivalent dimensions and parameters, 11-50–11-51
incomplete coupling, 11-53
joint effect, 11-54–11-55
leakage coefficient, 11-54
open magnetic circuit, 11-55–11-56
parameters, 11-49–11-50
power loss
definition, 11-56–11-57
dielectric loss, 11-59–11-60
magnetic loss, 11-57–11-59
resistive loss, 11-57
sinusoidally excited inductor, 11-60
Cramer's rule, 1-19
Critical temperature resistors (CTR), 11-10
Current-controlled current source (CCCS)
circuit implementation, 14-7
circuit schematic symbol, 18-3, 18-4
equivalent circuit, 14-6
SPICE format, 14-6
Current-controlled voltage source (CCVS), 14-4
circuit implementation, 14-5–14-6
circuit schematic symbol, 18-3
SPICE format, 14-5
Current feedback op-amp (CFOA)
bipolar realization, 17-5
buffer and CFOA implementation
frequency response, 17-7, 17-9
open-loop transimpedance, 17-7, 17-9
output impedance characteristics, 17-7–17-8
schematics and layout plot, 17-6–17-7
step response and harmonic distortion, 17-11
voltage buffer performance, 17-7
closed-loop gain, 17-3–17-4
CMOS compound device, 17-5
macromodel, 17-2–17-3
total impedance, 17-3
Current-mode op-amp
closed-loop characteristics, 16-35–16-36
dynamics, 16-37–16-38
higher-order effects, 16-38–16-40
Cutset-loop circuit analysis
analogous problem, mesh analysis, 19-31
circuit representation, 19-39–19-41
failure, nodal analysis, 19-30–19-31
fundamental cutset and loop, 19-32
graphical representation, 19-31–19-32
KCL equation, 19-32–19-34
KVL equation, 19-34
norator, 19-35–19-36
nullator and equivalent subcircuit, 19-37
passive element and equivalent circuits, 19-36–19-37
three-terminal/two-port element, 19-37–19-39
two-terminal element, 19-35–19-40

D

Determinant decision diagrams (DDD)
matrix determinant and representation, 25-24–25-25
nodal analysis, 25-23–25-24
vertex and vertex ordering, 25-24
Differential-input op-amp circuit, 21-7
Digital filter banks
basis function, 6-29–6-30
half-band filter, 6-27
paraunitary filter bank
parametrization, 6-27
reconstruction system, 6-25–6-27
polyphase representation, 6-24–6-25
subband reconstruction
conjugate quadrature filter, 6-23–6-24
PR property, 6-23
tree-structured system, 6-28–6-29
two-channel filter bank, 6-22
Digital filters
continuous-time frequency response, 10-15
discrete-time frequency response, 10-15–10-16
magnitude response, 10-16
quadrantal and octagonal symmetry, 10-29–10-30
rotational and diagonal symmetry, 10-30
Dirac delta function
definition, 3-42–3-43
doublet signal, 3-44
impulse function, 3-43
Discrete Fourier transform (DFT), 25-16
FFT algorithm
applications, 4-19
ordered inputs and bit-reversed outputs, 4-18–4-19
types and development, 4-17
properties, 4-15–4-17

Discrete-time Fourier transform (DTFT)
vs. CT spectra, 4-14–4-15
DTFT pairs
definition and bilateral *z*-transform, 4-12
types, 4-13
properties, 4-13–4-14
Doubly terminated passive filter, 21-4

E

Eigenvalues
inverse matrix, 1-18
singular-value decomposition (SVD), 1-17
Electrical circuit
classifications
active *vs.* passive, 18-12
linear *vs.* nonlinear, 18-11–18-12
lumped *vs.* distributed, 18-13–18-15
time-varying *vs.* time-invariant, 18-12–18-13
current polarity
charge transport, 18-5–18-6
current flow, 18-4–18-5
energy and voltage, 18-7–18-8
power
calculations, 18-8–18-9
conversation, 18-8–18-9
schematic symbols
controlled sources, 18-2–18-4
multiterminal elements, 18-1–18-2
two-terminal elements, 18-1, 18-2
voltage-mode operational amplifier, 18-4, 18-5
Electromagnetic interference (EMI), 11-49
Energetic power capacitor, 11-28, 11-29
Equivalence transformation, 1-14–1-15
Equivalent series inductance (ESI), 11-34
Equivalent series resistance (ESR), 11-25
Eternal signals, 26-3–26-4

F

Fast Fourier transform (FFT), 25-17
algorithms
applications, 4-19
ordered inputs and bit-reversed outputs, 4-18–4-19
types and development, 4-17
centrosymmetry, 10-13
quadrantal symmetry, 10-14
symmetry-based 1-D FFT, 10-11–10-12
Ferrodielectric capacitor, 11-31–11-32
Filter banks
digital filter banks
basis function, 6-29–6-30
half-band filter, 6-27
paraunitary properties, 6-25–6-27
polyphase representation, 6-24–6-25
subband construction, 6-22–6-24
tree-structured system, 6-28–6-29
two-channel filter bank, 6-22
paraunitary filter banks
dilation equation application, 6-63–6-64
infinite product convergence, 6-64–6-65
orthonormal wavelet basis, 6-65–6-70
parametrization, 6-27
wavelet tight frames, 6-70–6-71
synthesis and reconstruction, 6-13–6-14
time–frequency representation, 6-13
wavelet function design, 6-14–6-15
Filter transfer function
all-pole transfer function, 21-20
differential-input op-amp circuit, 21-17
dummy branch insertion, 21-13
four op-amp circuit, 21-18–21-19
gain function, 21-12–21-13
lowpass filter specifications, 21-19
Mason's gain formula, 21-12
RC-op-amp circuits, 21-14–21-15
state variable biquad, 21-19–21-20
transfer voltage ratio, 21-11
two op-amp circuit realizations, 21-15–21-16
First-order circuits, time domain analysis
network time constant, 26-10
transfer functions, 26-6–26-7
transient and steady-state responses, 26-9–26-10
zero-input response, 26-7–26-8
zero-state response, 26-8–26-9
Fixed capacitor, 11-26–11-27
Fixed resistor
chip resistor, 11-6, 11-7
dynamic tolerance Δ and resistor class, 11-2–11-3, 11-4
high-ohmic, high-voltage resistor, 11-6
high-power resistor, 11-6–11-7
resistive network, 11-5–11-6
temperature coefficient of resistance (TCR), 11-3, 11-5
Fourier transform
adaptive fault tolerance (AFT), 4-27–4-28
complex Fourier transform, 4-6–4-7
continuous-time signals
convergence, 4-10–4-11
CT and the DT sampled model, 4-5–4-6
exponential Fourier series, 4-7–4-8
Fourier spectrum, 4-6
periodic continuous-time signals, 4-11
properties, 4-2–4-5
trigonometric expansion, 4-8–4-10
discrete Fourier transform (DFT)
FFT algorithm, 4-17–4-19
properties, 4-15–4-17
discrete-time Fourier transform (DTFT)
vs. CT spectra, 4-14–4-15
DTFT pairs, 4-12–4-13
properties, 4-13–4-14

FFT spectral analysis
leakage and the picket-fence effect, 4-21–4-22
windowing and zero-padding, 4-21
FIR digital filter design
bandpass FIR filter, 4-22–4-23
frequency response, 4-22
functional relationships, 4-19–4-20
real-time filtering, Fourier block processing
block algorithms, 4-24
overlap-add processing, 4-23–4-24
overlap-save processing, 4-23
transform domain adaptive filter (TDAF)
adaptive tap vector, 4-25
autocorrelation matrix, 4-24–4-25
characteristics, 4-26–4-27
filter output and LMS adaptive algorithm, 4-24
Karhunen–Loe've transform (KLT), 4-26
power, 4-25–4-26
transformed outputs, 4-25
Fractal capacitors
boost factor *vs.* horizontal spacing, 12-7
bottom-plate parasitic capacitance, 12-5–12-6
fractal dimension, 12-4–12-6
high-density capacitor, 12-2
interdigitated capacitor, 12-7–12-8
lateral flux capacitors, 12-2–12-3
parallel-plate and MOS, 12-1–12-2
quality factor and lateral spacing, 12-7
three-dimensional representation, 12-5, 12-6
woven structure, 12-8
Frequency-dependent negative resistance (FDNR), 14-10
Frequency-domain analysis
advanced network analysis
CCCS and VCVS gain, 22-34–22-35
circuit branch admittance, 22-27–22-30
circuit branch impedance, 22-30
circuit transadmittance, 22-31–22-33
circuit transimpedance, 22-31
driving point impedances, 22-35–22-39
engineering utility, 22-18–22-25
fundamental concepts, 22-11–12-15
Kron–Bode formula, 22-25–22-27
Kron's formula, 22-15–22-18
sensitivity analysis, 22-39
transposed connection vector, 22-14
LLFT network
capacitor voltage and current, 24-2
dividers, 24-15–24-18
inductor voltage and current, 24-3
Middlebrook's extra element theorem, 24-38–24-42
Norton's theorem, 24-31–24-32
Ohm's law, 24-1–24-2
one-port LLFT network, 24-5
properties, 24-10–24-12
π–T conversion, 24-34–24-35
reciprocity, 24-35–24-37
network functions
Cramer's rule, 22-3
definition, 22-1
inductor current, 22-5–22-6
Laplace transform, 22-9–22-10
modified nodal, 22-4
network stability, 22-8–22-9
output and input impedance, 22-2
output voltage and input voltage, 22-1
ratio of two polynomials, 22-6
roots of the numerator and denominator, 22-7
stability, 22-8–22-9
time-domain response, 22-10
transfer function, 22-3
sinusoidal steady state analysis
AC circuit power, 24-62–24-73
network transfer function, 24-46–24-47
output voltage, 24-48
phase lead and lag, 24-55–24-57
phasor and network, 24-53–24-55
RC and RL network, 24-57–24-59
resonance, 24-60–24-62
steady-state output voltage, 24-49
transient and steady-state response, 24-45
source transformations, 24-12–24-15
substitution theorem, 24-43–24-44
superposition theorem
dependent and independent source, 24-19–24-20
loop impedance matrix, 24-18
Thévenin's theorem
application, 24-28–24-29
dependent sources, 24-24–24-25
impedance, 24-26
source transformation, 24-29–24-30
subnetworks, 24-24
two-port networks, 24-5–24-6
Front-end amplifier, *see* RF low-noise amplifiers

G

Gate-drain capacitance (C_{gd}) effect, 17-14–17-15
Generalized impedance convertor (GIC), 14-10
Geometric mean distance (GMD), 12-13
Graph theory
arc coloring theorem
graph painting, 7-25–7-26
induction hypothesis, 7-26
resistances and voltages, 7-27
circuit and cutset
circuit matrix, 7-12
cut matrix, 7-10–7-11

electrical networks
KCL and the KVL, 7-19
loop and cutset methods, 7-20–7-21
network elements and directed graph, 7-18
incidence matrix
determinant, 7-10
directed graph, 7-9
incidence vector, 7-9–7-10
orthogonality relation, 7-12–7-14
relative orientations, 7-12
Tellegen theorem
2-port network, 7-21–7-22
adjoint elements, 7-23–7-24
open-circuit voltage ratio, 7-24
voltage and current change, 7-22
tree enumeration spanning
Binet–Cauchy theorem, 7-15–7-16
directed graph, 7-17
undirected graph, 7-16–7-17

H

Haar wavelet basis, 6-11–6-12
Hierarchical methods
advantage, 25-17, 25-18
bipolar cascode stage, 25-19–25-20
modified nodal analysis and NAM, 25-19
resistance ladder network, 25-17–25-18
topological analysis, 25-18
Higher-order log-domain filters, 17-45
High-frequency (HF) amplifiers
current feedback op-amp (CFOA)
bipolar realization, 17-5
buffer and CFOA implementation, 17-7–17-11
closed-loop gain, 17-3–17-4
CMOS compound device, 17-5
macromodel, 17-2–17-3
total impedance, 17-3
high-Q resonators
gm-C bandpass biquad, 17-32–17-34
IF-sampling, 17-29–17-30
triode region transconductor, 17-31
unity-gain frequency, 17-32
log-domain process
basic log-domain integrator, 17-41–17-44
filter synthesis, 17-36–17-37
higher-order log-domain filters, 17-45
instantaneous companding, 17-34–17-36
performance aspects, 17-37–17-40
operation layout
feedback reduction, 17-23
flip-chip connection, 17-24
high-frequency (HF) ground, 17-23
input/output isolation, 17-23
I/O pads, 17-23
optical low-noise preamplifiers
front-end noise sources, 17-17–17-18
receiver performance criteria, 17-19–17-21
transimpedance (TZ) amplifiers, 17-21–17-22
RF low-noise amplifiers
cascode common source LNA, 17-15–17-16
CMOS common-source LNA, 17-12–17-15
gain, 17-10–17-12
input impedance matching and linearity, 17-12
noise, 17-8
RF power amplifier
classification, 17-24–17-25
PA requirements, 17-24
practical considerations, 17-27
wireless transceiver, 17-2
High-pass filter (HPF)
coefficients and specification, 10-34
contour and 3-D magnitude plots, 10-35
High-Q resonators
gm-C bandpass biquad
biquad bandpass and SFDR, 17-33
Q passive on-chip inductors, 17-32
simulated bandpass frequency response, 17-34
IF-sampling
ADC and sampling receiver, 17-29
continuous-time Sigma-Delta modulator, 17-30
linear region transconductor implementation
triode region transconductor, 17-31
unity-gain frequency, 17-32
High-voltage capacitor, 11-28–11-29
Hilbert space frame
arbitrary vector representation, 6-44
definition, 6-44
dual frame and sequence, 6-47
frame operator, 6-46
Riesz basis, 6-44–6-45
tight frame, 6-46
Hölder regularity index, 6-74
Hurwitz polynomials
analogs
2-V polynomials, zeroes continuity property, 9-7–9-10
broad-sense Hurwitz polynomial (BHP), 9-5
Hurwitz polynomial in the strict sense (HPSS), 9-6
immittance Hurwitz polynomial (IHP), 9-7
narrow-sense Hurwitz polynomial (NHP), 9-5
reactance Hurwitz polynomial (RHP), 9-7
scattering Hurwitz polynomial (SHP), 9-5–9-6
self-paraconjugate polynomial and SPHP, 9-5
very strict Hurwitz polynomial (VSHP), 9-6

preliminaries and notations
analog biplane, 9-2–9-3
infinite distant points, 9-2
isolated and continuum of zeroes, 9-3
two-dimensional Hurwitz polynomials, 9-11–9-12
two-variable polynomial, infinity
1-D complex plane, 9-3
2-D biplane, 9-4
nonessential singularities second kind (NSSK), 9-4
2-V very strict Hurwitz polynomial (VSHP), 9-10–9-11
Hybrid parameters
bipolar transistors, 20-11–20-12
circuit model, 20-10–20-11
common emitter configuration, 20-11

I

Ideal bandpass wavelet
Fourier transform, 6-5–6-6
frequency axis splitting, 6-6–6-7
impulse response, 6-8
vs. low-pass, 6-5–6-6
Ideal operational amplifier
circuit applications
frequency dependence, 16-5–16-6
instrumentation amplifiers, 16-8
inverting configuration, 16-5
multiple input circuits, 16-7–16-8
noninverting configuration, 16-4–16-5
nonlinear transfer functions, 16-6
simple attenuator, 16-4
unity gain buffer, 16-3
comparators, 16-9–16-10
inverting amplifier configuration, 16-10–16-11
open-loop equivalent circuit characterization, 16-1
types and configurations, 16-2–16-3
voltage op-amps, 16-3
Immittance Hurwitz polynomial (IHP), 9-7
Impact avalanche transit time (IMPATT) diode, 11-73
Indefinite admittance matrix (IAM) approach
basics, 25-10–25-11
resistive circuit, 25-11–25-12
Inductive link current
capacitive tree-branch voltages, 27-7–27-8
state equations, 27-5–27-6
Inductor
air inductor
advantages, 11-48
coil dimensions, 11-45–11-47
design guidance, 11-47
disadvantages, 11-49
frequency ranges, 11-48
quality factor, 11-47–11-48
circuit representation, 11-45
cored inductor
advanced inductor, 11-56
closed magnetic circuit, 11-50–11-55
open magnetic circuit, 11-55–11-56
parameters, 11-49–11-50
power loss, 11-56–11-60
definition, 11-63–11-64
functions and parameters, 11-44
fundamental observations, 11-42, 11-43
graphical symbols, 11-40
inductance, 11-40–11-42
magnetic core materials, 11-61–11-63
nonlinearity, 11-60–11-61
physical irregularities, 11-63
qualifiers and attributes, 11-42–11-43
Integrated circuit (IC) amplifiers, 15-3
active loads, 15-10–15-11
BJT differential pair, 15-20–15-21
cascode stage
advantages, 15-15–15-16
high-frequency equivalent circuit, 15-16–15-17
circuit efficiency, 15-31–15-32
class-B output stage
bias circuit, 15-33
Darlington output stages, 15-33
short-circuit proof, 15-32
class-D output stage, PWM amplifier
advantage and disadvantage, 15-35
architecture, 15-33, 15-34
configurations, 15-35–15-36
duty cycle, 15-33–15-34
common-base stage, 15-13–15-14
common emitter stage
configuration, 15-11
equivalent circuit, 15-11–15-12
upper corner frequency, 15-12–15-13
differential stage
advantages and configuration, 15-18
mode of operation, 15-19–15-20
emitter follower, 15-14–15-15
limitations, power output, 15-29–15-30
small-signal models
hybrid-π model, 15-8–15-9
unilateral equivalent circuit, 15-9–15-10
thermal resistance, 15-30–15-31
Interference suppression capacitor, 11-30
Inverse Laplace transform
partial fraction expansion (PFE)
repeated linear factors, 3-31
repeated quadratic factors, 3-31–3-32
simple linear factors, 3-27–3-29
simple quadratic factors, 3-29–3-30
residue theory
differentiated time function, 3-24
LT occurrence, 3-25
Taylor series expansion, 3-23
time function, 3-22
Inverse z-transform
contour integration
Cauchy's residue theorem, 5-7
order interchanging, 5-6–5-7
residues, 5-6–5-7

inspection method and power series expansion, 5-8
partial fraction expansion, 5-7–5-8
Inversion negative impedance converter (INIC), 14-11

K

Kirchhoff's current law (KCL), 13-10–13-11
circuit paths, 19-6–19-7
cutset-loop circuit analysis, 19-32–19-34
Gaussian surface, 19-5–19-6
illustration, 19-4–19-5
mesh analysis, 19-24, 19-25
voltage rise and drop, 19-7
Kirchhoff's voltage law (KVL)
circuit analysis, 19-8–19-9
circuit paths, 19-8
cutset-loop circuit analysis, 19-34
illustration, 19-7–19-8
mesh analysis, 19-24–19-26
Kron's formula
engineering utility
bipolar junction transistor amplifier, 22-23
Bode's theory, 22-21
common emitter amplifier, 22-21–22-22
perturbed voltage gain, 22-18–22-19
transfer function sensitivity, 22-20–22-21
generalization
I–O transfer characteristic node, 22-25
return ratio, 22-26–22-27
transfer function and return ratio, 22-26
impedance inclusion, 22-15–22-16
output voltage, 22-17–22-18
Thévenin voltage and impedance, 22-16–22-17

L

Laplace transformation
bilateral Laplace transform (BLT), 3-8–3-9
Dirac delta function
definition, 3-42–3-43
doublet signal, 3-44
impulse function, 3-43
discrete-time Laplace transform (DTLT), 3-45
forced response, 3-4
homogeneous solution and natural response, 3-2–3-3
inverse Laplace transform
partial fraction expansion (PFE), 3-27–3-32
residue theory, 3-23–3-27
time function, 3-22
Laplace-domain phasor circuits, 3-41–3-42
linear system
differential equation, 3-32–3-34
generalized phasor, 3-34–3-35
magnitude and phase response, 3-35–3-37
nonhomogeneous solution and forced response, 3-4–3-5
phasor concept generalization
complex number, 3-7
generalized phasor, 3-7–3-8
pole–zero analysis
BIBO stability, 3-38–3-39
general function, $F(s)$, 3-15
natural modes, 3-37–3-38
polynomial factorization, 3-16
ROC, 3-16–3-18
stability and natural modes, 3-39–3-41
properties
convolution property, 3-20–3-21
modulation property, 3-19
square wave, 3-19
time-differentiation property, 3-20
ULT, 3-18
series RLC circuit, 3-2
solution scrutinization, 3-6–3-7
total solution, 3-5–3-6
unilateral Laplace transforms
direct computation, 3-11–3-15
operational properties, 3-18–3-22
Linear capacitor
adjustable capacitor, 11-27–11-28
energetic power capacitor, 11-28, 11-29
fixed capacitor-class 1, 11-26–11-27
high-voltage capacitor, 11-28–11-29
interference suppression capacitor, 11-30
Linear operators and matrices
bijective function, 1-4
characteristics
eigenvalues, 1-15–1-17
eigenvectors, 1-16
singular values, 1-17–1-18
determinants
definition, 1-8
first-and second order matrix, 1-10
Laplace's expansion, 1-9
linear systems
Cramer's rule, 1-19
input and output system, 1-18
poles and zeros, 1-18–1-19
matrix operations
matrix addition, 1-7
matrix multiplication, 1-7–1-8
matrix transposition, 1-8
vector addition, 1-6
matrix representations, 1-6
transformations
equivalence transformation, 1-14–1-15
similarity transformation, 1-12–1-14
vector spaces over fields
field F, properties, 1-2
F-vector space, 1-2–1-3
n-dimensional vector space, 1-3–1-4
Linear resistor, 11-1–11-2
fixed resistor
chip resistor, 11-6, 11-7
dynamic tolerance Δ and resistor class, 11-2–11-3, 11-4
high-ohmic, high-voltage resistor, 11-6
high-power resistor, 11-6–11-7
resistive network, 11-5–11-6
temperature coefficient of resistance (TCR), 11-3, 11-5

variable resistor, potentiometers
electric circuit, 11-7, 11-8
tapers, 11-9
types, 11-7–11-9
Linear time-invariant systems, *z*-Transform
one-sided *z*-transform, 5-14
system transfer function, 5-13–5-14
Log-domain process
basic log-domain integrator
component mismatch, 17-41
frequency limitations, 17-43–17-44
noise, 17-44
Ohmic resistance, 17-42–17-43
filter synthesis, 17-36–17-37
higher-order log-domain filters, 17-45
instantaneous companding, 17-34–17-36
performance aspects
finite current gain, 17-38–17-39
frequency performance, 17-39–17-40
tuning range, 17-37–17-38
Low-pass filter (LPF)
coefficients and specification, 10-33
contour and 3-D magnitude plots, 10-34
Lumped, linear, finite, time-invariant (LLFT) networks
capacitor voltage and current, 24-2
impedance concept
one-port network, 24-3–24-5
two-port networks, 24-5–24-6
independent nodes, 24-6–24-7
inductor voltage and current, 24-3
inverse transform phasors, 24-52–24-53
LLFT networks, 24-3–24-4
loop impedance matrix, 24-9–24-10
Middlebrook's extra element theorem
parallel version, 24-38–24-40
series version, 24-40–24-43
nodal admittance matrix, 24-6–24-7
Ohm's law, 24-1–24-2
one-port LLFT network, 24-5
phasor transforms
Euler's identity, 24-49
integration constant, 24-52
sine and cosine function, 24-50
properties
driving-point network function polynomial, 24-11–24-12
poles and zeroes, 24-10
transfer function, 24-12
π–T conversion
loop equation, 24-35
networks, 24-34–24-35
T replacement, 24-35
reciprocity
circuit modification, 24-36–24-37
excitation and response, 24-37
reciprocal network, 24-35–24-36
sinusoidal steady state analysis
AC circuit power, 24-62–24-73
network transfer function, 24-46–24-47
output voltage, 24-48
phase lead and lag, 24-55–24-57
phasor and network, 24-53–24-55
RC and RL network, 24-57–24-59
resonance, 24-60–24-62
steady-state output voltage, 24-49
transient and steady-state response, 24-45
source transformations, 24-12–24-15
substitution theorem, 24-43–24-44
superposition
dependent and independent source, 24-19–24-20
loop impedance matrix, 24-18
Thévenin's theorem
application, 24-28–24-29
dependent sources, 24-24–24-25
impedance, 24-26
source transformation, 24-29–24-30
subnetworks, 24-24
two-port networks, 24-5–24-6
voltage excitation, 24-8–24-9

M

Magnitude response symmetry
2-D polynomials determination
diagonal symmetry, 10-25–10-26
fourfold rotational symmetry, 10-26
octagonal symmetry, 10-27
quadrantal symmetry, 10-21–10-25
analog 2-D filter
quadrantal and octagonal symmetry, 10-28–10-29
rotational and diagonal symmetry, 10-29
transfer function, 10-27
applications
continuous-time frequency response, 10-15
discrete-time frequency response, 10-15–10-16
magnitude-squared functions, 10-16
bandpass filter
MATLAB optimization, 10-32
specifications, 10-31–10-32
bandstop filter
coefficients and specification, 10-35
contour and 3-D magnitude plots, 10-36
digital filters
continuous-time frequency response, 10-15
discrete-time frequency response, 10-15–10-16
magnitude response, 10-16
quadrantal and octagonal symmetry, 10-29–10-30
rotational and diagonal symmetry, 10-30

high-pass filter
coefficients and specification, 10-34
contour and 3-D magnitude plots, 10-35
low-pass filter
coefficients and specification, 10-33
contour and 3-D magnitude plots, 10-34
magnitude-squared functions, 10-16
Mason's gain formula
feedback loops, 8-10
j self-loops, 8-9
Mason graph, 8-8–8-9
Matrix operations
addition, 1-7
multiplication, 1-7–1-8
transposition, 1-8
vector addition, 1-6
Maximum power transfer theorem
load resistance, 19-55–19-58
power *vs*. resistance, 19-55, 19-56
Mesh analysis
circuit graph, 19-22–19-23
current sources, 19-26
deactivated current, 19-27
dependent source, 19-28–19-30
features and constraints, 19-27–19-28
independent source, 19-30
KCL, 19-24, 19-25
KVL, 19-24–19-26
mesh current, 19-24, 19-25
one-mesh circuit, 19-23
two-mesh circuit, 19-23–19-24
Metal–insulator–metal (MIM), 11-14
Metal-oxide-semiconductor field-effect transistor (MOSFET), 18-1, 18-2
Middlebrook's extra element theorem
parallel version, 24-38–24-40
series version, 24-40–24-43
Miller effect capacitance, 15-9–15-10
Modified nodal analysis (MNA), 25-19
Modified *z*-transform, 5-16
Multidimensional *z*-transforms, 5-15

N

Narrow-sense Hurwitz polynomial (NHP), 9-5
Negative impedance converter (NIC), 14-8
Negative temperature coefficient (NTC) thermistor, 11-10
Network analysis
CCCS and VCVS gain, 22-34–22-35
circuit branch admittance, 22-27–22-30
circuit branch impedance, 22-30
circuit transadmittance
linear circuit, 22-31–22-32
voltage feedback amplifier, 22-33
circuit transimpedance, 22-31
driving point impedances
Blackman's formula, 22-37
critical circuit parameters, 22-35
transimpedance, 22-36
zero parameter, 22-36–22-37
functions
Cramer's rule, 22-3
definition, 22-1
inductor current, 22-5–22-6
Laplace transform, 22-9–22-10
modified nodal, 22-4
network stability, 22-8–22-9
output and input impedance, 22-2
output voltage and input voltage, 22-1
ratio of two polynomials, 22-6
roots of the numerator and denominator, 22-7
stability, 22-8–22-9
time-domain response, 22-10
transfer function, 22-3
fundamental concepts
branch linear network, 22-11–22-12
impedance and transimpedance, 22-14–12-15
nodal admittance matrix, 22-12–22-13
Norton equivalent circuit, 22-11–22-12
vector concept, 22-13
voltage gain, 12-15
Kron–Bode formula, 22-25–22-27
Kron's formula
bipolar junction transistor amplifier, 22-23
Bode's theory, 22-21
common emitter amplifier, 22-21–22-22
impedance inclusion, 22-15–22-16
I–O transfer characteristic node, 22-25
output voltage, 22-17–22-18
perturbed voltage gain, 22-18–22-19
return ratio, 22-26–22-27
Thévenin voltage and impedance, 22-16–22-17
transfer function and return ratio, 22-26
transfer function sensitivity, 22-20–22-21
sensitivity analysis, 22-39
transposed connection vector, 22-14
Network laws and theorems
basic circuit theory, 19-3
circuit analysis, 19-3–19-4
conductors and elements, 19-1, 19-2
current measurement, 19-2–19-3
cutset-loop circuit analysis
analogous problem, mesh analysis, 19-31
circuit graph, 19-31–19-32
example circuit, 19-39–19-41
failure, nodal analysis, 19-30–19-31
fundamental cutset and loop, 19-32
KCL equation, 19-32–19-34
KVL equation, 19-34
norator, 19-35–19-36
nullator and equivalent subcircuit, 19-37
passive element and equivalent circuits, 19-36–19-37
three-terminal/two-port element, 19-37–19-39
two-terminal element, 19-35–19-40

Kirchhoff's current law (KCL)
  circuit paths, 19-6–19-7
  Gaussian surface, 19-5–19-6
  illustration, 19-4–19-5
  voltage changes, 19-7
Kirchhoff's voltage law (KVL)
  circuit analysis, 19-8–19-9
  circuit paths, 19-8
  illustration, 19-7–19-8
linear circuit
  KCL, 19-42–19-43
  superposition and homogeneity condition, 19-43–19-44
maximum power transfer theorem
  load resistance, 19-55–19-58
  power *vs*. resistance, 19-55, 19-56
mesh analysis
  circuit graph, 19-22–19-23
  current sources, 19-26
  deactivated current, 19-27
  dependent source, 19-28–19-30
  features and constraints, 19-27–19-28
  independent source, 19-30
  KCL, 19-24, 19-25
  KVL, 19-24–19-26
  mesh current, 19-24, 19-25
  one-mesh circuit, 19-23
  two-mesh circuit, 19-23–19-24
nodal analysis
  deactivated circuit configuration, 19-14–19-15
  dependent source, 19-16–19-17
  differentiation and integration operators, 19-19–19-20
  dynamic circuit analysis, 19-20–19-22
  dynamic element relationships, 19-18–19-19
  floating element, 19-10–19-12
  independent voltage source, 19-13–19-14
  node voltage concept, 19-9–19-10
  nonessential node, 19-17–19-18
  reference node, 19-10, 19-11
  resistive circuit, 19-12–19-13
  supernode and essential node, 19-15, 19-18
Norton theorem
  current calculation and power dissipation, 19-54–19-55
  equivalent circuit, 19-51–19-52
  mesh equation method, 19-52–19-53
reciprocity theorem
  statement, 19-58
  verification, 19-58–19-59
superposition theorem
  current calculation, 19-46–19-47
  deactivated sources, 19-44–19-46
Thévenin theorem
  current calculation, 19-48–19-50
  voltage determination, 19-50–19-51
  voltage divider circuit, 19-47–19-48
voltage rise and drop, 19-7
voltmeter, 19-1–19-2
Nodal analysis
  deactivated circuit configuration, 19-14–19-15
  dependent source, 19-16–19-17
  differentiation and integration operators, 19-19–19-20
  dynamic circuit analysis, 19-20–19-22
  dynamic element relationships, 19-18–19-19
  floating element, 19-10–19-12
  independent voltage source, 19-13–19-14
  node voltage concept, 19-9–19-10
  nonessential node, 19-17–19-18
  reference node, 19-10, 19-11
  resistive circuit, 19-12–19-13
  supernode and essential node, 19-15, 19-18
Node admittance matrix (NAM)
  determinant, 25-5
Non ideal op-amp
  common-mode rejection ratio (CMRR)
    common-mode gain, 16-16–16-17
    differential and noninverting configuration, 16-17
    total output voltage, 16-17–16-18
  electrical noise
    amplifier noise, 16-21–16-22
    circuit's noise performance, 16-23
    white-noise voltage and density, 16-23–16-24
  finite differential gain
    frequency dependent gain, 16-11–16-12
    noninverting configuration, 16-13
    output saturation, 16-13–16-14
  finite input impedance, 16-18–16-19
  input bias currents
    input offset current, 16-20
    nonzero offset current, 16-20–16-21
  nonzero output impedance, 16-19–16-20
  offset voltage
    output offset voltage, 16-15–16-16
    transfer function, 16-14–16-15
  power-supply rejection ratio, 16-18
Norton theorem, 24-31–24-32
  current calculation and power dissipation, 19-54–19-55
  equivalent circuit, 19-51–19-52
  mesh equation method, 19-52–19-53

O

Ohm's law, 24-1–24-2
On-chip transformers
  analytical transformer models
    capacitances, 12-18
    stacked transformer, 12-17
    tapped transformer, 12-16

monolithic transformer realizations
models and coupling expression, 12-15–12-16
port applications, 12-16–12-7
Operational amplifier (op-amp)
BJT
analog computer, 15-21–15-22
classical architecture, 15-22–15-23
features, 15-22
high-gain differential stage, 15-23–15-24
second amplifier stage, 15-24–15-26
specifications, 15-26–15-27
circuit schematic symbol, 18-4, 18-5
current feedback op-amp (CFOA)
bipolar realization, 17-5
closed-loop gain, 17-3–17-4
CMOS compound device, 17-5
frequency response, 17-7, 17-9
macromodel, 17-2–17-3
open and close-loop transimpedance, 17-7, 17-9
output impedance characteristics, 17-7–17-8
schematics and layout plot, 17-6–17-7
step response and harmonic distortion, 17-11
total impedance, 17-3
voltage buffer performance, 17-7
current-mode op-amp
closed-loop characteristics, 16-35–16-36
current-mode op-amp dynamics, 16-37–16-38
higher-order effects, 16-38–16-40
ideal op-amp
circuit applications, 16-3–16-8
comparators, 16-9–16-10
inverting amplifier configuration, 16-10–16-11
open-loop equivalent circuit, 16-1–16-3
voltage op-amps, 16-3–16-11
nonideal op-amp
common-mode rejection ratio (CMRR), 16-16–16-18
electrical noise, 16-21–16-24
finite differential gain, 16-11–16-14
finite input impedance, 16-18–16-19
input bias currents, 16-20–16-21
nonzero output impedance, 16-19–16-20
offset voltage, 16-14–16-16
power-supply rejection ratio, 16-18
voltage-mode op-amps
closed-loop frequency response, 16-26–16-30
closed-loop transient response, 16-30–16-32
current-mode op amps, 16-32–16-34
open-loop voltage gain, 16-26
SPICE stimulation, 16-32
Optical low-noise preamplifiers
front-end noise sources
amplifier and resistor noise, 17-18
PIN photodiode noise, 17-17–17-18
receiver performance criteria
dynamic range, 17-21
equivalent input noise current, 17-18–17-19
intersymbol interference, 17-20–17-21
optical sensitivity, 17-19–17-20
photodiode terminal, SNR, 17-20
transimpedance (TZ) amplifiers, 17-21–17-22

P

Parameter extraction method
indefinite admittance matrix (IAM) approach
basics, 25-10–25-11
resistive circuit, 25-11–25-12
interpolation method, 25-16–25-17
techniques and limitations, 25-10
two graph-based tableau approach
determinants and cotree, 25-13–25-14
element stamps, 25-12
matrices, 25-13, 25-15
tableau equation, 25-13
voltage and current graph, 25-12, 25-14–25-15
Paraunitary filter banks
dilation equation application, 6-63–6-64
infinite product convergence, 6-64–6-65
multiresolution analysis
fast wavelet transform, 6-58
orthonormal multiresolution, 6-57
power symmetric filter, 6-56
wavelet generation, 6-57–6-58
orthonormal wavelet basis
Haar basis, 6-69
L^2 and pointwise convergence, 6-67–6-68
Lawton's eigenfunction condition, 6-68
nonorthonormal, 6-69–6-70
scaling and wavelet functions, 6-68
parametrization, 6-27
wavelet tight frames, 6-70–6-71
Partial fraction expansion (PFE)
repeated linear factors, 3-31
repeated quadratic factors, 3-31–3-32
simple linear factors, 3-27–3-30
simple quadratic factors, 3-29–3-30
Passive circuit elements
active filter synthesis, SFG
bandpass passive filter, 21-9
Butterworth low-pass filter, 21-7–21-9
Chebyshev low-pass response, 21-10
component op-amp circuits, 21-5–21-7
doubly terminated passive filter, 21-4

general ladder network, 21-4–21-5
leapfrog realization, 21-4–21-5
normalization, 21-10–21-11
capacitor
dielectric constant, 11-24
geometry, 11-34, 11-36–11-37
linear capacitor, 11-26–11-30
nonideal linear capacitor, 11-37–11-39
nonlinear capacitor, 11-31–11-33
power and quality factor, 11-25
properties, 11-33–11-35
v' index, 11-25–11-26
inductor
air inductor, 11-45–11-49
circuit representation, 11-45
cored inductor, 11-49–11-60
definition, 11-63–11-64
functions and parameters, 11-44
fundamental observations, 11-42, 11-43
graphical symbols, 11-40
inductance, 11-40–11-42
International Standards, 11-63
magnetic core materials, 11-61, 11-62
magnetic material measurements, 11-61, 11-63
nonlinearity, 11-60–11-61
physical irregularities, 11-63
qualifiers and attributes, 11-42–11-43
resistor
fixed resistor, 11-2–11-7
magnetic geometry, 11-17–11-20
magnetic properties, 11-11–11-17
nonideal linear resistor, 11-20–11-23
photoresistor and magnetoresistor, 11-11
thermistor, 11-10–11-11
variable resistor, 11-7–11-9
varistor, 11-9–11-10
semiconductor diode
depletion capacitances, 11-79–11-80
diode capacitances, 11-78–11-79
impurity profiles, 11-73, 11-74
nonlinear static I-V characteristics, 11-73–11-78
piecewise linear model, 11-83–11-84
as switch, 11-80–11-82
symbols, 11-84
temperature properties, 11-82–11-83
transformer
characteristics, 11-66–11-67
circuit model, 11-70–11-72
definition, 11-66, 11-72
ideal transformer model, 11-67–11-68
leakage and magnetizing inductances, 11-69–11-70
linear transformer model, 11-68–11-69
Phasor transforms
Euler's identity, 24-49
integration constant, 24-52
sine and cosine function, 24-50
Polar polymer dielectric capacitor, 11-32–11-33
Pole–zero analysis
linear system, LT
BIBO stability, 3-38–3-39
natural modes, 3-37–3-38
stability and natural modes, 3-39–3-41
ULT and BLT
general function, $F(s)$, 3-15
polynomial factorization, 3-16
ROC, 3-16–3-18
Port representations
feedback systems
Beta Network, 20-16–20-17
phase and gain margins, 20-15
Tow–Thomas biquad, 20-18–20-19
transfer function, 20-15
voltage-controlled voltage source (VCVS), 20-16
two-port network
bipolar transistor, 20-11–20-12
calculation, 20-7
Cramer's rule and circuit models, 20-10–20-11
hybrid parameters, 20-8–20-9
interconnections, 20-13–20-14
port voltages and current, 20-5
transmission parameters, 20-6, 20-14–20-15
y parameters, 20-6–20-8
z parameters, 20-6
Positive real function (PRF), 9-3
Positive temperature coefficient (PTC), 11-10
Power supply rejection ratio (PSRR), 15-27
Pulse-width modulation (PWM) amplifier
advantage and disadvantage, 15-35
architecture, 15-33, 15-34
configurations, 15-35–15-36
duty cycle, 15-33–15-34
IC amplifier, 15-3
Pulse-width modulation (PWM) IC amplifier, 15-3

R

Reactance function (RF), 9-3
Reactance Hurwitz polynomial (RHP), 9-7
Real-time filtering, Fourier block
block algorithms, 4-24
overlap-add processing, 4-23–4-24
overlap-save processing, 4-23
Reciprocity theorem
statement, 19-58
verification, 19-58–19-59
Resistive circuit element
bipolar NPN transistor, 13-5–13-6
op-amp, 13-6–13-7
photodiode, 13-4–13-5
tunnel diode, 13-5

Resistor
geometry
fail-safe resistor, 11-19–11-20
potentiometer, shape complication, 11-19
resistive element shape, 11-17–11-19
linear resistor
fixed resistor, 11-2–11-7
Ohm's law and types, 11-1–11-2
variable resistor, 11-7–11-9
nonideal linear resistor
noise, 11-20, 11-21
potentiometer, regulation curve, 11-21–11-23
rotational noise and CRV, 11-21
voltage coefficient of resistance (VCR), 11-20–11-22
nonlinear resistor
photoresistor and magnetoresistor, 11-11
thermistor, 11-10–11-11
varistor, 11-9–11-10
properties
ceramic substrate influence, 11-16–11-17
parameter comparison, 11-15–11-16
polymer resistor, 11-15
pyrolytic carbon resistor, 11-14
TCR, 11-11–11-17
thick film resistor, 11-14–11-15
thin film resistor, 11-13–11-14
RF low-noise amplifiers
cascode common source LNA
input matching, 17-15
noise figure, 17-16
voltage gain, 17-15–17-16
CMOS common-source LNA
CS input stage, noise, 17-13
gate-drain capacitance (C_{gd}) effect, 17-14–17-15
input impedance matching, 17-12–17-13
simplified analysis, 17-12
voltage amplifier, inductive load, 17-14
gain, 17-10–17-12
input impedance matching and linearity, 17-12
noise, 17-8
RF passive IC components
fractal capacitors
boost factor *vs.* horizontal spacing, 12-7
bottom-plate parasitic capacitance, 12-5–12-6
fractal dimension, 12-4–12-5, 12-6
high-density capacitor, 12-2
interdigitated capacitor, 12-7–12-8
lateral flux capacitors, 12-2–12-3
parallel-plate and MOS, 12-1–12-2
quality factor and lateral spacing, 12-7
three-dimensional representation, 12-5, 12-6
woven structure, 12-8
on-chip transformers, 12-4
analytical transformer models, 12-16–12-19
monolithic transformer realizations, 12-15–12-17
spiral inductors
noise coupling, 12-8–12-9
planar spiral inductors, 12-11–12-14
substrate effects, 12-9–12-11
RF power amplifier
classification, 17-24–17-25
PA requirements, 17-24
practical considerations, 17-27

S

Scattering Hurwitz polynomial (SHP), 9-5–9-6
Scattering matrix, circulator, 14-12–14-14
Second-order circuits, time domain analysis
network characterization, 26-17
transfer function, 26-11–26-12
transient and steady-state responses, 26-15–26-17
zero-input response, 26-13–26-14
zero-state response, 26-14–26-15
Semiconductor diode
depletion capacitances, 11-79–11-80
diode capacitances, 11-78–11-79
impurity profiles, 11-73, 11-74
nonlinear static I–V characteristics, 11-73
forward I–V diode characteristics, 11-75, 11-77–11-78
p–n junction equation, 11-74–11-76
reverse I–V characteristics, 11-75, 11-78
piecewise linear model, 11-83–11-84
as switch, 11-80–11-82
symbols, 11-84
temperature properties, 11-82–11-83
Sequence of expressions (SoE), *see* Hierarchical methods
SFG, *see* Signal flow graph
Short-time Fourier transform (STFT)
definition, 6-16–6-17
filter impulse response, 6-17
Gabor window, time–frequency resolution, 6-18–6-19
invertibility, 6-47–6-48
limitations, 6-30
orthonormality, 6-48–6-49
shifted filters, analysis bank, 6-18
time–frequency localization, 6-49–6-50
vs. wavelet transform
mathematical issues, 6-22
Morlet wavelet, 6-21–6-22
signal analysis, 6-20
time–frequency tiling scheme, 6-19–6-20
Signal converters
circulator, 14-12–14-15
2-port circulator, 14-13
3-port circulator, 14-13–14-14
nullator and norator, 14-14, 14-15
nullor, 14-14–14-15
terminated 2-port network, 14-12–14-13
current negative impedance converter, 14-11–14-12

definition, 14-15
gyrator, 14-8–14-10
Antoniou's circuit, 14-9–14-10
ideal 2-port gyrator, 14-8–14-9
network ports, 14-7–14-8
voltage negative impedance converter, 14-11
Signal flow graphs
active filter, filter transfer function
differential-Input op-amp circuit, 21-17
dummy branche insertion, 21-13
four op-amp circuit, 21-18–21-19
gain function, 21-12–21-13
lowpass filter specifications, 21-19
Mason's gain formula, 21-12
RC-op-amp circuits, 21-14–21-15
state variable biquad, 21-19–21-20
transfer voltage ratio, 21-11
two op-amp circuit realizations, 21-15–21-16
active filter synthesis, passive filter circuit
bandpass passive filter, 21-9
Butterworth low-pass filter, 21-7–21-9
Chebyshev low-pass response, 21-10
component op-amp circuits, 21-5–21-7
doubly terminated passive filter, 21-4
general ladder network, 21-4–21-5
leapfrog realization, 21-4–21-5
normalization, 21-10–21-11
adjacency matrix, directed graph
definition, 8-1
edge weight, 8-4
permutation, 8-2–8-3
Coates' gain formula
coefficient matrix, 8-7–8-8
n×n matrix, 8-6–8-7
nonsingular matrix, 8-5
topological formula, 8-4
definition, 25-8
formulation, 25-9, 25-10
Harary theorem, 8-3–8-4
linear network
construction, 21-1–21-2
network branches, 21-1
Mason's gain formula
feedback loops, 8-10
j self-loops, 8-9
Mason graph, 8-8–8-9
Mason's SFG, 25-7–25-8
Similarity transformation, 1-12–1-14
Singular-value decomposition (SVD), 1-17
Sinusoidal steady state analysis
AC circuit power
average power and RMS values, 24-65–24-66
average power delivered, 24-71
input impedance, 24-64
load impedance, 24-70–24-71
maximum average power, 24-72–24-73
power factor (PF), 24-63
power triangle, 24-67
sinusoidal sources, 24-62–24-63
network transfer function, 24-46–24-47
output voltage, 24-48
phase lead and lag, 24-55–24-57
phasor and network, 24-53–24-55
RC and RL network, 24-57–24-59
resonance
parallel resonant circuit, 24-60–24-61
series resonant circuit, 24-61–24-62
steady-state output voltage, 24-49
transient and steady-state response, 24-45
Sinusoids, 26-3
Spiral inductors
noise coupling, 12-8–12-9
planar spiral inductors
current sheet approximation, 12-3–12-14
data-fitted monomial expression, 12-14, 12-15
layouts, 12-11–12-12, 12-13
modified Wheeler formula, 12-12–12-3
substrate effects
patterned ground shield (PGS), 12-10–12-11
physical model, 12-9–12-10
State transition matrix
complete response, 27-20–27–21
Jordan canonical form, 27-16
Laplace transform technique, 27-17
normal mode approach, 27-18
parallel RLC circuit, 27-17
similarity transformation method, 27-19
zero-input response, 27-21
State-variable techniques
complete response, 27-20–27-21
concept of states
capacitive loop, 27-2–27-3
definition, 27-1
simple RC circuit, 27-2
network topology
capacitive tree-branch voltages, 27-8
first-order differential equation, 27-3
inductive link current, 27-8
KCL and KVL, 27-4
order of complexity, 27-1–27-2
resistive link currents, 27-7
state equations, 27-5–27-6
state transition matrix
Jordan canonical form, 27-16
Laplace transform technique, 27-17
normal mode approach, 27-18
parallel RLC circuit, 27-17
similarity transformation method, 27-19
zero-input response, 27-21
Superposition theorem
current calculation, 19-46–19-47
deactivated sources, 19-44–19-46

Symbolic analysis
approximate symbolic analysis
bipolar cascode stage, 25-21–25-23
classification, 25-20–25-21
error criteria, 25-21
determinant decision diagrams (DDD)
matrix determinant and representation, 25-24–25-25
nodal analysis, 25-23–25-24
vertex and vertex ordering, 25-24
frequency domain analysis, 25-3
hierarchical methods
advantage, 25-17, 25-18
bipolar cascode stage, 25-19–25-20
modified nodal analysis and NAM, 25-19
resistance ladder network, 25-17–25-18
topological analysis, 25-18
illustration, 25-1–25-2
time domain analysis
fully symbolic, 25-25
problems and limitation, 25-25
semisymbolic, 25-26
traditional methods
cancellation-free equation, 25-4–25-5
interpolation method, 25-16–25-17
parameter extraction method, 25-10–25-16
$RLCg_m$ circuit and term cancellations, 25-4
signal flowgraph method (SFG), 25-7–25-9, 25-10
tree enumeration methods, 25-5–25-7
Symmetric magnitude response
analog 2-D filter
quadrantal and octagonal symmetry, 10-28–10-29
rotational and diagonal symmetry, 10-29
transfer function, 10-27
bandpass filter
MATLAB optimization, 10-32
specifications, 10-31–10-32
bandstop filter
coefficients and specification, 10-35
contour and 3-D magnitude plots, 10-36
digital filters
quadrantal and octagonal symmetry, 10-29–10-30
rotational and diagonal symmetry, 10-30
high-pass filter
coefficients and specification, 10-34
contour and 3-D magnitude plots, 10-35
low-pass filter
coefficients and specification, 10-33
contour and 3-D magnitude plots, 10-34
magnitude-squared functions, 10-16
Symmetry applications
2-D Fourier transform
continuous-time continuous-frequency case, 10-7–10-8
continuous-time discrete-frequency case, 10-8
discrete-time continuous-frequency case, 10-9
discrete-time discrete-frequency case, 10-9–10-10
2-D magnitude response
continuous-time frequency response, 10-15
discrete-time frequency response, 10-15–10-16
magnitude-squared functions, 10-16
2-D polynomials determination
diagonal symmetry, 10-25–10-26
fourfold rotational symmetry, 10-26
octagonal symmetry, 10-27
quadrantal symmetry, 10-21–10-25
analog 2-D filter stability, magnitude response
quadrantal and octagonal symmetry, 10-28–10-29
rotational and diagonal symmetry, 10-29
transfer function, 10-27
digital filter stability, magnitude response
quadrantal and octagonal symmetry, 10-29–10-30
rotational and diagonal symmetry, 10-30
FFT
one dimensional, 10-11–10-12
symmetrical decomposition, 10-14
two dimensional, 10-12–10-14
filter design, magnitude response
bandpass filter, 10-31–10-33
bandstop filter, 10-35–10-36
high-pass filter, 10-34–10-35
low-pass filter, 10-33–10-34
objective function, 10-29–10-30
transfer function, 10-29
magnitude-squared function
polynomial P, 10-19–10-20
quadrantal symmetry, 10-20
spectral forms, 10-20–10-21
polynomial operations
anti-self-paraconjugate polynomial, 10-18–10-19
paraconjugate operation, 10-17
self-paraconjugate polynomial, 10-17–10-18
T-ψ symmetries
basic symmetries, 10-4–10-5
quadrantal and diagonal symmetry, 10-5–10-6
rotational and octagonal symmetry, 10-6–10-7

T

Taylor series, 3-23
Tellegen theorem
2-port network, 7-21–7-22
adjoint elements, 7-23–7-24
open-circuit voltage ratio, 7-24
voltage and current change, 7-22
Temperature coefficient of capacitance (TCC), 11-24

Tensor
  basis tensors
    finite-dimensional vector spaces, 2-5–2-8
    Kronecker product, 2-8
  tensor product
    arbitrary bilinear operator, 2-4
    isomorphism, 2-5
Terminal representation
  four-terminal network
    first application, 20-3
    passive circuit, 20-1–20-2
    second application, 20-4
  indefinite admittance matrix, 20-3
  network analysis, 20-5
  nodal admittance matrix, 20-2–20-3
Thévenin theorem
  application, 24-28–24-29
  current calculation, 19-48–19-50
  dependent sources, 24-24–24-25
  impedance, 24-26
  source transformation, 24-29–24-30
  subnetworks, 24-24
  voltage determination, 19-50–19-51
  voltage divider circuit, 19-47–19-48
Third-order intercept point (TOI), 15-5
Time domain analysis
  first-order circuits
    network time constant, 26-10
    transfer functions, 26-6–26-7
    transient and steady-state responses, 26-9–26-10
    zero-input response, 26-7–26-8
    zero-state response, 26-8–26-9
  second-order circuits
    network characterization, 26-17
    transfer function, 26-11–26-12
    transient and steady-state responses, 26-15–26-17
    zero-input response, 26-13–26-14
    zero-state response, 26-14–26-15
  signal types
    causal circuit, 26-4
    eternal signals, 26-3–26-4
    impulse, 26-2–26-3
    periodic and aperiodic waveforms, 26-4–26-6
    ramp function, 26-2
    sinusoids, 26-3
    unit step function, 26-1–26-2
z-Transform
  discrete-time FT, 5-3–5-4
  generating function, 5-2
  inverse *z*-transform
    contour integration, 5-5–5-7
    inspection method and power series expansion, 5-8
    partial fraction expansion, 5-7–5-8
  linear time-invariant systems
    one-sided *z*-transform, 5-14
    system transfer function, 5-13–5-14
  properties
    ROC, 5-8–5-9
    transform, 5-9–5-10
    z-domain convolution, 5-10–5-12
  ROC, unit circle, 5-3
  two-sided *z*-transform, 5-2
  variations
    chirp *z*-transform algorithm, 5-17
    modified *z*-transform, 5-16
    multidimensional *z*-transforms, 5-15
Transform domain adaptive filter (TDAF)
  adaptive tap vector, 4-25
  autocorrelation matrix, 4-24–4-25
  characteristics, 4-26–4-27
  filter output and LMS adaptive algorithm, 4-24
  Karhunen–Loe'ye transform (KLT), 4-26
  power, 4-25–4-26
  transformed outputs, 4-25
Transformer
  characteristics, 11-66–11-67
  definition, 11-66, 11-72
  ideal transformer model, 11-67–11-68
  nonideal transformer
    circuit model, 11-70–11-72
    leakage and magnetizing inductances, 11-69–11-70
    linear transformer model, 11-68–11-69
Transimpedance (TZ) amplifiers, 17-21–17-22
Tree enumeration methods
  augmented circuit, 25-6–25-7
  circuit construction, 25-5–25-6
  NAM determinant and cofactors, 25-5
Tree-structured filter banks
  analysis and synthesis bank, 6-28
  analysis filter response, 6-28–6-29
Two-dimensional (2-D) digital filters
  2-D symmetries
    centrosymmetry, 10-13
    quadrantal symmetry, 10-14
  magnitude response symmetry
    continuous-time frequency response, 10-15
    discrete-time frequency response, 10-15–10-16
    magnitude response, 10-16
  symmetry
    anti-self-paraconjugate polynomial, 10-18–10-19
    magnitude-squared function, 10-19–10-21
    octagonal, 10-6–10-7
    paraconjugate operation, 10-17
    quadrantal and diagonal, 10-5–10-6
    reflection and rotational, 10-4–10-5
    self-paraconjugate polynomial, 10-17–10-18
    symmetry-based 1-D FFT, 10-11–10-12
    T operations, 10-3
    T–ψ operations, 10-2–10-3
    ψ-operation, 10-4
  two-dimensional (2-D) Fourier transform
    continuous-time continuous-frequency, 10-7–10-8

discrete-time continuous-frequency, 10-9
discrete-time discrete-frequency, 10-9–10-10
Two graph-based tableau approach
determinants and cotree, 25-13–25-14
element stamps, 25-12
matrices, 25-13, 25-15
tableau equation, 25-13
voltage and current graph, 25-12, 25-14–25-15
Two-port network
bipolar transistor, 20-11–20-12
calculation, 20-7
Cramer's rule and circuit models, 20-10–20-11
hybrid parameters, 20-8–20-9
interconnections, 20-13–20-14
port voltages and current, 20-5
transmission parameters, 20-6, 20-14–20-15
y parameters, 20-6–20-8
z parameters, 20-6
Two-variable polynomial
1-D complex plane, 9-3
2-D biplane, 9-4
nonessential singularities second kind (NSSK), 9-4

U

Unilateral Laplace transforms
direct computation, 3-11–3-15
operational properties, 3-18–3-22

V

Variable resistor
electric circuit, 11-7, 11-8
tapers, 11-9
types, 11-7–11-9
Vector spaces over fields
field *F*, properties, 1-2
F-vector space, 1-2–1-3
n-dimensional vector space, 1-3–1-4
Very strict Hurwitz polynomial (VSHP), 9-6
Voltage-controlled current source (VCCS)
circuit implementation, 14-2–14-3
circuit schematic symbol, 18-3
equivalent circuit, MOSFET, 14-1–14-2
SPICE format, 14-2, 14-4
Voltage-controlled voltage source (VCVS), 14-3
circuit schematic symbol, 18-1–18-3
dynamic model, op-amp, 13-9, 13-10
SPICE format and circuit implementation, 14-4
Voltage-dependent resistor (VDR), 11-9
Voltage inversion negative impedance converter (VNIC), 14-11
Voltage-mode op-amps
closed-loop frequency response
closed-loop small-signal bandwidth, 16-28
crossover frequency, 16-29
feedback factor, 16-27
closed-loop transient response, 16-30–16-32
current-mode op amps, 16-32–16-34
open-loop voltage gain, 16-26
SPICE stimulation, 16-32
2-V very strict Hurwitz polynomial (VSHP), 9-10–9-11

W

Walsh–Hadamard transform (WHT), 4-20–4-21
Wavelet transform (WT) technique
basis functions, 6-10
digital filter banks
basis function, 6-29–6-30
half-band filter, 6-27
paraunitary filter banks, parametrization, 6-27
paraunitary properties, 6-25–6-27
polyphase representation, 6-24–6-25
subband construction, 6-22–6-24
tree-structured system, 6-28–6-29
two-channel filter bank, 6-22
filter banks
synthesis and reconstruction, 6-13–6-14
time–frequency representation, 6-13
wavelet function design, 6-14–6-15
Fourier basis, 6-15
Haar wavelet basis, 6-11–6-12
Hilbert space frame
arbitrary vector representation, 6-44
definition, 6-44
dual frame and sequence, 6-47
frame operator, 6-46
Riesz basis, 6-44–6-45
tight frame, 6-46
ideal bandpass wavelet
Fourier transform, 6-5–6-6
frequency axis splitting, 6-6–6-7
impulse response, 6-8
vs. low-pass, 6-5–6-6
L^1 and L^2 signal space
convolution integral, 6-36–6-37
Lebesgue integrals, 6-31–6-32
L^P signal, 6-32–6-34
L^2 functions, 6-8–6-10
multiresolution and wavelets
continuous time filter banks, 6-59–6-60
frequency domain, 6-51
ideal filter, 6-51–6-52
paraunitary filter banks, 6-56–6-58
time domain, 6-50
notations and acronyms, 6-3
octave-band splitting scheme, 6-8, 6-9
orthogonal projections, 6-10
orthonormal wavelet, 6-71–6-72
short-time Fourier transform
definition, 6-16–6-17
filter impulse response, 6-17
Gabor window, time–frequency resolution, 6-18–6-19
invertibility, 6-47–6-48
limitations, 6-30
orthonormality, 6-48–6-49

shifted filters, analysis bank, 6-18
time–frequency localization, 6-49–6-50
wavelet regularity
ninth-order FIR filter, 6-73–6-74
smoothness and Hölder regularity index, 6-74
time-domain decay and regularity, 6-77–6-78
two-scale dilation equation, 6-72–6-73
wavelet transforms
coefficient computation, 6-10–6-11
CWT and DWT, 6-15
definition, 6-11
frequency response, 6-16
properties, 6-12–6-13
vs. short-time Fourier transform, 6-19–6-22
Wideband amplifiers
composite stages, 15-27–15-28
feedback cascades, 15-28–15-29

Upon its initial publication, *The Circuits and Filters Handbook* broke new ground. It quickly became **the** resource for comprehensive coverage of issues and practical information that can be put to immediate use. Not content to rest on his laurels, in addition to updating the second edition, editor Wai-Kai Chen divided it into tightly focused texts that made the information easily accessible and digestible. These texts have been revised, updated, and expanded so that they continue to provide solid coverage of standard practices and enlightened perspectives on new and emerging techniques.

Fundamentals of Circuits and Filters draws together international contributors who provide a unified view of key concepts in the theory of linear circuits and filters, emphasizing interrelated concepts and providing a mathematical reference to the handbook itself. They include a brief introduction to the theory and application of the z-transform, explain the advantages of the wavelet transform and the reason for its recent popularity, present a self-contained development of graph theory, and develop signal flow graphs following the closely related methods of Coates and Mason. The book describes the limited palette of options available in passive energy storage elements and how to make the most use out of what is available. It presents amplifier designs, terminal and port representations of circuits, and an overview of the different approaches to network analysis, highlighting the advantages and disadvantages of each.

Revised chapters:

- Linear Operators and Matrices
- Bilinear Operators and Matrices
- Fourier Methods for Signal Analysis and Processing
- Application of Symmetry: Two-Dimensional Polynomials, Fourier Transforms, and Filter Design
- Passive Circuit Elements
- Bipolar Junction Transistor Amplifiers
- Operational Amplifiers
- Terminal and Port Representations
- Symbolic Analysis

This volume will undoubtedly take its place as the engineer's first choice in looking for solutions to problems encountered in the analysis and behavior predictions of circuits and filters.

58878

ISBN: 978-1-4200-5887-1
90000
9 781420 058871

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

The Circuits and Filters Handbook

Third Edition

Analog and VLSI Circuits

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

Analog and VLSI Circuits

The Circuits and Filters Handbook

Third Edition

Edited by

Wai-Kai Chen

Fundamentals of Circuits and Filters

Feedback, Nonlinear, and Distributed Circuits

Analog and VLSI Circuits

Computer Aided Design and Design Automation

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Analog and VLSI Circuits

Edited by

Wai-Kai Chen

University of Illinois

Chicago, U. S. A.

CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the

Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5891-8 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Analog and VLSI circuits / edited by Wai-Kai Chen.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-4200-5891-8
ISBN-10: 1-4200-5891-6
1. Linear integrated circuits. 2. Integrated circuits--Very large scale integration. 3. Electronic circuits. I. Chen, Wai-Kai, 1936- II. Title.

TK7874.654.A47 2009
621.39'5--dc22 2008048128

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface vii

Editor-in-Chief ix

Contributors xi

SECTION I Analog Integrated Circuits

1 Monolithic Device Models **1**-1
Bogdan M. Wilamowski, Guofu Niu, John Choma, Jr., Stephen I. Long, Nhat M. Nguyen, and Martin A. Brooke

2 Analog Circuit Cells **2**-1
Kenneth V. Noren, John Choma, Jr., J. Trujillo, David G. Haigh Bill Redman-White, Rahim Akbari-Dilmaghani, Mohammed Ismail, Shu-Chuan Huang, Chung-Chih Hung, and Trond Saether

3 High-Performance Analog Circuits **3**-1
Chris Toumazou, Alison Payne, John Lidgey, Alicja Konczakowska, and Bogdan M. Wilamowski

4 RF Communication Circuits **4**-1
Michiel Steyaert, Wouter De Cock, and Patrick Reynaert

5 PLL Circuits **5**-1
Muh-Tian Shiue and Chorng-Kuang Wang

6 Synthesis of Reactance Pulse-Forming Networks **6**-1
Igor M. Filanovsky

SECTION II The VLSI Circuits

7 Fundamentals of Digital Signal Processing **7**-1
Roland Priemer

8 Digital Circuits **8**-1
John P. Uyemura, Robert C. Chang, and Bing J. Sheu

9 Digital Systems **9**-1
Festus Gail Gray, Wayne D. Grover, Josephine C. Chang, Bing J. Sheu Roland Priemer, Kung Yao, and Flavio Lorenzelli

10 Data Converters **10**-1
Bang-Sup Song and Ramesh Harjani

Index **IN**-1

Preface

The purpose of this book is to provide in a single volume a comprehensive reference work covering the broad spectrum of monolithic device models, high-performance analog circuits, radio-frequency communications and PLL circuits, digital systems, and data converters. This book is written and developed for the practicing electrical engineers and computer scientists in industry, government, and academia. The goal is to provide the most up-to-date information in the field.

Over the years, the fundamentals of the field have evolved to include a wide range of topics and a broad range of practice. To encompass such a wide range of knowledge, this book focuses on the key concepts, models, and equations that enable the design engineer to analyze, design, and predict the behavior of large-scale circuits and systems. While design formulas and tables are listed, emphasis is placed on the key concepts and theories underlying the processes.

This book stresses fundamental theories behind professional applications and uses several examples to reinforce this point. Extensive development of theory and details of proofs have been omitted. The reader is assumed to have a certain degree of sophistication and experience. However, brief reviews of theories, principles, and mathematics of some subject areas are given. These reviews have been done concisely with perception.

The compilation of this book would not have been possible without the dedication and efforts of Professor John Choma, Jr., and most of all the contributing authors. I wish to thank them all.

Wai-Kai Chen

Editor-in-Chief



Wai-Kai Chen is a professor and head emeritus of the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. He received his BS and MS in electrical engineering at Ohio University, where he was later recognized as a distinguished professor. He earned his PhD in electrical engineering at the University of Illinois at Urbana–Champaign.

Professor Chen has extensive experience in education and industry and is very active professionally in the fields of circuits and systems. He has served as a visiting professor at Purdue University, the University of Hawaii at Manoa, and Chuo University in Tokyo, Japan. He was the editor-in-chief of the *IEEE Transactions on Circuits and Systems*, *Series I and II*, the president of the IEEE Circuits and Systems Society, and is the founding editor and the editor-in-chief of the *Journal of Circuits, Systems and Computers*. He received the Lester R. Ford Award from the Mathematical Association of America; the Alexander von Humboldt Award from Germany; the JSPS Fellowship Award from the Japan Society for the Promotion of Science; the National Taipei University of Science and Technology Distinguished Alumnus Award; the Ohio University Alumni Medal of Merit for Distinguished Achievement in Engineering Education; the Senior University Scholar Award and the 2000 Faculty Research Award from the University of Illinois at Chicago; and the Distinguished Alumnus Award from the University of Illinois at Urbana–Champaign. He is the recipient of the Golden Jubilee Medal, the Education Award, and the Meritorious Service Award from the IEEE Circuits and Systems Society, and the Third Millennium Medal from the IEEE. He has also received more than a dozen honorary professorship awards from major institutions in Taiwan and China.

A fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the American Association for the Advancement of Science (AAAS), Professor Chen is widely known in the profession for the following works: *Applied Graph Theory* (North-Holland), *Theory and Design of Broadband Matching Networks* (Pergamon Press), *Active Network and Feedback Amplifier Theory* (McGraw-Hill), *Linear Networks and Systems* (Brooks/Cole), *Passive and Active Filters: Theory and Implements* (John Wiley), *Theory of Nets: Flows in Networks* (Wiley-Interscience), *The Electrical Engineering Handbook* (Academic Press), and *The VLSI Handbook* (CRC Press).

Contributors

Rahim Akbari-Dilmaghani
Department of Electronic and Electrical Engineering
University College of London
London, United Kingdom

Martin A. Brooke
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia

Josephine C. Chang
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

Robert C. Chang
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

John Choma, Jr.
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

Wouter De Cock
Department of Electrical Engineering
Catholic University of Leuven
Leuven, Belgium

Igor M. Filanovsky
Department of Electrical Engineering
University of Alberta
Edmonton, Alberta, Canada

Festus Gail Gray
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia

Wayne D. Grover
Network Systems
TRLabs
Edmonton, Alberta, Canada

and

Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Alberta, Canada

David G. Haigh
Department of Electronic and Electrical Engineering
University College of London
London, United Kingdom

Ramesh Harjani
Department of Electrical Engineering
University of Minnesota
Minneapolis, Minnesota

Shu-Chuan Huang
Department of Electrical Engineering
Ohio State University
Columbus, Ohio

Chung-Chih Hung
Department of Electrical Engineering
Tatung Institute of Technology
Taipei, Taiwan

Mohammed Ismail
Department of Electrical Engineering
Ohio State University
Columbus, Ohio

Alicja Konczakowska
Department of Optoelectronics and Electronics Systems
Gdansk University of Technology
Gdansk, Poland

John Lidgey
School of Technology
Oxford Brookes University
London, United Kingdom

Stephen I. Long
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, California

Flavio Lorenzelli
SGS-Thomson Microelectronics
Milan, Italy

and

University of Milan, Crema
Crema, Italy

Nhat M. Nguyen
Rambus Inc.
Los Altos, California

Guofu Niu
Department of Electrical and Computer Engineering
Auburn University
Auburn, Alabama

Kenneth V. Noren
Department of Electrical and Computer Engineering
University of Idaho
Moscow, Idaho

Alison Payne
Institute of Biomedical Engineering
Imperial College of Science, Technology, and Medicine
London, United Kingdom

Roland Priemer
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, Illinois

Bill Redman-White
School of Electronics and Computer Science
University of Southampton
Southampton, United Kingdom

Patrick Reynaert
Department of Electrical Engineering
Catholic University of Leuven
Leuven, Belgium

Trond Saether
Nordic VLSI A/S
Flatasen, Norway

Bing J. Sheu
Taiwan Semiconductor Manufacturing Company
Hsin-Chu, Taiwan

Muh-Tian Shiue
Department of Electrical Engineering
National Central University
Chung-Li, Taiwan

Bang-Sup Song
Department of Electrical and Computer Engineering
University of California, San Diego
San Diego, California

Michiel Steyaert
Department of Electrical Engineering
Catholic University of Leuven
Leuven, Belgium

Chris Toumazou
Institute of Biomedical Engineering
Imperial College of Science, Technology, and Medicine
London, United Kingdom

J. Trujillo
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

John P. Uyemura
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia

Chorng-Kuang Wang
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

Bogdan M. Wilamowski
Alabama Nano/Micro Science and Technology Center
Department of Electrical and Computer Engineering
Auburn University
Auburn, Alabama

Kung Yao
Electrical Engineering Department
University of Southern California, Los Angeles
Los Angeles, California

I

Analog Integrated Circuits

John Choma, Jr.
University of Southern California

1 Monolithic Device Models *Bogdan M. Wilamowski, Guofu Niu, John Choma, Jr., Stephen I. Long, Nhat M. Nguyen, and Martin A. Brooke* **1**-1
Bipolar Junction Transistor • References • Metal–Oxide–Silicon Field Effect Transistor • References • JFET, MESFET, and HEMT Technology and Devices • References • Passive Components • References • Chip Parasitics in Analog Integrated Circuits • References

2 Analog Circuit Cells *Kenneth V. Noren, John Choma, Jr., J. Trujillo, David G. Haigh, Bill Redman-White, Rahim Akbari-Dilmaghani, Mohammed Ismail, Shu-Chuan Huang, Chung-Chih Hung, and Trond Saether* **2**-1
Bipolar Biasing Circuits • References • Canonic Cells of Linear Bipolar Technology • References • MOSFET Biasing Circuits • References • Canonical Cells of MOSFET Technology • References

3 High-Performance Analog Circuits *Chris Toumazou, Alison Payne, John Lidgey, Alicja Konczakowska, and Bogdan M. Wilamowski* **3**-1
Broadband Bipolar Networks • Appendix A: Transfer Function and Bandwidth Characteristic of Current-Feedback • Appendix B: Transfer Function and Bandwidth Characteristic of Voltage-Feedback • Appendix C: Transconductance of the Current-Feedback Op-Amp Input Stage • Appendix D: Transfer Function of Widlar Current Mirror • Appendix E: Transfer Function of Widlar Current Mirror with Emitter Degeneration Resistors • References • Bipolar Noise • References

4 RF Communication Circuits *Michiel Steyaert, Wouter De Cock, and Patrick Reynaert* **4**-1
Introduction • System Level RF Design • Technology • Receiver • Synthesizer • Transmitter • References

5 PLL Circuits *Muh-Tian Shiue and Chorng-Kuang Wang* **5**-1
Introduction • PLL Techniques • Building Blocks of PLL Circuit • PLL Applications • Bibliography

6 Synthesis of Reactance Pulse-Forming Networks *Igor M. Filanovsky* **6**-1
Introduction • Networks Forming Quasi-Rectangular Output Pulses • Transfer Functions of Wideband Amplifiers • Forming a Sinusoidal Pulse • Summary • References

1

Monolithic Device Models

1.1 Bipolar Junction Transistor **1**-1
Ebers–Moll Model • Gummel–Poon Model • Current Gains of Bipolar Transistors • High-Current Phenomena • Small-Signal Model • Technologies • Model Parameters • SiGe HBTs
References **1**-20
1.2 Metal–Oxide–Silicon Field Effect Transistor **1**-21
Introduction • Channel Charge • Volt–Ampere Characteristics • Transistor Capacitances • Small-Signal Operation • Design-Oriented Analysis Strategy
References **1**-81
1.3 JFET, MESFET, and HEMT Technology and Devices **1**-82
Introduction • Silicon JFET Device Operation and Technology • Compound Semiconductor FET Technologies • Conclusion
References **1**-101
1.4 Passive Components **1**-103
Resistors • Capacitors • Inductors
References **1**-131
1.5 Chip Parasitics in Analog Integrated Circuits **1**-132
Interconnect Parasitics • Pad and Packaging Parasitics • Parasitic Measurement
References **1**-145

Bogdan M. Wilamowski
Auburn University

Guofu Niu
Auburn University

John Choma, Jr.
University of Southern California

Stephen I. Long
University of California, Santa Barbara

Nhat M. Nguyen
Rambus Inc.

Martin A. Brooke
Georgia Institute of Technology

1.1 Bipolar Junction Transistor

Bogdan M. Wilamowski and Guofu Niu

The bipolar junction transistor (BJT) is historically the first solid-state analog amplifier and digital switch, and formed the basis of integrated circuits (ICs) in the 1970s. Starting in the early 1980s, the MOSFET had gradually taken over, particularly for main stream digital ICs. However, in the 1990s, the invention of silicon–germanium base heterojunction bipolar transistor (SiGe HBT) brought the bipolar transistor back into high-volume commercial production, mainly for the now widespread wireless and wire line communications applications. Today, SiGe HBTs are used to design radio-frequency (RF) ICs and systems for cell phones, wireless local area network (WLAN), automobile collision avoidance

radar, wireless distribution of cable television, millimeter wave radios, and many more applications, due to its outstanding high-frequency performance and ability to integrate with CMOS for realizing digital, analog, and RF functions on the same chip.

Below we first introduce the basic concepts of BJT using a historically important equivalent circuit model, the Ebers–Moll model. Then the Gummel–Poon model is introduced, as it is widely used for computer-aided design, and is the basis of modern BJT models like the VBIC, Mextram, and HICUM models. Current gain, high-current phenomena, fabrication technologies, and SiGe HBTs are then discussed.

1.1.1 Ebers–Moll Model

A NPN BJT consists of two closely spaced PN junctions connected back to back sharing the same p-type region, as shown in Figure 1.1a. The drawing is not drawn to scale. The emitter and base layers are thin, typically less than 1 μm, and the collector is much thicker to support a high output voltage swing. For forward mode operation, the emitter–base (EB) junction is forward biased, and the collector–base (CB) junction is reverse biased. Minority carriers are injected from emitter to base, travel across the base, and are then collected by the reverse biased CB junction. Therefore, the collector current is transported from the EB junction, and thus proportional to the EB junction current. In the forward-active mode, the current–voltage characteristic of the EB junction is described by the well-known diode equation

$$I_{EF} = I_{E0}\left[\exp\left(\frac{V_{BE}}{V_T}\right) - 1\right] \tag{1.1}$$

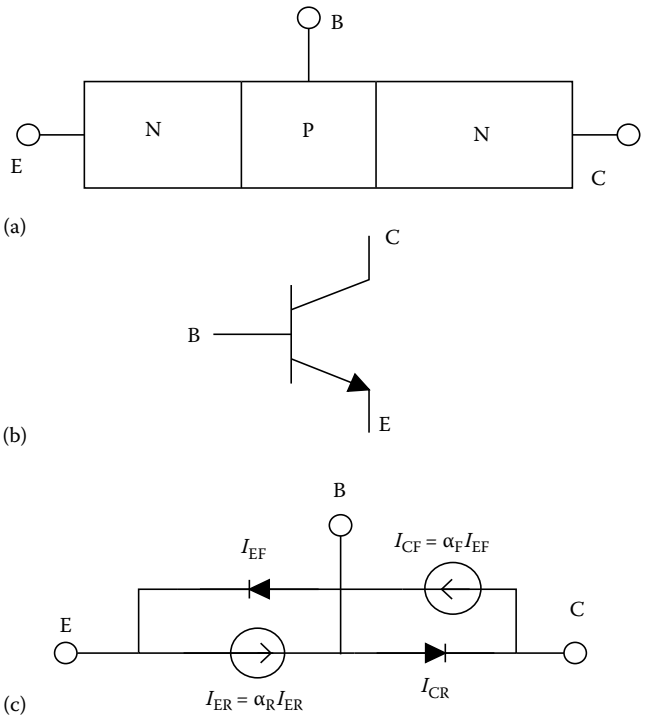


FIGURE 1.1 (a) Cross-sectional view of a NPN BJT. (b) Circuit symbol. (c) The Ebers–Moll equivalent circuit model.

where

I_{E0} is the EB junction saturation current

$V_T = kT/q$ is the thermal potential (about 25 mV at room temperature)

The collector current is typically smaller than the emitter current $I_{CF} = \alpha_F I_{EF}$, where α_F is the forward current gain.

Under reverse mode operation, the CB junction is forward biased and the EB junction is reverse biased. Like in the forward mode, the forward biased CB junction current gives the collector current

$$I_{CF} = I_{C0}\left[\exp\left(\frac{V_{BC}}{V_T}\right) - 1\right] \tag{1.2}$$

where I_{C0} is the CB junction saturation current. Similarly $I_{ER} = \alpha_R I_R$, where α_R is the reverse current gain. Under general biasing conditions, it can be proven that to first order, a superposition of the above described forward and reverse mode equivalent circuits can be used to describe transistor operation, as shown in Figure 1.1b. The forward transistor operation is described by Equation 1.1, and the reverse transistor operation is described by Equation 1.2. From the Kirchoff's current law one can write $I_C = I_{CF} - I_{CR}$, $I_E = I_{EF} - I_{ER}$, and $I_B = I_E - I_C$. Using Equations 1.1 and 1.2 the emitter and collector currents can be described as

$$\begin{aligned} I_E &= a_{11}\left(\exp\frac{V_{BE}}{V_T} - 1\right) - a_{12}\left(\exp\frac{V_{BC}}{V_T} - 1\right) \\ I_C &= a_{21}\left(\exp\frac{V_{BE}}{V_T} - 1\right) - a_{22}\left(\exp\frac{V_{BC}}{V_T} - 1\right) \end{aligned} \tag{1.3}$$

which are known as the Ebers–Moll equations [1]. The Ebers–Moll coefficients a_{ij} are given as

$$a_{11} = I_{E0}, \quad a_{12} = \alpha_R I_{C0}, \quad a_{21} = \alpha_F I_{E0}, \quad a_{22} = I_{C0} \tag{1.4}$$

The Ebers–Moll coefficients are a very strong function of the temperature

$$a_{ij} = K_x T^m \exp\frac{V_{go}}{V_T} \tag{1.5}$$

where

K_x is proportional to the junction area and independent of the temperature

$V_{go} = 1.21$ V is the bandgap voltage in silicon (extrapolated to 0 K)

m is a material constant with a value between 2.5 and 4

When both EB and CB junctions are forward biased, the transistor is called to be working in the saturation region. Current injection through the collector junction may activate parasitic transistors in ICs using p-type substrate, where base acts as emitter, collector as base, and substrate as collector. In typical ICs, bipolar transistors must not operate in saturation. Therefore, for the integrated bipolar transistor the Ebers–Moll equations can be simplified to the form

$$\begin{aligned} I_E &= a_{11}\left(\exp\frac{V_{BE}}{V_T} - 1\right) \\ I_C &= a_{21}\left(\exp\frac{V_{BE}}{V_T} - 1\right) \end{aligned} \tag{1.6}$$

where $a_{21}/a_{11} = \alpha_F$. This equation corresponds to the circuit diagram shown in Figure 1.1c.

1.1.2 Gummel–Poon Model

In real bipolar transistors the current voltage characteristics are more complex than those described by the Ebers–Moll equations. Typical current–voltage characteristics of the bipolar transistor, plotted in semilogarithmic scale, are shown in Figure 1.2. At small-base emitter voltages, due to the generation–recombination phenomena, the base current is proportional to

$$I_{BL} \propto \exp\frac{V_{BE}}{2V_T} \tag{1.7}$$

Also, due to the base conductivity modulation at high-level injections, the collector current for larger voltages can be expressed by the similar relation

$$I_{CH} \propto \exp\frac{V_{BE}}{2V_T} \tag{1.8}$$

Note, that the collector current for wide range is given by

$$I_C = I_s \exp\frac{V_{BE}}{V_T} \tag{1.9}$$

The saturation current is a function of device structure parameters

$$I_s = \frac{qAn_i^2 V_T \mu_B}{\int_0^{w_B} N_B(x)\mathrm{d}x} \tag{1.10}$$

where

$q = 1.6 \times 10^{-19}$ C is the electron charge
A is the EB junction area
n_i is the intrinsic concentration ($n_i = 1.5 \times 10^{10}$ at 300 K)
μ_B is the mobility of the majority carriers in the transistor base
w_B is the effective base thickness
$N_B(x)$ is the distribution of impurities in the base

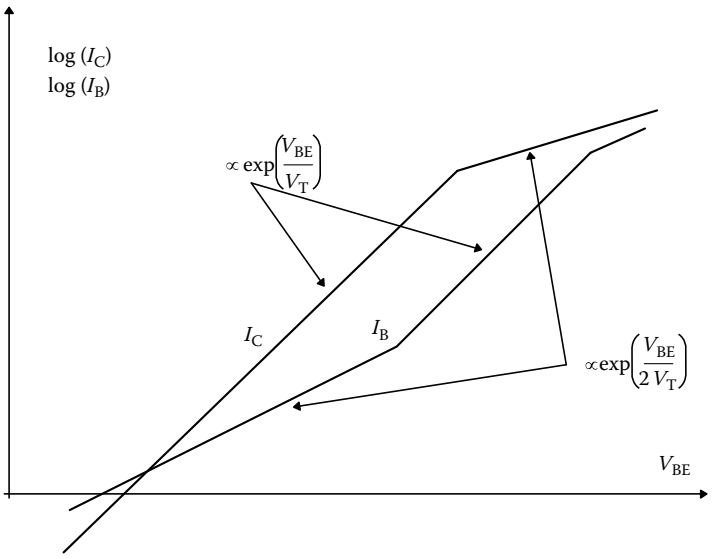


FIGURE 1.2 Collector and base currents as a function of base–emitter voltage.

Note, that the saturation current is inversely proportional to the total impurity dose in the base. In the transistor with the uniform base, the saturation current is given by

$$I_s = \frac{qAn_i^2 V_T \mu_B}{w_B N_B} \tag{1.11}$$

When a transistor operates in the reverse-active mode (emitter and collector are switched) then the current of such biased transistor is given by

$$I_E = I_s \exp \frac{V_{BC}}{V_T} \tag{1.12}$$

Note, that the I_s parameter is the same for forward and reverse mode of operation. The Gummel–Poon transistor model [2] was derived from the Ebers–Moll model using the assumption that $a_{12} = a_{21} = I_s$. For the Gummel–Poon model, Equations 1.3 are simplified to the form

$$\begin{aligned} I_E &= I_s \left(\frac{1}{\alpha_F} \exp \frac{V_{BE}}{V_T} - \exp \frac{V_{BC}}{V_T} \right) \\ I_C &= I_s \left(\exp \frac{V_{BE}}{V_T} - \frac{1}{\alpha_R} \exp \frac{V_{BC}}{V_T} \right) \end{aligned} \tag{1.13}$$

These equations require only three coefficients, while the Ebers–Moll requires four. The saturation current I_s is constant for a wide range of currents. The current gain coefficients α_F and α_R have values smaller, but close to unity. Often instead of using the current gain as $\alpha = I_C/I_E$, the current gain β as a ratio of the collector current to the base current $\beta = I_C/I_B$ is used. The mutual relationships between α and β coefficients are given by

$$\alpha_F = \frac{\beta_F}{\beta_F + 1}, \quad \beta_F = \frac{\alpha_F}{1 - \alpha_F}, \quad \alpha_R = \frac{\beta_R}{\beta_R + 1}, \quad \beta_R = \frac{\alpha_R}{1 - \alpha_R} \tag{1.14}$$

The Gummel–Poon model was implemented in Simulation Program with Integrated Circuit Emphasis (SPICE) [3] and other computer programs for circuit analysis. To make the equations more general, the material parameters η_F and η_R were introduced

$$I_C = I_s \left[\exp \frac{V_{BE}}{\eta_F V_T} - \left(1 + \frac{1}{\beta_R} \right) \exp \frac{V_{BC}}{\eta_R V_T} \right] \tag{1.15}$$

The values of η_F and η_R vary from 1 to 2.

1.1.3 Current Gains of Bipolar Transistors

The transistor current gain β is limited by two phenomena: base transport efficiency and emitter injection efficiency. The effective current gain β can be expressed as

$$\frac{1}{\beta} = \frac{1}{\beta_I} + \frac{1}{\beta_T} + \frac{1}{\beta_R} \tag{1.16}$$

where

β_I is the transistor current gain caused by emitter injection efficiency
β_T is the transistor current gain caused by base transport efficiency
β_R is the recombination component of the current gain

As one can see from Equation 1.16, smaller values of β_I, β_T, and β_R dominate. The base transport efficiency can be defined as a ratio of injected carriers into the base, to the carriers that recombine within the base. This ratio is also equal to the ratio of the minority carrier life time, to the transit time of carriers through the base. The carrier transit time can be approximated by an empirical relationship

$$\tau_{transit} = \frac{w_B^2}{V_T \mu_B (2 + 0.9\eta)}, \quad \eta = \ln\left(\frac{N_{BE}}{N_{BC}}\right) \tag{1.17}$$

where
- μ_B is the mobility of the minority carriers in base
- w_B is the base thickness
- N_{BE} is the impurity doping level at the emitter side of the base
- N_{BC} is the impurity doping level at the collector side of the base

Therefore, the current gain due to the transport efficiency is

$$\beta_T = \frac{\tau_{life}}{\tau_{transit}} = (2 + 0.9\eta)\left(\frac{L_B}{w_B}\right)^2 \tag{1.18}$$

where $L_B = \sqrt{V_T \mu_B \tau_{life}}$ is the diffusion length of minority carriers in the base.

The current gain β_I, due to the emitter injection efficiency, is given

$$\beta_I = \frac{\mu_B \int_0^{w_E} N_{Eeff}(x)dx}{\mu_E \int_0^{w_B} N_B(x)dx} \tag{1.19}$$

where
- μ_B and μ_E are minority carrier mobilities in the base and in the emitter
- $N_B(x)$ is impurity distribution in the base
- N_{Eeff} is the effective impurity distribution in the emitter

The recombination component of current gain β_R is caused by the different current–voltage relationship of base and collector currents as can be seen in Figure 1.2. The slower base current increase is due to the recombination phenomenon within the depletion layer of the base–emitter junction. Since the current gain is a ratio of the collector current to the base current, the relation for β_R can be found as

$$\beta_R = K_{R0} I_C^{1-(1/\eta_R)} \tag{1.20}$$

As it can be seen from Figure 1.2, the current gain β is a function of the current. This gain–current relationship is illustrated in Figure 1.3. The range of a constant current gain is wide for bipolar transistors with a technology characterized by a lower number of generation–recombination centers.

With an increase of CB voltage, the depletion layer penetrates deeper into the base. Therefore, the effective thickness of the base decreases. This leads to an increase of transistor current gain with applied collector voltages. Figure 1.4 illustrates this phenomenon, which is known as the Early's effect. The extensions of transistor characteristics (dotted lines in Figure 1.4) are crossing the voltage axis at

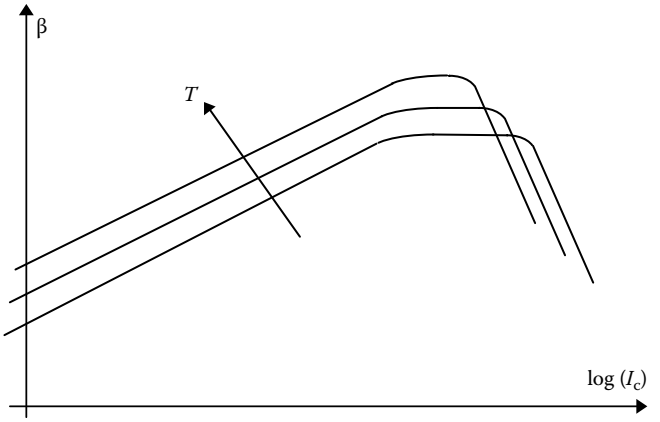


FIGURE 1.3 Current gain β as a function of collector current.

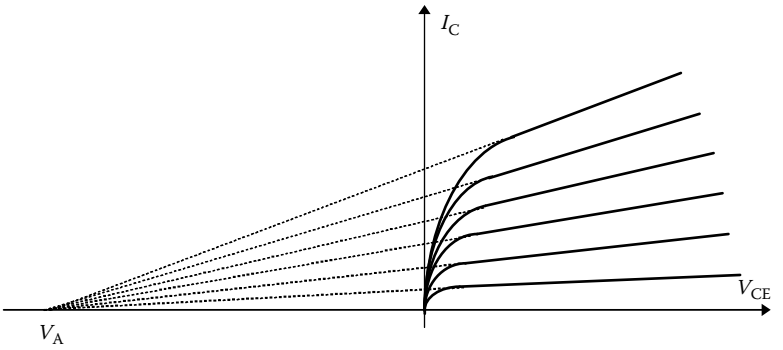


FIGURE 1.4 Current–voltage characteristics of a bipolar transistor.

the point $-V_A$, where V_A is known as the Early voltage. The current gain β, as a function of collector voltage, is usually expressed using the relation

$$\beta = \beta_o\left(1 + \frac{V_{CE}}{V_A}\right) \tag{1.21}$$

Similar equation can be defined for the reverse mode of operation.

1.1.4 High-Current Phenomena

The concentration of minority carriers increases with the rise of transistor currents. When the concentration of moving carriers exceeds a certain limit, the transistor property degenerates. Two phenomena are responsible for this limitation. The first is related to the high concentration of moving carriers (electrons in the NPN transistor) in the base–collector depletion region. This is known as the Kirk effect. The second phenomenon is caused by a high level of carriers injected into the base. When the concentration of injected minority carriers in the base exceeds the impurity concentration there, then the base conductivity modulation limits the transistor performance.

To understand the Kirk effect consider the NPN transistor in forward-active mode with the base–collector junction reversely biased. The depletion layer consists of the negative lattice charge of the base

region and the positive lattice charge of the collector region. Boundaries of the depletion layer are such that total the positive and negative charges are equal. When a collector current, carrying negatively charged electrons, flows through the junction, effective negative charge on the base side of junction increases. Also, the positive lattice charge of the collector side of the junction is compensated by negative charge of moving electrons. This way, the CB space charge region moves toward the collector, resulting in a thicker effective base. With a large current level, the thickness of the base may be doubled or tripled. This phenomenon, known as the Kirk effect, becomes very significant when the charge of moving electrons exceeds the charge of the lightly doped collector N_C. The threshold current for the Kirk effect is given by

$$I_{max} = qAv_{sat}N_C \tag{1.22}$$

where v_{sat} is the saturation velocity for electrons ($v_{sat} = 10^7$ cm/s for silicon).

The conductivity modulation in the base, or high-level injection, starts when the concentration of injected electrons into the base exceeds the lowest impurity concentration in the base N_{Bmin}. This occurs for the collector current I_{max} given by

$$I_{max} < qAN_{Bmax}, \quad \nu = \frac{qAV_T\mu_B N_{Bmax}(2 + 0.9\eta)}{w_B} \tag{1.23}$$

The above equation is derived using Equation 1.17 for the estimation of base transient time.

The high-current phenomena are significantly enlarged by the current crowding effect. The typical cross section of bipolar transistor is shown in Figure 1.5. The horizontal flow of the base current results in the voltage drop across the base region under the emitter. This small voltage difference on the base–emitter junction causes a significant difference in the current densities at the junction. This is due to the very nonlinear junction current–voltage characteristics. As a result, the base–emitter junction has very nonuniform current distribution across the junction. Most of the current flows through the part of the junction closest to base contact. For transistors with larger emitter areas, the current crowding effect is more significant. This nonuniform transistor current distribution makes the high-current phenomena, such as the base conductivity modulation and the Kirk effect, start for smaller currents than given by Equations 1.22 and 1.23. The current crowding effect is also responsible for the change of the effective base resistance with a current. As base current increases, the larger part of emitter current flows closer to the base contact, and the effective base resistance decreases.

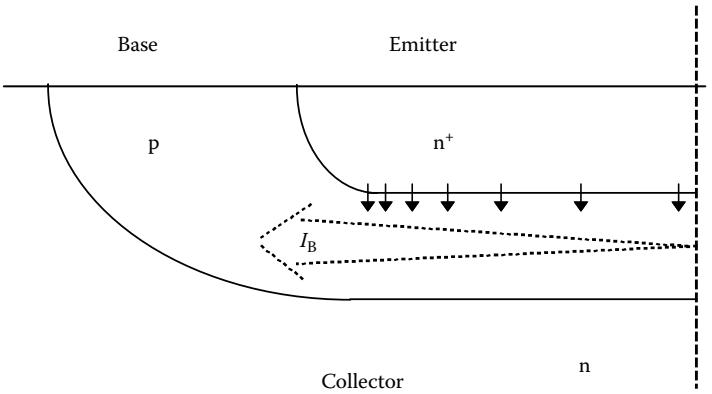


FIGURE 1.5 Current crowding effect.

1.1.5 Small-Signal Model

Small-signal transistor models are essential for AC circuit design. The small-signal equivalent circuit of the bipolar transistor is shown in Figure 1.6a. The lumped circuit shown in Figure 1.6a is only an approximation. In real transistors resistances and capacitances have a distributed character. For most design tasks, this lumped model is adequate, or even the simple equivalent transistor model shown in Figure 1.6b can be considered. The small-signal resistances, r_π and r_o, are inversely proportional to the transistor currents, and the transconductance g_m is directly proportional to the transistor currents

$$r_\pi = \frac{\eta_F V_T}{I_B} = \frac{\eta_F V_T \beta_F}{I_C}, \quad r_o = \frac{V_A}{I_C}, \quad g_m = \frac{I_C}{\eta_F V_T} \tag{1.24}$$

where

η_F is the forward emission coefficient, ranging form 1.0 to 2.0
V_T is the thermal potential ($V_T = 25$ mV at room temperature)

Similar equations to Equation 1.24 can be written for the reverse transistor operation as well.

The series base, emitter, and collector resistances R_B, R_E, and R_C are usually neglected for simple analysis (Figure 1.6b). However, for high-frequency analysis it is essential to use at least the base series resistance R_B. The series emitter resistance R_E usually has a constant, bias-independent value. The collector resistance R_C may significantly vary with the biasing current. The value of the series collector resistance may lower by one or two orders of magnitude if the collector junction becomes forward biased. A large series collector resistance may force the transistor into the saturation mode. Usually, when collector–emitter voltage is large enough, the effect of collector resistance is not significant. The SPICE model assumes constant value for the collector resistance R_C.

The series base resistance R_B may significantly limit the transistor performance at high frequencies. Due to the current crowding effect and the base conductivity modulation, the series base resistance is a function of the collector current I_C [4]

$$R_B = R_{Bmin} + \frac{R_{B0} - R_{Bmin}}{0.5 + \sqrt{0.25 + \frac{i_C}{I_{KF}}}} \tag{1.25}$$

where

I_{KF} is β_F high-current roll-off corner
R_{B0} is the base resistance at very small currents
R_{Bmin} is the minimum base resistance at high currents

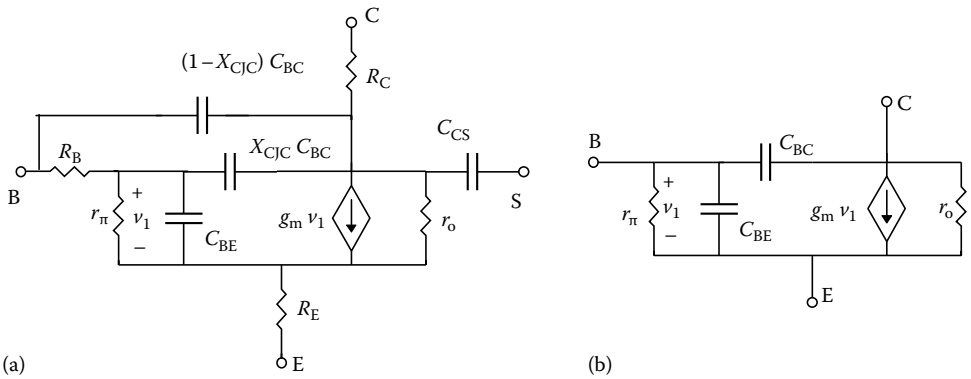


FIGURE 1.6 Bipolar transistor equivalent diagrams. (a) SPICE model and (b) simplified model.

Another possible approximation of the base series resistance R_B, as a function of the base current I_B, is [4]

$$R_B = 3(R_{B0} - R_{Bmin})\frac{\tan z - z}{z\tan^2 z} + R_{Bmin}, \quad z = \frac{\sqrt{1 + \frac{1.44 I_B}{\pi^2 I_{RB}}} - 1}{\frac{24}{\pi^2}\sqrt{\frac{I_B}{I_{RB}}}} \tag{1.26}$$

where I_{RB} is the base current for which the base resistance falls halfway to its minimum value.

The base–emitter capacitance C_{BE} is composed of two terms: the diffusion capacitance, which is proportional to the collector current, and the depletion capacitance, which is a function of the base–emitter voltage V_{BE}. The C_{BE} capacitance is given by

$$C_{BE} = \tau_F \frac{I_C}{\eta_F V_T} + C_{JE0}\left(1 - \frac{v_{BE}}{V_{JE0}}\right)^{-m_{JE}} \tag{1.27}$$

where

V_{JE0} is the base–emitter junction potential
τ_F is the base transit time for forward direction
C_{JE0} is the base–emitter zero-bias junction capacitance
m_{JE} is the base–emitter grading coefficient

The base–collector capacitance C_{BC} is given by a similar expression as Equation 1.27. In the case when the transistor operates in forward-active mode, it can be simplified to

$$C_{BC} = C_{JC0}\left(1 - \frac{v_{BC}}{V_{JC0}}\right)^{-m_{JC}} \tag{1.28}$$

where

V_{JC0} is the base–collector junction potential
C_{JC0} is the base–collector zero-bias junction capacitance
m_{JC} is the base–collector grading coefficient

In the case when the bipolar transistor is in the integrated form, the collector–substrate capacitance C_{CS} has to be considered

$$C_{CS} = C_{JS0}\left(1 - \frac{V_{CS}}{V_{JS0}}\right)^{-m_{JS}} \tag{1.29}$$

where

V_{JS0} is the collector–substrate junction potential
C_{JS0} the collector–substrate zero-bias junction capacitance
m_{JS} is the collector–substrate grading coefficient

When the transistor enters saturation, or it operates in the reverse-active mode, Equations 1.27 and 1.28 should be modified to

$$C_{BE} = \tau_F \frac{I_S \exp\left(\frac{V_{BE}}{\eta_F V_T}\right)}{\eta_F V_T} + C_{JE0}\left(1 - \frac{V_{BE}}{V_{JE0}}\right)^{-m_{JE}} \tag{1.30}$$

$$C_{BC} = \tau_R \frac{I_S \exp\left(\frac{V_{BC}}{\eta_R V_T}\right)}{\eta_R V_T} + C_{JC0}\left(1 - \frac{V_{BC}}{V_{JC0}}\right)^{-m_{JC}} \tag{1.31}$$

1.1.6 Technologies

The bipolar technology was used to fabricate the first ICs more than 40 years ago. A similar standard bipolar process is still used. In recent years, for high-performance circuits and for BiCMOS technology, the standard bipolar process was modified by using the thick selective silicon oxidation instead of the p-type isolation diffusion. Also, the diffusion process was substituted by the ion implantation process, low-temperature epitaxy, and Chemical Vapor Deposition (CVD).

1.1.6.1 Integrated NPN Bipolar Transistor

The structure of the typical integrated bipolar transistor is shown in Figure 1.7. The typical impurity profile of the bipolar transistor is shown in Figure 1.8. The emitter doping level is much higher than the base doping, so large current gains are possible (see Equation 1.19). The base is narrow and it has an impurity gradient, so the carrier transit time through the base is short (see Equation 1.17). Collector concentration near the base–collector junction is low, therefore, the transistor has a large breakdown voltage, large Early voltage V_{AF}, and CB depletion capacitance is low. High impurity concentration in the buried layer leads to a small collector series resistance. The emitter strips have to be as narrow as technology allows, reducing the base series resistance and the current crowding effect. If large emitter area is required, many narrow emitter strips interlaced with base contacts have to be used in a single

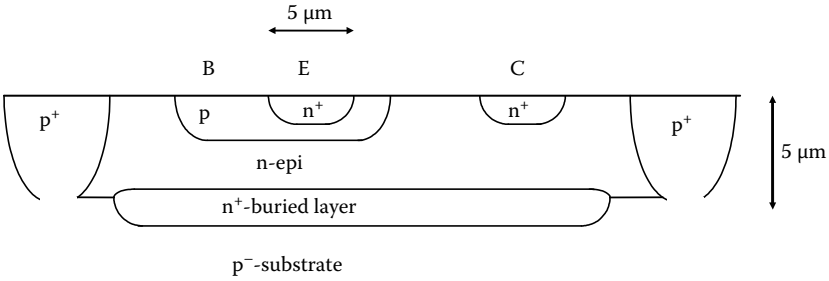


FIGURE 1.7 NPN bipolar structure.

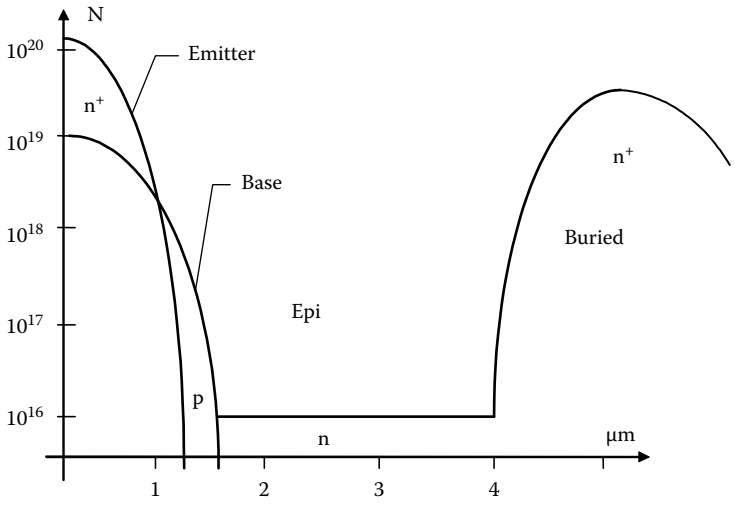


FIGURE 1.8 Cross section of a typical bipolar transistor.

transistor. Special attention has to be taken during the circuit design, so the base–collector junction is not forward biased. If the base–collector junction is forward biased, then the parasitic PNP transistors activate. This leads to undesired circuit operation. Thus, the integrated bipolar transistors must not operate in reverse or in saturation modes.

1.1.6.2 Lateral and Vertical PNP Transistors

The standard bipolar technology is oriented for fabrication of the NPN transistors with the structure shown in Figure 1.7. Using the same process, other circuit elements, such as resistors and PNP transistors, can be fabricated as well.

The lateral transistor, shown in Figure 1.9a uses the base p-type layer for both emitter and collector fabrication. The vertical transistor, shown in Figure 1.9b uses the p-type base layer for emitter, and the p-type substrate as collector. This transistor is sometimes known as the substrate transistor. In both transistors the base is made of the n-type epitaxial layer. Such transistors with a uniform and thick base are slow. Also, the current gain β of such transistors is small. Note, that the vertical transistor has the collector shorted to the substrate as Figure 1.10b illustrates. When a PNP transistor with a large current gain is required, then the concept of the composite transistor can be implemented. Such a composite transistor, known also as superbeta transistor, consists a PNP lateral transistor, and the standard NPN transistor connected as shown in Figure 1.10c. The composed transistor acts as the PNP transistor and it has a current gain β approximately equal to $\beta_{pnp}\beta_{npn}$.

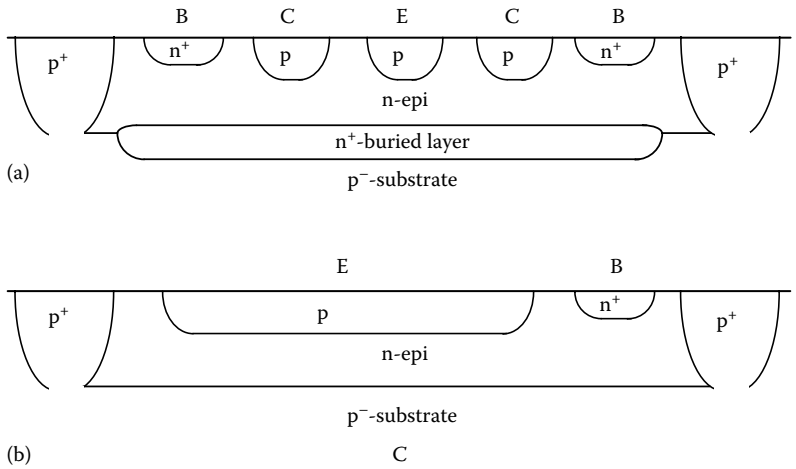


FIGURE 1.9 Integrated PNP transistors: (a) lateral PNP transistor, and (b) substrate PNP transistor.

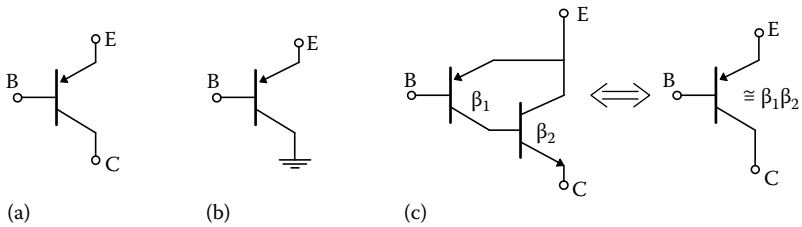


FIGURE 1.10 Integrated PNP transistors: (a) lateral transistor, (b) substrate transistor, and (c) composed transistor.

1.1.7 Model Parameters

It is essential to use proper transistor models in the computer aided design tools. The accuracy of simulation results depends on the model accuracy, and on the values of the model parameters used. In Section 1.1, the thermal and second-order effect in the transistor model are discussed. The SPICE bipolar transistor model parameters are discussed.

1.1.7.1 Thermal Sensitivity

All parameters of the transistor model are temperature dependent. Some parameters are very strong functions of temperature. To simplify the model description, the temperature dependence of some parameters are often neglected. In this chapter, the temperature dependence of the transistor model is described based on the model of the SPICE program [3–5]. Deviations from the actual temperature dependence will also be discussed. The temperature dependence of junction capacitance is given by

$$C_J(T) = C_J\left\{1 + m_J\left[4.010^{-4}(T - T_{NOM}) + 1 - \frac{V_J(T)}{V_J}\right]\right\} \tag{1.32}$$

where T_{NOM} is the nominal temperature, which is specified in the SPICE program in the .OPTIONS statement. The junction potential $V_J(T)$ is a function of temperature

$$V_J(T) = V_J\frac{T}{T_{NOM}} - 3V_T\ln\left(\frac{T}{T_{NOM}}\right) - E_G(T) + E_G\frac{T}{T_{NOM}} \tag{1.33}$$

The value of 3 in the multiplication coefficient of above equation is from the temperature dependence of the effective state densities in the valence and conduction bands. The temperature dependence of the energy gap is computed in the SPICE program from

$$E_G(T) = E_G - \frac{7.0210^{-4}T^2}{T + 1108} \tag{1.34}$$

The transistor saturation current as a function of temperature is calculated as

$$I_s(T) = I_s\left(\frac{T}{T_{NOM}}\right)^{X_{TI}}\exp\left[\frac{E_G(T - T_{NOM})}{V_T T_{NOM}}\right] \tag{1.35}$$

where E_G is the energy gap at the nominal temperature. The junction leakage currents I_{SE} and I_{SC} are calculated using

$$I_{SE}(T) = I_{SE}\left(\frac{T}{T_{NOM}}\right)^{X_{TI}-X_{TB}}\exp\left[\frac{E_G(T - T_{NOM})}{\eta_E V_T T_{NOM}}\right] \tag{1.36}$$

and

$$I_{SC}(T) = I_{SC}\left(\frac{T}{T_{NOM}}\right)^{X_{TI}-X_{TB}}\exp\left[\frac{E_G(T - T_{NOM})}{\eta_C V_T T_{NOM}}\right] \tag{1.37}$$

The temperature dependence of the transistor current gains β_F and β_R are modeled in the SPICE as

$$\beta_F(T) = \beta_F\left(\frac{T}{T_{NOM}}\right)^{X_{TB}}, \quad \beta_R(T) = \beta_R\left(\frac{T}{T_{NOM}}\right)^{X_{TB}} \tag{1.38}$$

The SPICE model does not give accurate results for the temperature relationship of the current gain β at high currents. For high current levels the current gain decreases sharply with the temperature, as can be seen from Figure 1.3. Also, the knee current parameters IKF, IKR, IKB are temperature-dependent, and this is not implemented in the SPICE program.

1.1.7.2 Second-Order Effects

The current gain β is sometimes modeled indirectly by using different equations for the collector and base currents [4,5]

$$I_C = \frac{I_S(T)}{Q_b}\left(\exp\frac{V_{BE}}{\eta_F V_T} - \exp\frac{V_{BC}}{\eta_R V_T}\right) - \frac{I_S(T)}{\beta_R(T)}\left(\exp\frac{V_{BC}}{\eta_R V_T} - 1\right) - I_{SC}(T)\left(\exp\frac{V_{BC}}{\eta_C V_T} - 1\right) \tag{1.39}$$

where

$$Q_b = \frac{1 + \sqrt{1 + 4Q_X}}{2\left(1 - \frac{V_{BC}}{V_{AF}} - \frac{V_{BE}}{V_{AR}}\right)} \tag{1.40}$$

$$Q_X = \frac{I_s(T)}{I_{KF}}\left(\exp\frac{V_{BE}}{\eta_F V_T} - 1\right) + \frac{I_s(T)}{I_{KR}}\left(\exp\frac{V_{BC}}{\eta_R V_T} - 1\right) \tag{1.41}$$

and

$$I_B = \frac{I_S}{\beta_F}\left(\exp\frac{V_{BE}}{\eta_F V_T} - 1\right) + I_{SE}\left(\exp\frac{V_{BE}}{\eta_E V_T} - 1\right) + \frac{I_S}{\beta_R}\left(\exp\frac{V_{BC}}{\eta_R V_T} - 1\right) + I_{SC}\left(\exp\frac{V_{BC}}{\eta_C V_T} - 1\right) \tag{1.42}$$

where

I_{SE} is the base–emitter junction leakage current
I_{SC} is the base–collector junction leakage current
η_E is the base–emitter junction leakage emission coefficient
η_C is the base–collector junction leakage emission coefficient

The forward transit time τ_F is a function of biasing conditions. In the SPICE program the τ_F parameter is computed using

$$\tau_F = \tau_{F0}\left[1 + X_{TF}\left(\frac{I_{CC}}{I_{CC} + I_{TF}}\right)^2 \exp\frac{V_{BC}}{1.44 V_{TF}}\right], \quad I_{CC} = I_s\left(\exp\frac{V_{BE}}{\eta_F V_T} - 1\right) \tag{1.43}$$

At high frequencies the phase of the collector current shifts. This phase shift is computed in the SPICE program following way

$$I_C(\omega) = I_C \exp(j\omega P_{TF}\tau_F) \tag{1.44}$$

where P_{TF} is a coefficient for excess phase calculation.

Noise is usually modeled as the thermal noise for parasitic series resistances, and as shot and flicker noise for collector and base currents

$$\overline{i_B^2} = \frac{4kT\Delta f}{R} \tag{1.45}$$

$$\overline{i_B^2} = \left(2qI_B + \frac{K_F I_B^{A_F}}{F}\right)\Delta f \tag{1.46}$$

$$\overline{i_C^2} = 2qI_C\Delta f \tag{1.47}$$

where K_F and A_F are the flicker-noise coefficient and flicker-noise exponent. More detailed information about noise modeling is given in Section 3.2.

1.1.7.3 SPICE Model of the Bipolar Transistor

The SPICE model of bipolar transistor uses similar or identical equations as described in this chapter [3–5]. Table 1.1 shows the parameters of the bipolar transistor model and its relation to the parameters used in this chapter.

TABLE 1.1 Parameters of SPICE Bipolar Transistor Model

Name Used	Equations	SPICE Name	Parameter Description	Unit	Typical Value	SPICE Default
I_s	1.10, 1.11	IS	Saturation current	A	10^{-15}	10^{-16}
I_{SE}	1.39	ISE	B–E leakage saturation current	A	10^{-12}	0
I_{SC}	1.39	ICS	B–C leakage saturation current	A	10^{-12}	0
β_F	1.14, 1.16, 1.21	BF	Forward current gain	—	100	100
β_R	1.14, 1.16, 1.21	BF	Reverse current gain	—	0.1	1
η_F	1.15, 1.24, 1.30, 1.31, 1.39 through 1.41	NF	Forward current emission coefficient	—	1.2	1.0
η_R	1.15, 1.24, 1.30, 1.31, 1.39 through 1.42	NR	Reverse current emission coefficient	—	1.3	1.0
η_E	1.39	NE	B–E leakage emission coefficient	—	1.4	1.5
η_C	1.39	NC	B–C leakage emission coefficient	—	1.4	1.5
V_{AF}	1.21, 1.40	VAF	Forward Early voltage	V	200	∞
V_{AR}	1.21, 1.40	VAR	Reverse Early voltage	V	50	∞
I_{KF}	1.22, 1.23, 1.40	IKF	β_F high-current roll-off corner	A	0.05	∞
I_{KR}	1.22, 1.23, 1.40	IKR	β_R high-current roll-off corner	A	0.01	∞
I_{RB}	1.26	IRB	Current where base resistance falls by half	A	0.1	∞
R_B	1.25, 1.26	RB	Zero base resistance	Ω	100	0
R_{Bmin}	1.25, 1.26	RBM	Minimum base resistance	Ω	10	RB
R_E	Figure 1.6	RE	Emitter series resistance	Ω	1	0
R_C	Figure 1.6	RC	Collector series resistance	Ω	50	0
C_{JE0}	1.27	CJE	B–E zero-bias depletion capacitance	F	10^{-12}	0
C_{JC0}	1.28	CJC	B–C zero-bias depletion capacitance	F	10^{-12}	0
C_{JS0}	1.29	CJS	Zero-bias collector–substrate capacitance	F	10^{-12}	0
V_{JE0}	1.27	VJE	B–E built-in potential	V	0.8	0.75
V_{JC0}	1.28	VJC	B–C built-in potential	V	0.7	0.75
V_{JS0}	1.29	VJS	Substrate junction built-in potential	V	0.7	0.75
m_{JE}	1.27	MJE	B–E junction exponential factor	—	0.33	0.33
m_{JC}	1.28	MJC	B–C junction exponential factor	—	0.5	0.33

(*continued*)

TABLE 1.1 (continued) Parameters of SPICE Bipolar Transistor Model

Name Used	Equations	SPICE Name	Parameter Description	Unit	Typical Value	SPICE Default
m_{JS}	1.29	MJS	Substrate junction exponential factor	—	0.5	0
X_{CJC}	Figure 1.6	XCJC	Fraction of B–C capacitance connected to internal base node (see Figure 1.6)	—	0.5	0
τ_F	1.17, 1.28, 1.30, 1.42	TF	Ideal forward transit time	s	10^{-10}	0
τ_R	1.31	TR	Reverse transit time	s	10^{-8}	0
X_{TF}	1.43	XTF	Coefficient for bias dependence of τ_F	—		0
V_{TF}	1.43	VTF	Voltage for τ_F dependence on V_{BC}	V		∞
I_{TF}	1.43	ITF	Current where $\tau_F = f(I_C, V_{BC})$ starts	A		0
P_{TF}	1.44	PTF	Excess phase at freq $= 1/(2\pi\tau_F)$ Hz	°		0
X_{TB}	1.38	XTB	Forward and reverse beta temperature exponent			0
E_G	1.34	EG	Energy gap	eV	1.1	1.11
X_{TI}	1.35 through 1.37	XTI	Temperature exponent for effect on I_s	—	3.5	3
K_F	1.46	KF	Flicker-noise coefficient	—		0
A_F	1.46	AF	Flicker-noise exponent	—		1
F_C		FC	Coefficient for the forward biased depletion capacitance formula	—	0.5	0.5
T_{NOM}	1.32 through 1.38	TNOM	Nominal temperature specified in .OPTION statement	K	300	300

The SPICE [3] was developed mainly for analysis of ICs. During the analysis it is assumed that the temperatures of all circuit elements are the same. This is not true for power ICs where the junction temperatures may differ by 30 K or more. This is obviously not true for circuits composed of the discrete elements where the junction temperatures may differ by 100 K and more. These temperature effects, which can significantly affect the analysis results, are not implemented in the SPICE program.

Although the SPICE bipolar transistor model uses more than 40 parameters, many features of the bipolar transistor are not included in the model. For example, the reverse junction characteristics are described by Equation 1.32. This model does not give accurate results. In the real silicon junction the leakage current is proportional to the thickness of the depletion layer, which is proportional to $V^{1/m}$. Also the SPICE model of the bipolar transistor assumes that there is no junction breakdown voltage. A more accurate model of the reverse junction characteristics is described in Section 11.5 of *Fundamentals of Circuits and Filters*. The reverse transit time τ_R is very important to model the switching property of the lumped bipolar transistor, and it is a strong function of the biasing condition and temperature. Both phenomena are not implemented in the SPICE model.

1.1.8 SiGe HBTs

The performance of the Si bipolar transistor can be greatly enhanced with proper engineering of the base bandgap profile using a narrower bandgap material, SiGe, an alloy of Si and Ge. Structure wise, a SiGe

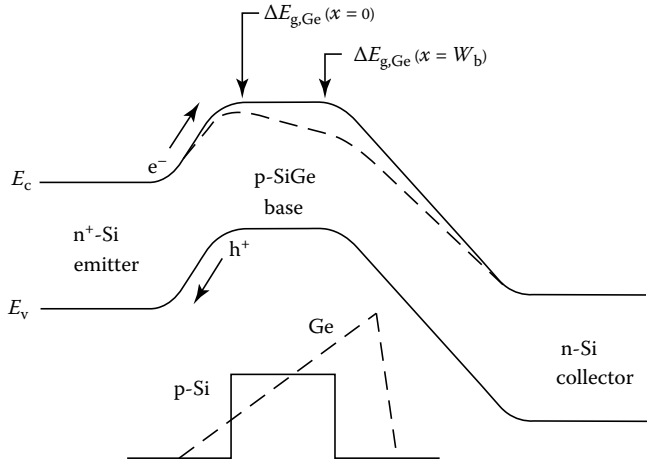


FIGURE 1.11 Energy band diagram of a graded base SiGe HBT and a comparably constructed Si BJT.

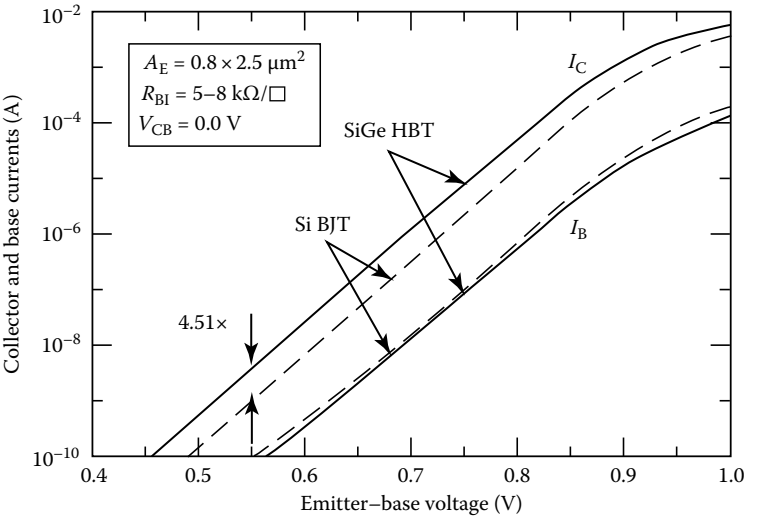


FIGURE 1.12 Experimental collector and base currents versus EB voltage for SiGe HBT and Si BJT.

HBT is essentially a Si BJT with a SiGe base. Its operation and circuit level performance advantages can be illustrated with the energy band diagram in Figure 1.11 [13]. Here the Ge content is linearly graded from emitter toward collector to create a large accelerating electric field that speeds up minority carrier transport across the base, thus making transistor speed much faster and cutoff frequency much higher. Everything else being the same, the potential barrier for electron injection into the base is reduced, thus exponentially enhancing the collector current. The base current is the same for SiGe HBT and Si BJT, as the emitter is typically made the same. Beta is thus higher in SiGe HBT. Figure 1.12 confirms these expectations experimentally with data from a typical first-generation SiGe HBT technology. The measured doping and Ge profiles are shown in Figure 1.13. The metallurgical base width is only 90 nm, and the neutral base width is around 50 nm. Figure 1.14 shows experimental cutoff frequency f_T improvement from using a graded SiGe base, which also directly translates into maximum oscillation frequency f_{max} improvement.

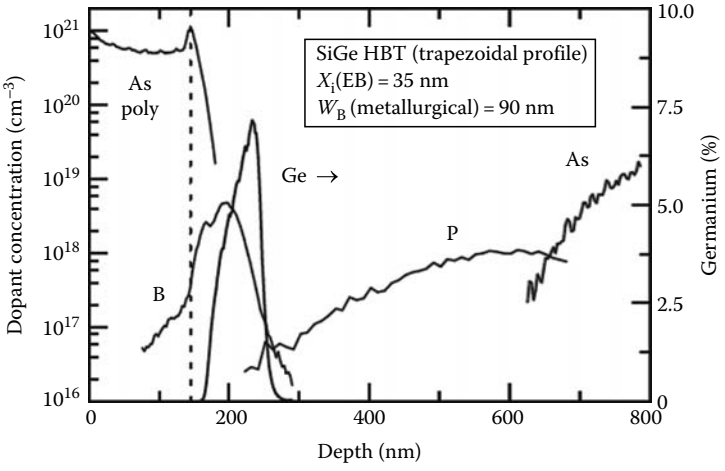


FIGURE 1.13 Measured doping and Ge profiles of a modern SiGe HBT.

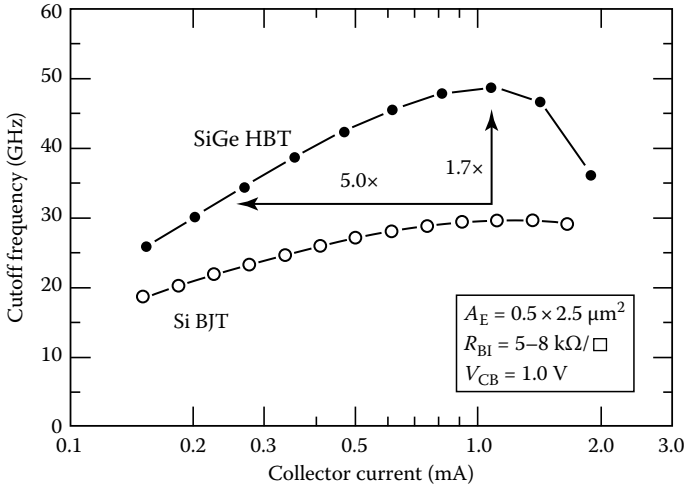


FIGURE 1.14 Experimental cutoff frequency versus collector current for SiGe HBT and Si BJT.

1.1.8.1 Operation Principle and Performance Advantages over Si BJT

In modern transistors, particularly with the use of polysilicon emitter, beta may be sufficient. If so, the higher beta potential of SiGe HBT can then be traded for reduced base resistance, through the use of higher base doping. The unique ability of simultaneously achieving high beta, low base resistance, and high cutoff frequency makes SiGe HBT attractive for many RF circuits. Broadband noise is naturally reduced, as low base resistance reduces transistor input noise voltage, and high beta as well as high f_T reduces transistor input noise current [13]. Experimentally, $1/f$ noise at the same base current was found to be approximately the same for SiGe HBT and Si BJT [14]. Consequently, $1/f$ noise is often naturally reduced in SiGe HBT circuits for the same biasing collector current, as base current is often smaller due to higher beta, as shown in Figure 1.15 using corner frequency as a figure-of-merit.

These, together with circuit-level optimization, can lead to excellent low-phase noise oscillators and frequency synthesizers suitable for both wireless and wire line communication circuits. Another less

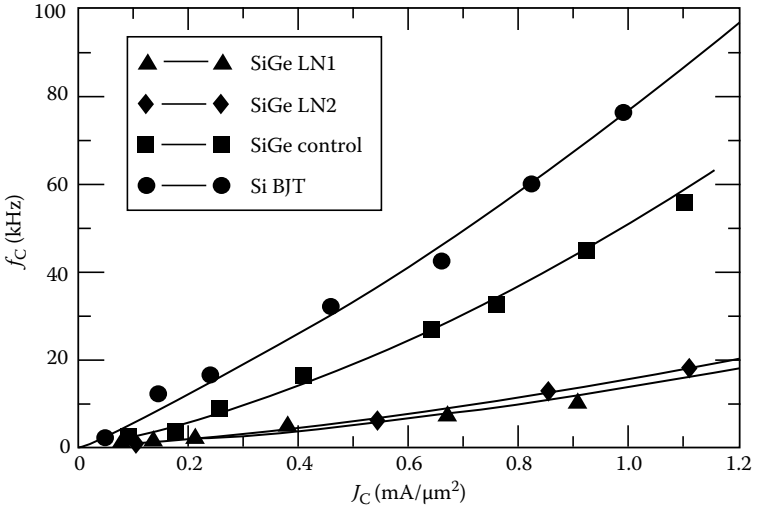


FIGURE 1.15 Experimentally measured corner frequency as a function of collector current density for three SiGe HBTs with different base SiGe designs, and a comparatively constructed Si BJT.

obvious advantage from grading Ge is the collector side of the neutral base has less impact on the collector current than the emitter side of the neutral base. Consequently, as collector voltage varies and the collector side of the neutral base is shifted toward the emitter due to increased CB junction depletion layer thickness, the collector current is increased to a much lesser extent than in a comparably constructed Si BJT, leading to a much higher output impedance or Early voltage. The $\beta \times V_A$ product is thus much higher in SiGe HBT than in Si BJT.

1.1.8.2 Industry Practice and Fabrication Technology

The standard industry practice today is to integrate SiGe HBT with CMOS, to form a SiGe BiCMOS technology. The ability to integrate with CMOS is also a significant advantage of SiGe HBT over III–V HBT. Modern SiGe BiCMOS combines the analog and RF performance advantages of the SiGe HBT, and the lower power logic, high integration level, and memory density of Si CMOS, into a single cost-effective system-on-chip (SoC) solution. Typically, SiGe HBTs with multiple breakdown voltages are offered through selective collector implantation, to provide more flexibility in circuit design.

The fabrication process of SiGe HBT and its integration with CMOS has been constantly evolving in the past two decades, and varies from company to company. Below are some common fabrication elements and modules shared by many if not all commercial first-generation (also most wide spread in manufacturing at present) SiGe technologies:

1. A starting N^+ subcollector around 5 Ω/sq on a p-type substrate at $5 \times 10^{15}/cm^3$, typically patterned to allow CMOS integration.
2. A high-temperature, lightly doped n-type collector, around 0.4–0.6 μm thick at $5 \times 10^{15}/cm^3$.
3. Polysilicon-filled deep trenches for isolation from adjacent devices, typically 1 μm wide and 7–10 μm deep.
4. Oxide filled shallow trenches or LOCOS for local device isolation, typically 0.3–0.6 μm deep.
5. An implanted collector reach through to the subcollector, typically at 10–20 Ωμm^2.
6. A composite SiGe epi layer consisting of a 10–20 nm Si buffer, a 70–100 nm boron-doped SiGe active layer, with or without C doping to help suppress boron out diffusion, and a 10–30 nm Si cap. The integrated boron dose is typically $1–3 \times 10^{13}/cm^2$.

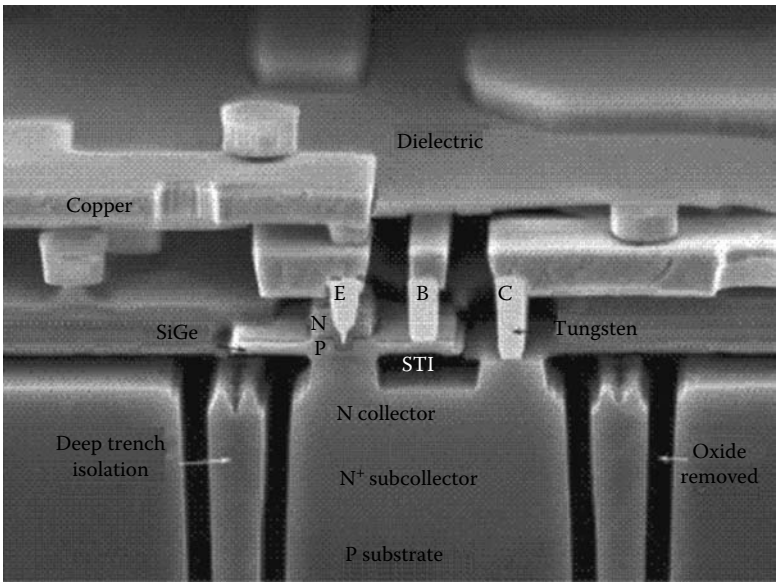


FIGURE 1.16 Structure of a modern SiGe HBT.

7. A variety of EB self-alignment scheme, depending on device structure and SiGe growth approach. All of them utilize some sort of spacer that is 100–300 nm wide.
8. Multiple self-aligned collector implantation to allow multiple breakdown voltages on the same chip.
9. Polysilicon extrinsic base, usually formed during SiGe growth over shallow trench oxide, and additional self-aligned extrinsic implantation to lower base resistance.
10. A silicided extrinsic base.
11. A 100–200 nm thick heavily doped ($>5 \times 10^{20}/\text{cm}^3$) polysilicon emitter, either implanted or in situ doped.
12. A variety of multiple level back-end-of-line metallization schemes using Al or Cu, typically borrowed from parent CMOS process.

These technological elements can also be seen in the electronic image of a second-generation SiGe HBT shown in Figure 1.16.

References

1. J. J. Ebers and J. M. Moll, Large signal behavior of bipolar transistors. *Proceedings IRE* 42, 1761–1772, December 1954.
2. H. K. Gummel and H. C. Poon, An integral charge-control model of bipolar transistors. *Bell System Technical Journal* 49, 827–852, May 1970.
3. L. W. Nagel and D. O. Pederson, SPICE (Simulation Program with Integrated Circuit Emphasis). University of California, Berkeley, ERL Memo No. ERL M382, April 1973.
4. P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*, McGraw-Hill, New York, 1988.
5. A. Vadimiresku, *The SPICE Book*, John Wiley & Sons, Hoboken, NJ, 1994.
6. A. S. Grove, *Physics and Technology of Semiconductor Devices*, John Wiley & Sons, Hoboken, NJ, 1967.
7. S. M. Sze, *Physics of Semiconductor Devices*, 2nd ed., John Wiley & Sons, Hoboken, NJ, 1981.
8. G. W. Neudeck, *The PN Junction Diode*, Vol II, Modular Series on Solid-State Devices, Addison-Wesley, Upper Saddle River, NJ, 1983.

9. R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, 2nd ed., John Wiley & Sons, Hoboken, NJ, 1986.
10. E. S. Yang, *Microelectronic Devices*, McGraw-Hill, New York, 1988.
11. B. G. Streetman, *Solid State Electronic Devices*. 3rd ed., Prentice Hall, Upper Saddle River, NJ, 1990.
12. D. A. Neamen, *Semiconductor Physics and Devices*, Irwin, 1992.
13. J. D. Cressler and G. Niu, *Silicon–Germanium Heterojunction Bipolar Transistor*, Artech House, Norwood, MA, 2003.
14. G. Niu, Noise in SiGe HBT RF technology: Physics, modeling and circuit implications, *Proceedings of the IEEE*, pp. 1583–1597, September 2005.

1.2 Metal–Oxide–Silicon Field Effect Transistor

John Choma, Jr.

1.2.1 Introduction

Integrated electronic circuits realized in metal–oxide–silicon field effect transistor (MOSFET) technology are ubiquitous in both the commercial and military sectors of the technical community. To be sure, transistors manufactured in certain bipolar and III–V compound transistor technologies compete successfully with their MOSFET counterparts from such performance perspectives as switching speed, wideband frequency response, and insensitivity to electromagnetic interference and irradiated environments. Nevertheless, the MOSFET reigns supreme in the extant state of the electronics art for several reasons. The first of these reasons derives from the fact that the cross-section geometry of a MOSFET, when compared to that of most other solid-state transistors, is simpler. This simplicity affords a relative ease of foundry processing, which in turn promotes high device yield and therefore, cost-effective manufacturing. A second reason is that the surface area consumed on chip, or *footprint*, of a MOSFET is generally smaller than that of a comparably performing bipolar or III–V compound transistors. This feature allows increased packing density, which is particularly advantageous for digital signal processors that commonly require upwards of millions of transistors for system functionality. Third, MOSFETs can deliver acceptable circuit performance at low standby power levels, which is a laudable attribute in light of the aforementioned high device density digital architectures and the portability culture in which society is immersed presently. Finally, the native insulating oxide indigenous to the monolithic processing of silicon semiconductors renders MOSFET technologies amenable to the implementation of complex electronic systems on a single chip. No such native oxide prevails in III–V compound technologies, thereby rendering awkward the electrical isolation among the various components, subsystems, and subcircuits that comprise the overall electronic system.

The penchant toward adopting MOSFET technology for analog signal processing applications can also be rationalized. In particular, the nature of modern integrated systems is rarely exclusively digital or exclusively analog. Such systems are, in fact, ''mixed signal'' architectures that embody both digital and analog signal processing on the same chip. Because of the simplicity, packing density, and power dissipation attributes of MOSFETs, virtually 100% of digital architectures are realized in MOSFET technology. Prudence alone accordingly dictates a MOSFET technology realization of the analog cells implicit to a mixed signal framework if only to facilitate the electrical interface between the analog and digital units.

Aside from the operating flexibility and programmability advantages boasted by digital circuit schema, digital circuits in mixed signal architectures are often required to assure and sustain performance optimality of the analog signal flow paths in an electronic system. Unlike most digital networks, high-performance analog circuits are sensitive to specific values, or at least specific ranges of values, of several of the key physical and electrical parameters that effectively define the electrical properties of MOSFETs. Unfortunately, attaining the requisite accuracy in the numerical delineation of these parameters becomes

progressively more daunting as the performance metrics imposed on an analog network become more challenging and as device geometries scale to meet omnipresent quests for wider signal processing passbands. In these high-performance systems, digital subsystems are often deployed to sense the observable performance metrics of an analog signal flow path, compare said metrics to their respective optimal design goals, and then appropriately adjust the relevant electrical parameters or signal excitations implicit to the signal path. In effect, the combined digital controller and analog network behave as a seamless adaptive system that automatically corrects for manufacturing vagaries, increased device operating temperatures, and certain environmental effects.

The most commonly utilized MOSFETs in modern electronic systems come in two flavors: the N-channel MOSFET (NMOS), diagrammed in Figure 1.17 and the P-channel MOSFET (PMOS) shown in Figure 1.18. In the NMOS device of Figure 1.17, the bulk substrate is P-type and is doped to an average acceptor impurity concentration of N_A, for which a representative range of values is $5(10^{14})$ atoms/cm^3 $< N_A <$ 10^{16} atoms/cm^3. Its vertical depth, which is not expressly highlighted in the figure, is many times larger than the depth, Y_d, (of the order of a few tenths of microns) of either the source or drain diffusions or implants. These regions, whose widths are indicated as L_{diff} and which are connected electrically to the source (S) and drain (D) terminals of the MOSFET, are very strongly doped in that their donor impurity concentrations are $N_D = 10^{20}$ atoms/cm^3 or larger. The width, L_{diff}, is typically two- or three-times the channel length, indicated as L in the diagram. The metallization contact that forms the electrical terminal of the semiconductor bulk (B) is generally connected to the most negative potential available in the circuit into which the subject transistor is embedded. Such a connection reverse biases the PN junctions formed between the bulk and source regions and between the bulk and drain regions. This reverse biasing ensures that for at least low signal frequencies, the source and drain regions are electrically isolated from each other and from the bulk substrate. In certain types of multiwell IC processes, bulk–source and bulk–drain reverse biasing is assured simply by returning the bulk terminal directly to the source region contact.

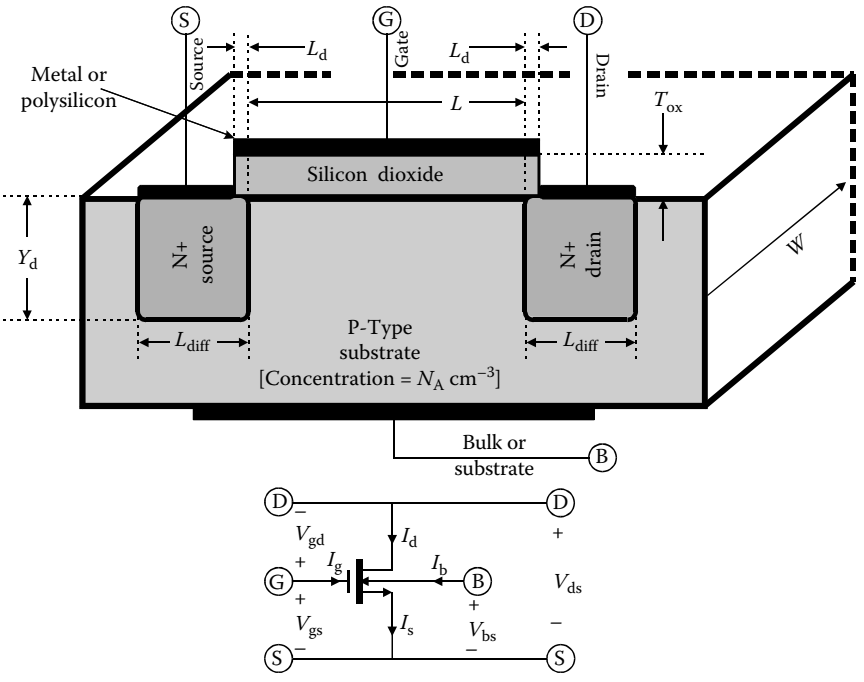


FIGURE 1.17 A simplified three-dimensional depiction of an N-channel MOSFET (NMOS) and its corresponding electrical schematic symbol. The diagram is not drawn to scale.

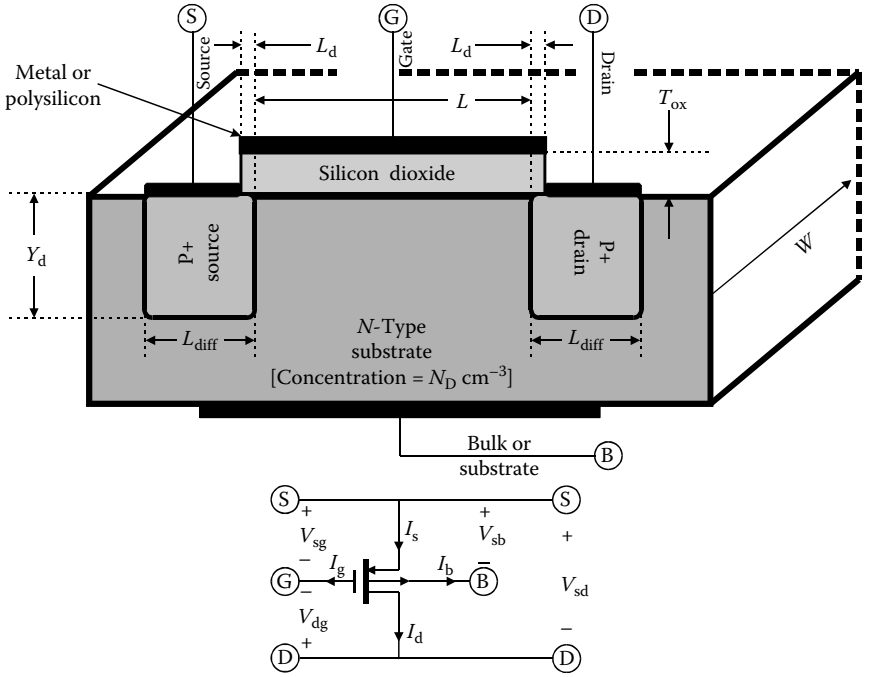


FIGURE 1.18 A simplified three-dimensional depiction of an P-channel MOSFET (PMOS) and its corresponding electrical schematic symbol. The diagram is not drawn to scale.

Lying atop the P-type bulk substrate is an insulating silicon dioxide layer of thickness T_{ox} that extends into the page as shown by a gate width, W. The oxide thickness in the extant state of the art is of the order of several tens of angstroms, where 1 Å is 10^{-8} cm. This oxide layer entirely covers the channel length, L, that separates the source region from the drain region, and it may overlap the source and drain regions by the amount, L_d, indicated in the diagram. The overlap of the source and drain regions is undesirable in that it limits broadband frequency responses in certain types of MOSFET amplifiers. In processes boasting self-aligned gate capabilities, L_d is ideally reduced to zero. But for state of the art processes delivering channel lengths as small as 65–130 nM, gate self-alignment focused on reducing L_d to no more than 5% of L is a challenging undertaking. The gate width, W, can be no smaller than the minimum channel length that can be produced by the identified foundry process. Subject to this proviso, the gate aspect ratio, W/L, is a designable parameter selected in accordance with the operating requirements of the circuit application for which the considered MOSFET is utilized.

The gate terminal (G) is formed by a contact made of a metallic or a polycrystalline silicon layer deposited directly atop the gate oxide. The gate metal of choice is aluminum. If the MOSFET under consideration is used in high-temperature environments and/or in applications that exploit low power supply voltages, polycrystalline silicon, which is commonly referred to as polysilicon, supplants the aluminum gate.

In addition to the simplified cross-section diagram of the N-channel MOSFET, Figure 1.17 inserts the electrical schematic symbol of the NMOS transistor. Of particular interest are the positive reference conventions adopted for four device currents and four device voltages. Specifically, positive drain current, I_d, flows into the transistor, as do the gate current, I_g, and the bulk, or substrate, current, I_b, while positive source current, I_s, flows out of the transistor. It follows from Kirchhoff's current law that

$$I_s = I_d + I_g + I_b. \tag{1.48}$$

However, since the gate contact is isolated from the semiconductor bulk by an insulating oxide layer, I_g is zero at the low frequencies for which capacitive phenomena associated with the insulating gate dielectric are insignificant. Moreover, the bulk current, I_b, is likewise almost zero at low signal frequencies, provided, as is usually the case, that care is taken to ensure reverse biasing of the bulk–drain and bulk–source PN junctions. Accordingly, the source and drain currents, I_s and I_d, respectively, are essentially identical when the frequencies of signals applied to the MOSFET are low. The pages that follow demonstrate that the static and low-frequency value of the drain, and hence the source, current is controlled by the gate-to-source voltage, V_{gs}, the drain-to-source voltage, V_{ds}, and, to a somewhat lesser extent, the bulk-to-source voltage V_{bs}. Stipulating an additional dependence of drain current on gate-to-drain voltage V_{gd} is superfluous, for by Kirchhoff's voltage law,

$$V_{ds} = V_{gs} - V_{gd}. \tag{1.49}$$

The P-channel MOSFET abstracted in Figure 1.18 is architecturally identical to its N-channel counterpart. The notable differences are that the bulk substrate in PMOS is N-type and the source and drain regions are heavily doped with P-type impurities. It follows that electrical isolation between the source region and the bulk, as well as between the drain region and the bulk, requires that the bulk substrate terminal of a PMOS device be connected either to the most positive of available circuit potentials or, if the process allows, to the source terminal. All of the geometrical parameters and their representative values remain the same as stipulated in conjunction with the NMOS unit. The PMOS electrical schematic symbol, which is also shown in the figure at hand, differs from the NMOS symbol in that the directions of the source terminal and bulk terminal arrows are reversed, as are the positive reference directions of all four transistor currents. While Equation 1.48 remains applicable, the analytical expression for the drain current, I_d, which now flows out of the transistor, is more conveniently couched in terms of the source-to-gate voltage, V_{sg}, the source-to-drain voltage, V_{sd}, and the source-to-bulk voltage, V_{sb}. The drain-to-gate voltage, V_{dg}, derives from

$$V_{sd} = V_{sg} - V_{dg}, \tag{1.50}$$

which mirrors Equation 1.49 subsequent to multiplying both sides of this equation by -1.

1.2.2 Channel Charge

A fundamental understanding of the physical charge storage and charge transport mechanisms that underpin the observable volt–ampere characteristics of considered transistors facilitates the reliable and reproducible design of high-performance analog networks in MOSFET technology. Aside from establishing a foundation upon which the static characteristic curves of a MOSFET can be constructed in a physically sound framework, these charge profiles also serve to define the voltage-dependent nature of the capacitance characteristics of a MOSFET. In effect, the subject charge profiles posture the MOSFET as a plausible varactor, which is useful in the monolithic design of voltage controlled oscillators, active filters, and other electronic networks.

The profile of charge stored in the channel between the source and drain regions of a MOSFET is best examined in terms of the simple circuit given in Figure 1.19a. In this circuit, the drain terminal is short circuited to the source to pin the drain–source voltage, V_{ds}, to zero. A zero bias is applied as indicated between the bulk and source, thereby establishing a charge depletion region about the PN junction formed between the substrate and source regions. Since the source and the drain are electrically connected to one another, the zero bias applied between bulk and source establishes an identical depletion zone about the bulk–drain PN junction. These depletion layers are delineated in the companion cross-section diagram of Figure 1.19b, as are the surface potential, φ_o, and the potential, V_{ox}, dropped across the gate silicon dioxide layer. With $V_{ds} = 0$, Equation 1.49 ensures a gate–source voltage, V_{gs}, that

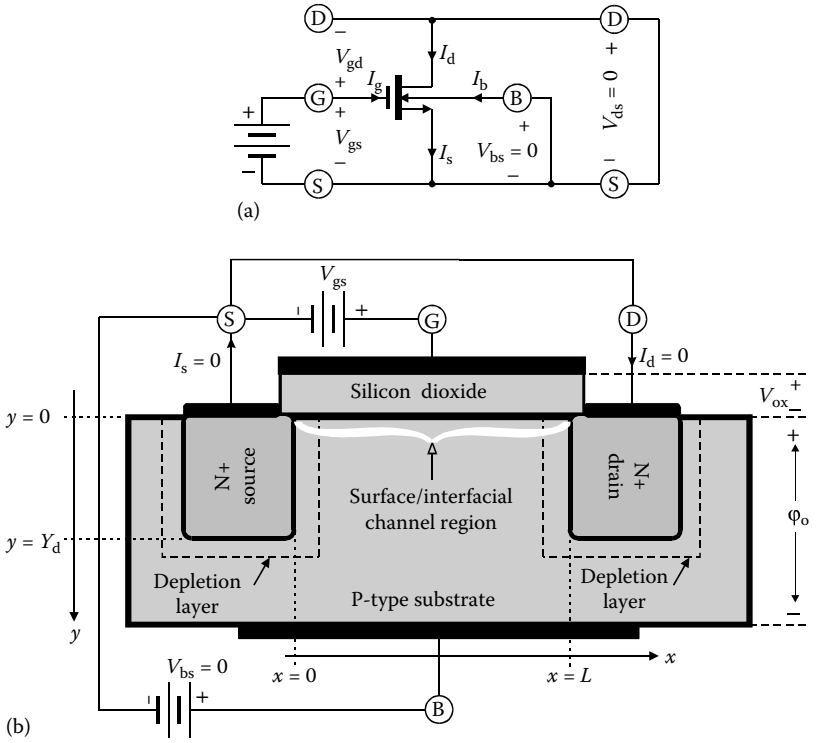


FIGURE 1.19 (a) NMOS transistor operated with $V_{ds} = 0$ and $V_{bs} = 0$. Although the battery connected between the gate and the source ensures $V_{gs} > 0$, $V_{gs} \leq 0$ is allowed in the discussion that references this circuit. (b) Cross-section diagram corresponding to the circuit in (a). Note that all applied voltages are referred to the source terminal. The diagram in (b) is not drawn to scale.

mirrors the gate–drain voltage, V_{gd}, regardless of the voltage applied between gate and source or gate and bulk terminals. In the absence of drain, source, bulk, and gate currents, $V_{ds} = 0$ also guarantees that surface potential φ_o, measured from the oxide semiconductor interface-to-the neutral zone of the bulk substrate, is the same throughout the channel region extending from $x = 0$-to-$x = L$ in the subject diagram. The aforementioned voltage, V_{ox}, includes the effects of parasitic trapped charge in the gate oxide, but it does not include the ramifications of work function differences that unavoidably prevail between the gate contact and the oxide and at the oxide–semiconductor interface. Note then that the voltage, V_{gb}, measured at the gate terminal with respect to the bulk terminal is, ignoring work function phenomena, simply

$$V_{gb} = V_{ox} + \varphi_o. \tag{1.51}$$

1.2.2.1 Surface Charge Density

A pivotally important analytical tool serving to define the charge, capacitance, and static volt–ampere characteristics of a MOSFET, is the charge density, $Q_o(\varphi_o)$, in units of coulombs per unit area, established at the semiconductor surface as a function of the surface potential, φ_o. Several authors have identified this charge profile as [1–3]

$$Q_o(\varphi_o) = -\text{sgn}(\varphi_o)\left(\frac{\sqrt{2}\varepsilon_s V_T}{D_b}\right)\sqrt{G(-\varphi_o) + G(\varphi_o)e^{-2V_F/V_T}}, \tag{1.52}$$

where $\varepsilon_s = 1.037$ pF/cm denotes the dielectric constant of silicon, and

$$\text{sgn}(\varphi_o) = \begin{cases} +1 & \text{for } \varphi_o > 0 \\ -1 & \text{for } \varphi_o < 0 \end{cases}. \tag{1.53}$$

In Equation 1.52,

$$V_T = kT/q \tag{1.54}$$

is the familiar semiconductor thermal voltage for which $k = (1.38)(10^{-23})$ J/K is Boltzmann's constant, $q = (1.60)(10^{-19})$ C is the magnitude of electron charge, and T is the absolute temperature of the semiconductor surface. The voltage, V_F, in the radical on the right-hand side of Equation 1.52 is the Fermi potential, which is given by

$$V_F \triangleq V_T \ln\left(\frac{N_A}{N_i}\right), \tag{1.55}$$

where N_A is the previously defined average acceptor impurity concentration of the bulk substrate in NMOS and $N_i = (1.45)(10^{10})$ atoms/cm^3 is the intrinsic carrier concentration of silicon at $T = 27°$C. The parameter, D_b, is known as the electron Debye length and is given by

$$D_b \triangleq \sqrt{\frac{\varepsilon_s V_T}{qN_A}}. \tag{1.56}$$

Finally, the function, $G(\varphi_o)$, in Equation 1.52 is

$$G(\varphi_o) = e^{\varphi_o/V_T} - 1 - \frac{\varphi_o}{V_T}, \tag{1.57}$$

where it is understood that the surface potential, φ_o, measured with respect to the charge neutral zone in the bulk in Figure 1.19 is established in response to an applied gate–bulk voltage, V_{gb}, or an applied gate–source voltage, V_{gs}. Observe that $G(\varphi_o) = G(-\varphi_o) = 0$ for $\varphi_o = 0$, which delivers the expected result in Equation 1.52 of $Q_o(0) = 0$. It should be understood that Equation 1.52 is premised on Poisson's equation and the Boltzmann carrier relationship,

$$p(0) = N_A e^{-\varphi_o/V_T}, \tag{1.58}$$

where $p(0)$ signifies the hole concentration at the surface if complete ionization of substrate dopant atoms is tacitly presumed. Since

$$p(0)n(0) = N_i^2, \tag{1.59}$$

the corresponding concentration of free surface electrons, $n(0)$, is

$$n(0) = \frac{N_i^2}{N_A} e^{\varphi_o/V_T} = N_A e^{(\varphi_o - 2V_F)/V_T}, \tag{1.60}$$

where Equation 1.55 has been exploited.

Because $G(\varphi_o)$ in Equation 1.57, as well as its companion relationship, $G(-\varphi_o)$, is a nonnegative number for all positive and negative values of the surface potential, the radical on the right-hand side of Equation 1.52 is a positive real number. Accordingly, Equation 1.53 forces $Q_o(\varphi_o) > 0$ for $\varphi_o < 0$ and $Q_o(\varphi_o) < 0$ for $\varphi_o > 0$. The positive nature of the surface charge density for negative surface potentials is indicative of bulk substrate holes attracted to the semiconductor surface because of the force exerted by the surface electric field established in response to negative surface potential. From Gauss' law, this field, say $E_o(\varphi_o)$ is simply

$$E_o(\varphi_o) = -\frac{Q_o(\varphi_o)}{\varepsilon_s} = \text{sgn}(\varphi_o)\left(\frac{\sqrt{2}V_T}{D_b}\right)\sqrt{G(-\varphi_o) + G(\varphi_o)e^{-2V_F/V_T}}, \tag{1.61}$$

which is indeed negative for $\varphi_o < 0$. Observe that Equation 1.58 supports the contention of an enhanced surface hole concentration when the potential established at the semiconductor surface is negative.

An equilibrium condition, which is more commonly referred to in the literature as the *flatband operating condition*, is reached when the applied gate–bulk or gate–source voltage produces a null surface potential, that is, $\varphi_o = 0$. For $\varphi_o = 0$, the net surface charge, $Q_o(\varphi_o)$, in Equation 1.52 is zero, as is the surface electric field, $E_o(\varphi_o)$, in Equation 1.61. Note further that by Equation 1.58, $p(0) = N_A$, which is the equilibrium hole concentration indicative of the NMOS bulk substrate for the transistor abstracted in Figure 1.17, assuming complete ionization of all substrate acceptor impurity atoms.

The negative surface charge prevailing for positive surface potentials, which gives rise to positive surface fields (field lines directed from the surface-to-the bulk substrate), reflects the surface charge depletion forged in response to holes repelled from the surface by $\varphi_o > 0$. Once again, Equation 1.58 is supportive of the proffered rationale in that it confirms a diminished surface hole concentration for progressively larger φ_o. Since departed holes leave in their wake a depletion zone of negative acceptor ions, the negative surface charge density resulting from positive surface potential is hardly surprising.

In addition to repelling holes from the semiconductor surface, Equation 1.60 indicates that the surface electron concentration increases as the surface potential, φ_o, rises above zero. Moreover, Equation 1.52 lends credence to this enhanced electron concentration claim since $Q_o(\varphi_o)$ is seen as becoming monotonically more negative as surface potential φ_o rises above zero. Indeed, the impact of the positive electric field associated with $\varphi_o > 0$ is to establish a force serving to attract the minority carriers (electrons) in the bulk substrate to the surface. For a surface potential in the range, $0 < \varphi_o < V_F$, the depletion charge contribution to the net surface charge continues to dominate over the charge associated with electrons cajoled to the surface, and the surface is said to operate in *depletion mode*. But as φ_o approaches and ultimately surpasses the Fermi potential, V_F, the impact on the nature of the surface charge becomes increasingly more interesting. For example, consider $\varphi_o = V_F$, for which Equations 1.58 and 1.60 yield $p(0) \equiv n(0) = N_i$, that is, the hole and electron concentrations at the surface are identically equal to the intrinsic carrier concentration. In effect, the surface region of the semiconductor changes from obviously P-type-to-intrinsic type, which is to say that the surface at $\varphi_o = V_F$ is neither P-type nor N-type.

For $V_F < \varphi_o < 2V_F$, Equations 1.58 and 1.60 project a surface electron concentration that actually exceeds the surface hole concentration, despite the originally P-type character of the semiconductor surface. In this range of surface potentials, the depletion layer at the surface continues to expand into the substrate but because of the enhanced electron concentration, the surface is said to operate in a condition of *weak inversion*. Weak inversion is significant from an engineering perspective in that it begins to establish the necessary condition for promoting observable drain and source current flow. In particular, suppose that the drain–source voltage, V_{ds}, were to be increased from its present null value to a suitably positive value. The presence of a significant mobile surface charge density in the form of free electrons allows said electrons to be transported from the source-to-the drain by the force associated with the lateral electric field established in response to $V_{ds} > 0$. In turn, this charge transport gives rise to a drain current flowing into the transistor and a source current flowing out of the device.

When φ_o rises to the value, $2V_F$, Equation 1.60 confirms a surface electron concentration that is numerically equal to the substrate doping concentration, N_A. In other words, the surface electron concentration precipitated by the strong positive electric fields implicit to $\varphi_o = 2V_F$ is identical to the equilibrium hole concentration evidenced in a silicon mass whose impurity concentration of completely ionized acceptor atoms is N_A. The surface has effectively changed its sex from its former P-type state to a field-induced (hence the terminology, ''field-effect,'' in the FET nomenclature) N-type state. Since the resultant surface electron concentration, $n(0)$, is rendered substantive, appreciable drain and source currents can flow for even modest values of applied drain–source voltages. In effect, the transistor can be said to be ''turned on'' when φ_o rises to twice the Fermi potential in the sense that a capability for substantial drain current flow is forged. When $\varphi_o \geq 2V_F$, the semiconductor surface is *strongly inverted*, or simply *inverted*.

Figure 1.20 displays a representative surface charge density profile as a function of the surface potential. Since a logarithmic charge scale is required to display all salient features of the charge density, the negative nature of the surface charge for positive surface potentials compels plotting the magnitude of the surface charge density on the vertical (charge) scale in the subject figure. The horizontal (voltage) scale is normalized to the thermal voltage, V_T. The plot invokes the presumptions of a 27°C semiconductor surface temperature and a substrate impurity concentration of $N_A = 10^{15}$ atoms/cm^3. For these stipulations, the thermal voltage is $V_T = 25.89$ mV, and the Fermi potential is $V_F = 288.4$ mV, whence $V_F/V_T = 11.14$. The plot displayed in Figure 1.20 clearly identifies the regions of hole accumulation ($\varphi_o < 0$), surface depletion ($0 < \varphi_o < 2V_F$), weak inversion, as typified by the increased concentration of free electrons at the surface ($V_F < \varphi_o < 2V_F$), and strong inversion, for which $\varphi_o \geq 2V_F$.

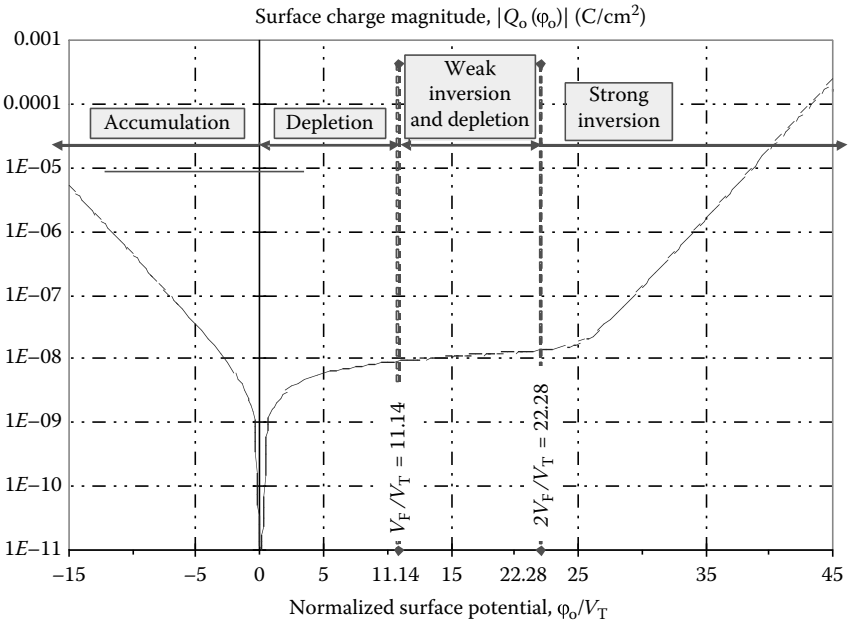


FIGURE 1.20 The magnitude of the surface charge density in the channel interfacial region for the MOSFET configured as shown in Figure 1.19b. A surface temperature of 27°C is assumed, as is a substrate impurity concentration of $N_A = 10^{15}$ atoms/cm^2.

1.2.2.2 Gate–Bulk Capacitance

The density of the net gate-to-bulk capacitance, $C_{gb}(\varphi_o)$, of the MOSFET whose cross-section diagram appears in Figure 1.19b is a series combination of the oxide capacitance density, C_{ox}, and the density of capacitance $C_d(\varphi_o)$, which is established between the oxide–substrate interface and the charge neutral region of the bulk. The pertinent equivalent circuit for $V_{ds} = 0$ is the structure depicted in Figure 1.21b, for which

$$C_{gb}(\varphi_o) = \frac{C_{ox}C_d(\varphi_o)}{C_{ox} + C_d(\varphi_o)}. \tag{1.62}$$

In Equation 1.62,

$$C_{ox} = \frac{\varepsilon_{ox}}{T_{ox}}, \tag{1.63}$$

where $\varepsilon_{ox} = 345$ fF/cm is the dielectric constant of silicon dioxide. Moreover,

$$C_d(\varphi_o) = \frac{d|Q_o(\varphi_o)|}{d\varphi_o}, \tag{1.64}$$

where the surface charge density, $Q_o(\varphi_o)$, is defined by Equation 1.52. After a trifle of differential calculus pain, it can be shown that

$$C_d(\varphi_o) = \text{sgn}(\varphi_o)\frac{\varepsilon_s}{\sqrt{2}D_b}\left[\frac{\left(e^{\varphi_o/V_T} - 1\right)e^{-2V_F/V_T} - \left(e^{-\varphi_o/V_T} - 1\right)}{\sqrt{G(-\varphi_o) + G(\varphi_o)e^{-2V_F/V_T}}}\right]. \tag{1.65}$$

The result at hand defines the surface capacitance density for all values of the surface potential, φ_o. A problem arises for $\varphi_o = 0$ in that the right-hand side becomes an indeterminate 0/0 form. This problem is circumvented by supplanting the exponential terms on the right-hand side of Equation 1.65, inclusive of those embedded in the functions, $G(\varphi_o)$ and $G(-\varphi_o)$, by their second order MacLaurin series expansions. Upon replacement of these exponential terms by said expansions, the surface capacitance density at the flatband condition, $\varphi_o = 0$, is found to be

$$C_d(0) = \frac{\varepsilon_s}{D_b}\sqrt{1 + e^{-2V_F/V_T}} \approx \frac{\varepsilon_s}{D_b} \triangleq C_{FB}, \tag{1.66}$$

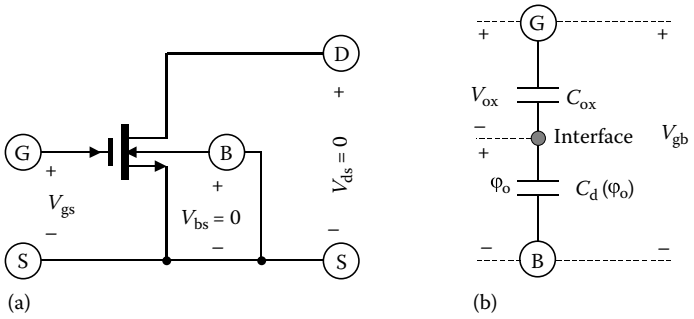


FIGURE 1.21 (a) NMOS transistor of Figure 1.19a operated with $V_{ds} = 0$ and $V_{bs} = 0$. (b) Circuit model between the gate and bulk terminals of the transistor in (a).

where C_{FB} is termed the surface flatband capacitance. The indicated approximation exploits the presumption that the impurity concentration, N_A, in the bulk substrate is significantly larger than the intrinsic carrier concentration, N_i, of silicon. It follows from Equation 1.62 that

$$\frac{C_{gb}(\varphi_o)}{C_{ox}} = \frac{1}{1 + \frac{C_{ox}}{C_d(\varphi_o)}}, \tag{1.67}$$

for which

$$\frac{C_{gb}(0)}{C_{ox}} \approx \frac{1}{1 + \left(\frac{\varepsilon_{ox}}{\varepsilon_s}\right)\left(\frac{D_b}{T_{ox}}\right)}. \tag{1.68}$$

Figure 1.22 displays a plot of the normalized gate–bulk capacitance, $C_{gb}(\varphi_o)/C_{ox}$, as a function of the normalized surface potential, φ_o/V_T, at room temperature (27°C) conditions. The relevant MOSFET is presumed to have an acceptor impurity concentration, N_A, in the bulk of 10^{15} atoms/cm^2, and a gate silicon dioxide thickness, T_{ox}, of 30 Å. The curve shows that in strong accumulation where $\varphi_o \ll 0$, the gate–bulk capacitance per unit area approaches the density of the oxide capacitance, C_{ox}. This observation reflects engineering expectations in that $\varphi_o \ll 0$ attracts a very large concentration of holes to the surface, for which the associated charge density serves to increase dramatically the surface density of capacitance, $C_d(\varphi_o)$. Indeed, for $\varphi_o \ll 0$, it is a simple matter to show that Equation 1.65 collapses to

$$C_d(\varphi_o)\big|_{\varphi_o \ll 0} \approx \frac{\varepsilon_s}{\sqrt{2}D_b}\sqrt{e^{-\varphi_o/V_T} - 1} \approx \frac{\varepsilon_s}{\sqrt{2}D_b}e^{|\varphi_o|/2V_T}, \tag{1.69}$$

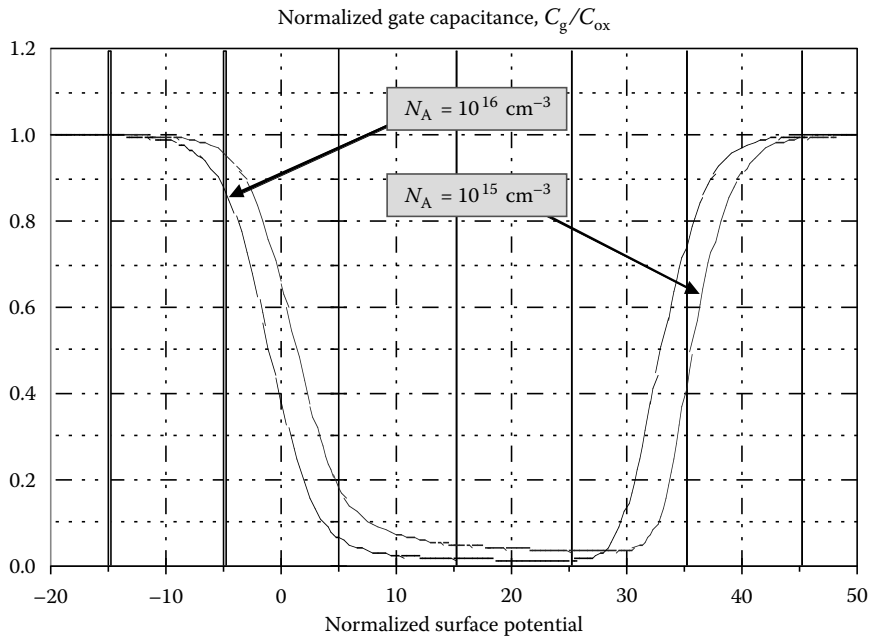


FIGURE 1.22 The normalized gate-to-bulk capacitance of the N-channel MOSFET shown in Figure 1.21a as a function of the indicated normalized surface potential. The temperature of the oxide–semiconductor interface is taken to be $T = 27$°C, the oxide thickness is $T_{ox} = 30$ Å, and the acceptor impurity concentration in the bulk is $N_A = 10^{15}$ atoms/cm^2. The curve is applicable to only low signal frequencies.

which clearly suggests a sharp rise in capacitance density with the absolute value of the negative surface potential. Since the surface capacitance density can be viewed as a ratio of the silicon dielectric constant, ε_s, to an effective and voltage-dependent dielectric thickness, say $y(\varphi_o)$, observe a dielectric thickness associated with Equation 1.69 of

$$y(\varphi_o)\big|_{\varphi_o \ll 0} \approx \sqrt{2} D_b e^{-|\varphi_o|/2V_T}, \tag{1.70}$$

which diminishes rapidly with progressively more negative surface potentials.

As the surface potential increases toward and beyond zero, the normalized capacitance plotted in Figure 1.22 decreases because a depletion layer begins to form at the interface. This depletion layer acts as a dielectric whose thickness increases as a nominal square root function of the surface potential. Under the depletion condition, the surface capacitance given by Equation 1.65 can be approximated by

$$C_d(\varphi_o)\big|_{\text{Depletion}} \approx \frac{\varepsilon_s}{\sqrt{2} D_b \sqrt{\frac{\varphi_o}{V_T} - 1}}, \tag{1.71}$$

which implies a depletion layer thickness, say y_d, (not to be confused with Y_d, the depth of the source and drain regions) of

$$y_d \triangleq y(\varphi_o)\big|_{\text{Depletion}} \approx D_b \sqrt{2\left(\frac{\varphi_o}{V_T} - 1\right)}. \tag{1.72}$$

Equation 1.71 approximates the actual surface capacitance density to within an error magnitude of nominally less than 10% for $3V_T \le \varphi_o \le 20V_T$. This allowable range of surface potential can actually be extended to embrace $\varphi_o \le 2V_F$ since for $\varphi_o > V_F$, an appreciable portion of the charge observed at the interface can be attributed to free electrons, and not simply to the ionic charge in the depletion layer forged by holes repelled from the interface.

As φ_o continues to increase, $C_d(\varphi_o)$, and hence $C_{gb}(\varphi_o)$, continues decreasing toward a minimum value that is achieved at a value close to a surface potential of $2V_F$, which is the threshold of strong surface inversion. At this potential, the thickness of the depletion layer implicit to the interfacial capacitance density is maximized since further increases in the surface charge density derive dominantly from electrons attracted to the surface. Rather than indulge in the academic propriety of using Equation 1.65 to compute the exact surface potential commensurate with minimal surface capacitance density, engineering prudence encourages the simplified approach of presuming $\varphi_o = 2V_F$ to be a sufficiently accurate requirement for minimal depletion capacitance. Upon adoption of this stance, Equation 1.72 is suitable for computing the maximum thickness, say W_d, of the depletion layer. Accordingly,

$$W_d \triangleq y(2V_F) \approx D_b \sqrt{2\left(\frac{2V_F}{V_T} - 1\right)}, \tag{1.73}$$

and since $2V_F$ is invariably much larger than the thermal voltage, V_T, Equation 1.56 allows this result to be written as

$$W_d \approx 2\sqrt{\frac{\varepsilon_s V_F}{qN_A}}. \tag{1.74}$$

The resultant minimum density of surface depletion capacitance is

$$C_{\rm d}(2V_{\rm F}) = \frac{\varepsilon_{\rm s}}{W_{\rm d}} \approx \frac{1}{2}\sqrt{\frac{qN_{\rm A}\varepsilon_{\rm s}}{V_{\rm F}}}. \tag{1.75}$$

Using Equations 1.67 and 1.63, the corresponding density, $C_{\rm min}$, of minimum gate–bulk capacitance is

$$C_{\rm min} \approx C_{\rm gb}(2V_{\rm F}) \approx \frac{\varepsilon_{\rm ox}/T_{\rm ox}}{1 + 2\left(\frac{\varepsilon_{\rm ox}}{T_{\rm ox}}\right)\sqrt{\frac{V_{\rm F}}{qN_{\rm A}\varepsilon_{\rm s}}}}. \tag{1.76}$$

Observe that the maximum factor by which the effective gate–bulk capacitance can be reduced is

$$\frac{C_{\rm ox}}{C_{\rm min}} \approx 1 + 2\left(\frac{\varepsilon_{\rm ox}}{T_{\rm ox}}\right)\sqrt{\frac{V_{\rm F}}{qN_{\rm A}\varepsilon_{\rm s}}}, \tag{1.77}$$

which involves parameters that are largely out of the control of the circuit designer. By inspection of Figure 1.22, this capacitance perturbation requires a surface potential swing extending from roughly $-15V_{\rm T}$ (about -400 mV at 27°C) to $2V_{\rm F}$ (generally smaller than 600 mV). Although the requisite surface potential excursion is somewhat large for maximal capacitance modulation, it should be noted that the maximum capacitance change factor predicted by Equation 1.77 can be as large as almost 100.

As $\varphi_{\rm o}$ increases beyond twice the Fermi potential, the interface charge density increases robustly as the semiconductor surface begins to invert strongly. Figure 1.22 resultantly displays an increasing bulk–gate capacitance density, not unlike the increased capacitance prevailing in strong accumulation because of holes attracted to the interface. Under actual measurement conditions, however, the indicated increased capacitance for $\varphi_{\rm o} > 2V_{\rm F}$ is observed only when the frequencies of signals established between the gate and bulk are below a few tens of hertz [4–6]. The problem is that for most practical signal frequencies, the recombination–generation rates of electrons in NMOS devices are unable to track with the signal-induced exchanges in charge between the neutral bulk and the inversion layer. Figure 1.23 displays the true gate–bulk capacitance characteristics for practical signal frequencies, wherein the dashed segment drawn for $\varphi_{\rm o} > 2V_{\rm F}$ is the applicable high-voltage capacitance trace for frequencies above a few tens of hertz.

1.2.2.3 Approximate Depletion Zone Analysis

As delineated in the discussion pertaining to the surface charge density defined by Equation 1.52, the MOSFET in Figure 1.21a exhibits depletion at the oxide–semiconductor interface for surface potentials satisfying the constraint, $0 < \varphi_{\rm o} < 2V_{\rm F}$. The formation of the surface depletion zone is critically important to the establishment of the volt–ampere characteristics of a MOSFET because it serves as a precursor to the surface inversion that comprises the necessary condition for drain and source current conduction. Recall, for example, that weak inversion is said to initiate at $\varphi_{\rm o} = V_{\rm F}$, in the sense that the original P-type character of the interfacial semiconductor is transformed to intrinsic material. When $\varphi_{\rm o}$ is elevated to $2V_{\rm F}$, the surface is strongly inverted in that the concentration of free electrons at the surface increases to a value that is identical to the average impurity concentration in the substrate. Although the concentration of surface electrons begins to increase for $\varphi_{\rm o}$ barely above zero, as is highlighted by Equation 1.60, an electron concentration commensurate with the possibility of substantial drain and source current flow does not materialize until the surface potential, $\varphi_{\rm o}$, reaches the immediate neighborhood of twice the Fermi potential. An attribute of Equation 1.52 is that this relationship does not explicitly distinguish between immobile depletion charge and mobile electron charge, both of which contribute to the observed surface charge density. A shortfall of Equation 1.52 is that its analytically cumbersome nature all but precludes the development of mathematically tractable expressions for the volt–ampere characteristics of

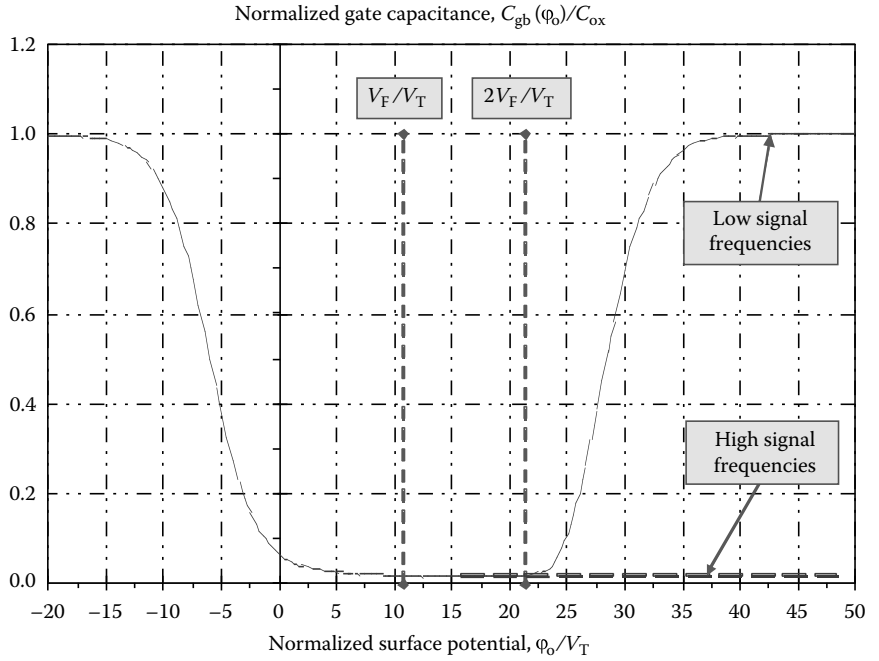


FIGURE 1.23 The capacitance characteristics of Figure 1.22 for the conditions of both low and high signal frequencies.

a MOSFET. Fortunately, the awkwardness of Equation 1.52 is mitigated if the reasonable approximation is made that for $0 < \varphi_o < 2V_F$, the charge in the surface channel region derives exclusively from depletion phenomena, that is, a substantive density of free electron charge does not materialize at the surface until $\varphi_o = 2V_F$.

To the extent that the entire substrate region is uniformly doped at the indicated impurity concentration of N_A and assuming complete ionization of all substrate impurity atoms, the resultant concentration, say $\rho(y)$, of immobile ionic charge in the depletion zone throughout the channel region from source-to-drain is nominally constant at the value, $-qN_A$. Of course, the electron concentration at the surface increases in proportion to the decreased hole population therein but as long as φ_o remains smaller than $2V_F$, Equation 1.60 confirms that the free electron concentration is significantly smaller than N_A.

Figure 1.24a depicts the depletion charge density, $\rho(y)$, beneath the oxide–semiconductor interface, where W_d represents the depth of the depletion layer established at the interface. Using Gauss' law, the electric field, $E(y)$, promoted by this charge concentration profile derives from

$$\frac{dE(y)}{dy} = \frac{\rho(y)}{\varepsilon_s}. \tag{1.78}$$

Since $\rho(y) = -qN_A$ for $0 \le y \le W_d$, Equation 1.78 implies

$$\int_{E(y)}^{E(W_d)} dE(y) = -\frac{qN_A}{\varepsilon_s} \int_{y}^{W_d} dy. \tag{1.79}$$

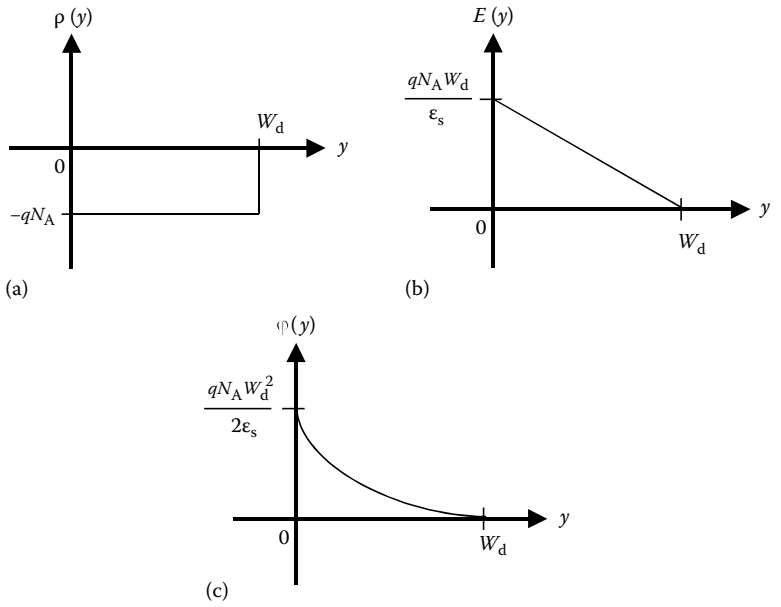


FIGURE 1.24 (a) The approximate profile of the depletion charge concentration at the surface of the MOSFET depicted in Figure 1.21a. (b) The electric field intensity as a function of bulk substrate depth measured with respect to the interfacial surface, corresponding to the charge profile in (a). (c) The potential implied by the electric field plot in (b).

In view of the fact that $E(W_d)$ is zero in the undepleted, charge neutral substrate region corresponding to $W \geq W_d$, Equation 1.79 produces the linear electric field relationship,

$$E(y) = \frac{qN_A W_d}{\varepsilon_s}\left(1 - \frac{y}{W_d}\right) = \frac{V_T W_d}{D_b^2}\left(1 - \frac{y}{W_d}\right), \tag{1.80}$$

where Equation 1.56 for the Debye length is invoked. Equation 1.80 is sketched as a function of the substrate depth variable, y, in Figure 1.24b. Observe that maximum field intensity prevails at surface where

$$E(0) \triangleq E_o = \frac{V_T}{D_b}\left(\frac{W_d}{D_b}\right). \tag{1.81}$$

The potential, $\varphi(y)$, corresponding to the field intensity, $E(y)$, stipulated by Equation 1.80 satisfies

$$-\frac{d\varphi(y)}{dy} = E(y) = \frac{qN_A}{\varepsilon_s}(W_d - y). \tag{1.82}$$

If zero reference potential is ascribed to the substrate depth, W_d, beyond which the substrate is charge neutral, Equation 1.82 sets forth

$$\int_{\varphi(y)}^{0} d\varphi(y) = \frac{qN_A}{\varepsilon_s}\int_{y}^{W_d}(W_d - y)dy, \tag{1.83}$$

whence the bulk substrate potential, $\varphi(y)$, referenced to the potential evidenced at the substrate depletion depth, W_d, is

$$\varphi(y) = \frac{qN_A W_d^2}{2\varepsilon_s}\left(1 - \frac{y}{W_d}\right)^2 = \frac{1}{2}V_T\left(\frac{W_d}{D_b}\right)^2\left(1 - \frac{y}{W_d}\right)^2, \tag{1.84}$$

whose functional dependence on variable y is sketched in Figure 1.24c. Equation 1.84 suggests that the surface potential, $\varphi(0)$, which is effectively the net voltage dropped across the depleted region of the bulk substrate, is

$$\varphi(0) \triangleq \varphi_o = \frac{qN_A W_d^2}{2\varepsilon_s} = \frac{1}{2}V_T\left(\frac{W_d}{D_b}\right)^2 \equiv \frac{E_o W_d}{2}, \tag{1.85}$$

Since the interface potential, φ_o, and hence the depletion depth, W_d, is controlled externally by the applied gate-to-bulk voltage, V_{gb}, it is of interest to determine φ_o as an explicit function of V_{gb}. To this end, the electric field intensity, E_{ox}, in the silicon dioxide layer of the structure of Figure 1.19b is uniform throughout the oxide thickness by virtue of the insulating nature of the oxide. Ignoring work function phenomena prevailing between the gate contact and gate oxide, as well as between the oxide and semiconductor surface, this field is simply

$$E_{ox} = \frac{V_{gb} - \varphi_o}{T_{ox}}. \tag{1.86}$$

Equation 1.86 also invokes the approximation that the voltage dropped from the bottom of the depletion region-to-the bulk terminal is essentially zero. This assumption is reasonable in that the substrate is ultimately reverse biased to preclude substantive bulk current flow. Moreover, the holes displaced from the interface region-to-the neutral bulk render the neutral substrate zone a low resistivity volume. Because the field immediately below the interface is E_o, as defined by Equation 1.81, continuity constraints mandate

$$\varepsilon_{ox}E_{ox} = \varepsilon_s E_o, \tag{1.87}$$

or

$$\varepsilon_{ox}\left(\frac{V_{gb} - \varphi_o}{T_{ox}}\right) = \varepsilon_s\left(\frac{V_T W_d}{D_b^2}\right) \approx C_{FB}\left(\frac{W_d}{D_b}\right)V_T. \tag{1.88}$$

Armed with Equations 1.86, 1.63, and 1.85, parameter W_d in Equation 1.88 can be eliminated to arrive at the utilitarian expression,

$$V_{gb} = \varphi_o + \sqrt{2V_\theta\varphi_o}, \tag{1.89}$$

where the voltage metric, V_θ, termed the *body effect voltage*, is given by

$$V_\theta = V_T\left(\frac{C_{FB}}{C_{ox}}\right)^2 = V_T\left(\frac{\varepsilon_s}{\varepsilon_{ox}}\right)^2\left(\frac{T_{ox}}{D_b}\right)^2 = \frac{qN_A\varepsilon_s}{C_{ox}^2}. \tag{1.90}$$

Parameter V_θ is generally of the order of the mid-tens of microvolts.* Observe that V_θ is proportional to the square of the oxide thickness, T_{ox} and is therefore reduced sharply with diminishing gate oxide thickness.

The final step to the problem of determining the dependence of interface potential φ_o on applied gate-to-bulk voltage V_{gb} involves a straightforward solution of Equation 1.89 for φ_o. The result is

$$\varphi_o = V_{gb} + V_\theta - \sqrt{V_\theta\left(2V_{gb} + V_\theta\right)}. \tag{1.91}$$

As expected, $\varphi_o = 0$ for $V_{gb} = 0$. Ordinarily, V_θ is much smaller than practical values of the gate-to-bulk voltage, V_{gb}, so that Equation 1.91 can be approximated as

$$\varphi_o \approx V_{gb} - \sqrt{2V_\theta V_{gb}}, \tag{1.92}$$

which is similar in form to Equation 1.89.

1.2.2.4 Threshold

The approximate depletion regime analysis executed in Section 1.2.2.3 conveniently precipitates an analytical definition of the threshold condition, that is, the condition whereby strong inversion materializes at the oxide–semiconductor interface. It has been demonstrated that the onset of the threshold condition corresponds to a surface potential, φ_o, of twice the Fermi potential. Accordingly, threshold requires that the gate–bulk voltage V_{gb}, in Equation 1.89 rise to a value, say V_{gbh}, such that

$$V_{gbh} = 2V_F + 2\sqrt{V_\theta V_F}. \tag{1.93}$$

It is to be understood that a gate-to-bulk voltage, V_{gb}, satisfying the constraint, $V_{gb} \geq V_{gbh}$, is commensurate with instilling strong inversion at the surface of the MOSFET depicted in Figure 1.21a. On the presumption that the depth, W_d, of the depletion layer, corresponding to $\varphi_o = 2V_F$ is unaltered by gate–bulk voltage increases beyond the threshold value, V_{gbh}, the resultant charge profile offered in Figure 1.24a changes into the form diagrammed in Figure 1.25. In this diagram, N_s is the concentration of free

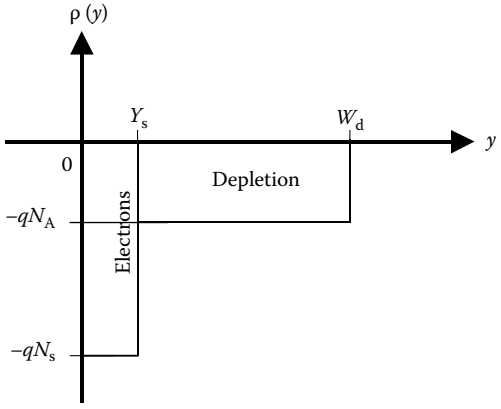


FIGURE 1.25 The approximate profile of the charge concentration for a strongly inverted surface in the MOSFET depicted in Figure 1.21a.

* Most HSPICE and other SPICE simulators use a *body effect parameter* to compute the extent to which the bulk–source voltage perturbs the gate–source threshold voltage. This body effect parameter, γ, derives from $\gamma^2 = 2V_\theta$.

electrons in the surface inversion layer, and Y_s is the thickness of the inversion layer. Depending on the value of the gate–bulk voltage excess, $(V_{gb} - V_{gbh})$, Y_s is typically 20%–50% larger than the electron Debye length.

Two circumstances limit the utility of Equation 1.93. The first of these is that MOSFETs are often operated with nonzero bulk–source bias, as opposed to the zero bias presumed to this juncture. If a bulk–source voltage, V_{bs}, is applied to the MOSFET in Figure 1.21a, the potentials at both the oxide–semiconductor interface and the bottom of the depletion layer are elevated by an amount, V_{bs}, which, in concert with earlier admonitions, is invariably a negative voltage to ensure reverse biasing of the bulk–source and bulk–drain PN junctions. Thus, the results documented in Section 1.2.2.3 remain valid because the voltage developed across the depletion layer is still the surface potential, φ_o, exploited therein. However, since the surface potential rises by V_{bs}, one of the necessary modifications to results disclosed earlier is that Equation 1.60 for the free electron concentration at the surface must be modified as

$$n(0) = \frac{N_i^2}{N_A} e^{(\varphi_o + V_{bs})/V_T} = N_A e^{(\varphi_o + V_{bs} - 2V_F)/V_T}. \tag{1.94}$$

Recall that the measure for the onset of strong inversion in a MOSFET is a surface electron concentration, $n(0)$, that equals to the hole concentration, N_A, in the equilibrium substrate. In order to effect this strong inversion condition, Equation 1.94 suggests the necessity of a surface potential that is at least as large as $(2V_F - V_{bs})$, as opposed to merely $2V_F$. Accordingly, the effect of bulk–source biasing on the gate–bulk threshold voltage can be embraced by replacing the voltage, $2V_F$, in Equation 1.93 by the voltage $(2V_F - V_{bs})$ so that

$$V_{gbh} = (2V_F - V_{bs}) + \sqrt{2V_\theta(2V_F - V_{bs})}. \tag{1.95}$$

The second shortfall of Equation 1.93 stems from the fact in actual circuit design environments, it is far more convenient to stipulate the minimum gate–source voltage, V_{gs}, and not the minimum gate–bulk voltage, V_{bs}, that establishes the onset of strong inversion. Since V_{gs} is the voltage sum, $(V_{gb} + V_{bs})$, adding V_{bs} to both sides of Equation 1.95 delivers a gate-to-source threshold voltage, say V_h, of the form,

$$V_h = V_{ho} + 2\sqrt{V_\theta V_F}\left(\sqrt{1 - \frac{V_{bs}}{2V_F}} - 1\right), \tag{1.96}$$

where V_{ho}, which represents the zero bias ($V_{bs} = 0$) value of the gate–source threshold potential, is

$$V_{ho} = 2\left(V_F + \sqrt{V_\theta V_F}\right). \tag{1.97}$$

In practice, the zero bias value, V_{ho}, of gate–source threshold voltage is best evaluated through measurement since it is strongly influenced by gate region work function phenomena and parasitic charges trapped in the gate oxide layer, whose engineering effects are difficult to quantify accurately and reliably. Observe in Equation 1.96 that the effect of an increasing bulk-to-source reverse bias ($V_{bs} < 0$) is to increase the threshold voltage above its zero bias value, V_{ho}, as a square root function of V_{bs}. This bulk-induced modulation of the threshold potential is rendered small by small values of the square root of parameter V_θ, which Equation 1.90 projects as directly dependent on the gate oxide thickness, T_{ox}. One reason for the current penchant toward progressively decreased oxide thickness is the minimization of threshold voltage modulation, which is generally an undesirable effect in MOSFETs deployed in analog network applications.

1.2.3 Volt–Ampere Characteristics

The static volt–ampere characteristics of the N-channel MOSFET in Figure 1.17 stipulate the dependence of the static drain current, I_d, on the static values of the gate–source voltage, V_{gs}, the drain–source voltage, V_{ds}, and the bulk–source voltage, V_{bs}, which is invariably a nonpositive voltage. It is convenient to partition these characteristics into three segments; namely, the *cutoff regime*, the *ohmic regime*, and the *saturation regime*.

1.2.3.1 Cutoff Regime

The cutoff regime is the most boring of the three MOSFET operating domains in that no drain current flows in cutoff, despite all reasonable positive value of the drain–source voltage, V_{ds}. Since drain current conduction requires surface inversion at the oxide–substrate interface, zero current is assured when no such charge inversion prevails. In turn, no inversion layer is formed when the gate–source voltage, V_{gs}, lies below its threshold value, V_h. Thus, in cutoff,

$$I_d = 0, \quad \text{if } V_{gs} < V_h. \tag{1.98}$$

As V_{gs} rises above zero but remains below V_h, a subthreshold current is induced from the drain-to-the source regions for $V_{ds} > 0$ [7]. This current is manifested by the fact that, as is conveyed by Equation 1.94, the interfacial free electron concentration increases slightly for surface potentials above 0 V. However, the current evidenced in the subthreshold regime is small because of the limited availability of free surface electrons. As a result, the gain and frequency response, in addition to the actual drain current, that the transistor is capable of mustering are limited. Although subthreshold operation enjoys utility in certain types of low-power system applications, such as hearing aids, where neither gain nor bandwidth are daunting requirements, it is rarely exploited in broadband and other high-performance applications.

1.2.3.2 Ohmic Regime

In the ohmic regime, which is sometimes called the *triode regime*, $V_{gs} \geq V_h$ and $V_{ds} \leq (V_{gs} - V_h)$. Thus, the interfacial surface of a MOSFET is strongly inverted in the ohmic domain, and simultaneously, a relatively small voltage is applied from the drain-to-the source. The voltage difference, $(V_{gs} - V_h)$, is commonly referenced as the *drain saturation voltage*, V_{dsat}, that is,

$$V_{dsat} \triangleq V_{gs} - V_h. \tag{1.99}$$

Because of Equation 1.49, observe that the provision, $V_{ds} \leq (V_{gs} - V_h)$, is equivalent to the requirement, $V_{gd} \geq V_h$. This is to say that a MOSFET operates in the ohmic regime if and only if both the gate–source and the gate–drain voltages are larger than the threshold potential. Viewed in yet another fashion, $V_{gs} \geq V_h$ and $V_{gd} \geq V_h$ ensure that both the source and the drain ends of the interfacial surface between the source and the drain regions are strongly inverted. A conduit, or channel, of free electrons that electrically couples the source to the drain is thereby established.

The aforementioned channel of free electrons is highlighted in the device cross section abstracted in Figure 1.26. Because $V_{gs} \geq V_h$ and $V_{gd} \geq V_h$, the electron inversion layer extends throughout the entire surface region from the source-to-the drain. But since $V_{gd} = (V_{gs} - V_{ds})$ and $V_{ds} > 0$, the gate-to-drain bias, V_{gd}, is necessarily smaller than its gate-to-source counterpart, V_{gs}. It follows that the surface potential in the neighborhood of the drain region is smaller than that prevailing near the source region, whence the electron concentration near the drain is smaller than it is at the source. Accordingly, the channel of electrons depicted in the figure at hand does not have a uniform depth (y-direction) and is, in fact, deeper at the source site, where $x = 0$, than it is at the drain site, which is typified analytically by $x = L$. For analogous reasons, the depletion region established about the source, at the interface, and at the drain is widest near the drain.

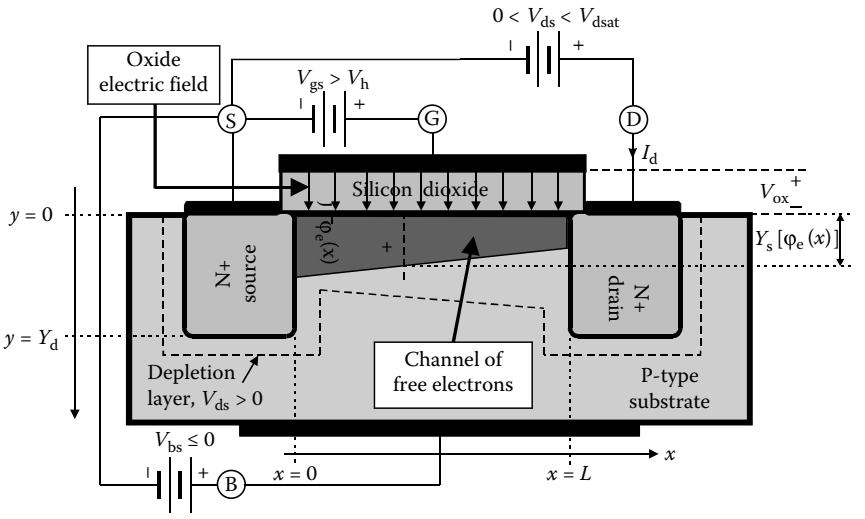


FIGURE 1.26 Cross section of the N-channel MOSFET operated in its ohmic regime. Note that all applied voltages are referred to the source terminal. The diagram is not drawn to scale.

An additionally important point is that the channel potential, symbolized in Figure 1.26 as $\varphi_c(x)$, is measured with respect to the source site. This notation is not to be confused with the previously invoked variable, $\varphi_o(y)$, which measures the potential at the interfacial surface with respect to the neutral region of the bulk substrate. The change in symbolism is reasonable and is encouraged by two issues addressed in the Section 1.2.3.3. The first of these issues is that the bulk–source biasing voltage, V_{bs}, has been absorbed into the threshold voltage metric stipulated by Equation 1.96. Second, this threshold voltage has been defined in terms of the gate voltage, V_{gs}, measured with respect to the source, as opposed to the gate voltage, V_{gb}, referenced to the bulk terminal. Because V_{ds} is nonzero, the channel potential, $\varphi_c(x)$, is not a constant but instead, it varies continuously from $\varphi_c(0) = 0$ at the source site where $x = 0$ to $\varphi_c(L) = V_{ds}$ at the drain site where $x = L$.

An applied drain–source voltage, V_{ds}, launches a lateral electric field, say $E_x(\varphi_c(x))$, that is directed from the drain site-to-the source site and is functionally dependent on the channel potential, $\varphi_c(x)$. This electric field is given by the familiar relationship,

$$E_x(\varphi_c(x)) = -\frac{d\varphi_c(x)}{dx}. \tag{1.100}$$

If μ_n denotes the mobility of electrons, whose concentration within the surface inversion layer postulated in Figures 1.26 and 1.25 is $N_s(\varphi_c(x))$, the static drain current, I_d, promoted by this lateral field is

$$\begin{aligned} I_d &= -q\mu_n W[Y_s(\varphi_c(x))][N_s(\varphi_c(x))]E_x(\varphi_c(x)) \\ &= q\mu_n W[Y_s(\varphi_c(x))][N_s(\varphi_c(x))]\frac{d\varphi_c(x)}{dx}, \end{aligned} \tag{1.101}$$

where $Y_s(\varphi_c(x))$ is the inversion layer thickness introduced in Figure 1.25 and depicted as dependent on the channel potential, $\varphi_c(x)$, in Figure 1.26. It is worthwhile noting that the product, $\mu_n E_x(\varphi_c(x))$, is the velocity of electrons propagated through the inversion layer. This velocity is zero, thereby implying zero drain current, if the gradient, $d\varphi_c(x)/dx$, of channel potential is null. In turn, the channel potential gradient is zero if the applied drain–source voltage, V_{ds}, is zero.

If it is assumed that increases in the channel potential over and above the threshold level incur no change in the geometry of the interfacial depletion region and instead, only cause electrons to be attracted to the surface, Gauss's law predicts

$$q[Y_s(\varphi_c(x))][N_s(\varphi_c(x))] = \varepsilon_{ox}E_{ox}(\varphi_c(x)), \tag{1.102}$$

with $E_{ox}(\varphi_c(x))$ symbolizing the oxide electric field, which is given by

$$E_{ox}(\varphi_c(x)) = \frac{V_{ox}}{T_{ox}} = \frac{V_{gs} - V_h - \varphi_c(x)}{T_{ox}}. \tag{1.103}$$

Recalling Equations 1.63, 1.103, and 1.102 combine with Equation 1.101 to deliver

$$I_d dx = \mu_n C_{ox} W\left[V_{gs} - V_h - \varphi_c(x)\right] d\varphi_c(x). \tag{1.104}$$

An integration of the left-hand side of this result from $x = 0$ to $x = L$ is tantamount to integrating the right-hand side of said result from $\varphi_c(0) = 0$ to $\varphi_c(L) = V_{ds}$. Assuming constant electron mobility through the channel, the requisite integration is straightforward and leads to the desired volt–ampere relationship,

$$I_d = K_n\left(\frac{W}{L}\right)V_{ds}\left(V_{gs} - V_h - \frac{V_{ds}}{2}\right), \tag{1.105}$$

where

$$K_n \triangleq \mu_n C_{ox} \tag{1.106}$$

is the so-called *transconductance coefficient* of the MOSFET. Although K_n is termed a transconductance coefficient, it is not actually a transconductance in that its physical dimension is that of siemens/volt, or mhos/volt.

Several interesting and enlightening features are advanced by Equation 1.105. The first of these is that zero drain current prevails if $V_{ds} = 0$, which is reassuring in that a current flow for null drain–source voltage violates engineering reason, if not the minor issue of conservation of energy. A second, and more significant, point is that the drain current is directly proportional to the gate aspect ratio, W/L. Thus, for fixed gate–source and drain–source voltages, the drain current can be increased or decreased in proportion to this geometric ratio. This controllability over the drain current renders the gate aspect ratio a designable circuit parameter, subject to the proviso that the circuit designer not attempt to make the gate width, W, smaller than the minimum channel length, L, that the process foundry is capable of producing. Thus, if the foundry boasts a 130 nm channel length process, the smallest practical value of W is, in fact, also 130 nm.

A third important feature of Equation 1.105 is the existence of a value of V_{ds} for which drain current I_d is maximized. By setting to zero the partial derivative of I_d in Equation 1.105 with respect to V_{ds}, this extremum is determined to lie at $V_{ds} = (V_{gs} - V_h) = V_{dsat}$, for which the corresponding maximum current, say I_{dsat}, is

$$I_{dsat} = \frac{K_n}{2}\left(\frac{W}{L}\right)\left(V_{gs} - V_h\right)^2 = \frac{K_n}{2}\left(\frac{W}{L}\right)V_{dsat}^2. \tag{1.107}$$

Recalling Equation 1.99, $V_{ds} = V_{dsat}$ corresponds to a gate–drain voltage, V_{gd}, of $V_{gd} = V_h$, which implies that the surface potential at the drain end of the channel barely sustains the onset of strong inversion. In effect, the depth of the electron channel is reduced to zero at the drain site for $V_{ds} = V_{dsat}$,

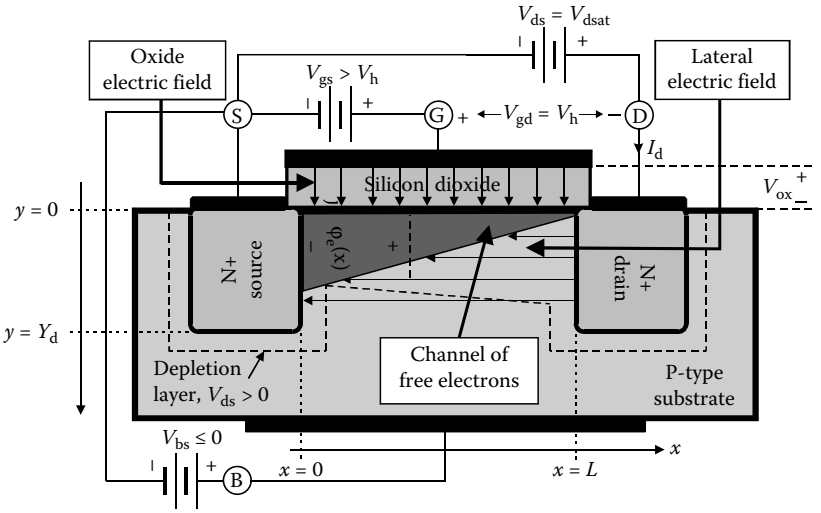


FIGURE 1.27 Cross section of the N-channel MOSFET operated in channel pinch off. The diagram is not drawn to scale.

which justifies the common vernacular of a channel that is *pinched off* at the drain. The situation at hand is diagrammed in Figure 1.27.

At first blush, it may appear incongruous that a pinched off channel, which might be viewed as a means to cut off the supply of electrons to the drain site, can sustain a drain current, yet alone the maximum drain current postulated by Equation 1.107. The current is indeed sustained because of two prevailing phenomena. First, the electric field, $-\mathrm{d}\varphi_c(x)/\mathrm{d}x$, within the inversion layer encourages the transit of electrons toward the tapered edge of the channel at the drain site. Second, electrons reaching the channel edge are influenced immediately by the lateral electric field established by the applied positive drain-to-source voltage. This field, which is abstracted in Figure 1.27 by the indicated horizontal vectors directed from the drain region-to-the source region, sweeps those electrons at the inversion layer boundary into the drain region. The resultant current arising from the transport of electrons across the depletion zone between the tapered channel edge and the drain is, like the current within the inversion layer, proportional to the mobility of electrons. In the depletion zone, this carrier mobility is minority carrier mobility, which is inversely proportional to the background impurity concentration of the bulk substrate. A fundamental reason for maintaining relatively low impurity concentration in the bulk is the assurance of relatively high minority carrier (electron) mobility therein so that carriers are swept across the depletion zone at high velocity, thereby facilitating fast transistor switching and broadband circuit responses.

The fourth interesting feature surrounding Equation 1.105 lends credence to the term, "ohmic," as a descriptive for the operating regime at hand. In particular, Equation 1.105 can be expressed in the form,

$$I_d = K_n\left(\frac{W}{L}\right)V_{ds}\left(V_{gs} - V_h - \frac{V_{ds}}{2}\right) = \frac{V_{ds}}{R_{ds}(V_{gs})}, \tag{1.108}$$

with

$$R_{ds}(V_{gs}) = \frac{1}{K_n(W/L)\left(V_{gs} - V_h - \frac{V_{ds}}{2}\right)}. \tag{1.109}$$

In other words, and as is proffered in Figure 1.28, a MOSFET operated in its ohmic regime, where $V_{gs} \ge V_h$ and $V_{ds} \le V_{dsat}$, behaves as a drain-to-source resistance, $R_{ds}(V_{gs})$, whose resistance value is

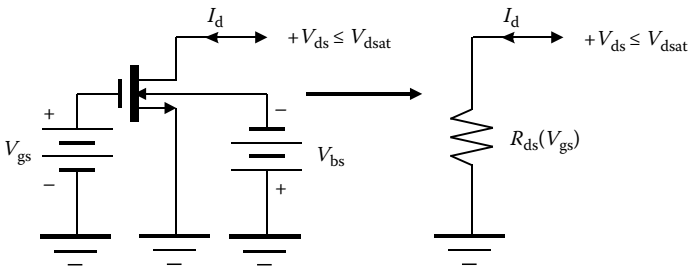


FIGURE 1.28 Static circuit model of an N-channel MOSFET operated in its ohmic regime. The transistor can be operated in such a way that its drain–source terminals emulate a voltage-controlled, nominally linear resistance.

controlled by the applied gate–source voltage, V_{gs}. Moreover, the synthesized resistance is approximately independent of the voltage, V_{ds}, developed across its terminals, and therefore emulates a linear resistance, if $V_{ds} \ll 2(V_{gs} - V_h) \equiv 2V_{dsat}$. In effect, the ohmic regime MOSFET is an electronic approximation of a linear potentiometer whose resistance setting is inversely proportional to the applied gate–source voltage.

1.2.3.3 Saturation Regime

In saturation, which is the volt–ampere domain in which MOSFETs embedded in high-performance analog circuits function, $V_{gs} \ge V_h$ and $V_{ds} \ge (V_{gs} - V_h)$. To first order, the drain current in saturation is taken to be the drain saturation current given by Equation 1.107, which is independent of drain–source voltage, V_{ds}, that is,

$$I_d = \frac{K_n}{2}\left(\frac{W}{L}\right)\left(V_{gs} - V_h\right)^2 \quad \text{for } V_{gs} > V_h, \quad \text{and} \quad V_{ds} \ge \left(V_{gs} - V_h\right). \tag{1.110}$$

The logic underlying this approximation is that the drain current in saturation is determined by the surface electron concentration established for the drain–source voltage, $V_{ds} = (V_{gs} - V_h) = V_{dsat}$, which barely allows for an electron channel spanning the entire source-to-drain spacing. Any increase in the drain-to-source voltage above its saturated value, V_{dsat}, simply adds impetus to the attractive force exerted on inversion layer electrons by the lateral electric field promoted by the drain–source voltage.

The problem with the foregoing logic is that the drain current given by Equation 1.107 is premised on Equation 1.105, which in turn invokes the presumption of an electron inversion layer length that is identical to the channel spacing length, L, separating the source region from the drain region. If $V_{ds} = V_{dsat}$ incurs pinch off at the drain site, and hence an inversion layer length equal to the channel length, L, as illustrated in Figures 1.27 and 1.29a, $V_{ds} > V_{dsat}$ necessarily incurs pinch off within the source–drain spacing, as is suggested in Figure 1.29b. Because of the indicated reduction in the effective channel length from L to $(L - \Delta L)$, the integrated form of Equation 1.104 is now

$$\int_0^{L-\Delta L} I_d dx = \int_0^{V_{dsat}} \mu_n C_{ox} W\left[V_{gs} - V_h - \varphi_c(x)\right] d\varphi_c(x). \tag{1.111}$$

The result of this integration exercise is easily demonstrated to be

$$I_d = \frac{K_n}{2}\left(\frac{W}{L - \Delta L}\right)\left(V_{gs} - V_h\right)^2 = I_{dsat}\left(\frac{L}{L - \Delta L}\right), \tag{1.112}$$

where the current, I_{dsat}, is given by Equation 1.107 and represents the drain current at the transition boundary between ohmic and saturation operational regimes. For most practical applications of

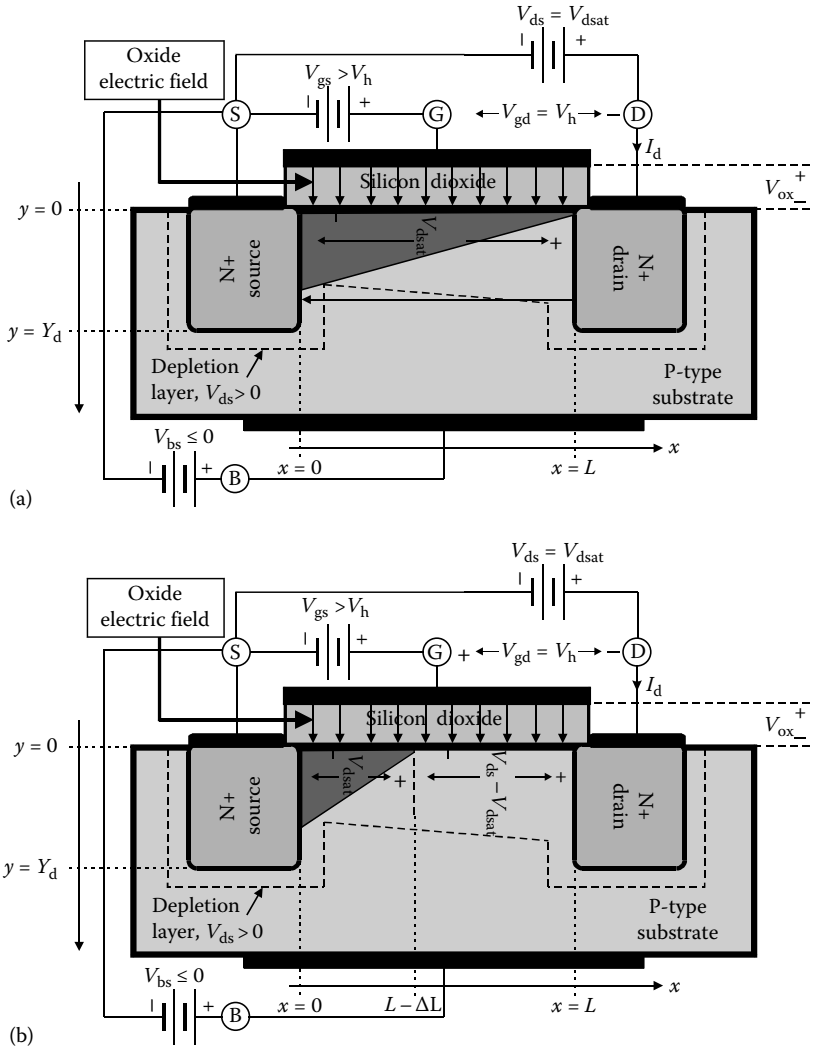


FIGURE 1.29 (a) Cross section of N-channel MOSFET operated in strong inversion and with $V_{ds} = V_{dsat}$. (b) Cross section of the MOSFET in (a) operated with $V_{ds} > V_{dsat}$. The diagrams are not drawn to scale.

MOSFETs [8], the effective reduction, $(L - \Delta L)$, in channel length relates to the drawn channel length, L, as

$$\frac{L}{L - \Delta L} \approx 1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}, \tag{1.113}$$

where V_λ, termed the *channel length modulation voltage*,* is given by the semiempirical expression,

$$V_\lambda = \left(\frac{L}{D_b}\right)\left(\frac{V_j}{V_F}\right)^2 \sqrt{32 V_T \left(V_{ds} - V_{dsat} + V_j\right)}. \tag{1.114}$$

* Most HSPICE and other SPICE simulators use a *channel length parameter* to compute the degree to which the drain–source voltage affects the drain saturation current. This channel length parameter, λ, is $\lambda = 1/V_\lambda$.

In Equation 1.114, V_T is the familiar thermal voltage, D_b is the electron Debye length delineated in Equation 1.56, and V_j is the *built-in potential* of the bulk–drain PN junction. Specifically,

$$V_j = V_T \ln\left(\frac{N_A N_D}{N_i^2}\right), \tag{1.115}$$

with N_A, N_D, and N_i respectively denoting the average impurity concentration in the bulk substrate, the average impurity concentration of the drain diffusion (or implant), and the intrinsic carrier concentration of silicon. Equation 1.114 delivers acceptable analytical accuracy for channel lengths, L, that are no smaller than 0.09 μ and drain–source voltages, V_{ds}, that lie within breakdown ratings of the considered transistor.

The drain current in the saturation regime is now expressible as

$$I_d \approx \frac{K_n}{2}\left(\frac{W}{L}\right)(V_{gs} - V_h)^2\left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right) = I_{dsat}\left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right), \tag{1.116}$$

where it is understood that the gate–source and drain–source voltages, V_{gs} and V_{ds}, respectively, are constrained to satisfy the saturation requirements, $V_{gs} > V_h$ and $V_{ds} \geq (V_{gs} - V_h) = V_{dsat}$. Clearly, the saturation regime drain current is no longer independent of the drain–source voltage. The current is seen to rise with V_{ds} with a slope of I_{dsat}/V_λ. Note, however, that this slope is not constant owing to its square root dependence on V_{ds}. For large V_λ, which is manifested by long transistor channel length, L, this rate of current rise with V_{ds} is modest and indeed, the slope of the current–voltage characteristic curve approaches zero in the limit as V_λ approaches infinity. These observations and Equation 1.116 itself suggest that the drain–source port of a MOSFET does not behave as a constant current source whose value, I_{dsat}, is controlled exclusively by gate–source voltage V_{gs}. Instead, the drain–source port is a practical controlled current source comprised of a constant current generator, albeit controlled by gate–source voltage V_{gs}, in shunt with a resistive branch. To wit, Equation 1.116 can be written as

$$I_d \approx I_{dsat} + \frac{V_{ds} - V_{dsat}}{V_\lambda / I_{dsat}}, \tag{1.117}$$

which suggests the static circuit model provided in Figure 1.30. The subject model is more useful conceptually than computationally since a change made to V_{gs} for the purpose of adjusting the nominal drain current, I_{dsat}, influences the resistance value, V_λ/I_{dsat}, and the voltage offset, V_{dsat}, introduced in the drain–source port.

A complication of the channel length embellishment to the saturation drain current expression is that Equation 1.116 is discontinuous with the ohmic domain drain current expression in Equation 1.105 at the transition boundary between respective operating domains. Simple software fixes in commonly

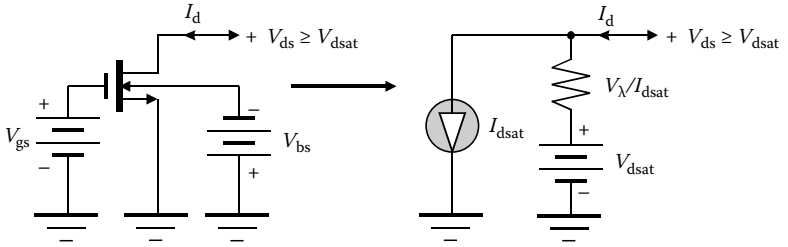


FIGURE 1.30 A large-signal circuit model for an N-channel MOSFET biased to operate in its saturation domain.

available circuit simulators rectify this incongruity. From an analytical perspective, the problem can be tacitly ignored, if V_λ in Equation 1.114 abides by the previously disclosed channel length and voltage restrictions.

For the convenience of the reader, the relevant expressions for the volt–ampere characteristic curves of an N-channel MOSFET are synopsized herewith. In particular,

$$I_d \approx \begin{cases} 0, & V_{gs} < V_h \\ K_n\left(\frac{W}{L}\right) V_{ds}\left(V_{gs} - V_h - \frac{V_{ds}}{2}\right), & V_{gs} \geq V_h;\ V_{ds} < V_{dsat} \\ \frac{K_n}{2}\left(\frac{W}{L}\right)\left(V_{gs} - V_h\right)^2\left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right), & V_{gs} \geq V_h;\ V_{ds} \geq V_{dsat} \end{cases}, \tag{1.118}$$

where V_{dsat} is the voltage difference, $(V_{gs} - V_h)$. It is to be understood that the positive reference direction of the drain current in NMOS is a current flowing into the drain, while the positive reference voltage polarities reflect those highlighted in Figure 1.17. Moreover, V_h is recalled as a threshold level dependent on the bulk–source voltage, V_{bs}, in accordance with Equation 1.96. A representative plot of the static volt–ampere characteristics of an NMOS transistor appear in Figure 1.31.

In the interests of clarity and completeness, the PMOS counterpart to Equation 1.118 is

$$I_d \approx \begin{cases} 0, & V_{sg} < V_h \\ K_p\left(\frac{W}{L}\right) V_{sd}\left(V_{sg} - V_h - \frac{V_{sd}}{2}\right), & V_{sg} \geq V_h;\ V_{sd} < V_{dsat} \\ \frac{K_p}{2}\left(\frac{W}{L}\right)\left(V_{sg} - V_h\right)^2\left(1 + \frac{V_{sg} - V_{dsat}}{V_\lambda}\right), & V_{sg} \geq V_h;\ V_{sd} \geq V_{dsat} \end{cases}, \tag{1.119}$$

where, in terms of the source–gate voltage, V_{sg}, V_{dsat} is now given by, $(V_{sg} - V_h)$, the threshold voltage, which is dependent on source–bulk voltage V_{sb} in Equation 1.96, remains a positive number, and

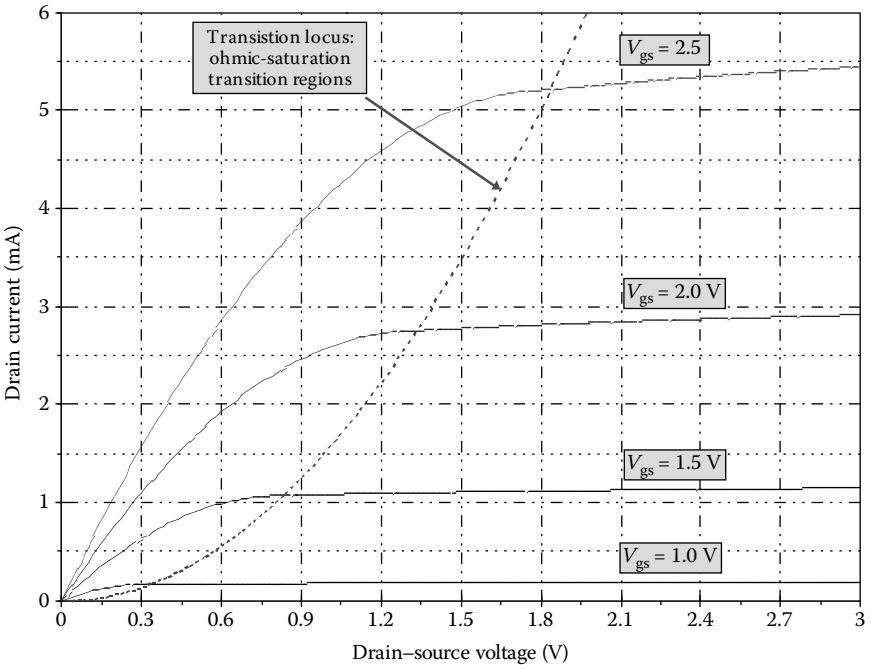


FIGURE 1.31 Common-source volt–ampere characteristic curves for an NMOS transistor.

transconductance parameter K_p is now the product of oxide capacitance density and hole mobility. The positive reference direction of the drain current in PMOS is a current flowing out of the drain, while the positive reference voltage polarities pertain to those delineated in Figure 1.18.

1.2.3.4 Refinements to the Static Model

The static volt–ampere characteristic in Equations 1.118 and 1.119 exhibit observable errors when computed currents are compared to experimental measurements executed on deep submicron MOSFET technology transistors. The principle source of these errors is two types of mobility degradation to which carriers in the source-to-drain channel are subjected. The first form of mobility degradation derives from the large lateral electric fields evidenced when even relatively small drain-to-source voltages are applied across channels whose lengths are smaller than approximately 0.25 μ. The second form of mobility impairment is caused by the strong vertical electric fields established by gate–source voltages applied across thin oxide layers.

1.2.3.4.1 Lateral Electric Fields

The NMOS and PMOS volt–ampere characteristic equations in Equations 1.118 and 1.119 are predicated on the presumption that the drift velocity, say v_c, of carriers propagated through the inverted channel at the oxide–semiconductor interface is proportional to the lateral electric field, $E_x[\varphi_c(x)]$. This field is, of course, established in the channel by applied drain–source voltage, V_{ds}, (in the case of NMOS) or applied source–drain voltage V_{sd} (in the case of PMOS). In particular,

$$v_c = \mu_o |E_x|, \tag{1.120}$$

where μ_o represents either the low field value of the electron mobility, μ_n, in N-channel devices or the low field value of the hole mobility, μ_p, in PMOS. The simpler notation, E_x is adopted in Equation 1.120 to represent the potential-dependent field function, $E_x[\varphi_c(x)]$. The need for the absolute value operation on the right-hand side of Equation 1.120 materializes from the fact that the carrier velocity, which is always a positive metric, is directed against the direction of the channel field in NMOS. In the case of NMOS transistors, carriers drift in the direction of the source-to-the drain, whereas the field is directed from drain-to-source and is therefore negative. For PMOS, no algebraic sign problems are manifested, since carriers drift in the same direction as the lateral field, whence E_x is positive.

The simplicity of Equation 1.120 belies the fact that the carrier drift velocity does not continually increase in proportion to the electric field. In fact, the carrier velocity saturates at a value, say v_{max}, which is of the order of 0.15 μm/ps in silicon, when electric fields are excessive. In recognition of this physical phenomenon, Equation 1.120 is supplanted by the empirical relationship,

$$v_c = \frac{\mu_o |E_x|}{1 + |E_x|/E_c}, \tag{1.121}$$

where

$$E_c = \frac{v_{max}}{\mu_o} \tag{1.122}$$

is termed the *critical electric field*. Typically, E_c is in the range of 3–5 V/μm. A comparison of Equation 1.121 with Equation 1.120 suggests an effective mobility, μ_e, of

$$\mu_e = \frac{\mu_o}{1 + |E_x|/E_c}. \tag{1.123}$$

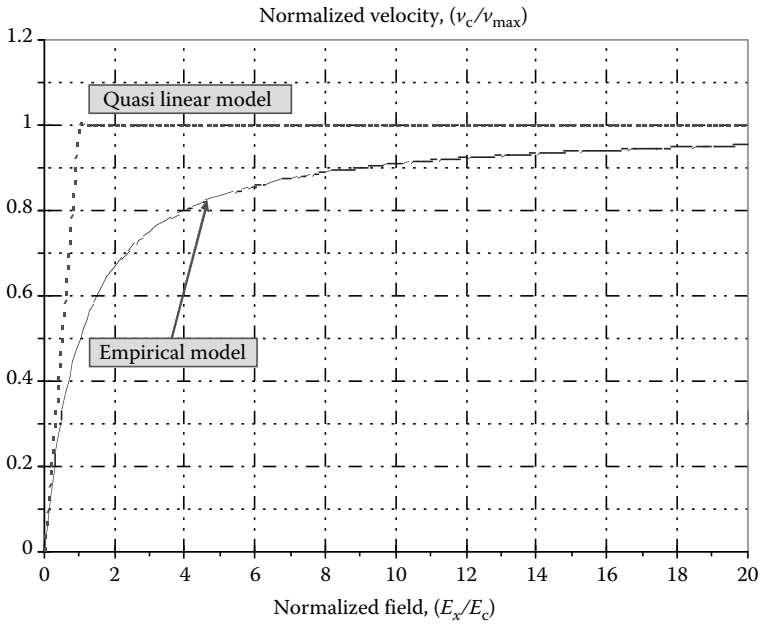


FIGURE 1.32 The dependence of carrier velocity on electric field in a semiconductor. The dashed curve represents the elementary low field approximation to the velocity-field relationship.

The mobility degradation inferred by the last disclosure bodes potentially decreased frequency response attributes of considered transistors, since the less mobile free electrons are in the inverted channel, the longer is the average time required for their transport from the source-to-the drain. Figure 1.32 sketches the velocity-field relationship implied by Equation 1.121. Note in this plot that the linear, or low field, approximation to the velocity characteristic is reasonably accurate up to about only 30% of the saturated limited velocity.

The effect on the ohmic regime drain current of the mobility degradation incurred by strong lateral electric fields can be studied by returning to Equation 1.104 and replacing the electron mobility, μ_n, therein by an adjusted mobility, μ_{ne}, such that

$$\mu_{ne} = \frac{\mu_n}{1 - E_x/E_c} = \frac{\mu_n}{1 + \frac{1}{E_c}\frac{d\varphi_c(x)}{dx}}, \tag{1.124}$$

where Equation 1.100 is applied and μ_n is understood to be the low field value of electron mobility in the inverted source-to-drain channel. Equation 1.104 becomes

$$I_d = \frac{\mu_n C_{ox} W\left[V_{gs} - V_h - \varphi_c(x)\right]}{1 + \frac{1}{E_c}\frac{d\varphi_c(x)}{dx}}\frac{d\varphi_c(x)}{dx}, \tag{1.125}$$

which leads to

$$I_d\left[\int_0^L dx + \frac{1}{E_c}\int_0^{V_{ds}} d\varphi_c(x)\right] = \mu_n C_{ox} W \int_0^{V_{ds}} \left[V_{gs} - V_h - \varphi_c(x)\right] d\varphi_c(x). \tag{1.126}$$

The requisite integrations produce

$$I_{\mathrm{d}} = K_n\left(\frac{W}{L}\right)\left[\frac{V_{\mathrm{ds}}\left(V_{\mathrm{gs}} - V_{\mathrm{h}} - \frac{V_{\mathrm{ds}}}{2}\right)}{1 + \frac{V_{\mathrm{ds}}}{V_{\mathrm{le}}}}\right], \tag{1.127}$$

where

$$V_{\mathrm{le}} = E_{\mathrm{c}}L = \left(\frac{v_{\max}}{\mu_n}\right)L \tag{1.128}$$

might be termed the *lateral electric field modulation voltage*. Observe that Equation 1.127 differs from the ohmic region volt–ampere relationship in Equation 1.118 by only the dimensionless factor in the denominator on the right-hand side of Equation 1.127. Appealing to Equation 1.128, this factor is seen to approach one when the channel length, L, is long. Of course, the subject factor also tends toward unity if the drain–source voltage, V_{ds}, is small. The latter point reflects engineering expectations in that small V_{ds} incurs lateral electric fields that are small enough to minimize field-induced mobility degradation.

A complication spawned by Equation 1.127 is that it no longer delivers the simple relationship for the drain saturation voltage witnessed in Equation 1.99. By definition, the drain saturation voltage, V_{dsat}, is the value of the drain–source voltage, V_{ds}, for which the slope of the ohmic regime I_{d} versus V_{ds} characteristic is zero. An application of this definition to Equation 1.127 leads to the revised drain saturation voltage,

$$V_{\mathrm{dsat}} = M_{\mathrm{sat}}\left(V_{\mathrm{gs}} - V_{\mathrm{h}}\right), \tag{1.129}$$

where, with

$$\alpha \triangleq \frac{V_{\mathrm{gs}} - V_{\mathrm{h}}}{V_{\mathrm{le}}}, \tag{1.130}$$

$$M_{\mathrm{sat}} = \frac{\sqrt{1 + 2\alpha} - 1}{\alpha}. \tag{1.131}$$

It can be demonstrated that $M_{\mathrm{sat}} \leq 1$ for $\alpha \geq 0$ and thus, an impact of carrier mobility degradation incurred by strong lateral fields in the inverted channel is a decrease in the low field value of the drain saturation voltage. While mobility degradation is generally an undesirable phenomenon, the drain saturation voltage decrease is actually good news in low-voltage applications that require MOSFETs to function in their saturated regimes.

The drain saturation current corresponding to the revised estimate of the drain saturation voltage can be determined by substituting Equation 1.129 into 1.127. This activity produces the aesthetically pleasing result,

$$I_{\mathrm{dsat}} = \frac{K_n}{2}\left(\frac{W}{L}\right)V_{\mathrm{dsat}}^2 = \frac{K_n}{2}\left(\frac{W}{L}\right)M_{\mathrm{sat}}^2\left(V_{\mathrm{gs}} - V_{\mathrm{h}}\right)^2. \tag{1.132}$$

In the limit of large channel lengths, V_{le} in Equation 1.128 is large, thereby rendering parameter α in Equation 1.130 small. But for very small α, M_{sat} in Equation 1.131 approaches unity. It is therefore reassuring that in the limit of large channel lengths, which are incapable of supporting large electric fields in the inverted channel, I_{dsat} in Equation 1.132 collapses to Equation 1.107, a relationship that implicitly

reflects tacit neglect of field-induced carrier mobility degradation. In contrast, very small channel lengths give rise to small V_{le} and large α, whence M_{sat} in Equation 1.131 reduces to

$$M_{sat}\big|_{\text{small L}} = \frac{\sqrt{1+2\alpha}-1}{\alpha}\bigg|_{\text{large }\alpha} \approx \sqrt{\frac{2}{\alpha}}. \tag{1.133}$$

Upon combining the last result with Equation 1.132, the short channel value of I_{dsat} is found to be

$$I_{dsat}\big|_{\text{small L}} \approx WC_{ox}v_{max}\left(V_{gs}-V_h\right), \tag{1.134}$$

where Equations 1.128 and 1.106 are exploited. Observe that the resultant drain saturation current is independent of the channel length, L. This independence stems from the fact that in the limit of very small channel lengths, carriers (electrons in the present case of an NMOS transistor) are transported through the inverted channel at their saturated limited, or maximum, velocity. This maximum velocity of carrier propagation renders L inconsequential with respect to the average time of carrier transport from the source region-to-the drain region. But perhaps the most interesting aspect of Equation 1.134 is that the short channel drain saturation current is a linear function of the gate–source voltage, V_{gs}. The linearity posed by Equation 1.134 is an obvious advantage for most analog signal processing applications, but achieving the velocity saturation implicit to this observed linearity may present voltage biasing challenges.

Of course, Equation 1.129 through 1.133 apply to the saturation regime of device operation in that in saturation, the drain current is merely the transistor current, I_{dsat}, evidenced at the boundary of ohmic and saturation regimes, corrected by channel length modulation effects. To wit, short channel phenomena imply that for $V_{gs} \geq V_h$ and $V_{ds} \geq V_{dsat}$,

$$I_d = \frac{K_n}{2}\left(\frac{W}{L}\right)M_{sat}^2\left(V_{gs}-V_h\right)^2\left(1+\frac{V_{ds}-V_{dsat}}{V_\lambda}\right), \tag{1.135}$$

where it is essential to remember that the drain saturation voltage, V_{dsat}, is now given by Equation 1.129. It is clear that M_{sat} in Equation 1.129 is properly viewed as a drain saturation voltage correction factor in a short channel (indeed, deep submicron) environment. Because of Equation 1.135, the square of M_{sat} can be accorded the stature of a current correction factor pertinent to short channel drain currents in the saturation regime. The dependence on parameter α of these correction factors is displayed in the plots submitted in Figure 1.33. The indicated correction factors are significant. For example, consider $\alpha = 2$, which might typically represent a gate–source voltage, V_{gs}, that is about a volt over the threshold potential. The curves in the figure at hand suggest an approximate 38% reduction in the drain saturation voltage predicted by the simple long channel model, which corresponds to $\alpha = 0$, as well as about a 62% attenuation of the corresponding drain saturation current.

Although Equations 1.135 and 1.129 are analytically elegant, their utility in a design-oriented environment is questionable in light of the dependence of factor M_{sat} on parameter α set forth by Equation 1.131. In light of this dilemma, an approximate curve fit of both M_{sat} and its square is judicious from an engineering design perspective. A numerical study of Equation 1.131 reveals that the empirical approximation,

$$M_{sat} = \frac{\sqrt{1+2\alpha}-1}{\alpha} \approx 1-\frac{\sqrt{\alpha}}{4}, \tag{1.136}$$

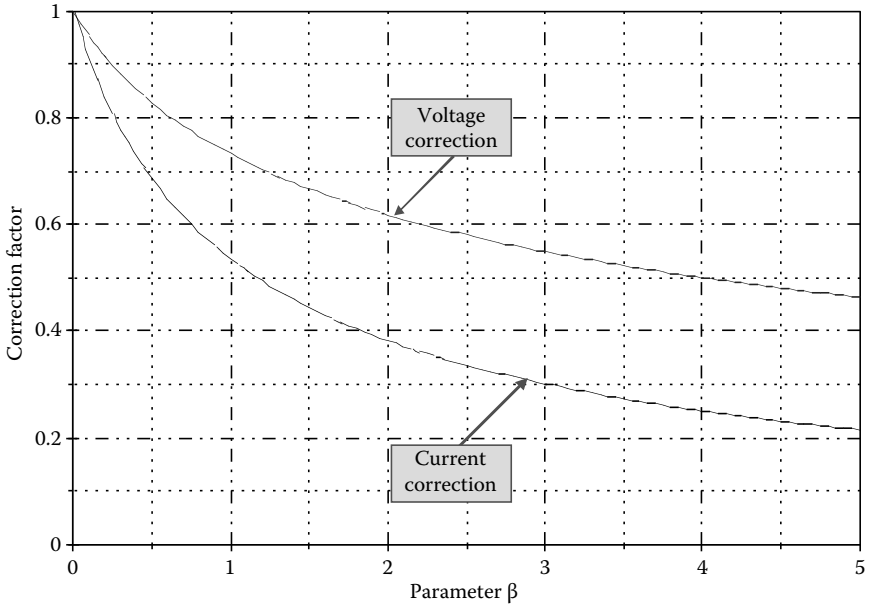


FIGURE 1.33 Voltage and current correction factors precipitated by large lateral electric fields in short channel MOSFETs. The parameter, α, is the effective gate–source voltage, $(V_{gs} - V_h)$, normalized to the lateral electric field modulation voltage, V_{le}.

results in an error of at most 4.8% for $0 \leq \alpha \leq 5$. On the other hand, a similar numerical exercise produces

$$M_{sat}^2 = \left(\frac{\sqrt{1+2\alpha}-1}{\alpha}\right)^2 \approx \frac{1}{1+0.78\alpha} \tag{1.137}$$

to a computational error of at most 5.1% for $0 \leq \alpha \leq 5$. For most design-oriented purposes, Equation 1.129 can therefore be supplanted by

$$\begin{aligned} V_{dsat} = M_{sat}(V_{gs} - V_h) &\approx \left(1 - \frac{\sqrt{\alpha}}{4}\right)(V_{gs} - V_h) \\ &= \left(1 - \frac{1}{4}\sqrt{\frac{V_{gs} - V_h}{V_{le}}}\right)(V_{gs} - V_h), \end{aligned} \tag{1.138}$$

while Equation 1.135 becomes for circuit design applications of MOSFETs operated in saturated regimes,

$$\begin{aligned} I_d &= \frac{K_n}{2}\left(\frac{W}{L}\right)M_{sat}^2(V_{gs} - V_h)^2\left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right) \\ &\approx \frac{K_n}{2}\left(\frac{W}{L}\right)(V_{gs} - V_h)^2\left(\frac{1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}}{1 + 0.78\alpha}\right) \\ &= \frac{K_n}{2}\left(\frac{W}{L}\right)(V_{gs} - V_h)^2\left(\frac{1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}}{1 + 0.78\left(\frac{V_{gs} - V_h}{V_{le}}\right)}\right). \end{aligned} \tag{1.139}$$

The academic purist who may understandably balk at the foregoing numerical empiricisms is respectfully reminded that the mobility expression in Equation 1.123 and the "long channel" velocity relationship of Equation 1.120 are hardly grounded in sound physical phenomenology. Moreover, it is interesting to note that of the more than 275 parameters indigenous to the commonly exploited Level 49 HSPICE model of a MOSFET, most are curve fit disclosures that bear no clarion relationship to the physical charge storage and charge transport mechanisms that underpin the volt–ampere characteristics of a MOSFET.

1.2.3.4.2 Vertical Electric Fields

Apart from the carrier mobility degradation incurred by strong lateral fields in the inverted channel of a MOSFET, mobility is impacted by the vertical electric field resulting from the applied effective interface potential, $(V_{gs} - V_h)$, in the case of NMOS or $(V_{sg} - V_h)$ for PMOS devices. In NMOS, increases in V_{gs} strengthens this vertical field so that free electrons transported from the source-to-the drain are encouraged to drift ever closer to the oxide–semiconductor interface. Unfortunately, the interface is far from a perfectly smooth boundary, if for no other reason than routine device processing invariably produces ionic contamination therein. The imperfect boundary causes potentially significant carrier scattering, which in turn results in diminished carrier mobility.

To first order, the mobility attenuation resulting from increased gate overdrive can be addressed analytically by replacing the low field mobility, μ_n (for NMOS), to which K_n in Equation 1.139 is directly proportional, by an effective carrier mobility, μ_{eff}, such that

$$\mu_{eff} = \frac{\mu_n}{1 + \frac{V_{gs} - V_h}{V_{ve}}}. \tag{1.140}$$

In this expression, V_{ve} is the *vertical electric field modulation voltage*, which is nominally directly proportional to the thickness, T_{ox}, of the oxide layer. Of course, an expression analogous to Equation 1.140 prevails for hole mobility in the inverted channel of PMOS transistors. To a very rough approximation,

$$V_{ve} = T_{ox}/15, \tag{1.141}$$

where T_{ox} in units of angstroms returns V_{ve} in units of volts. Because of Equations 1.140 and 1.139 for the saturation domain current becomes

$$\begin{aligned} I_d &= \frac{K_n}{2}\left(\frac{W}{L}\right) M_{sat}^2 \frac{\left(V_{gs} - V_h\right)^2}{\left(1 + \frac{V_{gs} - V_h}{V_{ve}}\right)} \left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right) \\ &\approx \frac{K_n}{2}\left(\frac{W}{L}\right) \frac{\left(V_{gs} - V_h\right)^2}{\left(1 + \frac{V_{gs} - V_h}{V_{ve}}\right)} \left(\frac{1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}}{1 + 0.78\left(\frac{V_{gs} - V_h}{V_{le}}\right)}\right). \end{aligned} \tag{1.142}$$

An analogous modification, which amounts to an effective reduction of the transconductance parameter, K_n, can be made to the ohmic domain current.

Obviously, Equation 1.142 is inordinately more cumbersome than is the simple, square law, volt–ampere characteristic advanced by Equation 1.110 for device operation in the saturation domain. As a result, the design-oriented determination of a suitable gate–source voltage for a desired drain current and corresponding drain–source voltage can be a daunting challenge. But in addition to the computational problems precipitated merely by algebraic complexity, engineering difficulties are additionally encountered with respect to the accurate numerical delineation of the model metrics, K_n, V_h, V_{ve}, V_λ, and V_{le}. These

latter difficulties derive from the unfortunate fact that the physical device and charge transport properties (saturation velocity, carrier mobility, regional concentrations, etc.) on which these and other model parameters depend are invariably unavailable to the circuit designer. At best, the circuit designer can reasonably expect to have presumably reliable, detailed device model parameters suitable for computer-aided simulation of transistor performance. For example, process foundries routinely supply their customers with device models in the form of Level 49 HSPICE or other computer-based files. Unfortunately, many, if not most, of the hundreds of numerical entries indigenous to these files are themselves nonphysical entities that defy satisfying mathematical relationships to the physical model metrics discussed in earlier paragraphs. These and related other design-oriented problems can prove exasperating. The aforementioned issues are best mitigated by coalescing manual design strategies and calculations with suitable computer-based simulations of device properties and volt–ampere characteristics.

1.2.3.5 Temperature Effects

The operating temperature of the inverted interfacial channel affects the drain current of a transistor in three ways. First, because thermal energy imparted to free carriers increases their scattering, the carrier mobility decreases in response to increased operating temperatures. To first order, the electron mobility, $\mu_n(T)$, at absolute temperature T relates to the mobility, $\mu_n(T_o)$, at a reference temperature, T_o, in accordance with the three-halves power law,

$$\mu_n(T) = \mu_n(T_o)\left(\frac{T_o}{T}\right)^{3/2}. \tag{1.143}$$

Because parameter K_n in Equation 1.142 is directly proportional to carrier mobility, Equation 1.143 implies that the drain current of a MOSFET is characterized by a negative temperature coefficient, that is, the drain current, I_d, decreases with increasing operating temperature.

A second effect of increased thermal energy is a perturbation of threshold voltage. A computation of this perturbation is best initiated by returning to Equation 1.96 to evaluate the derivative of the threshold voltage, V_h, with respect to the Fermi potential, V_F. Recalling Equations 1.96 and 1.97, and noting that the body effect voltage, V_θ, in Equation 1.90 is independent of temperature,

$$\frac{dV_h}{dV_F} = 2 + \sqrt{\frac{V_\theta}{V_F}} + \frac{V_h - V_{ho}}{2V_F} + \left(\frac{V_\theta}{2V_F}\right)\left(\frac{V_{bs}}{V_h - V_{ho} + 2\sqrt{V_\theta V_F}}\right), \tag{1.144}$$

where V_{ho} is recalled as signifying the zero bias ($V_{bs} = 0$) value of the threshold potential. Note that the last two terms on the right-hand side of this expression vanish when a MOSFET is operated with $V_{bs} = 0$. The sensitivity of the threshold voltage with respect to temperature follows as

$$\frac{dV_h}{dT} = \frac{dV_h}{dV_F} \times \frac{dV_F}{dT}. \tag{1.145}$$

The temperature derivative of the Fermi potential derives from Equation 1.55, with the proviso that due account be made of the temperature dependence of the intrinsic carrier concentration, N_i. To this end, a commonly used empiricism is

$$N_i = N_{io} 2^{(T - T_o)/T_n}, \tag{1.146}$$

where T_n is generally taken to be 10°C and, assuming the reference temperature, T_o, is 27°C, N_{io}, the intrinsic carrier concentration at $T = T_o$, is the previously used number, $(1.45)(10^{10})$ atoms/°C. With

$T_n = 10°C$, Equation 1.146 allows N_i to double for each 10°C rise above the reference temperature. Armed with Equations 1.146 and 1.55 produces

$$\frac{dV_F}{dT} = \frac{V_F}{T} - \frac{V_T}{T_n} \ln 2. \tag{1.147}$$

Equations 1.144 and 1.147 combine to yield the final result,

$$\frac{dV_h}{dT} = \left[2 + \sqrt{\frac{V_\theta}{V_F}} + \frac{V_h - V_{ho}}{2V_F} + \left(\frac{V_\theta}{2V_F}\right)\left(\frac{V_{bs}}{V_h - V_{ho} + 2\sqrt{V_\theta V_F}}\right)\right]\left(\frac{V_F}{T} - \frac{V_T}{T_n} \ln 2\right), \tag{1.148}$$

where parameters V_{ho}, V_h, and V_T, are computed at the reference temperature, T_o. The indicated temperature derivative of the threshold voltage is invariably a positive number in the range of 1.5–2.4 mV/°C. Thus, the threshold voltage increases with increasing operating temperatures, thereby leading to a decrease in the drain current. In other words, the temperature dependence of both the carrier mobility and the threshold voltage conduce a drain current exuding a negative temperature coefficient.

The algebraic form of Equation 1.148 is thoroughly depressing and is hardly a relationship stored in the human memories of circuit designers. Fortunately, for MOSFETs featuring thin gate oxides (under 50 Å) and substrate doping concentrations no smaller than 10^{15} atoms/cm^3, the terms in V_θ, $(V_h - V_{ho})$, and V_T/T_n are generally negligible, especially if the bulk–source bias, V_{bs}, is no more negative than 1.5 V. In this event,

$$\frac{dV_h}{dT} \approx \frac{2V_F}{T}, \tag{1.149}$$

which can be shown to be always larger—generally by no more than 5% or 6%—than the result predicted by Equation 1.148. For a substrate doping concentration of $N_A = (5)(10^{15})$ atoms/cm^3, Equation 1.149 predicts a threshold voltage sensitivity at $T = 27°C = 300.16$ K of 2.2 mV/°C.

1.2.4 Transistor Capacitances

At this juncture, the volt–ampere characteristic equations given by Equations 1.118, 1.119, 1.139, and 1.142 pertain to MOSFETs operated exclusively under static or low-frequency signal conditions. Specifically, the drain currents predicted by these relationships are unrealistically cavalier in that they respond instantaneously to applied gate–source, drain–source, and bulk–source excitations. When high-frequency signals are applied, the current responses are slowed by device capacitances arising from the charge storage that prevails in the inverted channel and within the depletion regions formed about the source and drain diffusions or implants. The engineering implications of this inherent inability of drain currents to respond instantly to signal excitations are MOSFET circuits exuding constrained bandwidths, nonzero input/output (I/O) delays and phase shifts, and nonzero rise and fall times in transient responses. In extreme cases, the interaction of these device capacitances with the energy storage elements of the peripheral circuit can produce excessive response peaking in either the frequency or time domains and even outright instability.

1.2.4.1 Depletion Capacitances

The first of the two principle sources of transistor capacitances is the depletion capacitance indigenous to both of the PN junctions formed respectively between the bulk and drain and between the bulk and source. In turn, each of these two transition region capacitances consists of a planar component and a peripheral, or sidewall, component. The planar component embodies the depletion layer established

between the bulk substrate region and the underside of the source and drain regions. On the other hand, the sidewall capacitance embraces the depletion layers in the areas of the source and drain regions that are proximate to the front surface, the back surface, and the side surface area adjacent to the active channel region. For the bulk–drain depletion capacitance, C_{bd},

$$C_{bd} = \frac{A_d C_j}{\left(1 - \frac{V_{bd}}{V_j}\right)^{M_j}} + \frac{P_d C_{jsw}}{\left(1 - \frac{V_{bd}}{V_j}\right)^{M_{jsw}}}, \tag{1.150}$$

where the forms of each of the terms on the right-hand side are observed to mirror the traditional depletion capacitance associated with a back biased PN junction. In Equation 1.150, C_j is the zero bias (meaning, $V_{bd} = 0$), value of the capacitance density, in units of farads/meter^2, associated with the planar component of the bulk–drain capacitance, while C_{jsw} is the zero bias lineal capacitance, in units of farads/meter, of the aforementioned sidewall areas. The planar drain area, A_d, is

$$A_d = WL_{dif}, \tag{1.151}$$

where L_{dif} is recalled in Figures 1.17 and 1.18 to represent the width of the drain region, which is generally identical to the width of the source implant. Generally, the dimension, L_{dif}, must be extracted empirically from measured data but as a rule of thumb, L_{dif} is nominally of the order of twice the channel length, L. Parameter P_d, is the effective length of the perimeter of the sidewall area and is stipulated by

$$P_d = W + 2L_{dif}. \tag{1.152}$$

Voltage V_j in Equation 1.150 is the built-in potential given by Equation 1.115, while M_j and M_{jsw} are the grading coefficients of the planar and sidewall PN junctions, respectively.* Typically $M_j = 0.5$ and $M_{jsw} = 0.33$. An analogous expression, whose terms convey equally analogous engineering interpretations, prevails for the net bulk–source depletion capacitance, C_{bs}. In particular,

$$C_{bs} = \frac{A_s C_j}{\left(1 - \frac{V_{bs}}{V_j}\right)^{M_j}} + \frac{P_s C_{jsw}}{\left(1 - \frac{V_{bs}}{V_j}\right)^{M_{jsw}}}, \tag{1.153}$$

where in general,

$$\left.\begin{aligned} A_s &\equiv A_d = WL_{dif} \approx 2WL \\ P_s &\equiv P_d = W + 2L_{dif} \approx W + 4L \end{aligned}\right\}. \tag{1.154}$$

1.2.4.2 Gate Capacitances

The second source of MOSFET capacitances is the gate capacitance, which itself is comprised of three distinct components. The first of these components appears between the gate and the bulk substrate. As is apparent from Figure 1.23, this particular capacitance has a very small nonzero frequency value in both weak and strong channel inversion modes, which suggests that the channel inversion layer effectively shields the gate from the bulk substrate. Because the gate–bulk capacitance is invariably very small, it bodes little consequence to achievable MOSFET circuit performance and therefore, it is usually ignored tacitly.

* In HSPICE and other forms of SPICE simulators, the built-in potential, V_j, is symbolized by P_B.

The other two components of net gate capacitance are the gate–source capacitance, C_{gs}, and the gate–drain capacitance, C_{gd}. Each of these energy storage elements is a superposition of an intrinsic module, which derives from the gate, gate oxide, and inverted channel, and an extrinsic constituent, which is attributed to gate oxide overlap at the source and drain sites. Since the inversion layer extends from source-to-drain in only the ohmic regime of operation, different values of these two capacitances prevail for ohmic and saturated operation. The maximum possible intrinsic capacitance established between the gate and the inversion layer is clearly WLC_{ox}. In the ohmic operating regime, this maximum capacitance is partitioned equally between the source and the drain to give identical intrinsic gate–source and gate–drain capacitance values; namely, $WLC_{ox}/2$. Accordingly, in the ohmic regime, the effective gate–source capacitance is

$$C_{gs} = \frac{WLC_{ox}}{2} + WC_{gso}, \tag{1.155}$$

where C_{gso} is the capacitance per unit length associated with the oxide–source overlap. Similarly, the effective gate–drain capacitance in the ohmic operating regime is

$$C_{gd} = \frac{WLC_{ox}}{2} + WC_{gdo}, \tag{1.156}$$

where C_{gdo} is the drain overlap capacitance counterpart to the source overlap region. Typically, C_{gso} and C_{gdo} are as small as 0.25 fF/μm in minimal geometry transistors. Thus, for a transistor characterized by $L = 180$ nm, $W/L = 20$, and an oxide thickness of $T_{ox} = 30$ Å, $C_{gs} = C_{gd} = 4.63$ fF. Observe herewith that the net overlap capacitance is $WC_{gso} = WC_{gdo} = 0.9$ fF, which is almost 20% of the total gate–source (or gate–drain) capacitance.

The capacitance situation in saturation is a bit more intricate than that which prevails in the ohmic regime. In saturated domains where $V_{ds} > V_{dsat}$, pinch off occurs within the source-to-drain channel, thereby leaving an effective depletion zone that is free of mobile carriers near the drain site. Accordingly, the drain–source voltage exerts no influence on the channel charge, and the resultant gate–drain capacitance derives exclusively from the oxide overlap with the drain, that is, the gate–drain capacitance, C_{gd}, in saturation is simply

$$C_{gd} = WC_{gdo}. \tag{1.157}$$

In contrast to the charge depletion prevailing in the channel region adjacent to the drain, a large free carrier population is concentrated near the source. Since this concentration is influenced strongly by interface potential, which is determined by the applied gate–source voltage, it is only logical to expect a comparatively substantial intrinsic gate–source capacitance.

An analytical disclosure of the foregoing gate-to-source capacitance commences with a return to Equations 1.102 and 1.103. If these two equations are combined and if Equation 1.63 is recalled,

$$q[Y_s(\varphi_c(x))][N_s(\varphi_c(x))] = C_{ox}\left[V_{gs} - V_h - \varphi_c(x)\right], \tag{1.158}$$

where the left-hand side of this relationship is understood to be the density of mobile charge in the inversion layer. Upon multiplication of both sides of Equation 1.158 by the gate width, W, the resultant left-hand side of the modified expression represents the net mobile charge per unit length of the inversion layer. It follows that the net differential mobile charge (amassed by electrons in NMOS), say $dq_n[\varphi_c(x)]$, contained in a differential channel volume of depth $Y_s[\varphi_c(x)]$, width W, and length extending from x to $(x + dx)$, is

$$dq_n(\varphi_c(x)) = WC_{ox}\left[V_{gs} - V_h - \varphi_c(x)\right]dx. \tag{1.159}$$

Ignoring mobility degradation incurred by lateral electric fields, Equation 1.104 can be used to recast Equation 1.58 in the form

$$\mathrm{d}q_n(\varphi_c(x)) = \frac{\mu_n(WC_{ox})^2}{I_d}\left[V_{gs} - V_h - \varphi_c(x)\right]^2 \mathrm{d}\varphi_c(x). \tag{1.160}$$

Equation 1.59 can be integrated conveniently from $\varphi_c(0)$ to $\varphi_c(V_{dsat})$, where the indicated interfacial potential limits correspond to the boundaries of the channel inversion layer evidenced in saturation. Such an integration of the left-hand side of Equation 1.59 brackets the net mobile charge, say $Q_n(V_{gs})$, observed in saturation for a stipulated gate–source voltage, V_{gs}. In particular,

$$\begin{aligned} Q_n(V_{gs}) &= \int_0^{V_{dsat}} \mathrm{d}q_n(\varphi_c(x)) = \frac{\mu_n(WC_{ox})^2}{I_d} \int_o^{V_{dsat}} \left[V_{gs} - V_h - \varphi_c(x)\right]^2 \mathrm{d}\varphi_c(x) \\ &= \frac{\mu_n(WC_{ox})^2}{3I_d}\left(V_{gs} - V_h\right)^3, \end{aligned} \tag{1.161}$$

where Equation 1.99 is exploited. Using Equation 1.109 to replace the drain current variable, I_d, in this relationship results in

$$Q_n(V_{gs}) = \frac{2}{3} WLC_{ox}\left(V_{gs} - V_h\right). \tag{1.162}$$

The saturation region value of the intrinsic gate–source capacitance now follows as

$$\frac{\mathrm{d}Q_n(V_{gs})}{\mathrm{d}V_{gs}} = \frac{2}{3} WLC_{ox}, \tag{1.163}$$

whereupon the saturation region value of the net gate–source capacitance, C_{gs}, inclusive of oxide overlap effects at the source site, is

$$C_{gs} = \frac{2}{3} WLC_{ox} + WC_{gso}. \tag{1.164}$$

1.2.4.3 Large-Signal Model

At this juncture, the large signal, or nonlinear, model of an N-channel MOSFET is the structure advanced in Figure 1.34. Depending on whether the transistor undergoing assessment is operated as an ohmic regime or as a saturated device, the equation for the indicated controlled current source, I_d, derives from expressions formulated in Section 1.2.3.2 or Section 1.2.3.3 or, for that matter, the model refinements addressed in Section 1.2.3.4. The depletion capacitances, C_{bs} and C_{bd}, are not affected by the domain of transistor operation, but the appropriate regional values of the capacitances, C_{gs} and C_{gd}, must be culled from the discourse in Section 1.2.4.2.

The model at hand also incorporates four resistive elements. The resistances, r_d and r_s, are respectively associated with the strongly doped drain and source regions, respectively. These resistances are specified in HSPICE simulation software by a sheet resistance parameter, R_{sh}, and drain and source geometric parameters, N_{rd} and N_{rs}. In particular,

$$\left.\begin{aligned} r_d &= N_{rd}R_{sh} \\ r_s &= N_{rs}R_{sh} \end{aligned}\right\}. \tag{1.165}$$

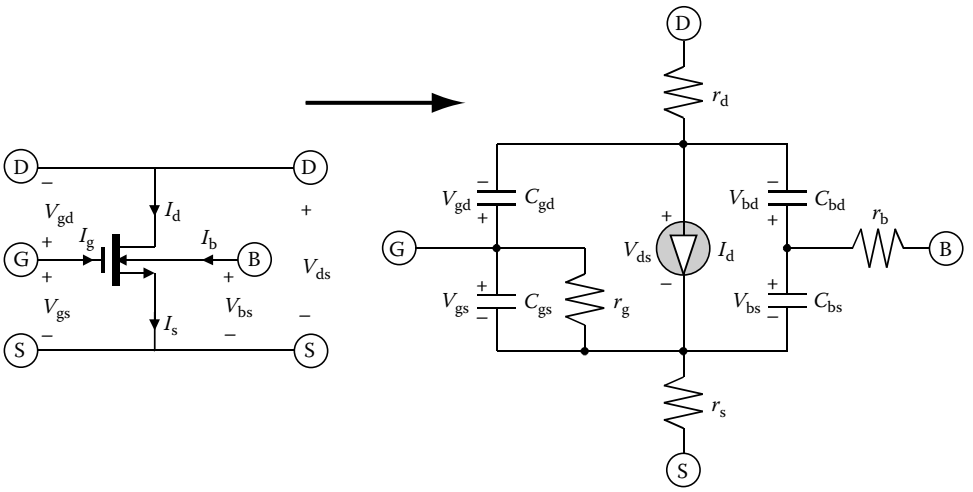


FIGURE 1.34 Large-signal model of an N-channel MOSFET. A topologically identical equivalent circuit prevails for P-channel MOSFETs.

Owing to the high doping concentrations of the drain and the source, which begets a small *sheet resistance* parameter, R_{sh}, resistances r_d and r_s are generally sufficiently small to justify their tacit neglect in most analog circuit applications. In contrast, resistance, r_b, which represents an effective spreading resistance in the bulk substrate, can be as large as the high tens to low hundreds of ohms. Despite its relatively large value, its impact on analog circuit performance is muted by the fact that the bulk rarely conducts significant currents, even at high signal frequencies. However, this resistance does influence the thermal noise characteristics of the drain–source channel.

Like resistance r_b, the gate resistance, r_g, is likewise important from a thermal noise perspective in that it captures the salient effects that thermally agitated mobile charge carriers exert on channel potential. It also looms significant with respect to design problems associated with maximum signal power transfer in radio frequency (RF) circuits [9]. This resistance is computed as [10]

$$r_g = \frac{5}{(\omega C_{gs})^2 R_{ch}}, \tag{1.166}$$

where R_{ch} represents the $V_{ds} = 0$ value of the drain–source channel resistance. Recalling Equation 1.109,

$$R_{ch} = \frac{1}{K_n\left(\frac{W}{L}\right)(V_{gs} - V_h)} = \frac{M_{sat} V_{dsat}}{2I_{dsat}}, \tag{1.167}$$

where Equations 1.129 and 1.132 have been used. Because of the inverse dependence of r_g on the square of radial signal frequency, ω, r_g is infinity for quiescent operating conditions and extremely large for low to even reasonably high frequencies.

1.2.5 Small-Signal Operation

As noted in Section 1.2.1, MOSFETs are the active device of choice in a plethora of high-performance analog integrated circuits. When the fundamental objective of these analog networks is linear I/O signal processing, each MOSFET therein is commonly biased in a saturated regime that ensures, for all applied signals of interest, an instantaneous drain–source voltage, v_{ds}, that is never any smaller than the

instantaneous drain–source saturation voltage, v_{dsat}. To be sure, linear signal processing can also be achieved when transistors operate in their ohmic regimes. But when high performance, in such senses as high gain, wide bandwidth, large dynamic range, and acceptable driving point I/O impedance levels, is a fundamental design objective, saturation is the regime of choice. Accordingly, ohmic linear equivalent circuits of transistors are ignored herewith and left as an investigation exercise for the reader.

A casual inspection of Equation 1.142 suggests that the instantaneous drain current, i_d, flowing in an N-channel MOSFET is a function of three device voltages: the instantaneous gate–source voltage, v_{gs}, the instantaneous drain–source voltage, v_{ds}, and the instantaneous bulk–source voltage, v_{bs} which covertly influences the threshold potential, V_h. An analogous statement applies to P-channel transistors, subject to the current and voltage conventions adopted earlier. Thus, Equation 1.142 can be generalized as

$$i_d \approx \frac{K_n}{2}\left(\frac{W}{L}\right)M_{sat}^2\left(v_{gs}-V_h\right)^2\left(\frac{1+\frac{v_{ds}-v_{dsat}}{V_\lambda}}{1+\frac{v_{gs}-V_h}{V_{ve}}}\right)=f\left(v_{gs},\, v_{ds},\, v_{bs}\right). \tag{1.168}$$

Under zero signal conditions, which is tantamount to operating the considered MOSFET at its quiescent operating point, it is understood that Equation 1.168 yields

$$I_d = f\left(V_{gs}, V_{ds}, V_{bs}\right), \tag{1.169}$$

where the indicated variables in capital letters designate static, or quiescent, device currents and voltages. In other words, the MOSFET described mathematically by Equation 1.169 is in a standby mode that awaits the application of dynamic, invariably time-varying, signals. Prior to signal excitation, the transistor maintains quiescent values of drain current, gate–source voltage, drain–source voltage, and bulk–source voltage that respectively equal I_d, V_{gs}, V_{ds}, and V_{bs}. Signals applied as a current, say I_{ds}, to the drain lead and/or a voltages, say V_1 to the gate–source port, V_2 to the bulk–source port, or V_3 to the drain–source port perturb the quiescent, or *Q-point*, counterparts of these electrical variables to deliver the observable net instantaneous current and voltage responses,

$$\left.\begin{aligned} i_d &= I_d + I_{ds} \\ v_{gs} &= V_{gs} + V_1 \\ v_{bs} &= V_{bs} + V_2 \\ v_{ds} &= V_{ds} + V_3 \end{aligned}\right\}. \tag{1.170}$$

In concert with these relationships, the MOSFET under consideration is said to operate linearly if and only if the signal-induced changes, I_{ds}, V_1, V_2, and V_3, interrelate linearly and if and only if the Q-point currents and voltages are independent of signal strengths. It is crucial to understand that operational linearity in an electronic device does not imply linear relationships among the instantaneous device variables, nor does it imply linearity among the corresponding quiescent values of these variables. Instead, operational linearity implies merely that a selected variable in the selected set of four perturbed variables in Equation 1.170 linearly superimpose with the remaining three electrical signal components.

1.2.5.1 Fundamental Small-Signal Model

Because of the obviously nonlinear nature of Equation 1.168, questions abound as to the plausibility of achieving the aforementioned linearity condition among the electrical perturbations induced by applied signals. Despite its inherently nonlinear nature, Equation 1.168 is a well-behaved functional relationship, which suggests that the desired linearity might be approximated adequately by limiting all signal excursions about respective operating point values to sufficiently small levels. This sufficiently small-signal mandate defines the concept of *small-signal analysis* and produces a *small-signal model* of the

MOSFET. A small-signal analysis reflective of a mathematical exploitation of the corresponding small-signal model are deemed both appropriate and useful if the retention of only the linear terms of the Taylor series expansion of Equation 1.168 about the operating point of the considered device leads to minimal errors in the resultant expression for the signal component of the drain current. Thus,

$$i_d \approx I_d + \left.\frac{\partial i_d}{\partial v_{gs}}\right|_Q (v_{gs} - V_{gs}) + \left.\frac{\partial i_d}{\partial v_{bs}}\right|_Q (v_{bs} - V_{bs}) + \left.\frac{\partial i_d}{\partial v_{ds}}\right|_Q (v_{ds} - V_{ds}), \tag{1.171}$$

where each of the three derivatives on the right-hand side of this relationship are evaluated at the *Q*-point of the MOSFET, that is, at $i_d = I_d$, $v_{gs} = V_{gs}$, $v_{bs} = V_{bs}$, and $v_{ds} = V_{ds}$. Using Equation 1.170 and noting that each of the three subject derivatives is a constant having units of conductance, Equation 1.171 can be couched in the form

$$I_{ds} \approx g_m V_1 + g_{mb} V_2 + \frac{V_3}{r_o}, \tag{1.172}$$

where

$$\left.\begin{aligned} g_m &\triangleq \left.\frac{\partial i_d}{\partial v_{gs}}\right|_Q \\ g_{mb} &\triangleq \left.\frac{\partial i_d}{\partial v_{bs}}\right|_Q \\ \frac{1}{r_o} &\triangleq \left.\frac{\partial i_d}{\partial v_{ds}}\right|_Q \end{aligned}\right\}. \tag{1.173}$$

Equation 1.71 gives rise to the small-signal, low-frequency equivalent circuit depicted in Figure 1.35a. The subject circuit becomes the small-signal, high-frequency MOSFET model if the four capacitances, C_{gs}, C_{gd}, C_{bd}, and C_{bs}, discussed in Section 1.2.4 are appended as indicated in Figure 1.35b. It is important to underscore the fact that either of the models in Figure 1.35 gives no information about the instantaneous electrical variables of a MOSFET, nor does either model allow for the computation of the quiescent values of these variables. Indeed, the models at hand require a priori knowledge of the *Q*-point since the small-signal parameters, g_m, g_{mb}, and r_o, depend on the operating point, as is implied by Equation 1.173. Moreover, the four capacitive elements in the model of Figure 1.35b likewise depend on the *Q*-point at which the considered transistor is biased. In short, the models in Figure 1.35 give first order approximations of the interrelationships among only the small-signal components of the net currents and voltages indigenous to a MOSFET. Although the topologies of both the models drawn in Figure 1.35 pertain to both the ohmic and saturation regimes of N-channel MOSFET operation, the equations to be developed shortly for the low-frequency parameters of these models apply exclusively to the saturation region. Moreover, while Figure 1.35 makes explicit reference to an N-channel transistor, or NMOS, the small-signal, low- and high-frequency equivalent circuits of PMOS units are identical to their NMOS counterparts. This topological identity stems from the fundamental fact that the small-signal models intertwine only signal-induced changes of device currents and voltages about their respective quiescent values. In an attempt to dispel possible confusion, the latter models are offered in Figure 1.36.

In the low-frequency models of either Figure 1.35a or Figure 1.36a the signal component of the bulk current, I_{bs}, flows into an open circuit because the bulk–drain and bulk–source junctions of devices embedded in analog networks are commonly reversed biased. The low-frequency signal component of the gate current, I_{gs}, is very nearly zero because the gate resistance, r_g, is, by Equation 1.166, inversely proportional to the square of the signal frequency. Of course, both of these currents are substantively

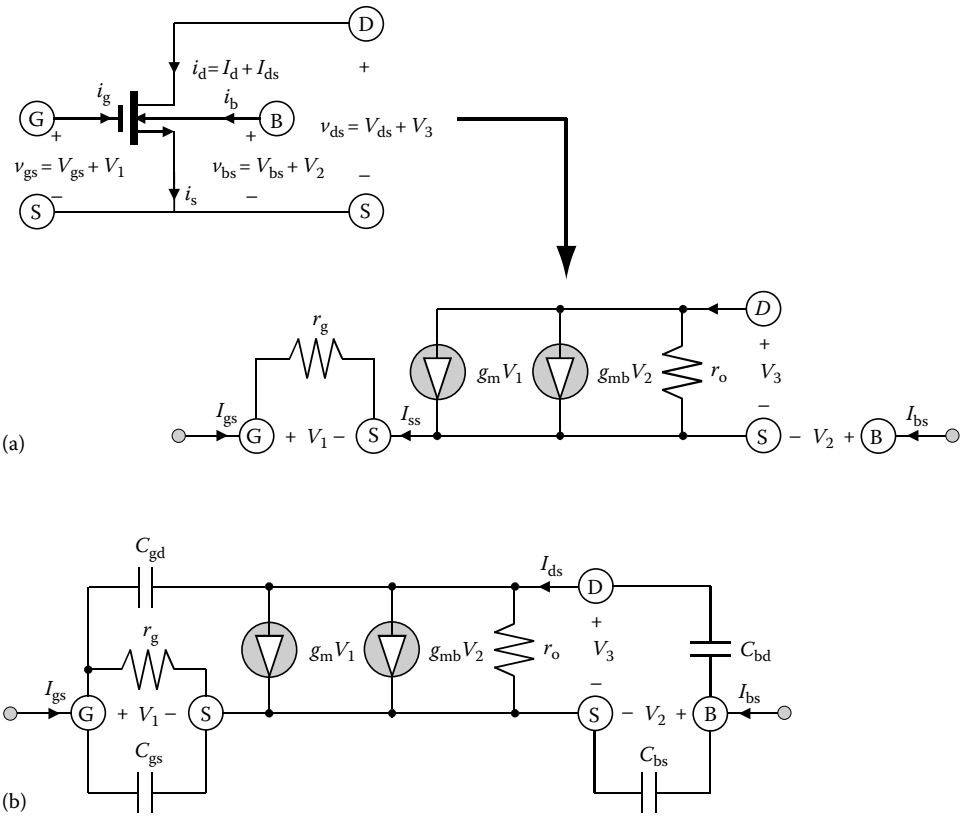


FIGURE 1.35 (a) Small-signal, low-frequency equivalent circuit of an N-channel MOSFET. (b) Small-signal, high-frequency equivalent circuit of an N-channel MOSFET. The topological structures of either model apply to both the ohmic and saturation regimes of operation.

larger at high frequencies where the various capacitances in the models of Figures 1.35b and 1.36b become poor approximations of the open circuits they mirror at low signal frequencies.

The parameter, g_m, is termed the *forward transconductance*. It is a critical analog circuit metric in that it serves as a measure of achievable forward gain. In particular, parameter g_m, when multiplied by the applied gate–source signal voltage, V_1, determines the amount of drain signal current, I_{ds}, manifested by the applied gate–source signal. On the other hand, the *bulk transconductance*, g_{mb}, measures the ability of a MOSFET to transfer applied bulk–source signal, V_2, to the drain signal current response. The controlled current, $g_{mb}V_2$, is negligible when it is much smaller than is its forward transconductance counterpart current, $g_m V_1$. However, it should be noted that depending on the selected quiescent operating point, parameter g_{mb} can be as much as 15%–25% of the forward transconductance, g_m. The current, $g_{mb}V_2$, is entirely inconsequential in numerous analog circuits that configure their utilized MOSFETs in such a way as to operate both the bulk and source terminals at signal ground, which renders $V_2 = 0$. Finally, r_o, the *drain–source channel resistance*, appears as a shunting resistive element across the drain and source terminals. If r_o is infinitely large (which, to be sure, it is not in practical MOSFETs) the drain-to-source small-signal port of a MOSFET behaves as an ideal Norton equivalent current source, that is, the current level determined largely by $g_m V_1$ is unaffected by modulations in the drain–source signal voltage, V_3. It follows that to the extent that the gate–source terminals serve as an input signal port boasting infinitely large impedance and the drain–source terminals function as the output port, the MOSFET emulates an ideal transconductance amplifier if the channel resistance, r_o, is large.

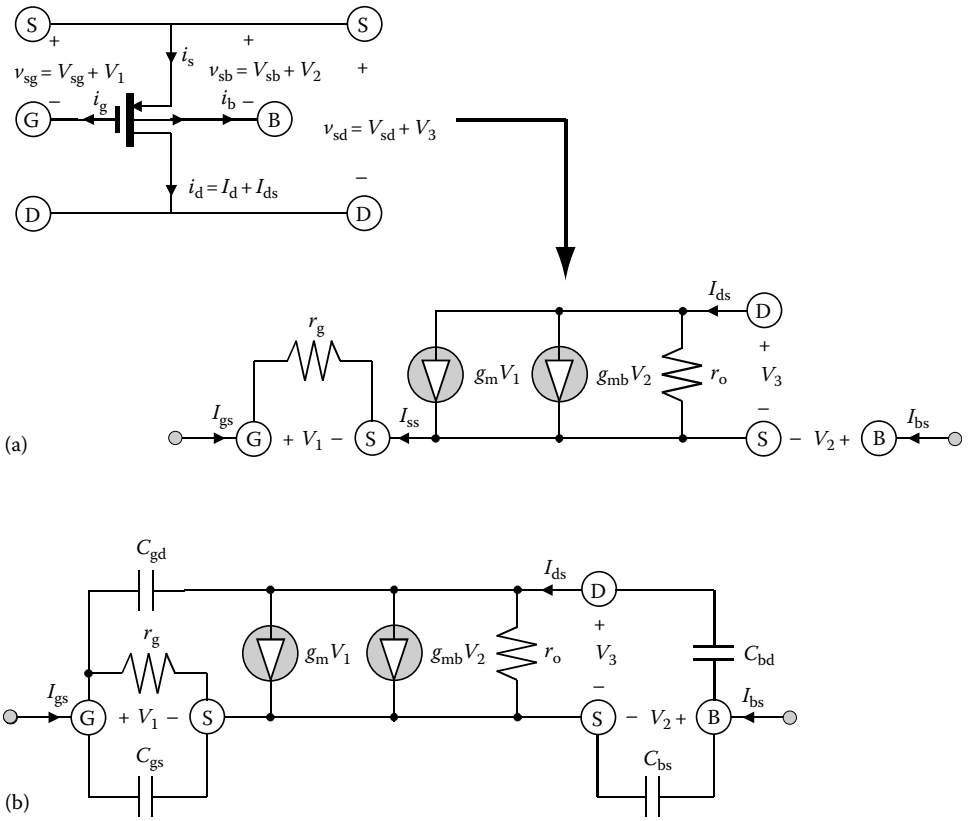


FIGURE 1.36 (a) Small-signal, low-frequency equivalent circuit of a P-channel MOSFET. (b) Small-signal, high-frequency equivalent circuit of a PMOS device.

The determination of the three low-frequency parameters defined in Equation 1.173 requires that the indicated derivatives of the drain current expression in Equation 1.168 be evaluated. This evaluation is an algebraically trying task that borders on a futile engineering enterprise in that many of the physical parameters implicit to Equation 1.168 are rarely disclosed to the circuit designer. It is therefore prudent to condescend to first order approximations of the subject small-signal parameters by replacing Equation 1.168 with the simpler expression,

$$i_d \approx \frac{K_n}{2}\left(\frac{W}{L}\right)(v_{gs} - V_h)^2\left(1 + \frac{v_{ds} - v_{dsat}}{V_\lambda}\right), \tag{1.174}$$

which effectively ignores the influence of both lateral and vertical electric fields in the MOSFET channel. By ignoring the effects of lateral fields, parameter M_{sat} in Equation 1.131 is one, whence the drain saturation voltage in Equation 1.129 is simply the voltage difference, $v_{dsat} = (v_{gs} - V_h)$. Accordingly, Equations 1.173 and 1.174 yield a forward transconductance of

$$g_m \triangleq \left.\frac{\partial i_d}{\partial v_{gs}}\right|_Q \approx \frac{2I_d}{V_{gs} - V_h} - \frac{I_d}{V_\lambda + V_{ds} - V_E}, \tag{1.175}$$

where it is understood that the variables, I_d, V_{gs}, V_h, V_λ, and V_{dsat} reflect the Q-point of the transistor undergoing study. Biasing voltage and standby power constraints ordinarily compel that the transistor be

biased at a drain–source voltage that is only slightly above the drain saturation voltage. Accordingly, V_λ is typically much larger than $(V_{ds} - V_{dsat})$. Moreover, V_λ is generally significantly larger than $V_{dsat}/2$. It follows that the second term on the right-hand side of Equation 1.175 is often negligible, whereupon Equations 1.168 and 1.175 combine for the case of large V_λ to deliver

$$g_m \triangleq \left.\frac{\partial i_d}{\partial v_{gs}}\right|_Q \approx \frac{2I_d}{V_{gs} - V_h} \approx \sqrt{2K_n(W/L)I_d}. \tag{1.176}$$

The result suggests that the forward transconductance of a MOSFET increases with the square root of the product of quiescent drain current and transistor gate aspect ratio. Accordingly, high gain requirements in certain MOSFET amplifiers compel relatively large standby drain currents and/or suitably large gate widths. The former tack conflicts with omnipresent desires for low power operation, while the latter begets increased device capacitances and hence, potentially degraded frequency responses. Observe that while the term in V_λ in Equation 1.175 is usually negligibly small, significant channel length modulation (which translates to small V_λ) is deleterious to high gain objectives.

An evaluation of the bulk transconductance, g_{mb}, requires that Equation 1.96 be considered analytically in conjunction with the threshold voltage term in Equation 1.174. After a bit of messy algebra, it can be shown that

$$g_{mb} \triangleq \left.\frac{\partial i_d}{\partial v_{bs}}\right|_Q \approx \lambda_b g_m, \tag{1.177}$$

where λ_b, which might be termed a *bulk modulation factor*, is

$$\lambda_b = \sqrt{\frac{V_\theta/2}{2V_F - V_{bs}}}. \tag{1.178}$$

Recall a previous assertion to the extent that the bulk transconductance, g_{mb}, may be insignificant in comparison to the small-signal impact of the forward transconductance, g_m. From Equations 1.177 and 1.178, λ_b, and hence g_{mb}, are small if V_θ, the body effect potential defined by Equation 1.90, is small. Since V_θ is proportional to the square of the gate oxide thickness, thin oxide layers conduce small bulk transconductances. It is interesting to note in Equation 1.178 that the small values of the bulk modulation factor that are precipitated by thin oxides are made even smaller by increases in the reverse bias applied between the bulk and source.

The drain–source channel resistance, r_o, in Equation 1.173 is readily confirmed to derive from

$$\frac{1}{r_o} \triangleq \left.\frac{\partial i_d}{\partial v_{ds}}\right|_Q \approx \frac{I_d}{V_\lambda + V_{ds} - V_{dsat}}. \tag{1.179}$$

The result shows that a large channel length modulation voltage, V_λ, gives rise to a large channel resistance, r_o, which in turn implies that the drain–source port of a MOSFET emulates the volt–ampere characteristics of an ideal current source. For conventional values of V_λ, large r_o is seen to require a small drain bias current, I_d.

1.2.5.2 Unity Gain Frequency

The models of Figures 1.35 and 1.36 provide an analytical path for computing a commonly invoked figure of merit for MOSFETs; namely, the *unity gain frequency*, which in radial units is symbolized as ω_T. Although this metric offers a meaningful basis for comparing the high-frequency signal processing

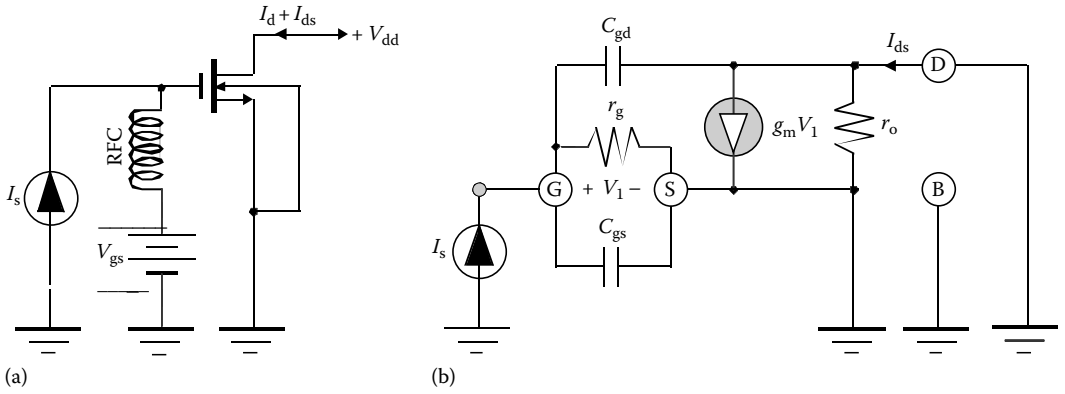


FIGURE 1.37 (a) N-channel common-source MOSFET configured for the evaluation of the unity gain frequency, ω_T, of the transistor. (b) Small-signal, high-frequency equivalent circuit of the network in (a).

capabilities of competing transistors and their associated fabrication processes, its value to bracketing the achievable bandwidths and response speeds of MOSFET circuits is dubious. The latter contention stems from the very definition of the metric. In particular, ω_T is the radial value of signal frequency at which the magnitude of the small-signal, short-circuit current gain of a common-source amplifier degrades to unity. The circuit of relevance is the topology of Figure 1.37a, in which a current signal, I_s is applied to the gate of a MOSFET whose source terminal is incident with signal ground. The radio frequency choke (RFC) provides a conduit for establishing a gate–source bias, V_{gs}, above threshold, while providing a dynamic impedance in series with the gate biasing voltage that is large enough to cajole most of the input signal current to enter the gate terminal.

An input current applied to a gate lead that inherently comprises an open circuit at low frequencies is hardly rational from a circuits perspective. This irrationality is exacerbated by the fact that the drain terminal, where the small-signal current signal response, I_{ds}, to input signal current I_s is extracted, is connected directly to the power supply rail, V_{dd}, thereby rendering the drain terminal short circuited to signal ground (hence the nomenclature, "short circuit" current gain). In other words, the current gain, I_{ds}/I_s, is computed for a common-source amplifier whose gate is driven by signal current and whose drain is short circuited to signal ground, which is hardly a viable analog circuit cell.

Assuming that the transistor at hand operates in saturation, the small-signal equivalent model of the circuit in Figure 1.37a is the structure given in Figure 1.37b. Because the drain, in addition to the bulk and the source, is grounded for signal conditions, the current gain, I_{ds}/I_s, tacitly ignores the high-frequency effects of bulk–drain and bulk–source transistor capacitances. Moreover, the connection of the bulk terminal to the source obviates the need for the bulk transconductance generator, $\lambda_b g_m V_2$, in the model at hand, while short circuiting the drain terminal to the source terminal renders the channel resistance, r_o, inconsequential. Accordingly, an analysis of the structure in Figure 1.37b yields

$$\left.\begin{aligned} I_{ds} &= g_m V_1 - j\omega C_{gd} V_1 \\ I_s &= \frac{V_1}{r_g} + j\omega C_{gs} V_1 + j\omega C_{gd} V_1 \end{aligned}\right\}, \tag{1.180}$$

whence

$$\frac{I_{ds}}{I_s} = \frac{g_m r_g \left(1 - j\omega C_{gd}/g_m\right)}{1 + j\omega r_g \left(C_{gs} + C_{gd}\right)}. \tag{1.181}$$

Since r_g in Equation 1.166 is infinitely large at zero signal frequency, the short circuit gain is seen to be infinity at zero frequency, which reflects engineering intuition in that the gate can conduct no current at zero frequency. In addition, r_g is likely to remain very large in the neighborhood of the 3 dB frequency projected by this gain relationship so that

$$\frac{I_{ds}}{I_s} \approx \frac{g_m}{j\omega\left(C_{gs} + C_{gd}\right)}. \tag{1.182}$$

Equation 1.182 also invokes the reasonable presumption that the frequency, g_m/C_{gd}, of the right half plane zero evidenced on the right-hand side of Equation 1.181 is significantly larger than the aforementioned 3 dB bandwidth. This presumption is tantamount to neglecting the gate-to-drain feedforward through the gate–drain capacitance, C_{gd}, in comparison to the I/O feedforward promoted by the transistor transconductance, g_m. While this approximation is suspect at very high signal frequencies, the approximations leading to Equation 1.182 allow an extrapolated value of the unity gain frequency of

$$\omega_T = 2\pi f_T = \frac{g_m}{C_{gs} + C_{gd}}. \tag{1.183}$$

Clearly, f_T is a highly optimistic estimate of achievable circuit performance, for it pertains expressly to the special case of a drain that is short circuited to the source terminal, thereby quashing the impact on bandwidth of bulk–drain capacitance and any load capacitance that might be driven by the subject transistor. Using Equations 1.106, 1.156, 1.164, 1.176, and 1.174 with V_λ presumed large, the last result is expressible as

$$\omega_T = 2\pi f_T \approx \frac{3\mu_n\left(V_{gs} - V_h\right)}{2L^2\left[1 + \frac{3\left(C_{gso} + C_{gdo}\right)}{2LC_{ox}}\right]}, \tag{1.184}$$

which reflects a direct dependence of the unity gain frequency on free carrier mobility. The result also infers that to the extent that $2LC_{ox} \gg 3(C_{gso} + C_{gdo})$, which may indeed be an engineering stretch for minimal geometry, deep submicron devices, f_T is inversely proportional to the square of channel length. A significant increase in f_T is therefore portended by even a relatively modest shortening of the channel length. The fact that the unity gain frequency value is common corporate banter in the marketing of state of the art transistors and processes arguably underpins the widespread process-engineering penchant for progressively decreased channel lengths.

With the gate resistance, r_g, presumed very large over the signal frequency range of interest, the resultant short circuit current gain in Equation 1.182 is dependent on only frequency-invariant small-signal transistor parameters. Accordingly, Equation 1.183 allows Equation 1.182 to be generalized as the complex frequency domain expression,

$$\frac{I_{ds}}{I_s} \approx \frac{\omega_T}{s}. \tag{1.185}$$

Moreover, the model in Figure 1.35b reduces to the equivalent circuit shown in Figure 1.38a, where signal current I_1 is identified as the current conducted by the gate–source capacitance, C_{gs}. Since signal voltage V_1 is clearly I_1/sC_{gs}, the voltage controlled current, $g_m V_1$, is

$$g_m V_1 = \frac{g_m I_1}{sC_{gs}} = \frac{\omega_T\left(C_{gs} + C_{gd}\right)I_1}{sC_{gs}} = \left(\frac{k_g\omega_T}{s}\right)I_1, \tag{1.186}$$

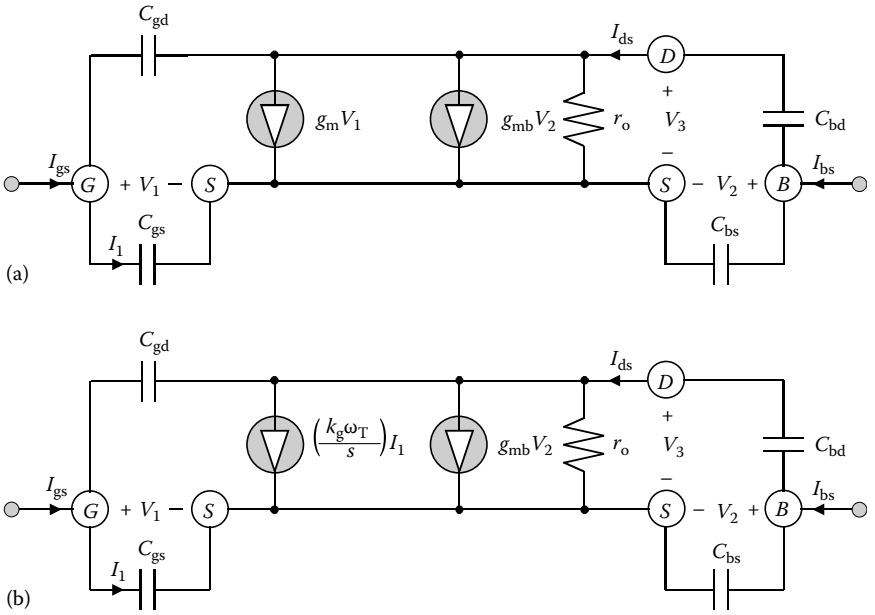


FIGURE 1.38 (a) Small-signal model of Figure 1.35b with gate resistance r_g ignored. (b) Alternative CCCS form of the equivalent circuit in (a).

where

$$k_g = 1 + \frac{C_{gd}}{C_{gs}}. \tag{1.187}$$

Equation 1.186 allows the voltage-controlled current source (VCCS) form of the model shown in Figure 1.38a to be transformed into the equivalent current-controlled structure offered in Figure 1.38b. The latter form proves useful in assessing the performance of amplifiers, such as certain forms of low noise bandpass structures, which utilize source degeneration inductances.

1.2.5.3 Small-Signal Model Development

While the small-signal transistor models shown in Figures 1.35 and 1.36 are topologically correct and conceptually useful from the perspective of linear active network design and first order performance assessment, their engineering utility is limited by two issues. First, the analytical expressions for the parameters embedded in these structures are predicated on a plethora of approximations stemming from the neglect of the effects of lateral and vertical electric fields and simplifications surrounding the charge storage mechanisms of devices and their associated capacitive profiles. These analytically simplifying approximations often place laboratory characterizations of device behavior at odds with physical reality. For example, the substrate doping concentration is not a constant, as is presumed in all foregoing analytical disclosures, but it is less than immediately clear if some sort of weighted average of this dopant level is appropriate for a satisfying voltage–current–charge characterization of a considered transistor. Second, the subject small-signal parameters are dependent on variables, such as carrier mobility, oxide overlap dimensions, doping concentrations, densities of charges trapped in the oxide, regional perimeter dimensions, and the like that are either not released to the circuit designer or are otherwise only vaguely known to the processing foundry.

Because of the foregoing parametric anomalies, reasonably accurate and physically sound assessments of small-signal device and associated circuit performance require that the numerical explication of all relevant small-signal parameters derive from appropriate laboratory measurements conducted on either test device structures or on entire test cells of the circuits undergoing development. A commonly used vehicle toward this characterization end is the *scattering parameters*, or *S-parameters*, measured for a grounded source, grounded gate, or grounded drain interconnection of a subject transistor excited for a suitable range of biasing levels and over an appropriate range of signal frequencies [11]. These parameters are extracted with fixed and known—generally 50 ohm—reference terminations at the input and output ports of the device undergoing test. The measured *S*-parameters, S_{ij}, are then converted into short circuit admittance (*y*-) parameters, y_{ij}. The latter two-port parameters are virtually impossible to discern directly in the laboratory because their numerical delineation mandates the imposition of input and output port signal short circuits, which are difficult to sustain over broad frequency passbands. Once the y_{ij} are determined, it is an involved, but nonetheless straightforward, matter to infer realistic values of most of the parameters implicit to the structures of Figures 1.35 and 1.36.

If the process foundry provides a reliable large-signal HSPICE model, such as the fundamentally heuristic Level 49 MOSFET model, of the device under consideration, the short circuit admittance parameters of the subject transistor can be deduced through appropriate small-signal computer-aided simulations. For example, consider the N-channel MOSFET in Figure 1.39, which is shown connected as a grounded source, three-port configuration. The battery voltage, V_{gg}, biases the gate–source terminals at a greater than threshold value of voltage that establishes the desired quiescent drain current, I_d. Of course, zero quiescent gate current flows in the gate lead of the transistor. On the other hand, the battery voltage, V_{dd}, which modestly influences the quiescent drain current, I_d, is chosen to ensure saturation regime operation of the transistor. Finally, the voltage, V_{bb}, biases the bulk substrate terminal, where it is understood that V_{bb} is ordinarily at most zero. In concert with the traditional stipulations of reverse-biased bulk–source and bulk–drain junctions, zero quiescent gate current is presumed to flow into the bulk. The application of any one or more of the three indicated signal voltages, V_{1s}, V_{2s}, and V_{3s}, produces signal current responses in the gate, bulk, and drain of I_{gs}, I_{bs}, and I_{ds}, respectively. Selecting the "AC" simulation option to manifest a strictly linear HSPICE analysis of the aforementioned current signal responses about respective quiescent values allows the applied signal voltages, V_{1s}, V_{2s}, and V_{3s}, to be set conveniently to amplitudes of 1 V.

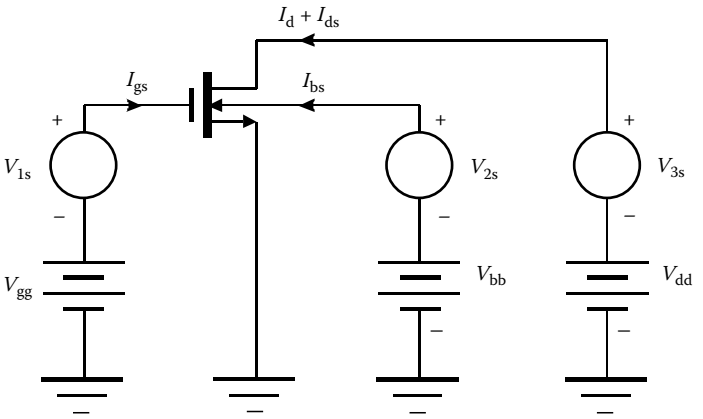


FIGURE 1.39 Common-source test cell of an N-channel transistor configured as a three-port network.

If the applied signal voltages subscribe to the small-signal, linear-operational constraint, superposition theory applies, and the three signal current responses can be described by the linear admittance parameter matrix,

$$\begin{bmatrix} I_{gs} \\ I_{bs} \\ I_{ds} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix} \begin{bmatrix} V_{1s} \\ V_{2s} \\ V_{3s} \end{bmatrix}. \tag{1.188}$$

In Equation 1.188, the short-circuit admittance parameters, y_{ij}, which are invariably complex numbers, are extracted over signal frequency. In particular,

$$\left.\begin{array}{lll} y_{11} = I_{gs}/V_{1s}|_{V_{2s}=V_{3s}=0}, & y_{12} = I_{gs}/V_{2s}|_{V_{1s}=V_{3s}=0}, & y_{13} = I_{gs}/V_{3s}|_{V_{1s}=V_{2s}=0} \\ y_{21} = I_{bs}/V_{1s}|_{V_{2s}=V_{3s}=0}, & y_{22} = I_{bs}/V_{2s}|_{V_{1s}=V_{3s}=0}, & y_{23} = I_{bs}/V_{3s}|_{V_{1s}=V_{2s}=0} \\ y_{31} = I_{ds}/V_{1s}|_{V_{2s}=V_{3s}=0}, & y_{32} = I_{ds}/V_{2s}|_{V_{1s}=V_{3s}=0}, & y_{33} = I_{ds}/V_{3s}|_{V_{1s}=V_{2s}=0} \end{array}\right\}. \tag{1.189}$$

The real and imaginary parts of all nine of these *y*-parameters can be readily evaluated from a small-signal HSPICE analysis of the structure in Figure 1.39 or alternatively, they can be discerned in terms of scattering parameters gleaned from measurements of a test structure analogous to that of the subject figure.

If the algebraic form of parameter y_{12} in Equation 1.188 is defined as

$$y_{12} \triangleq g_{12} - j\omega C_{12}, \tag{1.190}$$

the first of the equations in Equation 1.188 can be written as

$$I_{gs} = (y_{11} + y_{13} - j\omega C_{12})V_{1s} + g_{12}V_{2s} - y_{13}(V_{1s} - V_{3s}) + j\omega C_{12}(V_{1s} - V_{2s}), \tag{1.191}$$

which can be couched in the form,

$$I_{gs} = \left(\frac{1}{R_i} + j\omega C_i\right)V_{1s} + g_{12}V_{2s} + \left(\frac{1}{R_f} + j\omega C_f\right)(V_{1s} - V_{3s}) + j\omega C_{12}(V_{1s} - V_{2s}). \tag{1.192}$$

The first parenthesized term on the right-hand side of this expression represents a gate-to-ground shunt interconnection of a resistance,

$$R_i = \frac{1}{\mathrm{Re}(y_{11}) + \mathrm{Re}(y_{13})}, \tag{1.193}$$

and a capacitance,

$$C_i = \frac{\mathrm{Im}(y_{11}) + \mathrm{Im}(y_{13})}{\omega} - C_{12}, \tag{1.194}$$

where from Equation 1.190,

$$C_{12} = -\frac{\mathrm{Im}(y_{12})}{\omega}. \tag{1.195}$$

The resistance, R_i, tends to vary as the inverse square of the radial signal frequency. Thus, it is expedient to write R_i as

$$R_i = \frac{K_{Ri}}{\omega^2}, \tag{1.196}$$

where K_{Ri} is a constant boasting the strange dimensions of ohms–(rad/s)^2.

The second term in Equation 1.192 represents a VCCS whose bulk-to-gate transconductance is

$$g_{12} = \mathrm{Re}(y_{12}). \tag{1.197}$$

The second parenthesized factor on the right-hand side of the subject equation connotes a gate-to-drain shunt combination of resistance

$$R_\mathrm{f} = -\frac{1}{\mathrm{Re}(y_{13})} \tag{1.198}$$

and capacitance

$$C_\mathrm{f} = -\frac{\mathrm{Im}(y_{13})}{\omega}. \tag{1.199}$$

As is the case with resistance R_i, R_f also varies as the inverse square of the radial signal frequency. Accordingly,

$$R_\mathrm{f} = \frac{K_\mathrm{Rf}}{\omega^2}. \tag{1.200}$$

Finally, the last term in Equation 1.192 is merely a capacitance, C_{12}, incident between gate and bulk terminals. It is appropriate to interject that over a broad range of signal frequencies that do not exceed the transistor unity gain frequency, f_T, the capacitances, C_i, C_{12}, and C_f, are nearly constants, which suggests that $\mathrm{Im}(y_{11})$, $\mathrm{Im}(y_{12})$, and $\mathrm{Im}(y_{13})$ are nominally linear functions of the radial signal frequency. An analogous statement prevails for all of the other capacitances defined in the forthcoming paragraphs. While resistances R_i and R_f decrease sharply with signal frequency, they are so large (hundreds or even thousands of megohms) that they can usually be neglected in the course of most design-oriented analog circuit analyses.

Letting

$$y_{ij} = g_{ij} - j\omega C_{ij} \tag{1.201}$$

denote the general short circuit admittance parameter, y_{ij}, Equation 1.188 allows the signal current, I_bs, conducted by the bulk to be expressed as

$$I_\mathrm{bs} = (g_{21} - j\omega C_x)V_\mathrm{1s} + \left(\frac{1}{R_\mathrm{bb}} + j\omega C_\mathrm{bb}\right)V_\mathrm{2s} + (g_{23} - j\omega C_{23})V_\mathrm{3s} + j\omega C_{12}(V_\mathrm{2s} - V_\mathrm{1s}), \tag{1.202}$$

where, recalling Equation 1.201,

$$C_x = C_{21} - C_{12} = \frac{\mathrm{Im}(y_{12}) - \mathrm{Im}(y_{21})}{\omega}. \tag{1.203}$$

and

$$g_{21} = \mathrm{Re}(y_{21}). \tag{1.204}$$

The term, $(g_{21} - j\omega C_x)$ in Equation 1.202 is a transadmittance linking the signal gate voltage to the bulk signal current. The second parenthesized term on the right-hand side of Equation 1.202 reflects a bulk-to-ground parallel combination of a frequency-dependent resistance

$$R_{bb} = \frac{1}{\mathrm{Re}(y_{22})} = \frac{K_{Rbb}}{\omega^2}, \tag{1.205}$$

and capacitance

$$C_{bb} = \frac{\mathrm{Im}(y_{22})}{\omega} - C_{12}. \tag{1.206}$$

A second transadmittance factor, $(g_{23} - j\omega C_{23})$, surfaces to model the coupling of the drain signal voltage, V_{3s}, to the bulk signal current, I_{bs}, where

$$\left.\begin{aligned} g_{23} &= \mathrm{Re}(y_{23}) \\ C_{23} &= -\frac{\mathrm{Im}(y_{23})}{\omega} \end{aligned}\right\}. \tag{1.207}$$

Finally, the last term in Equation 1.200 complements its last term counterpart in Equation 1.192 in that it accounts for the bilateral capacitive coupling prevailing between the drain and bulk terminals.

The only current not yet addressed is the signal drain current, I_{ds}. From Equation 1.188,

$$I_{ds} = (g_m - j\omega C_m)V_{1s} + (g_{mb} - j\omega C_{mb})V_{2s} + \left(\frac{1}{R_o} + j\omega C_o\right)V_{3s} + \left(\frac{1}{R_f} + j\omega C_f\right)(V_{3s} - V_{1s}). \tag{1.208}$$

The factor, $(g_m - j\omega C_m)$, is the forward transadmittance that couples the gate signal voltage to the drain signal current. Its constituent variables are

$$\left.\begin{aligned} g_m &= \mathrm{Re}(y_{31}) - \mathrm{Re}(y_{13}) \\ C_m &= \frac{\mathrm{Im}(y_{13}) - \mathrm{Im}(y_{31})}{\omega} \end{aligned}\right\}. \tag{1.209}$$

On the other hand, $(g_{mb} - j\omega C_{mb})$ is the bulk transadmittance serving to bracket the signal drain current response to the signal bulk voltage, V_{2s}. The variables, g_{mb} and C_{mb}, are

$$\left.\begin{aligned} g_{mb} &= \mathrm{Re}(y_{32}) \\ C_{mb} &= -\frac{\mathrm{Im}(y_{32})}{\omega} \end{aligned}\right\}. \tag{1.210}$$

Following Equation 1.177, the bulk modulation factor, λ_b, can be discerned to be

$$\lambda_b = \frac{g_{mb}}{g_m} = \frac{\mathrm{Re}(y_{32})}{\mathrm{Re}(y_{31}) - \mathrm{Re}(y_{13})}. \tag{1.211}$$

The parenthesized factor of the third term on the right-hand side of Equation 1.206 is the drain-to-ground shunt interconnection of resistance R_o and capacitance C_o, such that

$$\left.\begin{aligned} R_o &= \frac{1}{\mathrm{Re}(y_{33}) + \mathrm{Re}(y_{13})} \\ C_o &= \frac{\mathrm{Im}(y_{33}) + \mathrm{Im}(y_{13})}{\omega} \end{aligned}\right\}. \tag{1.212}$$

The last term in Equation 1.208 reflects the previously introduced, bilateral R_f–C_f coupling between the gate and the drain terminals.

Equation 1.192 for the signal gate current, I_{gs}, Equation 1.202 for the signal bulk current, I_{bs}, and Equation 1.208 for the signal drain current, I_{ds}, can now be exploited to develop the foreboding three-port common-source MOSFET model diagrammed in Figure 1.40a. While the model is intractable for manual circuit analysis and considerably more complicated than its simplified brethren in Figures 1.35 and 1.36, it does serve to bolster circuit design insights. First, and perhaps most obviously, the model at hand illustrates the complex interactions of the bulk with the gate, drain, and source regions of a MOSFET. For example, the bulk signal voltage, V_{2s}, precipitates a real controlled source, $\lambda_b g_m V_{2s}$, in addition to a quadrature controlled source, $j\omega C_{mb} V_{2s}$, at the drain–source port. The first of these sources is the expected effect of bulk-induced modulation of MOSFET threshold voltage, but the latter controlled element is slightly south of transparent. The bulk also gives rise to a VCCS, $g_{12} V_{2s}$, in the gate–source port, which accounts for observable bulk-induced increases in high-frequency gate current. These intricacies, together with the complex transadmittance coupling, $(g_{23} + j\omega C_{23})V_{3s}$, from the drain-to-the bulk seemingly encourage, whenever possible and prudent, operating the MOSFET with its bulk terminal returned to the transistor source terminal. Under such a topological constraint, the model in Figure 1.40a collapses to the almost shockingly simpler network offered in Figure 1.40b.

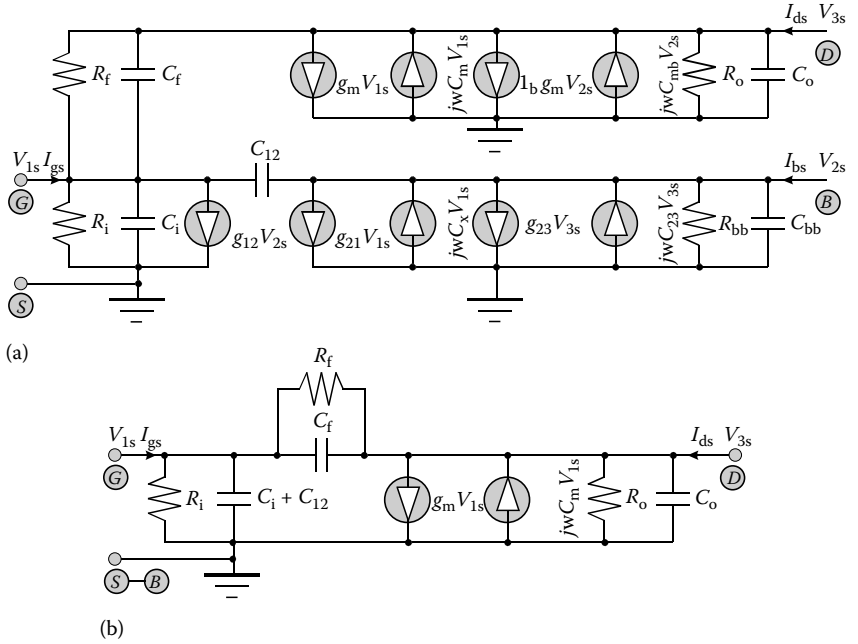


FIGURE 1.40 (a) Small-signal, three-port equivalent circuit for the common-source interconnection of a MOSFET. The three terminal voltages, V_{1s}, V_{2s}, and V_{3s}, denote signal voltages developed with respect to ground at the gate, bulk, and drain terminals, respectively. (b) The equivalent circuit of (a) with the bulk terminal connected directly to the MOSFET source terminal.

A comparison of the model in Figure 1.40b with that of Figure 1.35b suggests that the gate–source resistance, R_i, is effectively the gate resistance, r_g, introduced in Equation 1.166. The resistance, R_f, in Figure 1.40b has no counterpart in Figure 1.35b. Throughout the range of frequencies extending through the unity gain frequency of the considered transistor, both R_i and R_f are so large that they can be ignored for most small-signal analysis ventures, save possibly for a small-signal analysis entailing an assessment of the noise properties of a transistor. The resistance, R_o, is akin to the channel resistance, r_o, in Figure 1.35b. Unlike R_i and R_f, r_o is nominally frequency invariant through the device unity gain frequency metric. With R_i and R_f tacitly ignored and in view of the fact that r_o is independent of frequency, the steady state frequency variable, $j\omega$, in Figure 1.40 can be replaced by the Laplace operator, s, thereby allowing for small-signal step response and other transient investigations of MOSFET amplifiers.

The net capacitance, $(C_i + C_{12})$, in Figure 1.40 is the effective gate–source capacitance, C_{gs}, in Figure 1.35b. Because of the inclusion of capacitance C_{12}, this net gate–source capacitance accounts for gate-to-bulk capacitance, which earlier models presented in this discourse ignore tacitly, primarily because of the high-frequency capacitance characteristics advanced by Figure 1.23. The capacitance, C_f, is the effective gate–drain capacitance, C_{gd}, while capacitance C_o represents the effective bulk–drain capacitance, C_{bd}.

The model in Figure 1.35b highlights a real forward transconductance of g_m, while the models in Figure 1.40 project a complex forward transadmittance, Y_m, of

$$Y_m = g_m - j\omega C_m = g_m e^{j\varphi_m(\omega)} \sqrt{1 + \left(\frac{\omega C_m}{g_m}\right)^2}, \tag{1.213}$$

where

$$\varphi_m(\omega) = -\tan^{-1}\left(\frac{\omega C_m}{g_m}\right) \tag{1.214}$$

denotes an *excess phase angle* associated with the transport of minority carriers in the gate-induced channel extending from the source region-to-the drain region. Equivalently, the angle, $\varphi_m(\omega)$, is associated with an *excess envelope delay*, $T_m(\omega)$, such that

$$T_m(\omega) = -\frac{d\varphi_m(\omega)}{d\omega} = \frac{C_m/g_m}{1 + \left(\frac{\omega C_m}{g_m}\right)^2}, \tag{1.215}$$

whose low-frequency and, in this case, maximum value is obviously (C_m/g_m). Excess delay, for which no account prevails in the simpler models of Figures 1.35 and 1.36, looms potentially critical in feedback circuits in that it acts to degrade the achievable phase margin of the open loop response.

VCCSs having an imaginary transadmittance can be synthesized easily for small-signal, computer-based analyses through the use of a voltage-controlled voltage source (VCVS), a capacitor, and a current-controlled current source (CCCS), as depicted in Figure 1.41a. In this figure, the controlling current, I_i, of the CCCS, αI_i, is

$$I_i = j\omega C\mu V_i, \tag{1.216}$$

whence the indicated controlled current, I_o, is

$$I_o = \alpha I_i = j\omega C\alpha\mu V_i. \tag{1.217}$$

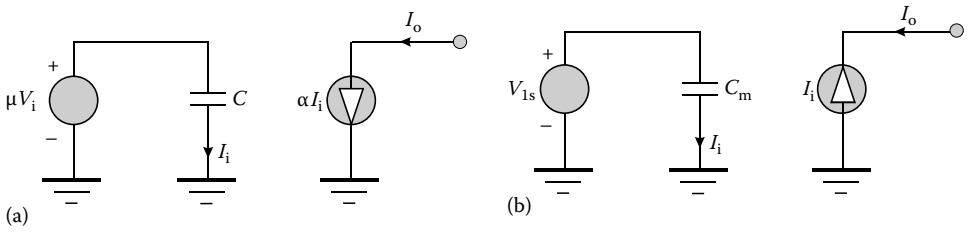


FIGURE 1.41 (a) Synthesis of a VCCS whose transadmittance is imaginary and proportional to radial signal frequency. (b) The synthesis of the controlled current, $-j\omega C_m V_{1s}$, in the models shown in Figure 1.40.

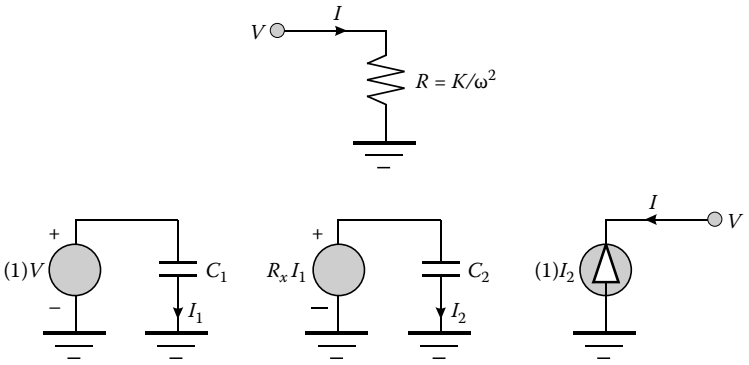


FIGURE 1.42 Synthesis of a branch resistance whose value is inversely proportional to the square of the radial signal frequency. The indicated resistance, R, is synthesized if $R_x = 1/KC_1C_2$.

Thus, for the imaginary component, $j\omega C_m$, of the forward transadmittance, Y_m, in Equation 1.211, $\mu = 1$, $C = C_m$, and $\alpha = -1$ gives the desired controlled current, $-j\omega C_m V_{1s}$, as is abstracted in Figure 1.41b.

Similarly, the frequency variant resistances, R_i, R_f, and R_{bb}, can be synthesized for small-signal, computer-aided analysis purposes using a VCVS, a current-controlled voltage source (CCVS), and a CCCS. This contention is illustrated in Figure 1.42 for the general case of a resistance, R, given by

$$R = -\frac{K}{s^2}, \tag{1.218}$$

which for steady state sinusoidal conditions is the generalized relationship,

$$R = \frac{K}{\omega^2}, \tag{1.219}$$

advanced by Equations 1.196, 1.200, and 1.205. To wit, the controlled current, I_1, generated by the VCVS, $(1)V$, is $I_1 = sC_1V$, while the current, I_2, established in response to the CCVS, (R_xI_1), is $I_2 = sC_2R_xI_1 = s^2C_1C_2R_xV$. It follows that the resistance, R, presented to the port driven by the CCCS, $(1)I_2$, is

$$R = \frac{V}{-I_2} = \frac{V}{-s^2C_1C_2R_xV} = -\frac{1}{s^2C_1C_2R_x}. \tag{1.220}$$

For arbitrary values of capacitances C_1 and C_2, selecting

$$R_x = \frac{1}{KC_1C_2} \tag{1.221}$$

achieves the desired resistance value set forth by Equation 1.218.

Parameterization example:
An N-channel transistor featuring a channel length of 180 nm has the Level 49 HSPICE parameters that appear in Table 1.2. The transistor is implemented with a gate aspect ratio of $W/L = 25$, and is biased at $V_{gs} = 1.1$ V, $V_{ds} = 1$ V and $V_{bs} = 0$ V. The device undergoing study is earmarked for analog small-signal applications that embrace a signal frequency range extending from 100 MHz to 10 GHz. For this frequency passband, determine nominal values of all of the parameters indigenous to the small-signal, common-source model of Figure 1.40b. Also, compute the extrapolated unity gain frequency of the transistor at the given quiescent operating point. Express these results as maximum value, minimum value, average value, and standard deviation (referred to the average value) over a frequency passband extending from 100 MHz to 10 GHz.

TABLE 1.2 Representative Level 49 HSPICE Parameters for an NMOS Transistor in a Fabrication Process Featuring a Nominal Channel Length of 180 nM

Model 180 nM NMOS (Level = 49)

+VERSION = 3.1	TNOM = 27	TOX = 4E-9	XJ = 1E-7
+NCH = 2.3549E17	VTH0 = 0.3627858	K1 = 0.5873035	K2 = 4.793052E-3
+K3 = 1E-3	K3B = 2.2736112	W0 = 1E-7	NLX = 1.675684E-7
+DVT0W = 0	DVT1W = 0	DVT2W = 0	DVT0 = 1.7838401
+DVT1 = 0.5354277	DVT2 = −1.243646E-3	U0 = 263.3294995	UA = −1.359749E-9
+UB = 2.250116E-18	UC = 5.204485E-11	VSAT = 1.083427E5	A0 = 2
+AGS = 0.4289385	B0 = −6.378671E-9	B1 = −1E-7	KETA = −0.0127717
+A1 = 5.347644E-4	A2 = 0.8370202	RDSW = 150	PRWG = 0.5
+PRWB = −0.2	WR = 1	WINT = 1.798714E-9	LINT = 7.631769E-9
+XL = −2E-8	XW = −1E-8	DWG = −3.268901E-9	DWB = 7.685893E-9
+VOFF = −0.0882278	NFACTOR = 2.5	CIT = 0	CDSC = 2.4E-4
+CDSCD = 0	CDSCB = 0	ETA0 = 2.455162E-3	ETAB = 1
+DSUB = 0.0173531	PCLM = 0.7303352	PDIBLC1 = 0.2246297	PDIBLC2 = 2.220529E-3
+PDIBLCB = −0.1	DROUT = 0.7685422	PSCBE1 = 8.697563E9	PSCBE2 = 5E-10
+PVAG = 0	DELTA = 0.01	RSH = 6.7	MOBMOD = 1
+PRT = 0	UTE = −1.5	KT1 = −0.11	KT1L = 0
+KT2 = 0.022	UA1 = 4.31E-9	UB1 = −7.61E-18	UC1 = −5.6E-11
+AT = 3.3E4	WL = 0	WLN = 1	WW = 0
+WWN = 1	WWL = 0	LL = 0	LLN = 1
+LW = 0	LWN = 1	LWL = 0	CAPMOD = 2
+XPART = 0.5	CGDO = 716E-12	CGSO = 716E-12	CGBO = 1E-12
+CJ = 9.725711E-4	PB = 0.7300537	MJ = 0.365507	CJSW = 2.604808E-10
+PBSW = 0.4	MJSW = 0.1	CJSWG = 3.3E-10	PBSWG = 0.4
+MJSWG = 0.1	CF = 0	PVTH0 = 4.289276E-4	PRDSW = −4.2003751
+PK2 = −4.920718E-4	WKETA = 6.938214E-4	LKETA = −0.0118628	PU0 = 24.2772783
+PUA = 9.138642E-11	PUB = 0	PVSAT = 1.680804E3	PETA0 = 2.44792E-6
+PKETA = 4.537962E-5)			

Results:

1. Before proceeding with the simulation, the planar source and drain areas, A_s and A_d, as well as the source and drain peripheral dimensions, P_s and P_d, must be computed through an appeal to Equation 1.154. These are $A_s = A_d = (1.62)(10^{-12})$ m^2 and $P_s = P_d = (5.22)(10^{-6})$ *m*. In arriving at these figures, use is made of the fact that for a channel length of $L = 180$ nm and a gate aspect ratio of $W/L = 25$, the gate width is $W = 4.5$ μm. The parameters, L, W, A_s, A_d, P_s, and P_d are inserted directly on the model line of the HSPICE net list. For example, the model line used in the simulations executed herewith is

$$M2\ 6\ 4\ 0\ 7\ 180\ \text{nM}, \quad L = 180\text{n}, \quad W = 4.5\ \text{u}, \quad A_S = 1.62\ \text{p}, \quad A_D = 1.62\ \text{P},$$
$$P_S = 5.22\ \text{u}, \quad P_D = 5.22\ \text{u}.$$

   In this model line, *M*2 identifies the transistor undergoing examination, ''6'' is the number of the drain node, ''4'' is the gate node number, ''0'' is the number of the grounded source node, and ''7'' is the number of the bulk substrate node. The insert, ''180 nM,'' identifies the name of the model used for the subject transistor.

   An HSPICE simulation of the simple test cell shown in Figure 1.39 can now be straightforwardly executed. In this test structure, $V_{gg} = 1.1$ V, $V_{dd} = 1$ V, and $V_{bb} = 0$ V combine to set the desired operating point of the transistor. The operating point information disclosed by the static HSPICE simulation is as follows.

   ID 8.9407E-04 (drain current is $I_d = 894.1$ μA)
   IS -8.9407E-04 (source current flows out of device and is virtually identical to the drain current)
   IB -1.0002E-12 (bulk current is about one picoampere and flows out of the device)
   IBD -9.9417E-13 (bulk current is sum of the bulk–drain and bulk–source junction currents)
   IBS -6.0237E-15 (bulk current is sum of the bulk–drain and bulk–source junction currents)
   VGS 1.1000 (desired gate–source quiescent voltage)
   VDS 1.0000 (desired drain–source quiescent voltage)
   VBS 0.0000 (desired bulk–source quiescent voltage)
   VTH 0.5102 (simulated threshold voltage is $V_h = 510.2$ mV)
   VDSAT 0.3149 (simulated drain saturation voltage is $V_{dsat} = 314.9$ mV)

   It should be noted that the quiescent drain source voltage, $V_{ds} = 1.0$ V, is certainly larger than the simulated drain saturation voltage, $V_{dsat} = 314.9$ mV. Accordingly, the device at hand operates in its saturation regime for suitable small-signal excitations.

2. A small-signal, computer-aided simulation of the test circuit in Figure 1.39 can now be executed at the quiescent operating point established in the preceding step of this exercise. The objective of this simulation is to ascertain the real and imaginary components of each of the nine short circuit admittance parameters, y_{ij}, introduced in Equation 1.188. The model parameters then derive from the pertinent equations given in Section 1.2.5.3.

   Gate-to-drain resistance coefficient, K_{Rf}:
   Maximum value is $(7.93)(10^{27})$
   Minimum value is $(7.92)(10^{27})$
   Average value is $(7.93)(10^{27})$
   Standard deviation is 0.03%

   Gate-to-drain capacitance, C_f:
   Maximum value is 3.22 fF
   Minimum value is 3.21 fF

Average value is 3.22 fF
Standard deviation is 0.02%

Gate-to-source resistance coefficient, K_{Ri}:
Maximum value is $(4.25)(10^{27})$
Minimum value is $(4.24)(10^{27})$
Average value is $(4.25)(10^{27})$
Standard deviation is 0.03%

Gate-to-source capacitance, C_i:
Maximum value is 7.56 fF
Minimum value is 7.55 fF
Average value is 7.55 fF
Standard deviation is 0.02%

Bulk–gate transconductance, g_{12}:
Maximum value is 1.01 μmho
Minimum value is 0 μmho
Average value is 0.11 μmho
Standard deviation is 191.97%

Gate-to-bulk capacitance, C_{12}:
Maximum value is 0.42 fF
Minimum value is 0.42 fF
Average value is 0.42 fF
Standard deviation is 0%

Forward transconductance, g_m:
Maximum value is 2.03 mmho
Minimum value is 2.03 mmho
Average value is 2.03 mmho
Standard deviation is 0.01%

Transadmittance capacitance, C_m:
Maximum value is 2.00 fF
Minimum value is 1.99 fF
Average value is 2.00 fF
Standard deviation is 0.05%

Gate–bulk transconductance, g_{21}:
Maximum value is 0.82 μmho
Minimum value is 0 μmho
Average value is 0.18 μmho
Standard deviation is 191.97%

Gate–bulk transadmittance capacitance, C_x:
Maximum value is 0.68 fF
Minimum value is 0.67 fF
Average value is 0.67 fF
Standard deviation is 0.03%

Bulk transconductance modulation factor, λ_b:
Maximum value is 0.21
Minimum value is 0.21

Average value is 0.21
Standard deviation is 0.03%

Drain–bulk transconductance, g_{23}:
Maximum value is 0 μmho
Minimum value is −0.41 μmho
Average value is −0.05 μmho
Standard deviation is 191.97%

Bulk transadmittance capacitance, C_{mb}:
Maximum value is 3.19 fF
Minimum value is 3.18 fF
Average value is 3.19 fF
Standard deviation is 0.02%

Drain–bulk transadmittance capacitance, C_{23}:
Maximum value is 2.63 fF
Minimum value is 2.63 fF
Average value is 2.63 fF
Standard deviation is 0.02%

Drain–source channel resistance, R_o:
Maximum value is 10.38 kΩ
Minimum value is 10.33 kΩ
Average value is 10.37 kΩ
Standard deviation is 0.09%

Bulk-to-source resistance coefficient, K_{Rbb}:
Maximum value is $(7.03)(10^{27})$
Minimum value is $(6.78)(10^{27})$
Average value is $(6.99)(10^{27})$
Standard deviation is 0.76%

Drain–source capacitance, C_o:
Maximum value is 2.62 fF
Minimum value is 2.62 fF
Average value is 2.62 fF
Standard deviation is 0.03%

Bulk–source capacitance, C_{bb}:
Maximum value is 6.83 fF
Minimum value is 6.82 fF
Average value is 6.83 fF
Standard deviation is 0.02%

3. Equation 1.183 is the pertinent equation for the computation of the extrapolated unity gain frequency. To this end, the average forward transconductance has been computed to be $g_m = 2.03$ mmho. The effective gate–source capacitance, C_{gs}, is the computed average value, $C_i = 7.55$ fF, which accounts for gate–source overlap and any other second order phenomena embraced by the utilized HSPICE model. On the other hand, the effective average gate–drain capacitance, C_{gd}, is $C_f = 3.22$ fF, which, like C_i, incorporates all pertinent high order device characterization phenomena. Accordingly

$$f_T = \frac{g_m}{2\pi(C_i + C_f)} = 30.0 \text{ GHz}.$$

Comments: With the exception of parameters g_{12}, g_{21}, and g_{23}, the quoted standard deviation numbers indicate an excellent model fit to circuit theoretic issues. These three transconductances can also be made to agree well with theoretical disclosures if they are each allowed to vary as the square of the radial signal frequency. However, their values are so small as to render overt concern of them unproductive.

The computed unity gain frequency, f_T, is within range of the expected frequency performance of representative MOSFETs manufactured in a 180 nM technology process. It is interesting to note, however, that the effective gate–drain capacitance (3.22 fF), which is traditionally ignored in first order, high-frequency circuit analysis ventures, is, in this case, almost 43% of the effective gate–source capacitance (7.55 fF).

1.2.6 Design-Oriented Analysis Strategy

When a MOSFET is exploited for a linear analog signal processing application, an essential early design requirement entails the implementation of suitable biasing. Generally, this biasing must ensure that for all pertinent signal levels, each transistor used to supply gain, impedance conversion, constant current, constant voltage, or other I/O properties operates in its saturated domain where its drain–source voltages, V_{ds}, is at least as large as its drain saturation voltage, V_{dsat}. When $V_{ds} \geq V_{dsat}$, Equation 1.142 is the applicable relationship for ascertaining a gate–source voltage, V_{gs}, commensurate with a target drain current, I_d, conducted at a given or desired value of drain–source voltage.

Unfortunately, academic satisfaction does not often resonate with the engineering reality that underlies predictable, reliable, and reproducible integrated circuit design. For the biasing issue at hand, Equation 1.142 is fraught with numerous shortfalls. Despite its algebraic cumbersomeness, Equation 1.142 is only an approximation of the static volt–ampere characteristics of a MOSFET operated in saturation, owing to a variety of analytical liberties exploited with respect to charge storage, charge transport, carrier mobility, and the other phenomenological issues discussed in preceding sections. Even if Equation 1.142 were an accurate disclosure of the aforementioned static characteristics, challenges surround its utilization because circuit and system designers are rarely privy to the physical and process parameters on which the metrics, K_n, V_h, V_{dsat}, V_{ve}, V_{le}, and V_λ, are dependent. These model variables can be discerned reliably through only laboratory measurement of static device responses or via analyses conducted on appropriate computer-based simulations founded on accurate and reliable transistor models.

On the tacit presumption that the foregoing six model variables can be extracted satisfactorily from measurement and/or simulation, Equation 1.142 might be supplanted by the more familiar, nominally square law relationship,

$$I_d = \frac{K_{ne}}{2}\left(\frac{W}{L}\right)\left(V_{gs} - V_h\right)^2\left(1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}\right), \tag{1.222}$$

where K_{ne} symbolizes the effective transconductance coefficient,

$$K_{ne} = \frac{K_n M_{sat}^2}{1 + \frac{V_{gs} - V_h}{V_{ve}}} \approx \frac{K_n}{\left(1 + \frac{V_{gs} - V_h}{V_{ve}}\right)\left[1 + 0.78\left(\frac{V_{gs} - V_h}{V_{le}}\right)\right]}. \tag{1.223}$$

This effective transconductance coefficient accounts for mobility degradation deriving from strong vertical (gate-to-channel) electric fields through the variable, V_{ve}, as well as mobility degradation caused by lateral (drain-to-source) electric fields, which is monitored by variable V_{le}. While Equation 1.222 suggests a relatively straightforward square law dependence of drain current on the so called *excess,* or *effective, gate–source voltage*, $(V_{gs} - V_h)$, particularly for the commonly encountered situation of $(V_{ds} - V_{dsat}) \ll V_\lambda$, it should be noted that K_{ne} is inversely proportional to a quadratic function of the

excess gate–source voltage. Typically, V_{ve} is of the order of 5- to 20-fold the value of V_{le} and thus, the possibility of simplifying Equation 1.223 to ease computational strain, while preserving computational accuracy, is dubious.

Example:

An N-channel transistor featuring a channel length of 180 nM has the Level 49 HSPICE parameters given in Table 1.2. The transistor is to be biased in saturation at $V_{ds} = 1$ V and $I_d \approx 1$ mA to achieve a small-signal transconductance, g_m, of at least 3 mmhos. Assuming that the bulk terminal is incident with the transistor source terminal, choose a reasonable gate aspect ratio, W/L, determine the required gate–source voltage bias, V_{gs}, and estimate the model parameters implicit to Equation 1.222.

Results:

1. The applicable circuit for computer-aided investigation is offered in Figure 1.43, where the transistor model parameters are those that appear in Table 1.2, and the gate aspect ratio, W/L, is to be determined. The null voltage source in the drain circuit of the device facilitates the extraction of the quiescent drain current, I_d. It is understood that for biasing purposes, the area and perimeter parameters, A_s, A_d, P_s, and P_d, are of no consequence and can therefore be defaulted to any convenient value. Initially, set $V_{gs} = 1$ V and $W/L = 1$ and, of course, $V_{ds} = 1$ V. The HSPICE static simulation reveals $I_d = 46.4$ μA, $V_{dsat} = 262.8$ mV, $V_h = 519.7$ mV, and $g_m = 136.5$ μmho. Since $V_{gs} = 1$ V is certainly larger than $V_h = 519.7$ mV and $V_{ds} = 1$ V $> V_{dsat} = 262.8$ mV, the transistor is clearly turned on and operates in its saturation domain.
2. With $W/L = 1$, the simulated drain current is a factor of 21.55 times smaller than the target current of 1 mA. This observation seemingly suggests the need for increasing the gate aspect ratio from 1 to 21.55, since the drain current is ostensibly proportional to W/L. In truth, the actual drain current is not directly proportional to W/L because of numerous second order effects, including weak dependencies of threshold voltage, drain saturation voltage, and parameter M_{sat} on gate width W. Experience shows that a more viable gate aspect ratio adjustment is about twice that computed or in this case, about 40. With $W/L = 40$ and $V_{gs} = V_{ds} = 1$ V, HSPICE delivers $I_d = 1.08$ mA, $V_{dsat} = 278.2$ mV, $V_h = 510.2$ mV, and $g_m = 3.17$ mmho. The simulated transconductance value satisfies its design target. Although V_{gs} can be decreased modestly to reduce the drain current to 1 mA, this exercise is unnecessary in view of the effects of routinely encountered device processing

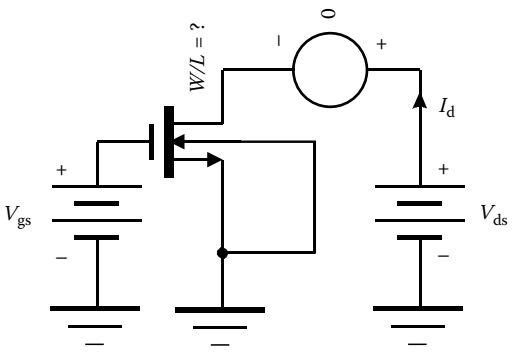


FIGURE 1.43 Circuit structure for MOSFET biasing simulation. The Level 49 HSPICE parameters of the transistor are delineated in Table 1.2.

vagaries and model parameter uncertainties. Thus the design requirement is satisfied for $W/L = 40$ and $V_{gs} = V_{ds} = 1$ V.

3. The model parameterization exercise begins by using Equation 1.141 to compute the voltage, V_{ve}. From Table 1.2, the oxide thickness is $T_{ox} = 4(10^{-9})$ m, which is 40 Å. Accordingly, $V_{ve} = 40/15 = 2.667$ V.
4. The next step in the parameterization process entails operating the transistor undergoing study at a V_{ds} value that equals its saturated value of 278.2 mV. This tack reduces the last parenthesized factor on the right-hand side of Equation 1.222 to unity, thereby simplifying the computation of the effective transconductance parameter, K_{ne}. With $W/L = 40$, $V_{gs} = 1$ V, and $V_{ds} = V_{dsat} = 278.2$ mV, HSPICE produces $I_d = 878.33$ μA and $V_h = 510.0$ mV. Appealing to Equation 1.222, parameter K_{ne} follows forthwith as $K_{ne} = 182.9$ μmho/V.
5. Recalling that $V_{dsat} = 278.2$ mV and $(V_{gs} - V_h) = (1 - 0.510)$ V $= 0.490$ V, Equation 1.129 delivers $M_{sat} = 0.5678$. The previously documented approximate equation, Equation 1.136, relating M_{sat} to variable α can be used to determine the numerical value of α for $V_{gs} = 1$ V and $V_h = 510.0$ mV. Alternatively, Equation 1.131 can be solved for α directly to yield

$$\alpha = \frac{2(1 - M_{sat})}{M_{sat}^2} = 2.682. \tag{E1.1}$$

   Using Equation 1.130, parameter V_{le} follows forthwith as $V_{le} = 182.7$ mV.
6. With $K_{ne} = 182.9$ μmho/V, $V_{ve} = 2.667$ V, $M_{sat} = 0.5678$, $V_{gs} = 1$ V, and $V_h = 510.0$ mV, the device transconductance parameter, K_n, follows from Equation 1.223 as $K_n = 671.7$ μmho/V. It is interesting to observe that the effective transconductance factor, K_{ne}, is almost 3.7 times smaller that the "actual" transconductance coefficient, K_n. Experience testifies to the apparent fact that for deep submicron devices, $2.5 \leq K_n/K_{ne} \leq 4$ is typical.
7. In principle, V_{ve}, V_{le}, V_h, V_{dsat}, K_n, and thus K_{ne}, do not vary with changes in the drain–source voltage, V_{ds}. Accordingly, the ratio of the drain current (1.08 mA) for $V_{ds} = 1$ volt to the drain current (878.33 μA) at $V_{ds} = V_{dsat} = 278.2$ mV is solely attributed to the last parenthesized factor on the right-hand side of Equation 1.222, that is,

$$\frac{I_d|_{V_{ds}=1\text{ V}}}{I_d|_{V_{ds}=V_{dsat}}} = \frac{1.08\text{ mA}}{878.33\ \mu\text{A}} = 1.230 = 1 + \frac{V_{ds} - V_{dsat}}{V_\lambda}. \tag{E1.2}$$

   It follows that the channel length modulation voltage is $V_\lambda = 3.144$ V.
8. In an attempt to demonstrate the propriety of the foregoing modeling exercise, the forward static transfer characteristic of the subject transistor is modeled in HSPICE for both $V_{ds} = 1$ V and $V_{ds} = 1.5$ V. The simulated results are then compared with calculations deriving from Equations 1.222 and 1.223 using the computed values of V_{ve}, V_{le}, and V_λ and the simulated disclosures for W/L, V_{dsat}, and V_h. Specifically, $V_{ve} = 2.667$ volts, $V_{le} = 182.7$ mV, $V_\lambda = 3.144$ volts, $W/L = 40$, $V_{dsat} = 278.2$ mV, and $V_h = 510.0$ mV.

Figures 1.44 and 1.45 display the results of the foregoing comparative study. In Figure 1.44, the simulated and calculated forward transistor characteristics in the saturation domain are displayed for a drain–source voltage, V_{ds}, of 1.0 V. The calculations corroborate reasonably well with pertinent simulations in that ±15% error is observed for $0.73\text{ V} < V_{gs} < 1.89$ V. It is notable that $V_{gs} = 0.73$ V is only slightly larger than 200 mV above threshold level, while at $V_{gs} = 1.89$ V, the transistor no longer operates in its saturation domain when $V_{ds} = 1$ V. Figure 1.45 confirms better corroboration between calculated and simulated results for $V_{ds} = 1.5$ V. In particular, the computational error is within ±9% for $0.92\text{ V} < V_{gs} < 2$ V and is within ±15% for $0.78\text{ V} < V_{gs} < 2$ V.

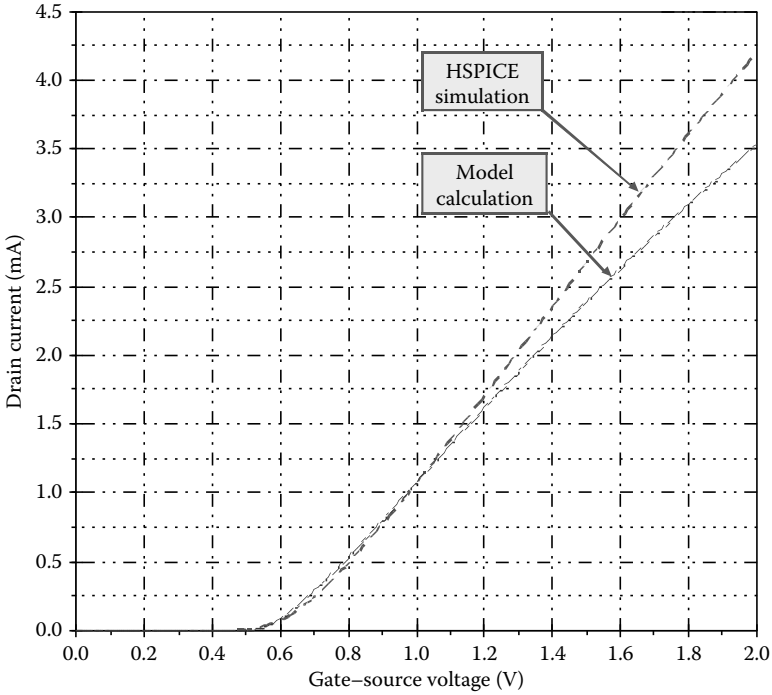


FIGURE 1.44 Simulated and calculated forward static transfer characteristic for the NMOS transistor whose model parameters are delineated in Table 1.2. The transistor is operated at a drain–source voltage, V_{ds}, of 1 V.

Comments: In Step #2 of the foregoing computational procedure, the gate aspect ratio, W/L, is the pivotal metric for achieving the desired transconductance and transistor drain current. If power dissipation is a dominant design concern, W/L can be increased above the value of 40 discerned in this example, with the understanding that the gate–source voltage, V_{gs}, can be reduced commensurately, thereby reducing the static drain current and hence, the power dissipation of the transistor. Of course, the primary penalty of large gate aspect ratio is a possible degradation of high-frequency circuit response since, as is confirmed by Equation 1.154, the capacitance area and peripheral dimensions increase in proportion to the gate width, W.

In Step #3, the metric, V_{ve}, is evaluated in terms of a purely empirical, and indeed crude first order, relationship to the oxide thickness, T_{ox}. A possible way around this dilemma is to compute V_{ve} and all of the other requisite modeling parameters by curve fitting Equation 1.222 to simulated or actually measured static data. While this approach may be academically satisfying, it may be imprudent from a design time perspective. Keep in mind that biasing is not the fundamental performance objective of an analog circuit; rather, biasing is the necessary condition that expedites the desired analog responses.

The drain saturation voltage, V_{dsat}, is obviously a nonlinear function of the excess gate voltage, $(V_{gs}-V_h)$, owing to the parameter, M_{sat}. But in addition, V_{dsat} changes slightly with the applied drain–source voltage, V_{ds}. Indeed, the Level 49 model parameters account for a slight sensitivity of threshold voltage on V_{ds}, which is as anticipated since the interface potential throughout the entire channel varies somewhat as a function of the lateral (drain-to-source) field engendered by V_{ds}.

Finally, it should be noted that the computed value (3.144 V) of the channel length modulation voltage, V_λ, is appreciably smaller than values often propounded in the textbook literature. However, V_λ is indeed a relatively small voltage for deep submicron MOS technology transistors. This anemic voltage is the principle cause of correspondingly small drain–source channel resistances, which renders the

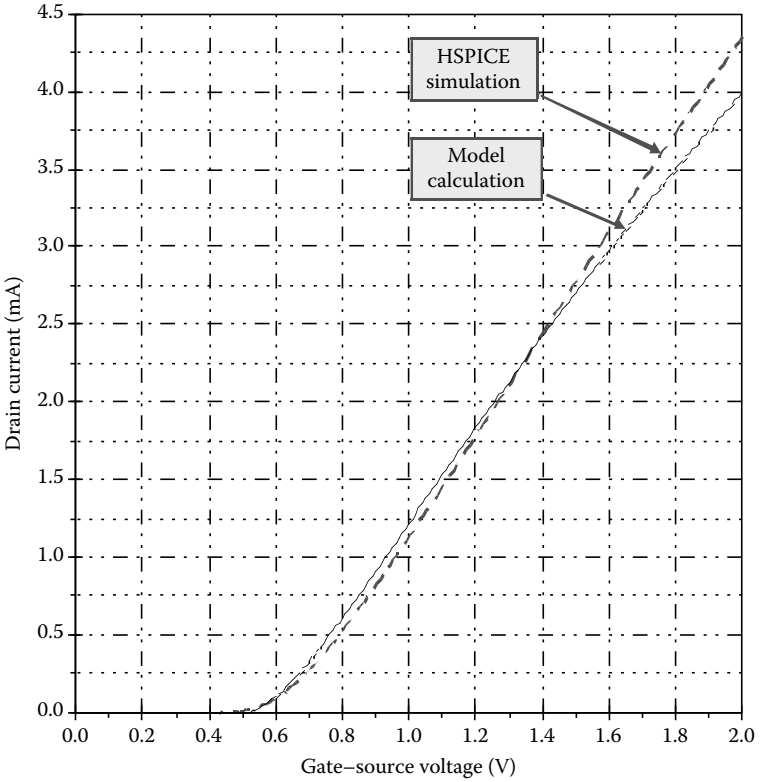


FIGURE 1.45 Simulated and calculated forward static transfer characteristic for the NMOS transistor whose model parameters are delineated in Table 1.2. The transistor is operated at a drain–source voltage, V_{ds}, of 1.5 V.

realization of transconductor amplifiers, as might be used in operational transconductor amplifier-capacitor (OTA-C) filters, a daunting challenge. The desire for accuracy surrounding the enumeration of V_λ is exacerbated by the fact that parameter V_λ is not the constant that is presumed tacitly in the foregoing demonstration. Instead, and as is suggested by Equation 1.114, V_λ is functionally dependent on drain–source voltage, drain saturation voltage, and threshold voltage. If V_λ or the drain–source channel resistance is critical in an analog circuit design endeavor, care must therefore be exercised to ensure that model parameters are extracted in terms of measured or simulated data that largely mirror the desired or expected operating state of the utilized transistor.

References

1. S. W. Sze, *Physics of Semiconductor Devices*. New York: John Wiley & Sons, 1969, pp. 366–379.
2. A. B. Glaser and G. E. Subak-Sharpe, *Integrated Circuit Engineering: Design, Fabrication, and Applications*. Reading, MA: Addison-Wesley Publishing Company, 1977, pp. 80–94.
3. A. S. Grove, *Physics and Technology of Semiconductor Devices*. New York: John Wiley & Sons, 1967, pp. 263–285.
4. A. Goetzberger, Ideal MOS curves for silicon, *Bell System Technical Journal*, 45, 1097, 1966.
5. S. R. Hofstein and G. Warfield, Physical limitation on the frequency response of a semiconductor surface inversion layer, *Solid State Electronics*, 8, 321, 1965.
6. A. S. Grove, E. H. Snow, B. E. Deal, and C. T. Sah, Simple physical model for the space-charge capacitance of metal–oxide–semiconductor structures, *Journal of Applied Physics*, 33, 2458, 1964.

7. R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Techniques for Analog and Digital Circuits*. New York: McGraw-Hill Publishing Company, 1990, pp. 174–177.
8. D. Johns and K. Martin, *Analog Integrated Circuit Design*. New York: John Wiley & Sons, Inc, 1997, pp. 24–27.
9. T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd Ed. Cambridge, United Kingdom: Cambridge University Press, 2004, Chaps. 11 and 12.
10. A. van der Ziel, *Noise in Solid State Devices and Circuits*. New York: John Wiley & Sons, Inc, 1986.
11. J. Choma and W.-K. Chen, *Feedback Networks: Theory and Applications*. Singapore: World Scientific Press, 2007, Chap. 3.

1.3 JFET, MESFET, and HEMT Technology and Devices

Stephen I. Long

1.3.1 Introduction

Many types of field effect devices are used in analog IC and RFIC design. Section 1.2 described the MOSFET and associated device models. MOSFETs are currently the predominant field effect device used in analog circuit applications due to the pervasive CMOS technology. CMOS fabrication is relatively inexpensive when not scaled below 0.25 μm. However, mask costs for 130 nm and below increase very rapidly, limiting applications to only those requiring extremely high volume. Also, drain breakdown voltage is quite low, of the order of 1 V for 65 nm CMOS. This constrains dynamic range or power output in certain applications.

Other field effect devices are available, but are considered niche market devices in most cases. This would include the legacy silicon JFET technology, still used in conjunction with bipolar transistors for some lower frequency analog applications. Compound semiconductor-based field effect devices (MESFET, HEMT, p-HEMT, m-HEMT) are often the FET of choice for applications requiring very wide bandwidth, extremely low noise, high gain at mm-wave frequencies, and high output powers at frequencies above 2 GHz. Cost of fabrication is frequently less than that of CMOS in smaller volume applications because the mask set costs are typically an order of magnitude less. Also, the compound semiconductor devices are grown on semi-insulating substrates. Passive components such as spiral inductors, MIM capacitors and deposited resistors have less parasitic capacitance and higher Q than is typical for silicon-based RFICs.

In this section, the silicon JFET and the main compound semiconductor HEMT devices will be described. Special emphasis will be placed on the GaN HEMTs whose performance is exceptionally good for microwave and mm-wave power amplifiers.

1.3.2 Silicon JFET Device Operation and Technology

Although the silicon JFET is today a legacy device, it is still used in some bipolar analog ICs to provide an inexpensive BiFET IC technology. Also, the description of its current–voltage characteristic is similar to any FET which uses a pn or Schottky metal–semiconductor junction for the gate electrode. The JFET consists of a conductive channel with source and drain contacts whose conductance is controlled by a gate electrode. The channel can be fabricated in either conductivity type, n or p, and both normally-on (depletion mode) and normally-off (enhancement mode) type devices are possible. The circuit symbols typically used for JFETs are shown in Figure 1.46 along with the bias polarities of active region operation for these four device possibilities. For analog circuit applications, the depletion mode is almost exclusively utilized because it provides a larger range of input voltage and therefore greater dynamic range. In silicon,

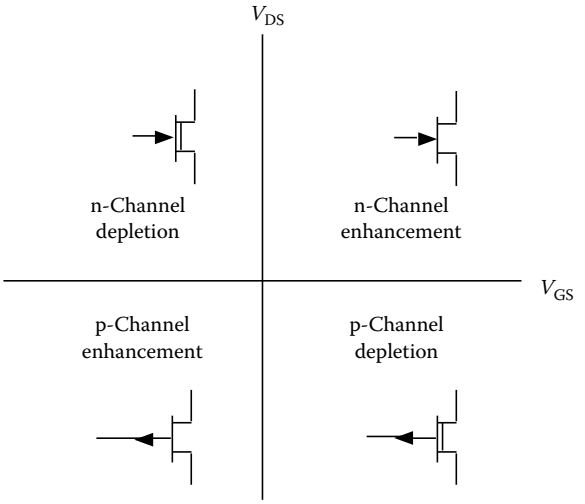


FIGURE 1.46 The circuit symbols typically used for JFETs are shown with the bias polarities for active region operation.

both p- and n-channel JFETs are used, but when compound semiconductor materials such as GaAs or InGaAs are used to build the FET, n-channel devices are used almost exclusively.

When fabricated with silicon, the JFET is used in analog IC processes for its high input impedance, limited by the depletion capacitance and leakage current of a reverse-biased pn junction. When the JFETs are used at the input stage, an op-amp with low input bias current, at least at room temperature, can be built. Fortunately, a p-channel JFET can be fabricated with a standard bipolar process with few additional process steps. This enables inexpensive BiFET processes to be employed for such applications. Unfortunately, the simple process modifications required for integrating JFETs and BJTs are not consistent with the requirements for high-performance devices. Short-gate lengths and high-channel doping levels are generally not possible. So the transconductance per channel width and the gain–bandwidth product of JFETs integrated with a traditional analog BJT process are not very good. The short-circuit current gain–bandwidth product (f_T) is about 50 MHz for an integrated p-channel JFET. The MOSFETs in a BiCMOS process are much better devices, however, a BiCMOS process does not often include both NPN and PNP BJTs needed for high-performance analog circuits.

Discrete silicon JFETs are available with much better performance because they can be fabricated with a process optimized for the JFET. Typical applications are for low-noise amplifiers up to the VHF/UHF range. Noise figures less than 0.1 dB can be obtained at low frequencies with high source impedances and 2 dB at high frequencies at the noise matching input condition with high performance discrete silicon JFETs. The low input gate current, I_G, which can be in the picoamp range, causes the shot noise (proportional to $\sqrt{I_G}$) component to be very low. The input equivalent noise current of the JFET is mainly due to input referred channel (Johnson) noise. This property gives very low noise performance when presented with a high source impedance. In this case, the JFET is often superior to a BJT for noise. For low source impedances, the BJT is generally better.

Compound semiconductor materials such as GaAs and InGaAs are used to fabricate JFET-like devices called metal-semiconductor FET (MESFETs) and high electron mobility transistor (HEMTs). The reason for using these materials is superior performance at high frequencies. These devices are unequaled for gain–bandwidth, ultralow noise, and power amplification at frequencies above 10 GHz and up to 300 GHz. Integrated analog microwave circuits are fabricated with these devices and are commercially available for use in low noise receiver and power amplifier applications. Some representative results will be summarized in Table 1.5.

1.3.2.1 JFET Static *I–V* Characteristics

The JFET differs in structure and in the details of its operation from the MOSFET discussed in Section 1.3.2. Figure 1.47 shows an idealized cross section of a JFET. The channel consists of a doped region, which can be either p- or n-type, with source and drain contacts at each end. The channel is generally isolated from its surrounding substrate material by a reverse biased p–n junction. The depletion regions are bounded in Figure 1.47 by dashed lines and are unshaded. The thin, doped channel region forms a resistor of width W into the page and height d. A gate electrode is located at the center of the channel, defined by a semiconductor region of opposite conductivity type of length L. An n-channel structure is shown here for purposes of illustration. The p-type gate constricts the channel, both through the depth of the diffusion or implant used to produce the gate and through the depletion layer formed at the p–n junction. The height of the channel can be varied by biasing the gate relative to the source (V_{GS}). A reverse bias increases the depletion layer thickness, reducing the channel height and the drain current. If V_{GS} is large enough that the channel is completely depleted, the drain current will become very small. This condition corresponds to the cutoff and subthreshold current regions of operation, and the V_{GS} required to cut-off the channel is called V_P, the pinch-off voltage. V_P corresponds to the threshold voltage that was defined for the MOSFET. Similarly, a forward bias between gate and channel can be used to increase drain current, up to the point where the gate junction begins to conduct. Most JFETs are designed to be depletion-mode (normally on); drain current can flow when $V_{GS} = 0$ and they are normally operated with a reverse-biased gate junction. It is also possible, however, to fabricate enhancement-mode JFETs by use of a thinner or more lightly doped channel.

The pinch-off voltage is a sensitive function of the doping and thickness of the channel region. It can be found if the channel-doping profile, $N(x)$, is known through Poisson's equation. For a nonuniform profile,

$$V_P = V_{BI} - \frac{q}{\varepsilon}\int_0^d xN(x)\mathrm{d}x \tag{1.224}$$

For uniform doping, $N(x) = N_D$ and the familiar result in Equation 1.225 shows that the pinch-off voltage depends on the square of the thickness. This result shows that very precise control of profile depth is needed if good matching and reproducibility of pinch-off voltage is to be obtained [7].

$$V_P - V_{BI} - \frac{qN_D d^2}{2\varepsilon} \tag{1.225}$$

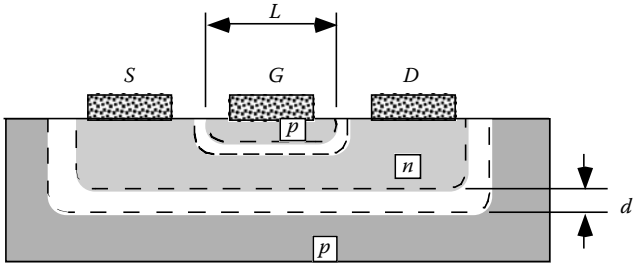


FIGURE 1.47 Idealized cross section of a JFET. The depletion regions are bounded with dashed lines and are unshaded.

1.3.2.2 JFET Operating Regions

The static current–voltage characteristics of the JFET can be categorized by the five regions of operation shown in Figure 1.48 for an n-channel device. The mechanisms that produce these regions can be qualitatively understood by referring to the channel cross sections in Figure 1.49. In these figures, the doped channel region is shaded, and the depletion region is white. First, consider the JFET in Figure 1.49a with small V_{DS} ($\ll V_{GS} - V_P$). This condition corresponds to the ohmic region (sometimes called linear or triode region) where current and voltage are linearly related. At small drain voltages, the depletion layer height is nearly uniform, the electric fields in the channel are too small to saturate the carrier velocity, and thus the channel behaves like a linear resistor. The resistance can be varied by changing V_{GS}. The channel height is reduced by increasing the reverse bias on the gate leading to an increased resistance.

As V_{DS} increases, the depletion layer thickness grows down the length of the channel as shown in Figure 1.49b. This occurs because the drain current causes a voltage increase along the channel as it flows through the channel resistance. Since the depletion layer thickness is governed by the gate-to-channel voltage (V_{GC}), there is an increasing reverse bias that leads to constriction of the channel at the drain end of the gate. Ideally, when $V_{DS} = V_{GS} - V_P$, then $V_{GC} = V_P$, and the channel height will approach zero (pinch-off). The constricted channel will cause the drain current to saturate as shown. Further increases in V_{DS} do not cause the drain current to increase since the channel has already constricted to a minimum height and the additional potential is accommodated by lateral extension of the depletion region at the drain end of the gate. This region of operation is generally described as the pinch-off region (rather than the saturation region in order to avoid confusion with BJT saturation). The height of the channel is not actually zero but is limited by the mobile channel charge, which travels at saturated drift velocity in this high field region.

If $V_{GS} < 0$, then the initial channel height at the source is reduced, I_D is less, and the pinch-off region occurs at a smaller drain voltage $V_{DS} = V_{GS} - V_P$. The saturation of drain current can also occur at smaller V_{DS} if the gate length is very small. In this case, the electric field in the channel is large, and the carrier velocity will saturate before the channel can reach pinch-off. Velocity saturation will also limit drain current.

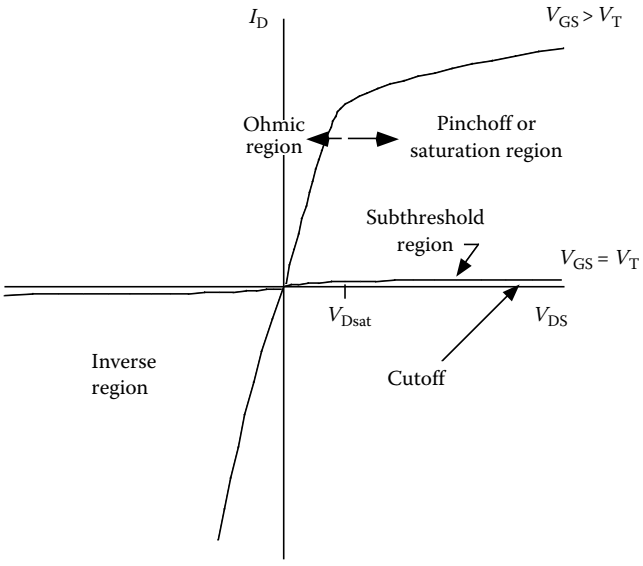


FIGURE 1.48 The static current–voltage characteristics of the JFET can be categorized by five regions of operation. An n-channel device is shown in this illustration.

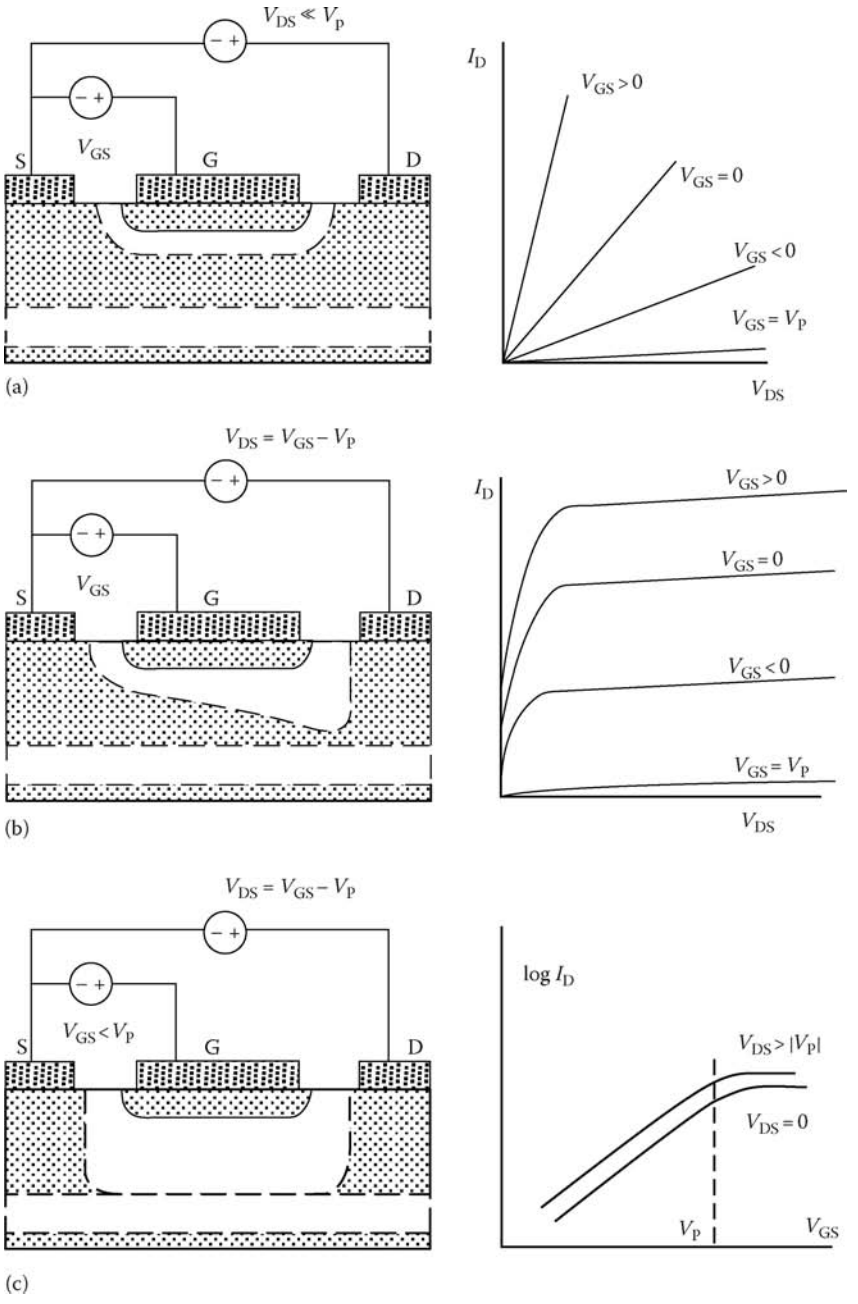


FIGURE 1.49 (a) Ohmic region with small V_{DS} ($\ll V_{GS} - V_P$). (b) When $V_{DS} = V_{GS} - V_P$, the channel height will become narrow at the drain end of the gate. The device enters pinch-off. The constricted channel will cause the drain current to saturate as shown. (c) Cutoff and subthreshold current regions occur when the depletion region extends through the channel.

The subthreshold region of operation, shown in Figure 1.49c is defined when small drain currents continue to flow even though $V_{GS} \leq V_P$. While technically this gate bias should produce cutoff, some small fraction of the electrons from the source region will have sufficient energy to overcome the

potential barrier caused by the gate depletion region and will drift into the drain region and produce a current. Since the energy distribution is exponential with potential, the current flow in this region varies exponentially with V_{GS}.

The inverse region occurs when the polarity of the drain bias is reversed. This region is of little interest for the JFET since gate-to-drain conduction of the gate diode limits the operation to the linear region only.

1.3.2.3 Channel-Length Modulation Effect

A close look at the *I–V* characteristic in the pinch-off region shows that the incremental conductivity or slope of this region is not equal to zero. There is some finite slope that is not expected from the simple velocity saturation or pinch-off models. Channel length modulation is one explanation for this increase; the position under the gate where pinch-off or velocity-saturation first occurs moves toward the source as V_{DS} increases. This is due to the expansion of the drain side depletion region at large V_{DS}. Figure 1.50 illustrates this point. Here, a channel cross section is shown for $V_{DS} = V_{GS} - V_P$ in Figure 1.50a and for $V_{DS} \gg V_{GS} - V_P$ in Figure 1.50b. While pinch-off always occurs when the gate-to-channel voltage is V_P, the higher drain voltage causes the location of this point ($x = L$) to move closer to the source end of the channel. Since the electric field in this region, E, is roughly proportional to $(V_{GS} - V_P)/L$ where L is now a function of V_{DS} and V_{GS} and the carrier velocity $v = \mu E$ (by assumption), then the current must increase as the channel length decreases due to increasing carrier velocity. If the channel length is short, velocity saturation may cause the drain current to saturate. In this case, the velocity saturation point moves closer to the source as drain voltage is increased. Since the length has decreased, less gate-to-channel voltage is needed to produce the critical field for velocity saturation. Less voltage implies a wider channel opening, hence more current.

1.3.2.4 Temperature Effects

There are two mechanisms that influence the drain current of the JFET when temperature is changed [8,9]. First, the pinch-off voltage becomes more negative (for n-channel) with increase in temperature, therefore requiring lower V_{GS} to cut off the channel or to enter the pinch-off region. Therefore, when the device is operating in the pinch-off region, and $V_{GS} - V_P$ is small, the drain current will increase with temperature. This effect is caused by the decrease in the built-in voltage of the gate-to-channel junction with increasing temperature. Second, the carrier mobility and saturated drift velocity decreases with temperature. This causes a reduction in drain current that is in opposition to the first effect. This effect dominates for large $V_{GS} - V_P$. Therefore, there is a V_{GS} value for which the drain current is exactly compensated by the two effects. This is illustrated qualitatively in Figure 1.51.

The gate current is also affected by temperature, as it is the reverse current of a pn junction. The current increases roughly by a factor of 2 for each 10°C increase in temperature. At high temperatures, the input current of a JFET input stage may become comparable to that of a well-designed BJT input stage of an op-amp, thus losing some of the benefit of the mixed BJT–JFET circuit design.

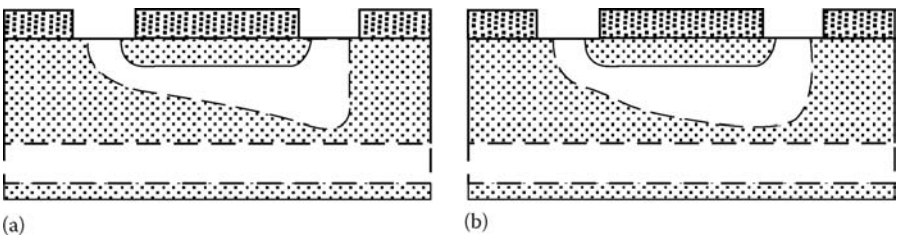


FIGURE 1.50 A channel cross section is shown for $V_{DS} = V_{GS} - V_P$ in (a) and for $V_{DS} \gg V_{GS} - V_P$ in (b). While pinch-off always occurs when the gate-to-channel voltage is V_P, the higher drain voltage causes the location of this point ($x = L$) to move closer to the source end of the channel.

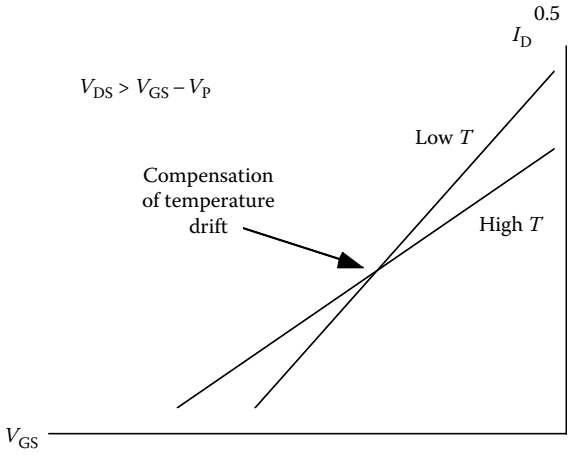


FIGURE 1.51 Effect of temperature on the drain current in the pinch-off region.

1.3.2.5 JFET Models

Most applications of the JFET in analog ICs employ the pinch-off region of operation. It is this region that provides power gain and buffer (source follower) capability for the device, so the models for the JFET presented below will concentrate on this region. It will also be assumed that the gate–source junction will not be biased into forward conduction. Although forward conduction is simple to model using the ideal diode equation within the FET equivalent circuit models, this bias condition is not useful for the principal analog circuit applications of the JFET and will also be avoided in the discussion that follows.

1.3.2.5.1 Large-Signal Model: Drain Current Equations

Equations modeling the large signal JFET $I_D - V_{GS}$ characteristic can be derived for the two extreme cases of FET operation in the pinch-off region. A gradually decreasing channel height and mobility limited drift velocity in the channel are appropriate assumptions for very long gate length FETs. A fixed channel height at pinch-off with velocity saturation limited drift velocity are more suitable for short gate lengths.

The square-law transfer characteristic [10] given by Equation 1.226 provides a good

$$I_D = I_{DSS}\left(1 - \frac{V_{GS}}{V_P}\right)^2 (1 + \lambda V_{DS}) \tag{1.226}$$

approximation to measured device characteristics in the case of long gate length (>5 μm) or very low electric fields in the channel $(V_{GS} - V_P)/L < E_{sat}$. In both cases, the channel height varies slowly and the velocity remains proportional to mobility. E_{sat} is the critical field for saturation of drift velocity, about 3.5 kV/cm for GaAs and 20 kV/cm for Si. I_{DSS} is defined as the drain current in the pinch-off region when $V_{GS} = 0$. The first two terms of the equation are useful for approximate calculation of DC biasing. The third term models the finite drain conductance caused by the channel length modulation effect. The parameter λ in this term is derived from the intercept of the drain current when extrapolated back to zero as shown in Figure 1.52.

Equation 1.226 is also used to represent the pinch-off region in the SPICE JFET model. It is parameterized in a slightly different form as shown below in Equation 1.227.

$$I_D = \beta(V_{GS,i} - V_{T0})^2 (1 + \lambda V_{DS}) \tag{1.227}$$

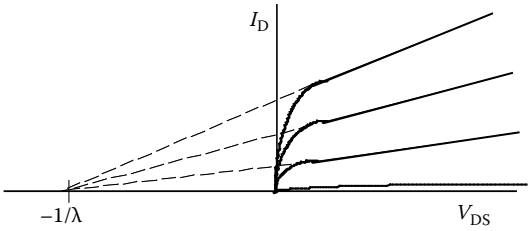


FIGURE 1.52 The channel length modulation parameter λ is defined by the extrapolation of the drain current in saturation to $I_D = 0$.

These equations are the same if $V_{T0} = V_P$, and

$$\beta = \frac{I_{DSS}}{V_P^2} \tag{1.228}$$

and

$$\begin{aligned} V_{GS,i} &= V_{GS} - I_D R_S \\ V_{DS,i} &= V_{DS} - I_E(R_S + R_D) \end{aligned} \tag{1.229}$$

The pinch-off region is defined for $V_{DS,i} \geq V_{GS,i} - V_{T0}$ as is usual for the gradual channel approximation. R_S and R_D are the parasitic source and drain resistances associated with the contacts and the part of the channel that is outside of the gate junction. These resistances will reduce the internal device voltages below the applied terminal voltages as shown in Equations 1.229.

For shorter gate length devices, improved models have been proposed and implemented in SPICE3 and some of the many commercial SPICE products, often in the MESFET model. The Statz model [11] is frequently used for this purpose. This model modifies the drain current dependence on V_{GS} by adding a velocity saturation model parameter b in the denominator as shown in Equation 1.230.

$$I_D = \left[\frac{\beta(V_{GS,i} - V_{T0})^2}{1 + b(V_{GS,i} - V_{T0})}\right](1 + \lambda V_{DS,i}) \tag{1.230}$$

This added term allows the drain current to be nearly square law in V_{GS} for small $V_{GS} - V_{T0}$, but it becomes almost linear when V_{GS} is large, effectively emulating the rapid rise in transconductance followed by saturation that is typical in short channel devices. Although the specific behavior of the drain current is sensitive to the vertical doping profile in the channel), Equation 1.230 is flexible enough to accommodate most short channel device characteristics with uniform or nonuniform channel doping. Another feature of short gate length FETs that this model predicts adequately is a saturation of I_D at $V_{DS,i} < V_{GS,i} - V_{T0}$. This early transition into the pinch-off region is also a consequence of velocity saturation and is widely observed.

1.3.2.5.2 Small-Signal Model

The small-signal model for the JFET in the pinch-off region is shown in Figure 1.53. The voltage dependent current source models the transconductance g_m as a constant which can be derived from the drain current equations above from

$$g_m = \frac{\partial I_D}{\partial V_{GS}} \tag{1.231}$$

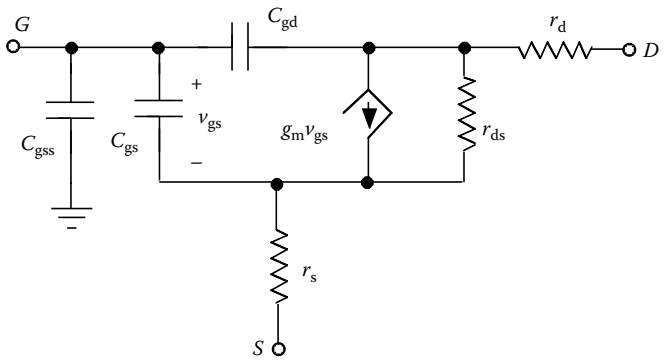


FIGURE 1.53 Small-signal model for the JFET in the pinch-off region.

The square-law current model (Equation 1.226) predicts a linearly increasing g_m with V_{GS}

$$g_m = -\frac{2I_{DSS}}{V_P}\left(1 - \frac{V_{GS}}{V_P}\right) \tag{1.232}$$

whereas a model which includes some velocity saturation effects such as Equation 1.230 would predict a saturation in g_m.

The small-signal output resistance, r_o, models the channel length modulation effect. This is also derived from the drain current equations through

$$r_o^{-1} = \frac{\partial I_D}{\partial V_{DS}} \tag{1.233}$$

For both models, r_o is determined by

$$r_o = \frac{1}{I_D \lambda} \tag{1.234}$$

The small-signal capacitors representing the nonlinear, voltage-dependent C_{gs}, C_{gd}, and C_{gss} are also shown in Figure 1.53. Parasitic source and drain resistances, R_S and R_D can also be included, as shown. If they are not included in the small-signal model, the effect of these parasitics can sometimes be produced in the intrinsic FET model by reducing the intrinsic g_m of the device.

The short-circuit current gain–bandwidth product, f_T, defined in Equation 1.235 is a high-frequency figure of merit for transistors. It is inversely proportional to the transit time τ of the channel charge, and it is increased by reducing the gate length. Reduced L also reduces the gate capacitance and increases transconductance. The material also affects f_T as higher drift velocity leads to higher g_m.

$$f_T = \frac{g_m}{2\pi\left(C_{gss} + C_{gs} + C_{gd}\right)} = \frac{1}{\tau} \tag{1.235}$$

1.3.2.6 Silicon JFET Technologies

The IC fabrication technology used to make JFETs depends primarily on the material. Discrete Si JFETs are available that provide f_T above 500 MHz and very low input rms noise currents through optimizing the channel design and minimizing parasitic capacitances, resistances, and gate diode leakage currents.

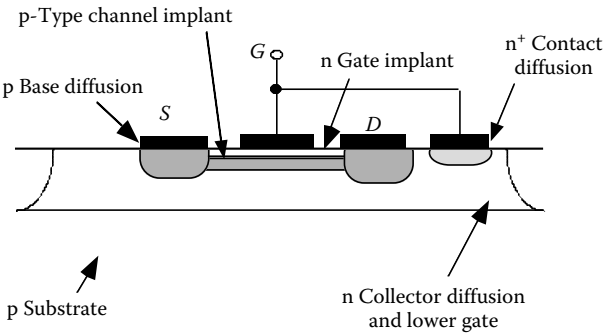


FIGURE 1.54 Cross section of an ion implanted silicon JFET (not to scale).

However, a silicon IC process is rarely designed to optimize the performance of the JFET; rather, the JFET is made to accommodate an existing bipolar process with as few modifications as possible [10]. Then, the extra circuit design flexibility and performance benefits of a relatively inexpensive mixed FET/BJT process (often called BiFET) can be obtained with small incremental cost.

In principle, it would be possible to build p-channel Si JFETs in a standard analog BJT process without additional mask steps if the base diffusion had suitable doping and thickness to give a useful pinch-off voltage when overlaid with the emitter diffusion. Unfortunately, this is usually not the case, since the emitter diffusion is too shallow, and the pinch-off voltage resulting from this approach would be too high (positive in the case of the p-channel device). Therefore, the channel of the JFET must be made thinner either through the use of an additional diffusion or by providing the channel and gate with ion implantations.

In analog ICs applications, silicon JFETs are passengers on a bipolar process; they must be compatible with the BJT process that they inhabit. Most flexibility in the JFET design is achieved using the ion implantation method. Figure 1.54 illustrates the cross section of an ion implanted JFET. In order to gain good control of the pinch-off voltage and transconductance, both the channel and the gate are formed by ion implantation. In addition, the forced compatibility with the BJT process requires use of the collector layer under the channel. This forms a lower gate electrode which is less heavily doped than the channel. Therefore, the depletion region at this interface extends primarily into the collector region, and the lower gate is less effective in contributing to the total transconductance of the JFET. It does add the parasitic capacitance C_{gss} to the device at the collector to substrate junction, limiting frequency response. In addition, the predeposition of channel and gate charge is much more repeatable with ion implantation than with earlier double diffusion methods, so device matching and reproducibility of pinch-off voltage is greatly improved. The f_T will be improved by the larger g_m per unit width and the slightly reduced gate capacitances, and the drain breakdown voltage will be increased as is often needed for an analog IC process. However, low-channel doping is not a good recipe for a high-frequency transistor with short gate length, so the f_T of these devices is still only 50 MHz or so.

1.3.3 Compound Semiconductor FET Technologies

An introduction to compound semiconductor materials will be presented in this section to establish the underlying rationale for using these materials for extremely high-performance MMIC and RFIC applications. The transport properties of typical III–V materials are compared with silicon and SiGe alloys.

There is no denying that silicon is the workhorse of the semiconductor industry. Large, high-quality substrates are relatively inexpensive, a highly stable oxide can be grown with low interface state density, and a highly advanced processing technology has enabled extremely large circuit density and extremely

fine lines to be achieved with low parasitic capacitances. Its greatest weakness for electronic device applications is the relatively low electron velocity and mobility. These intrinsic properties lead to higher transit times and access resistances, respectively, a limitation on high frequency device performance. The deeply scaled submicron technology has compensated for these deficiencies to some degree by aggressive reduction in gate length or base width. Also, p-SiGe has higher hole mobility than p-Si, so access resistance can be improved. And, using the strain induced by local depositions of SiGe in MOSFETs increases electron and hole mobilities. As good as Si IC technology is, there exist compound semiconductor materials whose intrinsic electron velocity and mobility are greatly superior to Si and so can potentially offer higher frequency, higher speed or higher power performance.

The III–V FET and bipolar device technology can provide the highest frequency and lowest noise circuit applications. Its main limitation is density. Device footprints are often significantly larger than those of similar Si devices. Thus, the high intrinsic performance of these devices is achieved in circuits of relatively low complexity.

1.3.3.1 Defining III–V Compound Semiconductors

The compound semiconductor family, as traditionally defined, is composed of the group III and group V elements shown in Table 1.3 [12]. Each semiconductor is formed from at least one group III and one group V element.

The main motivation for using the III–V compound semiconductors for device applications is found in their electronic properties when compared with those of the dominant semiconductor material, silicon. Figure 1.55 is a plot of steady-state electron velocity of several n-type semiconductors versus electric

TABLE 1.3 The Group II–VI Elements

II	III	IV	V	VI
Be	B	C	N	O
Mg	Al	Si	P	S
Zn	Ga	Ge	As	Se
Cd	In	Sn	Sb	Te

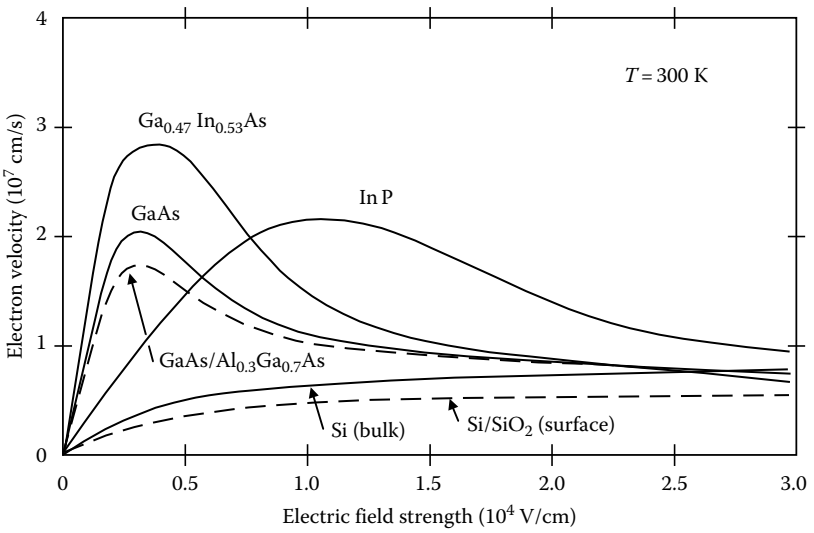


FIGURE 1.55 Electron velocity versus electric field for several n-type semiconductors.

field [12]. From this graph, we see that at low electric fields the slope of the III–V semiconductor curves (mobility) is higher than that of silicon. High mobility means that the semiconductor resistivity will be less for III–V n-type materials, and it therefore will be easier to achieve lower access resistance. Access resistance is the series resistance between the device contacts and the internal active region. An example would be the base resistance of a bipolar transistor or source resistance of a FET. Lower resistance will reduce some of the fundamental device time constants that often dominate device high frequency performance. Figure 1.55 also shows that the peak electron velocity is higher for the III–V's, and the peak velocity can be achieved at much lower electric fields. High velocity reduces transit time, the time required for a charge carrier to travel from its source to its destination, and improves device high-frequency performance. Achieving this high velocity at lower electric fields means that the devices will reach their peak performance at lower voltages, useful for low power, high-speed applications. Higher velocity of electrons also increases the current density of a device since current is the product of charge and velocity. Mobility and peak velocities of several semiconductors are compared in Table 1.4 [12].

The higher velocities are a consequence of the band structure of III–V materials. Since Si is an indirect bandgap material, conduction electrons reside in a high effective mass conduction band (CB). Mobility is dominated by the high effective mass. At high electric fields, the optical phonon generation process limits the maximum achievable electron drift velocity. GaAs, on the other hand, is direct gap, the electron mobility is high because of the lower energy, low effective mass CB where conduction electrons are confined at low fields. However, the average electron velocity will be reduced at higher electric fields due to scattering into the higher mass CB. This produces a saturated drift velocity less than the peak drift velocity, typical of the direct-gap III–V's.

To obtain significant transit velocity improvement over silicon, one must use a ternary III–III–V semiconductor such as InGaAs. The high effective mass CB is separated by 50% of the bandgap for InGaAs, whereas for GaAs it was only 20%. Thus, the peak velocity in InGaAs can be much higher than GaAs because more energy can be transferred to the conduction electrons before they begin scattering to the high mass CB. This results in higher peak velocity, 2.7×10^7 cm/s vs. 2×10^7 cm/s for GaAs.

Also shown in Table 1.4, p-type III–V semiconductors have rather poor hole mobility when compared with elemental semiconductor materials such as silicon or germanium. Holes also reach their peak velocities at much higher electric fields than electrons. Consequently, there has been very little use of p-channel III–V FET devices. The only reason to use compound semiconductor FETs is their superb high-frequency performance. The p-channel devices cannot provide this.

TABLE 1.4 Electronic Properties of Compound Semiconductors Compared with Si and Ge

Semiconductor	E_G (eV)	ε_r	Electron Mobility (cm^2/V-s)	Hole Mobility (cm^2/V-s)	Peak Electron Velocity (cm/s)
Si (bulk)	1.12	11.7	1,450	450	NA
Ge	0.66	15.8	3,900	1,900	NA
InP	1.35 D	12.4	4,600	150	2.1×10^7
GaAs	1.42 D	13.1	8,500	400	2×10^7
$Ga_{0.47}In_{0.53}As$	0.78 D	13.9	11,000	200	2.7×10^7
InAs	0.35 D	14.6	22,600	460	4×10^7
$Al_{0.3}Ga_{0.7}As$	1.80 D	12.2	1,000	100	—
AlAs	2.17	10.1	280	—	—
$Al_{0.48}In_{0.52}As$	1.92 D	12.3	800	100	—
GaN	3.39D	9.0	1,500	30	$2.5–2.7 \times 10^7$
SiC (4H)	3.26	9.8	500		2.2×10^7

Note: In bandgap energy column the symbol "D" indicates direct bandgap, otherwise it is indirect bandgap.

1.3.3.2 Heterojunctions

Heterojunctions provide an additional degree of freedom that is widely used to improve performance of compound semiconductor FET devices. The heterojunction formed by an atomically abrupt transition between AlGaAs and GaAs, shown in the energy band diagram of Figure 1.56 [12], creates discontinuities in the valence and CBs. The CB energy discontinuity is labeled ΔE_C and the valence band discontinuity, ΔE_V. Their sum equals the energy bandgap difference between the two materials. The potential energy steps caused by these discontinuities are used as barriers to electrons or holes. The relative sizes of these potential barriers depend on the composition of the semiconductor materials on each side of the heterojunction. In this example, an electron barrier in the CB is used to confine carriers into a narrow potential energy well with triangular shape. Quantum well structures such as these are used to improve device performance through two-dimensional charge transport channels, similar to the role played by the inversion layer in MOS devices. The structure and operation of heterojunctions in FETs will be described in Section 1.3.3.

The overall principle of the use of heterojunctions is summarized in a *Central Design Principle*:

> Heterostructures use energy gap variations in addition to electric fields as forces acting on holes and electrons to control their distribution and flow [13,14].

The energy barriers can control motion of charge both across the heterojunction and in the plane of the heterojunction. In addition, heterojunctions are most widely used in light emitting devices since the compositional differences also lead to either stepped or graded index of refraction, which can be used to confine, refract, and reflect light. The barriers also control the transport of holes and electrons in the light generating regions.

Figure 1.57 shows a plot of bandgap versus lattice constant for many of the III–V semiconductors [12]. Consider GaAs as an example. GaAs and AlAs have the same lattice constant (approximately 0.56 nm) but different band gaps (1.4 and 2.2 eV, respectively). An alloy semiconductor, AlGaAs, can be grown epitaxially on a GaAs substrate wafer using standard growth techniques. The composition can be selected by the Al to Ga ratio giving a bandgap that can be chosen across the entire range from GaAs to AlAs. Since both lattice constants are essentially the same, very low lattice mismatch can be achieved for any composition of $Al_xGa_{1-x}As$. Lattice matching permits low defect density, high quality materials to be grown that have good electronic and optical properties.

It quickly becomes apparent from Figure 1.57, however, that a requirement for lattice matching to the substrate greatly restricts the combinations of materials available to the device designer. For electron devices, the low mismatch GaAs/AlAs alloys, GaSb/AlSb alloys, and ternary combinations $GaAs/Ga_{0.49}In_{0.51}P$ and $InP/In_{0.53}Ga_{0.47}As/In_{0.52}Al_{0.48}As$ alone are available. Efforts to utilize

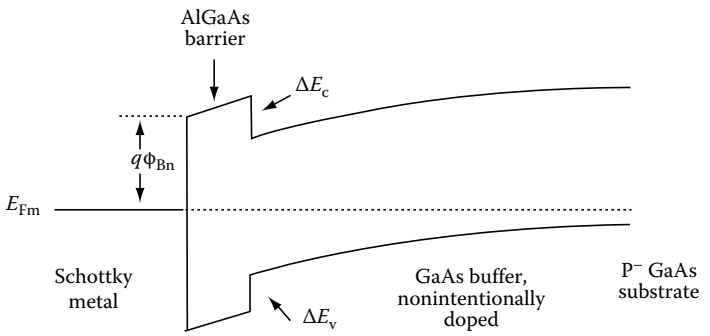


FIGURE 1.56 Energy band diagram of an abrupt heterojunction. Typical AlGaAs/GaAs HEMT band diagram.

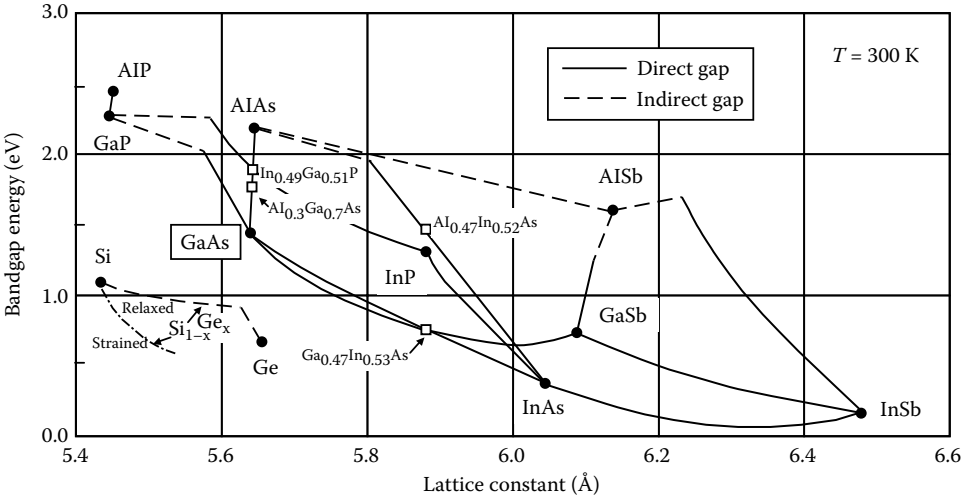


FIGURE 1.57 Energy bandgap versus lattice constant for compound semiconductor materials.

combinations such as GaP on Si or GaAs on Ge that lattice match have been generally unsuccessful because of problems with interface structure, polarization, and autodoping.

For several years, lattice matching was considered to be a necessary condition if mobility-damaging defects were to be avoided. This barrier was later broken when it was discovered that high quality semiconductor materials could still be obtained although lattice-mismatched if the thickness of the mismatched layer is sufficiently small [15,16]. This technique, called pseudomorphic growth, opened another dimension in III–V device technology, and allowed device structures to be optimized over a wider range of bandgap for better electron or hole dynamics and optical properties.

Two of the pseudomorphic systems that have been very successful in high performance millimeter-wave FETs are the InAlAs/InGaAs/GaAs and InAlAs/InGaAs/InP systems. The $In_xGa_{1-x}As$ layer is responsible for the high electron mobility and velocity which both improve as the In concentration x is increased. Up to $x = 0.25$ for GaAs substrates and $x = 0.80$ for InP substrates have been demonstrated and result in great performance enhancements when compared with lattice-matched combinations. [6]

InP substrates, however, are more expensive, smaller, and more easily broken than GaAs. And, the 3.8% lattice mismatch would seem to be too great for direct epitaxy of $In_{0.53}Ga_{0.47}As$ on GaAs substrates. It has been demonstrated, however, that good quality devices can be obtained using the metamorphic growth technique. A thick InP transition layer or a graded InGaP layer is grown directly upon a GaAs substrate. The defects caused by the lattice mismatch are largely contained in this layer, and low defect layers can be obtained when grown upon this transitional buffer layer [17,18].

1.3.3.3 Compound Semiconductor HEMT Devices

High performance GaAs MESFET* and HEMT† devices are constructed with a metal-to-semiconductor junction gate instead of a diffused or implanted pn junction gate. The metal gate forms a Schottky barrier diode directly on an n-type channel or on a wider bandgap barrier layer. In the case of the MESFET as shown in Figure 1.58 [19], the gate, directly on the n-type doped channel, forms a depletion layer which allows the channel height to be varied in the same manner as the JFET. No gate dielectric or p-type diffusion is necessary. Often the gate is deposited in a recess, etched below the surface of the channel. This allows for thicker and sometimes more highly doped regions at source and drain to be used to

* Metal–semiconductor FET.
† High electron mobility transistor.

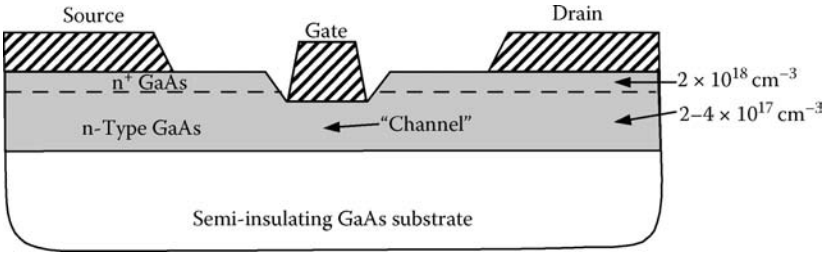


FIGURE 1.58 Cross section of recessed gate GaAs MESFET. (From Estreich, D., in *The VLSI Handbook*, CRC Press, Boca Raton, FL, 2006.)

reduce parasitic resistances. When gate cross sections are very small, for example less than 150 nm, the gate metal is taller than the width, and a thicker, wider region is often deposited on the top to reduce gate access resistance.

With the HEMT device, the gate potential modulates the height of a triangular potential well (Figure 1.56) thereby varying the channel charge available for source–drain conduction. The channel layer is confined by the triangular potential well formed at the interface between the higher bandgap barrier (InAlAs or AlGaAs) and channel (InGaAs or GaAs) as illustrated in the device cross sectional drawing in Figure 1.59 [19]. In some devices, the back side of the channel is also confined by a wide gap barrier. The confinement provided by these energy barriers provides large channel electron sheet concentrations, improving g_m and current density. The active region of the HEMT is formed by epitaxial growth of the channel and barrier region with molecular beam epitaxy.

In Figure 1.59, the device is also shown with a recessed gate. This type of structure enables the use of more highly doped, lower bandgap material on the surface to reduce parasitic source and drain resistances.

These compound semiconductor FETs are used as the primary active device in analog microwave and mm-wave monolithic integrated circuits (MMICs or RFICs). Extremely low noise figure and wide bandwidth have been obtained by the use of HEMT, p-HEMT (pseudomorphic HEMT), and m-HEMT (metamorphic HEMT) devices. These devices achieve their improved performance mainly through the high mobility, undoped InGaAs channel material. The electron velocity vs. electric field of $In_{0.53}Ga_{0.47}As$ is compared with GaAs and Si in Figure 1.55 where it can be seen that higher drift velocity is obtained in $In_{0.53}Ga_{0.47}As$ [20] than either GaAs [21] or Si [22]. The higher the In concentration in the InGaAs, the higher the mobility and velocity and the lower the noise.

Finally, the gate barrier heterojunction also enables good Schottky gate characteristics to be obtained even though the channel material itself has a low bandgap and would otherwise provide a poor barrier height if the metal were in direct contact.

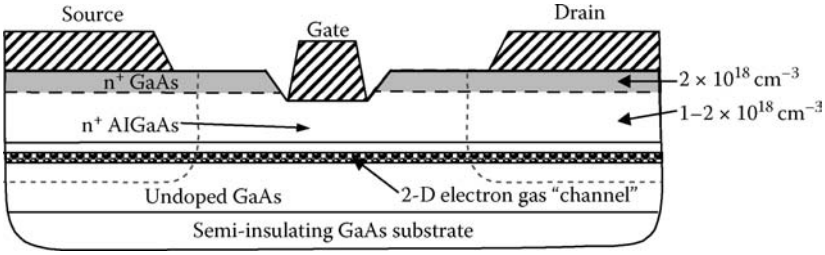


FIGURE 1.59 Cross section of recessed gate AlGaAs/GaAs HEMT device structure. (From Estreich, D., in *The VLSI Handbook*, CRC Press, Boca Raton, FL, 2006.)

TABLE 1.5 Microwave and mm-Wave Performance Comparison between Compound Semiconductor FETs

Frequency (GHz)	Device	Gain (dB)	Noise Figure (dB)	Reference
4–9	Dual Gate	21	<1.75	[1]
10–20	100 nm	17	<2.75	
20–40	GaAs pHEMT	20	<2.5	
23	130 nm	43	1.9	[2]
18–40	InP HEMT	>40	—	
0.5–80	100 nm	>17	<2.5	[3]
	InP HEMT			
70–105	50 nm	20	2.4	[4]
220	m-HEMT	21	9	
192–235	50 nm	>15	—	[5]
	m-HEMT			
270	35 nm	11.6	—	[6]
300	InP p-HEMT	6		

Excellent performance of HEMT, p-HEMT, and m-HEMT MMICs at microwave and millimeter wave frequencies has been reported. Table 1.5 presents a summary of some representative MMIC amplifiers where both narrowband and wideband amplifiers are reported. Gate lengths down to 35 nm have been successfully used for mm-wave and sub-mm-wave amplifiers.

GaAs HEMT devices also exhibit higher breakdown voltages (often 15 to 20 V) that make them suitable for power amplifier applications. In a recent article, a wideband distributed GaAs HEMT amplifier was reported that provided over 4 W of output power from 4 to 18 GHz at a 5 V drain bias with a power added efficiency of 23% [23]. Narrowband HEMT amplifiers can provide much higher efficiency and power.

1.3.3.4 Wide Bandgap Compound Semiconductors

In recent years there has been increasing interest in the wide bandgap compound semiconductors, SiC and GaN (and associated alloys of Al/In/Ga with N). The applications have been primarily for microwave power applications because the wider bandgap increases breakdown voltage while the band structure allows for high electron peak velocities in both materials [24,25]. Table 1.6 compares the fundamental physical properties of the wide-gap compound semiconductors with GaAs and Si [12]. It should be noted that there is not uniform agreement on the wide-gap parameter values from one reference to the next, but the numbers presented are representative of the current literature. As seen in Table 1.6, thermal conductivity is very high for both SiC and GaN, allowing for effective removal of heat from power devices. In fact, at room temperature, SiC has a higher thermal conductivity than any metal.

Figure 1.60 compares the electron velocity of GaN and SiC with GaAs and silicon [12,26]. The peak velocity of GaN is reached at electric fields above 150 kV/cm. Both SiC and GaN retain their good

TABLE 1.6 High Electron Mobility GaN

Material	Bandgap (eV)	Mobility (cm^2/V-s)	E_c (V/cm)	Saturation Drift Velocity (cm/s)	Thermal Conductivity (W/cm-K)
n-SiC (4H)	3.26	500	2.2×10^6	2×10^7	3.0–3.8
n-GaN	3.39	1500	3×10^6	1.5×10^7	2.2
n-GaAs	1.4	5000	3×10^5	0.6×10^7	0.45
n-Si	1.1	1300	2.5×10^5	1×10^7	1.45

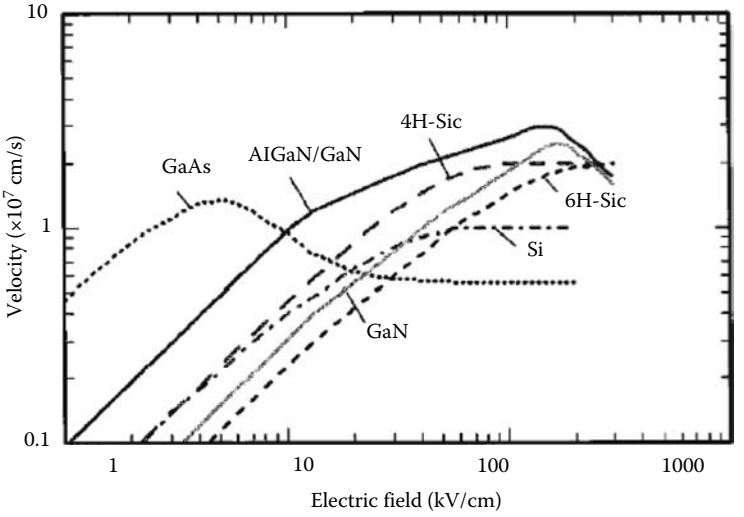


FIGURE 1.60 Electron velocity versus electric field of GaN and SiC compared with Silicon and GaAs. (From Trew, R.J., *Proc. IEEE*, 90, 1032, 2002. With permission.)

transport properties for high power applications. Table 1.6 shows that GaN has high electron mobility as well, which helps to reduce parasitic source resistance. Hole mobility for wide bandgap compound semiconductors is quite low, however, generally less than 50 $cm^2/V-s$.

There is significant lattice mismatch between a GaN channel in a heterojunction FET and the AlGaN barrier layer. However, the strain caused by this mismatch produces polarization and piezoelectric effects that induce large sheet charge densities, above $10^{13}\,cm^{-2}$ in the channel, beneficial for high current density operation of these devices [26]. This level of charge is about five times higher than what can be induced in GaAs channels in the AlGaAs/GaAs heterostructure.

Figures of merit are often employed when comparing materials for microwave power amplifier applications. Johnson's FOM [27]

$$\text{JFOM} = \frac{E_c v_{sat}}{2\pi} \tag{1.236}$$

has units of power–frequency. E_c is the maximum or critical electric field for breakdown and v_{sat} is the saturated drift velocity at high electric fields. This expresses the electronic merits of the material but neglects to consider thermal conductivity, also of importance for power electronics. Nevertheless, based on the electronic properties alone, GaN and SiC have a JFOM approximately 18 times greater than Si or GaAs. If the superior thermal conductivity is also considered, it becomes clear that these materials are extremely well suited for microwave power.

1.3.3.5 GaN HEMT Field Effect Transistors

The unusual electronic and thermal properties of the wide bandgap materials such as GaN are very attractive for applications requiring transistors with both high breakdown voltage and high frequency performance. This range of applications is focused especially on microwave power transistors. To understand why, consider first what is most desirable in a high power transistor's drain current–voltage characteristics. Figure 1.61 shows an idealized representation of this characteristic. To obtain the maximum output sinusoidal voltage and current amplitudes, one would like a device with high breakdown voltage, V_{break}, low "knee" voltage, V_{knee}, and high maximum current, I_{max}. The voltage swing is

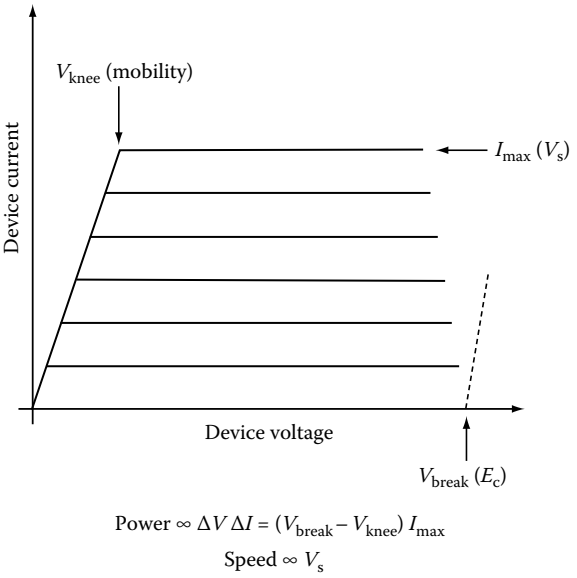


FIGURE 1.61 Idealized GaN HEMT drain current–drain voltage characteristic.

determined by $V_{break} - V_{knee}$, and the current swing by I_{max}. This combination provides high power density (W/mm) in a device, therefore requiring smaller device area. Power is proportional to

$$P_{OUT} \propto (V_{break} - V_{knee}) \times I_{max}.$$

DC to RF conversion efficiency, equally important in a power device, is proportional to

$$\text{Efficiency} \propto (1 - V_{knee}/V_{break}).$$

Thus, a large breakdown voltage is helpful for both power and efficiency.

But, Si LDMOS devices also have high breakdown voltage. So, why is GaN better? First, the current density is far higher in GaN. The high current density in the GaN HEMT is a result of the exceptionally high sheet charge, n_S, in the channel, typically 1×10^{13} cm^{-2} or higher. The high sheet charge density is a result of the static polarization that occurs at the interface between the GaN channel and the AlGaN barrier. To satisfy the charge balance at the interface, a high density of negative charge is required. A cross section of a GaN HEMT device is shown in Figure 1.62.

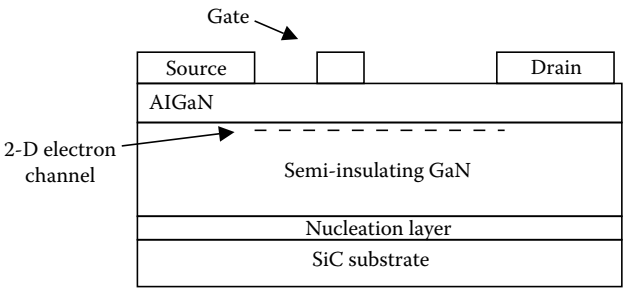


FIGURE 1.62 Cross section of GaN HEMT device structure.

Second, the high current and power density means that smaller device areas can be used to meet a particular power requirement. Smaller area translates into higher impedances because capacitances are proportional to device area. This simplifies the matching at the input and output, and therefore can lead to wider bandwidth and lower losses. The GaN devices are grown on a semi-insulating SiC substrate, thus the drain–source capacitance, C_{ds}, is small: typically about 0.25 pF/mm of device width. A 25 W device would require about 5 mm of channel width giving C_{ds} of about 1.25 pF. The Si device drain capacitance is generally quite high by comparison. For example, a 25 W LDMOS device designed for operation at 500 MHz has a C_{ds} of about 30 pF. This greatly limits the application of such devices for higher frequency or switching mode PA applications.

The current density and high frequency performance are also aided by the high saturated electron velocity in GaN. Refer once again to Figure 1.60 where electron velocity at high electric field in GaN is compared with other semiconductor materials. The electron transit time across the channel is inversely proportional to the carrier velocity; the current directly proportional.

The GaN HEMT gate structure often includes a field plate at the drain end of the gate as shown in Figure 1.63. The field plate distributes the electric field in the gate–drain region over a wider area leading to reduction in peak electric field and higher breakdown voltage. In addition, surface trap charge is less affected by potential variation on the surface, thus there is less depletion of channel charge by these traps leading to reduction in the low frequency dispersion effects.

Recent GaN HEMT microwave power amplifier performance highlights are presented in Table 1.7. Power outputs of several hundreds of watts have been obtained under low duty cycle pulsed operating conditions where thermal effects are less serious. CW output powers in the same range have been obtained by power combining of two or more amplifiers. Operation at 35 GHz was reported on MMIC PAs using devices with reduced gate length (0.15 μm) and smaller drain voltages.

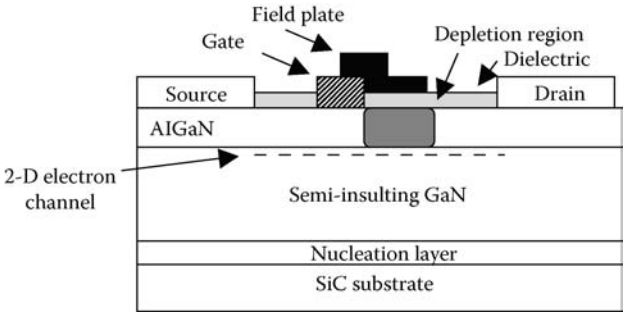


FIGURE 1.63 Cross section of GaN HEMT device structure with gate connected field plate.

TABLE 1.7 Recent GaN/HEMT Microwave Power Amplifier Performance Highlights

F (GHz)	P_{OUT} (W)	PAE (%)	V_{DC} (V)	W/mm	Reference
1.5	500[a]	49	65	13.9[a]	[28]
2.14	750[a]	—	50	7.8[a]	[29]
2.14	370	—	45	3.8	[30]
6	130[a]	45	50[a]	5.4[a]	[31]
9.5	80	34	30	3.5	[32]
35	4[b]	23	24	3.3	[33]
35	0.9[b]	51	20	4.5	[34]

[a] Pulsed CW unless otherwise noted.
[b] MMIC.

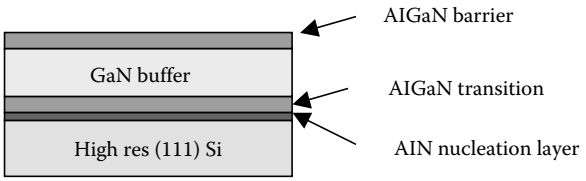


FIGURE 1.64 GaN on Si m-HEMT substrate.

GaN has also been successfully grown on high resistivity silicon substrates as illustrated by Figure 1.64. A thick nucleation layer leads to a metamorphic structure. Good power amplifier GaN on Si HEMT devices have been demonstrated to provide comparable performance in power and efficiency to those grown on Si [35,36]. The thermal conductivity of silicon is considerably less than that of SiC, however, so one would not expect the thermal resistance of the GaN on Si HEMT to be as low as the former.

1.3.4 Conclusion

While the mainstream semiconductor device and circuit technology is defined by silicon and its related materials, the superior electron transport properties of compound semiconductor materials offer unique advantages in applications requiring the highest frequency, speed and power or lowest noise figure. The range of device structural possibilities with compound semiconductor heterojunctions is far greater than what can be realized without this option, and this has led to high-performance FET structures, HEMT, p-HEMT, and m-HEMT, with excellent bandwidth, noise figure, and power.

References

1. Deal, W. R. et al., Design and analysis of broadband dual-gate balanced low-noise amplifiers, *IEEE J. Solid State Circuits*, 42(10), 2107–2115, Oct. 2007.
2. Matsuda, S. et al., Very compact high-gain broadband low-noise amplifier in InP HEMT technology, *IEEE Trans. Microwave Theory Tech*., 54(12) part 2, 4565–4571, Dec. 2006.
3. Grundbacher, R., et al., High performance millimeter wave 0.1/spl mu/m InP HEMT MMIC LNAs fabricated on 100 mm wafers, *International Conference on Indium Phosphide and Related Materials, 16th IPRM*, May 31–June 4; Kagoshima, Japan, pp. 284–287, 2004.
4. Schlechtweg, M. and A. Tessmann, From 100 GHz to terahertz electronics—activities in Europe, *IEEE Compound Semiconductor IC Symposium*, pp. 8–11, San Antonio, TX, Nov. 2006.
5. Leuther, A., et al., 50 nm MHEMT technology for G- and H-Band MMICs, *International Conference on Indium Phosphide and Related Materials, 19th IPRM*, 14–18 May, Matsue Japan, pp. 24–27, 2007.
6. Deal, W. R., et al., Demonstration of a 270-GHz MMIC amplifier using 35-nm InP HEMT technology, *IEEE Microwave Wireless Components Lett*., 17(5), 391–393, May 2007.
7. Sze, S. M., *Physics of Semiconductor Devices*. Wiley-Interscience, New York, 1981.
8. Taylor, G. W., H. M. Darley, et al., A device model for an ion-implanted MESFET. *IEEE Trans. Electron. Dev*., ED-26, 172–179, 1979.
9. Lee, S. J. and C. P. Lee, Temperature effect on low threshold voltage ion-implanted GaAs MESFETs. *Electron. Lett*., 17(20), 760–761, 1981.
10. Gray, P. R. and R. Meyer, *Analysis and Design of Analog Integrated Circuits*. 4th ed., John Wiley, New York, 2004.
11. Statz, H., P. Newman, et al., GaAs FET device and circuit simulation in SPICE, *IEEE Trans. Electron. Dev*., 34(2), 160–169, 1987.
12. Long, S. I., Compound semiconductor materials, Chap. 71 in *The VLSI Handbook*, 2nd edn., Ed. W.-K Chen, CRC Press, Boca Raton, FL, 2006.

13. Kroemer, H., Heterostructures for everything: Device principles of the 1980's? *Japanese J. Appl. Phys*., 20, 9, 1981.
14. Kroemer, H., Heterostructure bipolar transistors and integrated circuits, *Proc. IEEE*, 70, 13, 1982.
15. Matthews, J. W. and A. E. Blakeslee, Defects in epitaxial multilayers, III. Preparation of almost perfect layers, *J. Crystal Growth*, 32, 265, 1976.
16. Matthews, J. W. and A. E. Blakeslee, Coherent strain in epitaxially grown films, *J. Crystal Growth*, 27, 118, 1974.
17. Hoke, W. E., et al., Properties of metamorphic materials and device molecular beam epitaxy, *2002 International Conference on Molecular Beam Epitaxy*, 15–20 September, San Francisco, CA, pp. 69–70, 2002.
18. Schlechtweg, M., et al., Millimeter-wave and mixed-signal integrated circuits based on advanced metamorphic HEMT technology, *16th International Conference on Indium Phosphide and Related Materials*, May 31–June 4; Kagoshima, Japan, pp. 609–614, 2004.
19. Estreich, D., Compound semiconductor devices for analog and digital circuits, Chap. 72 in *The VLSI Handbook*, 2nd edn., Ed. W.-K Chen, CRC Press, Boca Raton, FL, 2006.
20. Littlejohn, M. A., et al., High-field transport in InGaAs and related heterostructures, in *Properties of Lattice-Matched and Strained Indium Gallium Arsenide*, Inspec—IEE, London, pp. 107–116, 1993.
21. Pozela, F. and A. Reklaitis, Electron transport properties in GaAs at high electric fields, *Solid-State Elect*., 23, 927–933, 1980.
22. Jacobini, C., et al., A review of some charge transport properties of Silicon, *Solid-State Elect*., 20, 77, 1977.
23. Meharry, D., et al., Multi-watt wideband MMICs in GaN and GaAs, *IEEE/MTT-S International Microwave Symposium*, June 3–June 8; Honolulu; Hawaii, pp. 631–634, 2007.
24. Gelmont, B., K. Kim, and M. Shur, Monte-Carlo simulation of electron transport in gallium nitride, *J. Appl. Phys*., 74, 1818–1821, 1993.
25. F. Schwierz, An electron mobility model for wurtzite GaN, *Solid State Elect*., 49, 889–895, 2005.
26. Trew, R. J., SiC and GaN transistors: Is there one winner for microwave power applications? *Proc. IEEE*, *Special Issue on Wide Bandgap Semiconductors*, 90, 1032–1047, June 2002
27. Johnson, E. O., Physical limitations on frequency and power parameters of transistors, *IRE International Convention Record*, Vol. 13; Part 5, pp. 27–34, March 1965.
28. Maekawa, A., et al., A 500W Push-Pull AlGaN/GaN HEMT amplifier for L band high power applications, *IEEE/MTT-S International Microwave Symposium*, IEEE Mircrowave Theory and Techniques Society, San Francisco, CA, June 11–16, pp. 722–725, 2006.
29. Wakejima, A., et al., Pulsed 0.75 kW output single-ended GaN-FET amplifier for L/S band applications, *Electron. Lett*., 42, 1349–1350, Nov. 2006.
30. Wakejima, A., et al., 370 W output power GaN-FET amplifier for W-CDMA cellular base stations, *Electron. Lett*., 41, 1371–1372, Dec. 2005.
31. Yamanaka, K., et al., C-band GaN HEMT power amplifier with 220W output power, *2007 IEEE International Microwave Symposium*, June 8, Honolulu, Hawaii, pp. 1251–1254, 2007.
32. Takagi, K., et al., Xband AlGaN/GaN HEMT with over 80 W output power, *IEEE Compound Semiconductor IC Symposium* (*CSICS*), Nov 12–15, San Antonio, TX, pp. 265–268, 2006.
33. Darwish, A., et al., 4 W Ka-band AlGaN/GaN power amplifier MMIC, *IEEE/MTT-S International Microwave Symposium*, IEEE Microwave Theory and Techniques Society, San Francisco, CA, June 11–16, pp. 730–733, 2006.
34. Kao, M.-Y., et al., AlGaN/GaN HEMTs with PAE of 53% at 35 GHz for HPA and multi-function MMIC applications, *2007 IEEE International Microwave Symposium*, Honolulu, Hawaii, June 3–8, pp. 627–629, 2007.
35. Johnson, J. W., Piner, E. L., Vescan, A., Therrien, R., Rajagopal, P., Roberts, J. C., Brown, J. D., Singhal, S., and Linthieum, K. J., 12 W/mm AlGaN/GaN HFETs on silicon substrates, *IEEE Elect. Dev. Lett*., 25(7), 459–461, July 2004.
36. Nagy, W. et al., 150 W GaN on Si RF power transistor, *IEEE/MIT-S International Microwave Symposium*, Microwave Theory and Techniques Society, Long Beach, CA, 4pp, June 2005.

1.4 Passive Components

Nhat M. Nguyen

1.4.1 Resistors

Resistors available in monolithic form are classified in general as semiconductor resistors and thin-film resistors. Semiconductor structures include diffused, pinched, epitaxial, and ion-implanted resistors. Commonly used thin-film resistors include tantalum, nickel–chromium (Ni–Cr), cermet (Cr–SiO), and tin oxide (SnO_2). Diffused, pinched, and epitaxial resistors can be fabricated along with other circuit elements without any additional processing steps. Ion-implanted and thin-film resistors require additional processing steps for monolithic integration but offer lower temperature coefficient, smaller absolute value variation, and superior high-frequency performance.

Resistor calculation. The simplified structure of a uniformly doped resistor of length L, width W, and thickness T is shown in Figure 1.65. The resistance is

$$R = \frac{1}{\sigma}\frac{L}{WT} = \left(\frac{\rho}{T}\right)\frac{L}{W} = R_{\mathrm{n}}\frac{L}{W} \tag{1.237}$$

where

σ and ρ are conductivity and resistivity of the sample, respectively
R_{N} is referred to as the *sheet resistance*

From the theory of semiconductor physics, the conductivity of a semiconductor sample is

$$\sigma = q\left(\mu_n n + \mu_p p\right) \tag{1.238}$$

where

q is the electron charge (1.6×10^{-19} C)
$\mu_n(\mathrm{cm}^2/\mathrm{V}\cdot\mathrm{s})$ is the electron mobility
$\mu_p(\mathrm{cm}^2/\mathrm{V}\cdot\mathrm{s})$ is the hole mobility
$n(\mathrm{cm}^{-3})$ is the electron concentration
$p(\mathrm{cm}^{-3})$ is the hole concentration
$\sigma(\Omega/\mathrm{cm})^{-1}$ is the electrical conductivity

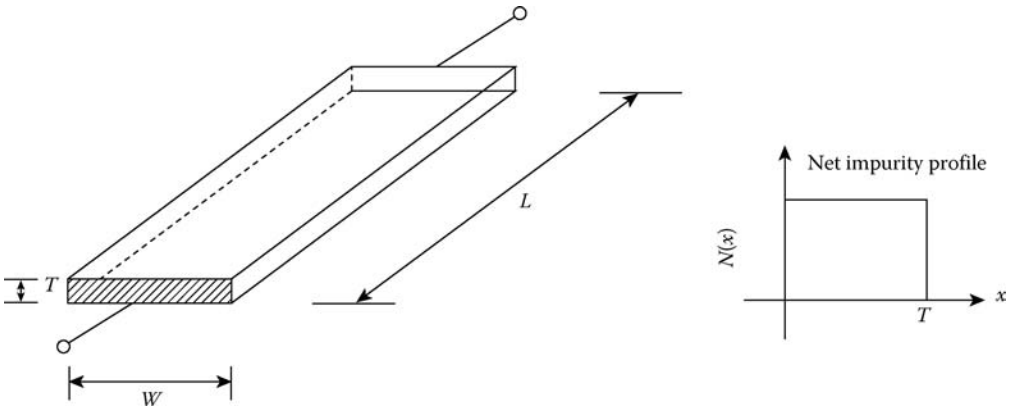


FIGURE 1.65 Simplified structure of a uniformly doped resistor.

For an n-type doped sample with a concentration $N_D(\text{cm}^{-3})$ of donor impurity atoms, the electron concentration n is approximately equal to N_D. Given the mass-action law $np = n_i^2$, the conductivity of an n-type doped sample is approximated by

$$\sigma = q\left(\mu_n N_D + \mu_p \frac{n_i^2}{N_D}\right) \approx q\mu_n N_D \tag{1.239}$$

where $n_i(\text{cm}^{-3})$ is the *intrinsic* concentration. For a p-type doped sample, the conductivity is

$$\sigma = q\left(\mu_n \frac{n_i^2}{N_A} + \mu_p N_A\right) \approx q\mu_p N_A \tag{1.240}$$

where $N_A(\text{cm}^{-3})$ is the concentration of p-type donor impurity atoms. The sheet resistance of an n-type uniformly doped resistor is thus

$$R_n = \left(\frac{1}{q\mu_n N_D T}\right) \tag{1.241}$$

For an n-type nonuniformly doped resistor as shown in Figure 1.66, where n-type impurity atoms are introduced into the p-type region by means of a high-temperature diffusion process, the sheet resistance [7] is

$$R_n = \left[\int_0^{x_j} q\mu_n N_D(x)\mathrm{d}x\right]^{-1} \tag{1.242}$$

where x_j is the distance from the surface to the edge of the junction depletion layer.

Measured values of electron mobility and hole mobility in silicon material as a function of impurity concentration are shown in Figure 1.67 [4]. The resistivity ρ (Ω-cm) of n-type and p-type silicon as a function of impurity concentration is shown in Figure 1.68 [12].

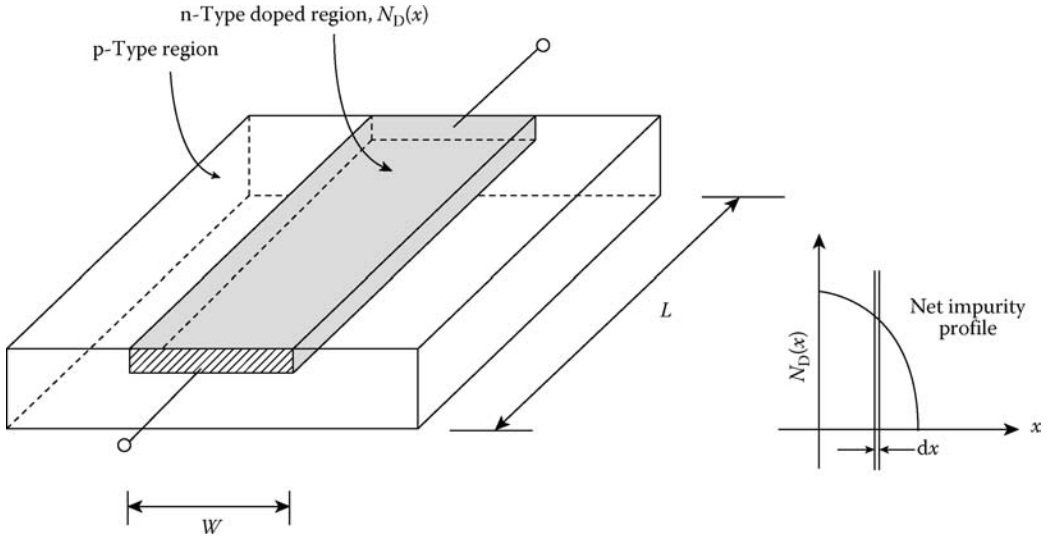


FIGURE 1.66 Simplified structure of an n-type nonuniformly doped resistor.

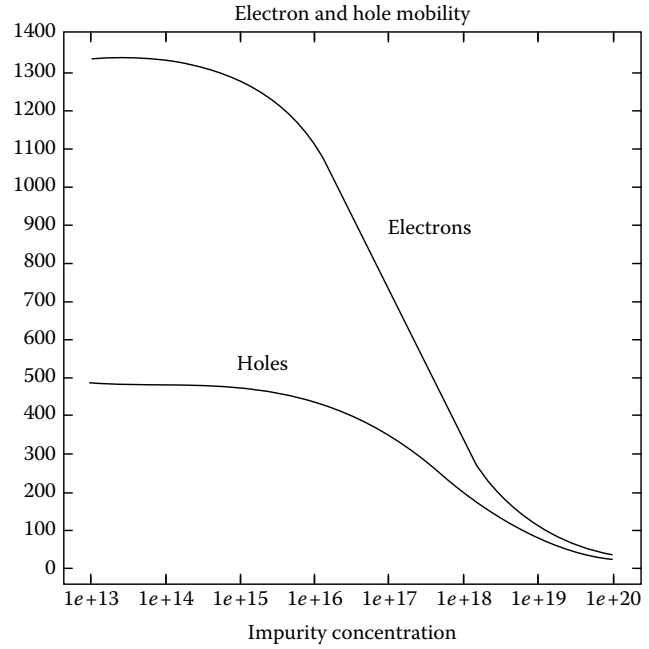


FIGURE 1.67 Electron and hole mobility vs. impurity concentration in silicon.

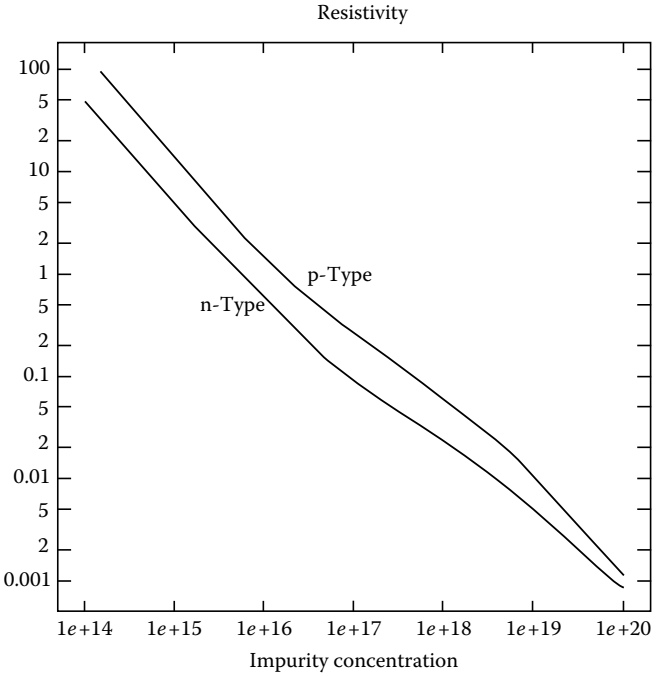


FIGURE 1.68 Resistivity of p-type and n-type silicon vs. impurity concentration.

The sheet resistance depends also on temperature since both electron mobility and hole mobility vary with temperature [17]. This effect is accounted for by utilizing a temperature coefficient quantity that measures the sheet resistance variation as a function of temperature. A mathematical model of the temperature effect is

$$R_v(T) = R_v(T_o)[(T - T_o)\text{TC}] \tag{1.243}$$

where

T_o is the room temperature

"TC" is the temperature coefficient

1.4.1.1 Diffused Resistors

In metal-oxide-semiconductor (MOS) technology, the diffused layer forming the source and drain of the MOS transistors can be used to form a diffused resistor. In silicon bipolar technology, the available diffused layers are base diffusion, emitter diffusion, active base region, and epitaxial layer.

Base-diffused resistors. The structure of a typical base diffused resistor is shown in Figure 1.69, where the substrate material is assumed of p-type silicon material. The diffused resistor is formed by using the p-type base diffusion of the npn transistors. The resistor contacts are formed by etching selected windows of the SiO_2 passivation layer and depositing thin films of conductive metallic material. The isolation region can be formed with either a p-type doped junction or a trench filled with SiO_2 dielectric material. The pn junction formed by the p-type resistor and the n-type epitaxial (epi) layer must be reverse biased

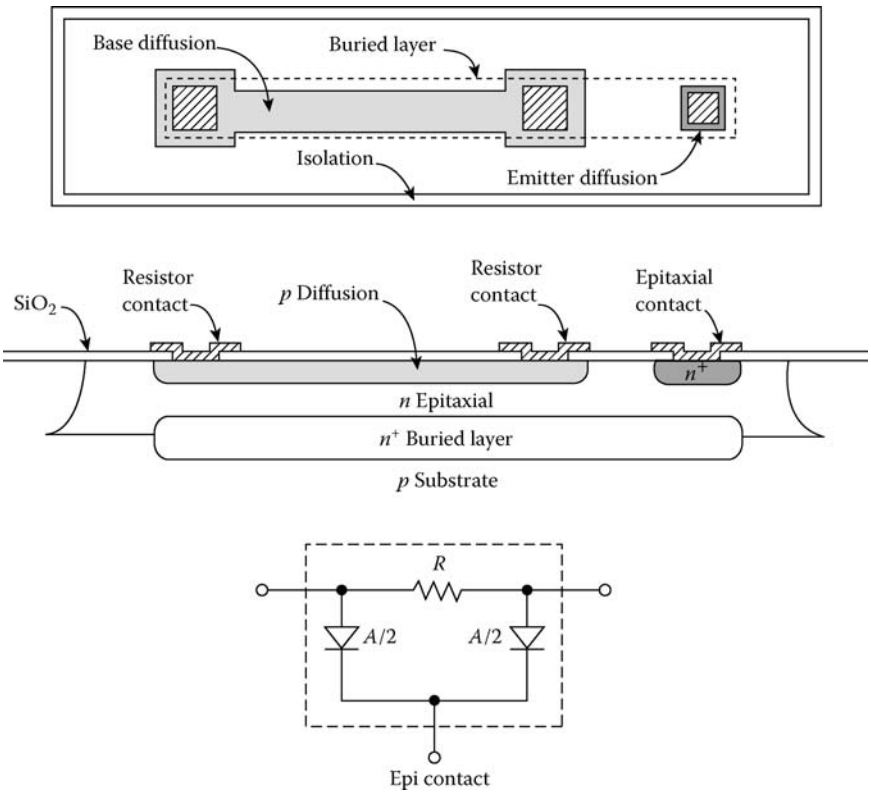


FIGURE 1.69 p-Type base-diffused resistor.

in order to eliminate the undesired dc current path through the pn junction. The impedance associated with a forward-biased pn junction is low and thus would also cause significant ac signal loss. To ensure this reverse bias constraint the epi region must be connected to a potential that is more positive than either end of the resistor contacts. Connecting the epi region to a relatively higher potential also eliminates the conductive action due to the parasitic pnp transistor formed by the p-type resistor, the n-type epi region, and the p-type substrate. When the base-diffused resistor is fabricated along with other circuit elements to form an integrated circuit (IC), the epitaxial contact is normally connected to the most positive supply of the circuit.

The resistance of a diffused resistor is given by Equation 1.237, where the diffused sheet resistance is between 100 and 200 Ω/N. Due to the lateral diffusion of impurity atoms, the effective cross-sectional area of the resistor is larger than the width determined by photomasking. This lateral or side diffusion effect can be accounted for by replacing the resistor width W by an effective width W_{eff}, where $W_{\text{eff}} \geq W$. The resistance from the two resistor contacts must also be accounted for, especially for small values of L/W [3]. Base-diffused resistors have a typical temperature coefficient between +1500 and +2000 ppm/°C.

The maximum allowable voltage for the base-diffused resistor of Figure 1.69 is limited by the breakdown voltage between the p-type base diffusion and the n-type epi. This voltage equals the breakdown voltage BV_{CBO} of the collector–base junction of the npn transistor and typically causes an *avalanche breakdown* mechanism across the base–epi junction. As the applied voltage approaches the breakdown voltage, a large leakage current flows from the epi region to the base region and can cause excessive heat dissipation.

For analog IC applications where good matching tolerance between adjacent resistors is required, the resistor width should be made as large as possible. Base-diffused resistors with 50 μm resistor widths can achieve a matching tolerance of ±0.2%. The minimum resistor width is limited by photolithographic consideration with typical values between 3 and 5 μm. Also, in order to avoid the self-heating problem of the resistor it is important to ensure a minimum resistor width for a given dc current level, with a typical value of about 3 μm for every 1 mA of current.

With respect to high-frequency performance, the reverse-biased pn junction between the p-type base diffusion and the n-type epi contributes a distributed depletion capacitance which in turn causes an impedance roll-off at 20 dB/decade. This capacitance depends on the voltage applied across the junction and the junction impurity-atom dopings. For most applications the electrical lumped model as shown in Figure 1.69 is adequate for characterizing this capacitive effect where the effective pn junction area is divided equally between the two diodes. Figure 1.70 shows a normalized impedance response as a function of the *RC* distributed stage. The frequency at which impedance value is reduced by 3 dB is given by

$$f_{-3\text{ dB}} = \begin{cases} \left(\dfrac{1}{2\pi}\right)\dfrac{2.0}{RC} & N = 1 \text{ (Circuit model of Figure 1.62)} \\ \left(\dfrac{1}{2\pi}\right)\dfrac{2.32}{RC} & N = 2 \\ \left(\dfrac{1}{2\pi}\right)\dfrac{2.42}{RC} & N = 3 \\ \left(\dfrac{1}{2\pi}\right)\dfrac{2.48}{RC} & N = 4 \end{cases} \tag{1.244}$$

Emitter-diffused resistors. Emitter-diffused resistors are formed by using the heavily doped n^+ emitter diffusion layer of the npn transistors. Due to the high doping concentration, the sheet resistance can be as low as 2 to 10 Ω/N with a typical absolute value tolerance of ±20%.

Figure 1.71 shows an emitter-diffused resistor structure where an n^+ diffusion layer is formed directly on top of the n-type epitaxial region and the ohmic contacts are composed of conductive

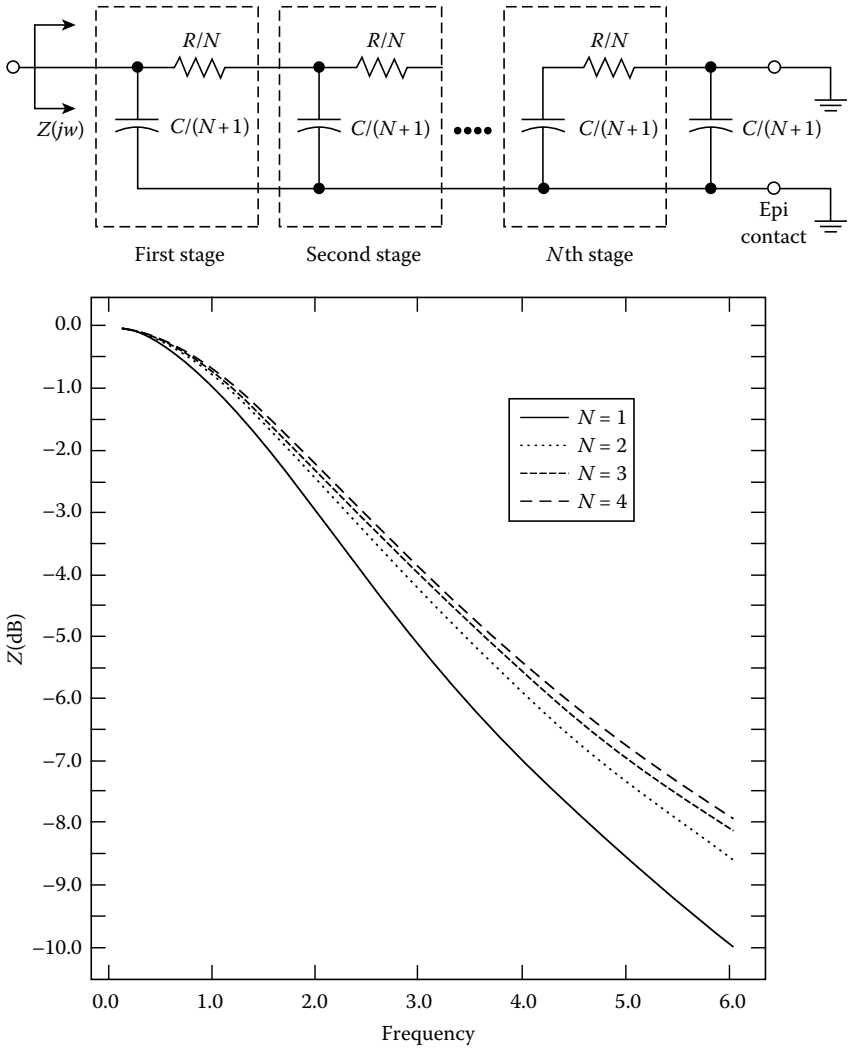


FIGURE 1.70 Normalized frequency response of a diffused resistor for $N = 1, 2, 3, 4$. The epi contact and one end of the resistor are grounded.

metal thin films. Since the resistor body and the epi layer are both n-type doped, they are electrically connected in parallel but the epi layer is of much higher resistivity due to its lower concentration doping, and thus the effective sheet resistance of the resistor structure is determined solely by the n^+ diffusion layer. The pn junction formed between the p-type substrate and the n-type epi region must always be reverse biased, which is accomplished by connecting the substrate to a most negative potential. Because of the common n-type epi layer, each resistor structure of Figure 1.71 requires a separate isolation region.

Figure 1.72 shows another emitter diffused resistor structure where the n^+ diffusion layer is situated within a p-type diffused well. Several such resistors can be fabricated in the same p-type well or in the same isolation region because the resistors are all electrically isolated. The p-type well and the n^+ diffusion region form a pn junction that must always be reverse biased for electrical isolation. In order to eliminate the conductive action due to the parasitic npn transistor formed by the n-type resistor body,

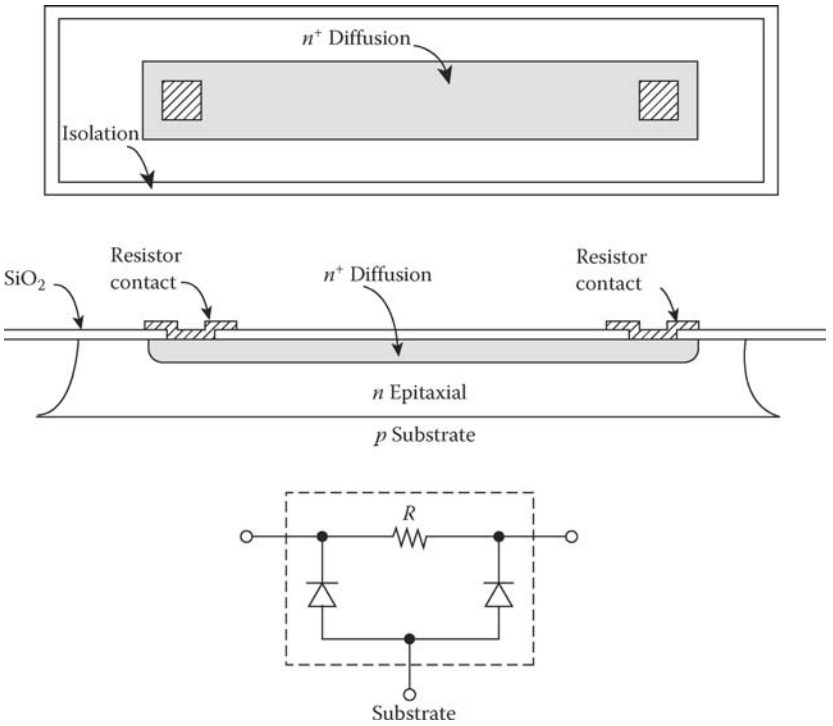


FIGURE 1.71 n-Type emitter-diffused resistor I.

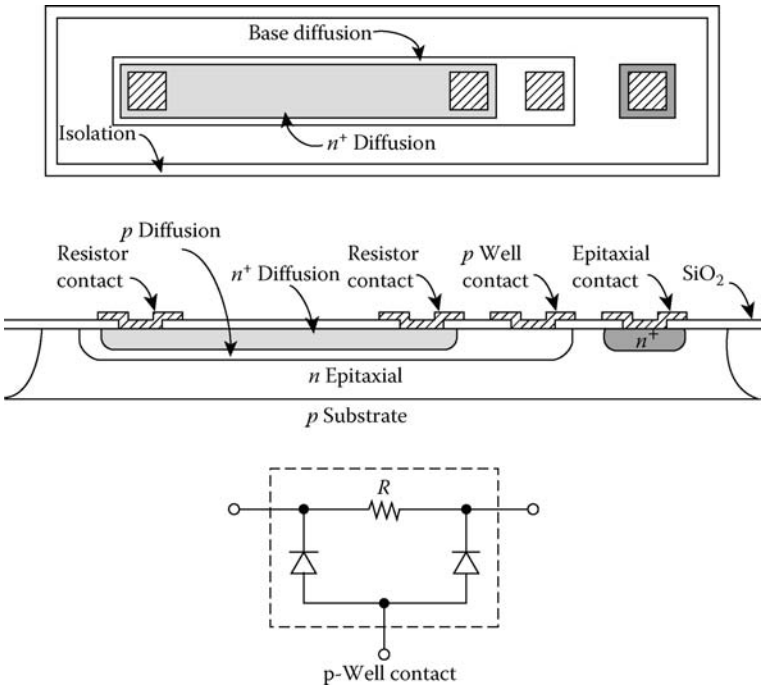


FIGURE 1.72 n-Type emitter-diffused resistor II.

the p-type well, and the n-type epi, the junction potential across the well contact and the epi contact must be either short-circuited or reverse-biased. The maximum voltage that can be applied across the emitter-diffused resistor of Figure 1.72 is limited by the breakdown voltage between the n^+ diffusion and the p-type well. This voltage equals the breakdown voltage BV_{EBO} of the emitter–base junction of the npn transistor, with typical values between 6 and 8 V.

1.4.1.2 Pinched Resistors

The active base region for the npn transistor can be used to construct pinched resistors with typical sheet resistance range from 2 to 10 KΩ/N. These high values can be achieved due to a thin cross-sectional area through which the resistor current traverses. The structure of a p-type base-pinched resistor is shown in Figure 1.73, where the p-type resistor body is ''pinched'' between the n^+ diffusion layer and the n-type epitaxial layer. The n^+ diffusion layer overlaps the p-type diffusion layer and is therefore electrically connected to the n-type epi. In many aspects the base-pinched resistor behaves like a p-channel JFET, in which the active base region functions as the p-channel, the two resistor contacts assume the drain and source, and the n^+ diffusion and the epi constitute the n-type gate. When the pn junction formed between the active base and the surrounding n^+ diffusion and n-epi is subject to a reverse bias potential, the carrier-free depletion region increases and extends into the active base region, effectively reducing the resistor cross section and consequently increasing the sheet resistance. Since the carrier-free depletion region varies with reverse bias potential, the pinched resistance is voltage controlled and is nonlinear.

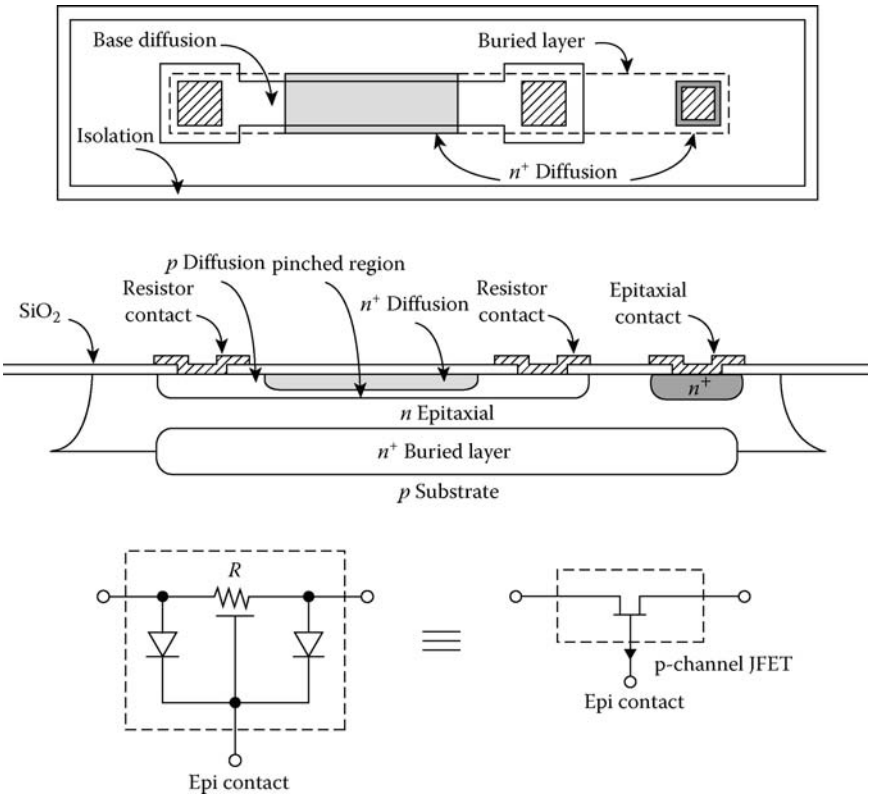


FIGURE 1.73 p-Type base-pinched resistor.

Absolute values for the base-pinched resistors can vary as much as ±50% due to large process variation in the fabrication of the active base region. The maximum voltage that can be applied across the base-pinched resistor of Figure 1.73 is restricted by the breakdown voltage between the n^+ diffusion layer and the p-type base diffusion. The breakdown voltage has a typical value around 6 V.

1.4.1.3 Epitaxial Resistors

Large values of sheet resistance can be obtained either by reducing the effective cross-sectional area of the resistor structure or by using a low doping concentration that forms the resistor body. The first technique is used to realize the pinched resistor while the second is used to realize the epitaxial resistor. Figure 1.74 shows an epitaxial resistor structure where the resistor is formed with a lightly doped epitaxial layer. For an epi thickness of 10 μm and a doping concentration of 10^{15} donor atoms/cm^3, this structure achieves a resistivity of 5 Ω-cm and an effective sheet resistance of 5 KΩ/N. The temperature coefficient of the epitaxial resistor is relatively high with typical values around +3000 ppm/°C. This large temperature variation is a direct consequence of the hole and electron mobilities undergoing more drastic variations against temperature at particularly low doping concentrations [13]. The maximum voltage that can be applied across the epitaxial resistor is significantly higher than that for the pinched resistor. This voltage

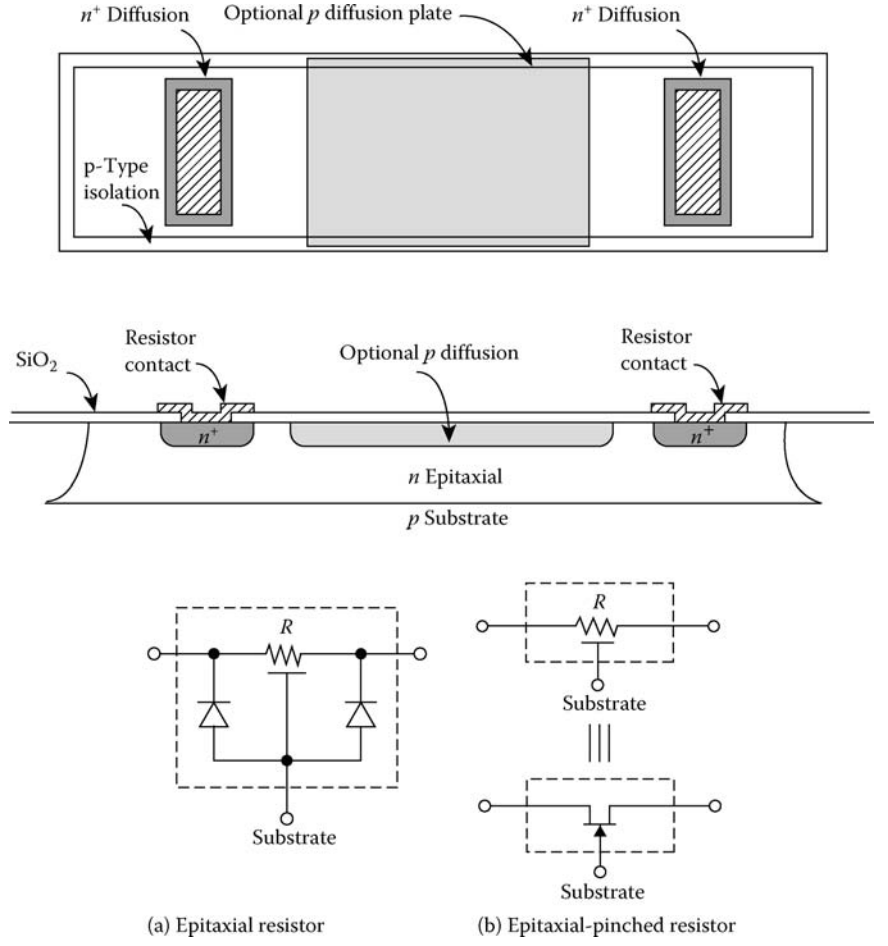


FIGURE 1.74 n-Type epitaxial and epitaxial-pinched resistors.

is set by the breakdown voltage between the n-type epi and the p-type substrate which varies inversely with the doping concentration of this pn junction.

Epitaxial-pinched resistors. By putting a p-type diffusion plate on top of the epitaxial resistor of Figure 1.74, even larger sheet resistance value can be obtained. The p-type diffusion plate overlaps the epi region and is electrically connected to the substrate through the p-type isolation. The epi layer is thus pinched between the p-type diffusion plate and the p-type substrate. When the n-type epi and the surrounding p-type regions is subject to a reverse bias potential, the junction depletion width extends into the epi region and effectively reduces the cross-sectional area. Typical sheet resistance values are between 4 and 5 KΩ/N. The epitaxial-pinched resistor behaves like an n-channel JFET, in which the effective channel width is controlled by the substrate voltage.

1.4.1.4 Ion-Implanted Resistors

Ion implantation is an alternative technique beside diffusion for inserting impurity atoms into a silicon wafer [17]. Commonly used impurities for implantation are the p-type boron atoms. The desired impurity atoms are first ionized and then accelerated to a high energy by an electric field. When a beam of these high-energy ions is directed at the wafer, the ions penetrate into exposed regions of the wafer surface. The penetration depth depends on the velocity at contact and is typically between 0.1 and 0.8 μm. The exposed regions of the wafer surface are defined by selectively etching a thick thermally grown SiO_2 layer that covers the wafer and functions as a barrier against the implanted ions. Unique characteristics of the ion-implantation technique include a precise control of the impurity concentration, uniformly implanted layers of impurity atoms, and no lateral diffusion. The structure of a p-type ion-implanted resistor is shown in Figure 1.75, where the p-type diffused regions at the contacts are used to achieve good ohmic contacts to the implanted resistor. The pn junction formed between the p-type implanted region and the n-type epitaxial layer must be reverse biased for electrical isolation. By connecting the epi region to a potential relatively more positive than the substrate potential, the conductive action due to the parasitic pnp transistor formed by the p-type implanted, the n-type epi, and the p-type substrate is also eliminated. Ion-implanted resistors exhibit relatively tight absolute value tolerance and excellent matching. Absolute value tolerance down to ±3% and matching tolerance of ±2% are typical performance.

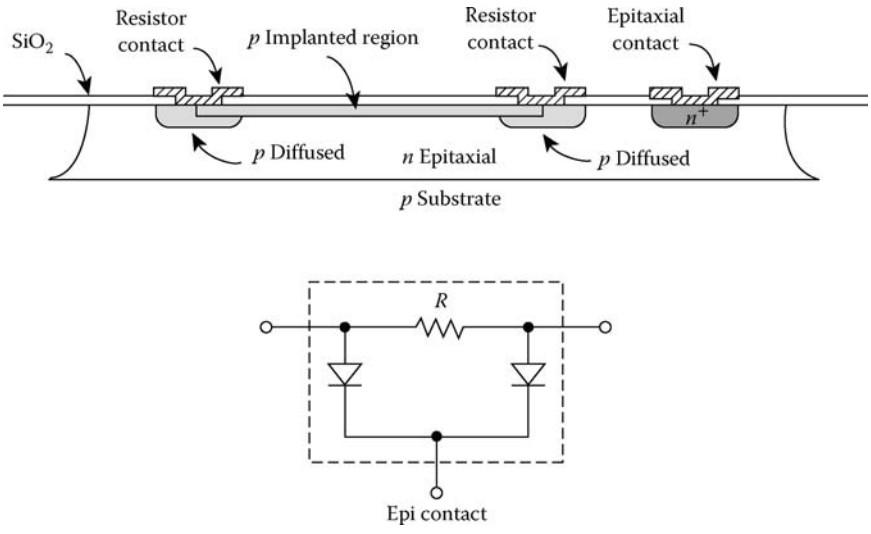


FIGURE 1.75 p-Type ion-implanted resistor.

TABLE 1.8 Typical Properties of Semiconductor Resistors

Resistor Type	Sheet ρ(Ω/N)	Absolute Tolerance (%)	Matching Tolerance (%)	Temperature Coefficient (ppm/°C)
Base-diffused	100–200	±20	±2 (5 μm wide)	+1500 to +2000
			±0.2 (50 μm wide)	—
Emitter-diffused	2–10	±20	±2	+600
Base-pinched	2–10 K	±50	±10	+2500
Epitaxial	2–5 K	±30	±5	+3000
Epitaxial-pinched	4–10 K	±50	±7	+3000
Ion-implanted	100–1000	±3	±2 (5 μm wide)	Controllable to ±100
			±0.15 (50 μm wide)	

Source: Gray, P.R. and Meyer, R.G., *Analysis and Design of Analog Integrated Circuits*, Wiley, New York, 1984, p. 119.

Table 1.8 provides a summary of the typical characteristics for the diffused, pinched, epitaxial, and ion-implanted resistors.

1.4.1.5 Thin-Film Resistors

Compared with diffused resistors, thin-film resistors offer advantages of a lower temperature coefficient, a smaller absolute value variation, and an excellent high-frequency characteristic. Commonly used resistive thin films are tantalum, Ni–Cr, Cr–SiO, and SnO_2. A typical thin-film resistor structure is shown in Figure 1.76, where a thin-film resistive layer is deposited on top of a thermally grown SiO_2 layer and a thin-film conductive metal layer is used to form the resistor contacts. The oxide layer functions as an insulating layer for the resistor. Various CVD techniques can be used to form the thin films [8]. The oxide passivation layer deposited on top of the resistive film and the conductive film protects the device surface from contamination. The electrical lumped model as shown in Figure 1.76 is adequate to characterize the high-frequency performance of the resistor. The parallel-plate capacitance formed

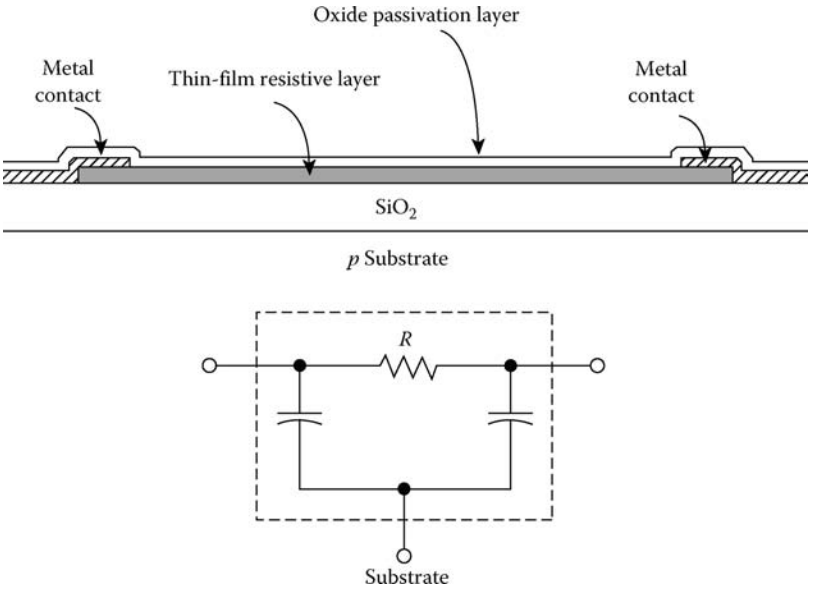


FIGURE 1.76 Thin-film resistor.

TABLE 1.9 Typical Characteristic of Thin-Film Resistors

Resistor Type	Sheet ρ(Ω/N)	Absolute Tolerance (%)	Matching Tolerance (%)	Temperature Coefficient (ppm/°C)
Ni–Cr	40–400	±5	±1	±100
Ta	10–1000	±5	±1	±100
SnO_2	80–4000	±8	±2	0–1500
Cr–SiO	30–2500	±10	±2	±50 to ±150

Source: Grebene, A.B., *Bipolar and MOS Analog Integrated Circuit Design*, Wiley, New York, 1984, p. 155.

between the thin-film resistive and the substrate is divided equally between the two capacitors. Table 1.9 provides a summary of the characteristics for some commonly used thin-film resistors.

1.4.2 Capacitors

Monolithic capacitors are widely used in analog and digital ICs for functions such as circuit stability, bandwidth enhancement, ac signal coupling, impedance matching, and charge storage cells. Capacitor structures available in monolithic form include pn junction, MOS, and polysilicon capacitors. pn junctions under reverse-biased conditions exhibit a nonlinear voltage-dependent capacitance. MOS and polysilicon capacitors, on the other hand, closely resemble the linear parallel-plate capacitor structure as shown in Figure 1.77. If the insulator thickness T of the parallel-plate structure is small compared with the plate width W and length L, the electric field between the plates is uniform (fringing field neglected). Under this condition the capacitance can be calculated by

$$C = \frac{\kappa\varepsilon_0}{T} WL \tag{1.245}$$

where κ is the relative dielectric constant of the insulating material and ε_0 is the permittivity constant in vacuum (8.854×10^{-14} F/cm).

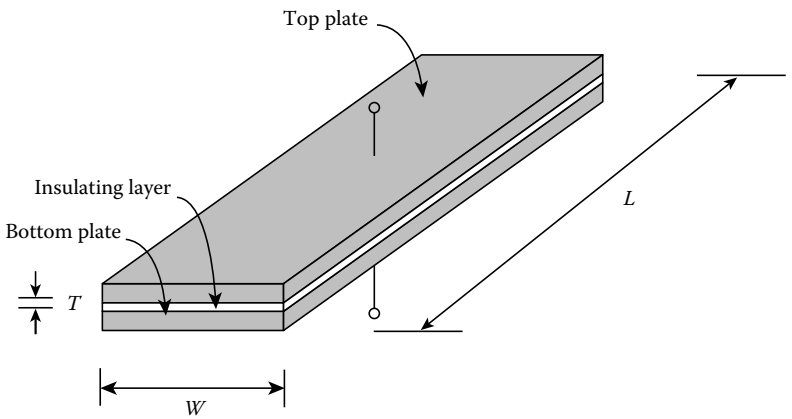


FIGURE 1.77 Structure of a parallel-plate capacitor.

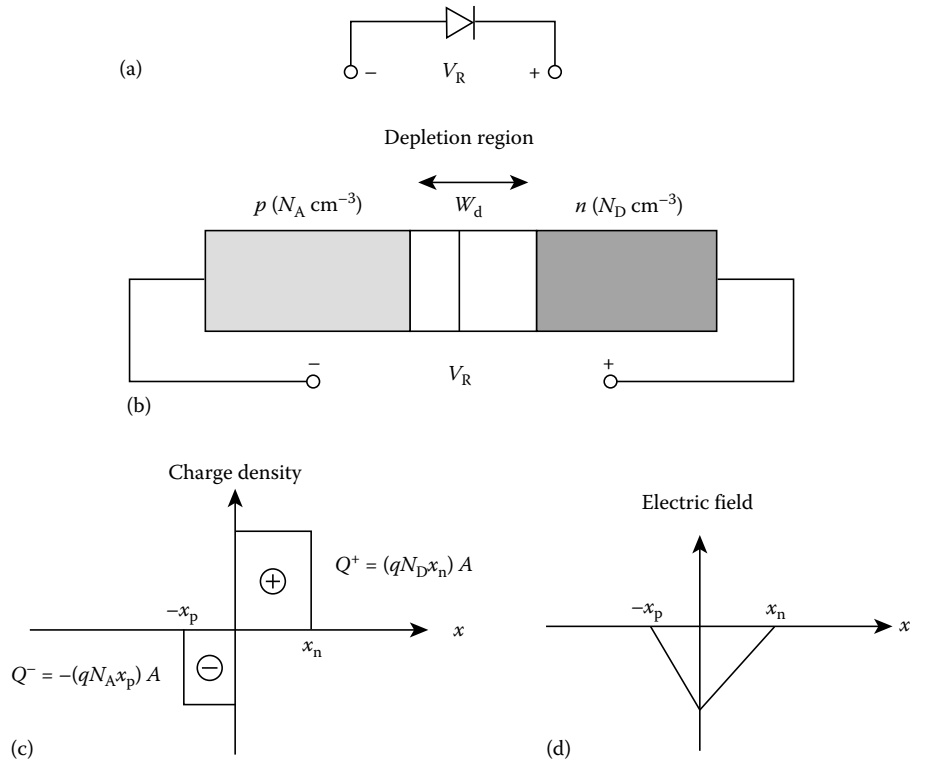


FIGURE 1.78 Abrupt p–n junction: (a) p–n junction symbol; (b) depletion region; (c) charge density within the depletion region; and (d) electric field.

1.4.2.1 Junction Capacitors

The structure of an *abrupt* pn junction is shown in Figure 1.78, where the doping is assumed uniform throughout the region on both sides. The acceptor impurity concentration of the p region is N_A atoms/cm^3 and the donor impurity concentration of the n region is N_D atoms/cm^3. When the two regions are brought in contact, mobile holes from the p region diffuse across the junction to the n region and mobile electrons diffuse from the n to the p region. This diffusion process creates a *depletion* region that is essentially free of mobile carriers (depletion approximation) and contains only fixed acceptor and donor ions. Ionized acceptor atoms are negatively charged and ionized donor atoms are positively charged. In equilibrium the diffusion process is balanced out by a drift process that arises from a *built-in* voltage ψ_o across the junction. This voltage is positive from the n region relative to the p region and is given by [17]

$$\psi_o = \frac{kT}{q} \ln \frac{N_A N_D}{n_i^2} \tag{1.246}$$

where

k is the Boltzmann constant (1.38×10^{-23} V · C/K)
T is the temperature in Kelvin (K)
q is the electron charge (1.60×10^{-19} C)
n_i(cm^{-3}) is the *intrinsic* carrier concentration in a pure semiconductor sample

For silicon at 300 K, $n_i \approx 1.5 \times 10^{10}$ cm^{-3}.

When the pn junction is subject to an applied reverse bias voltage V_R, the drift process is augmented by the external electric field and more mobile electrons and holes are pulled away from the junction. Because of this effect, the depletion width W_d and consequently the charge Q on each side of the junction vary with the applied voltage. A junction capacitor can thus be defined to correlate this charge–voltage relationship. The Poisson's equation relating the junction voltage $\phi(x)$ to the electric field $\xi(x)$ and the total charge Q is

$$\begin{aligned}\frac{d^2\varphi(x)}{dx^2} &= -\frac{d\xi(x)}{dx} = -\frac{q}{\varepsilon_S}(p - n + N_D - N_A) \\ &\approx \begin{cases} \dfrac{qN_A}{\varepsilon_S} & -x_p < x < 0 \\ -\dfrac{qN_D}{\varepsilon_S} & 0 < x < x_n \end{cases}\end{aligned} \tag{1.247}$$

where ε_S ($11.8\varepsilon_0 = 1.04 \times 10^{-12}$ F/cm) is the permittivity of the silicon material. The first integral of Equation 1.247 yields the electric field as

$$\xi(x) = \begin{cases} -\dfrac{qN_A}{\varepsilon_S}(x + x_p) & -x_p < x < 0 \\ -\dfrac{qN_D}{\varepsilon_S}(x_n + x) & 0 < x < x_n \end{cases} \tag{1.248}$$

The electric field is shown in Figure 1.78, where the maximum field strength occurs at the junction edge. This value is given by

$$|\xi_{max}| = \frac{qN_A}{\varepsilon_S}x_p = \frac{qN_D}{\varepsilon_S}x_n$$

The partial depletion width x_p on the p region and the partial depletion width x_n on the n region can then be related to the depletion width W_d as

$$\begin{aligned} x_p + x_n &= W_d \\ x_p &= \frac{N_D}{N_A + N_D}W_d \\ x_n &= \frac{N_A}{N_A + N_D}W_d \end{aligned}$$

Taking the second integral of Equation 1.247 yields the junction voltage

$$\varphi(x) = \begin{cases} \dfrac{qN_A}{\varepsilon_S}\left(\dfrac{x_p^2}{2} + x_p x + \dfrac{x^2}{2}\right) & -x_p < x < 0 \\ \dfrac{qN_D}{\varepsilon_S}\left(\dfrac{x_n x_p}{2} + x_n x - \dfrac{x^2}{2}\right) & 0 < x < x_n \end{cases} \tag{1.249}$$

where the voltage at x_p is arbitrarily assigned to be zero. The total voltage $\psi_0 + V_R$ can be expressed as

$$\psi_o + V_R = \varphi(x_n) = \frac{qN_D}{2\varepsilon_S}\left(1 + \frac{N_D}{N_A}\right)x_n^2$$

Finally, the depletion width W_d and the total charge Q in terms of the total voltage across the junction can be derived to be

$$W_d = \left[\frac{2\varepsilon_S}{q}(\psi_o + V_R)\left(\frac{1}{N_A} + \frac{1}{N_D}\right)\right]^{1/2}$$
$$|Q| = A(qN_A x_p) = A(qN_D x_n) = A\left[2q\varepsilon_S(\psi_o + V_R)\left(\frac{1}{N_A} + \frac{1}{N_D}\right)^{-1}\right]^{1/2} \tag{1.250}$$

The junction capacitance is thus

$$C_j = \left|\frac{dQ}{dV_R}\right| = A\left[\frac{q\varepsilon_S}{2}\left(\frac{1}{\psi_o + V_R}\right)\left(\frac{1}{N_A} + \frac{1}{N_D}\right)^{-1}\right]^{1/2}$$
$$= \frac{C_{jo}}{\left(1 + \frac{V_R}{\psi_o}\right)^{1/2}} \tag{1.251}$$

where

A is the effective cross-sectional junction area
C_{jo} is the value of C_j for $V_R = 0$

If the doping concentration in one side of the pn junction is much higher than that in the other, the depletion width and the junction capacitance can be simplified to

$$W_d = \left[\frac{2\varepsilon_S}{qN_L}(\psi_o + V_R)\right]^{1/2} \tag{1.252}$$

$$C_j = A\left[\frac{\varepsilon_S qN_L}{2}\left(\frac{1}{\psi_o + V_R}\right)\right]^{1/2} \tag{1.253}$$

where N_L is the concentration of the lightly doped side. Figure 1.79 displays the junction capacitance per unit area as a function of the total voltage $\psi_o + V_R$ and the concentration on the lightly doped side of the junction [3].

In silicon bipolar technology the base–emitter, the base–collector, and the collector–substrate junctions under reverse bias are often utilized for realizing a junction capacitance. The collector–substrate junction has only a limited use since it can only function as a shunt capacitor due to the substrate being connected to an ac ground.

Base–collector junction capacitor. A typical base–collector capacitor structure is shown in Figure 1.80 together with an equivalent lumped circuit model. A heavily doped n^+ buried layer is used to minimize the series resistance R_C. For the base–collector junction to operate in reverse bias, the n-type collector must be connected to a voltage relatively higher than the voltage at the p-type base. The junction breakdown voltage is determined by BV_{CBO} of the npn transistor, which has a typical value between 25 and 50 V.

Base–emitter junction capacitor. Figure 1.81 shows a typical base–emitter capacitor structure where the parasitic junctions D_{BC} and D_{SC} must always be in reverse bias. The base–emitter junction achieves the highest capacitance per unit area among the base–collector, base–emitter, and collector–substrate junctions due to the relatively higher doping concentrations in the base and emitter regions.

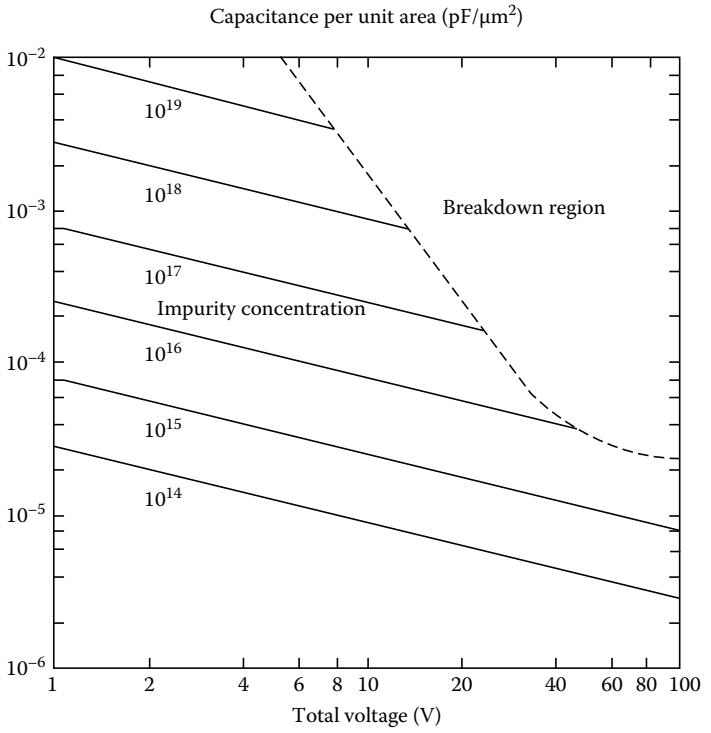


FIGURE 1.79 Junction capacitance as a function of the total voltage and the concentration on the lightly doped side.

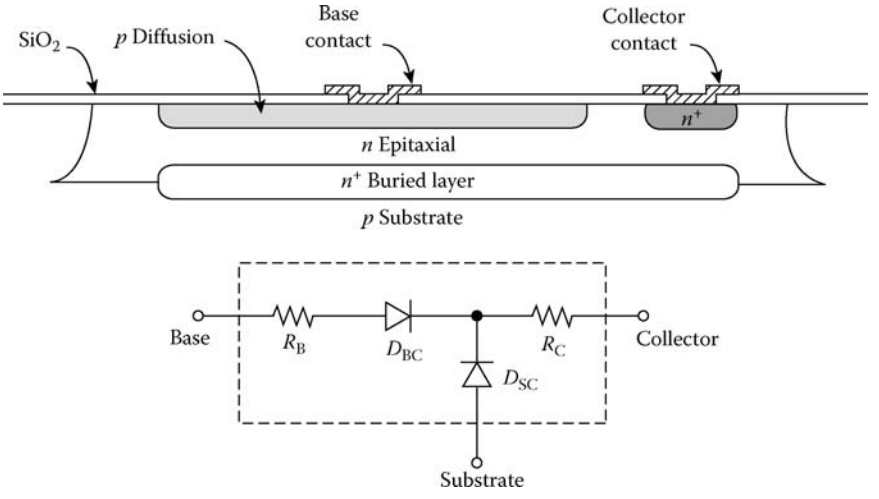


FIGURE 1.80 Base–collector junction capacitor.

For the base–emitter junction to operate in reverse bias, the n-type emitter must be connected to a voltage relatively higher than the voltage at the p-type base. The breakdown voltage of the base–emitter junction is relatively low, determined by the BV_{EBO} of the npn transistor, which has a typical value of about 6 V.

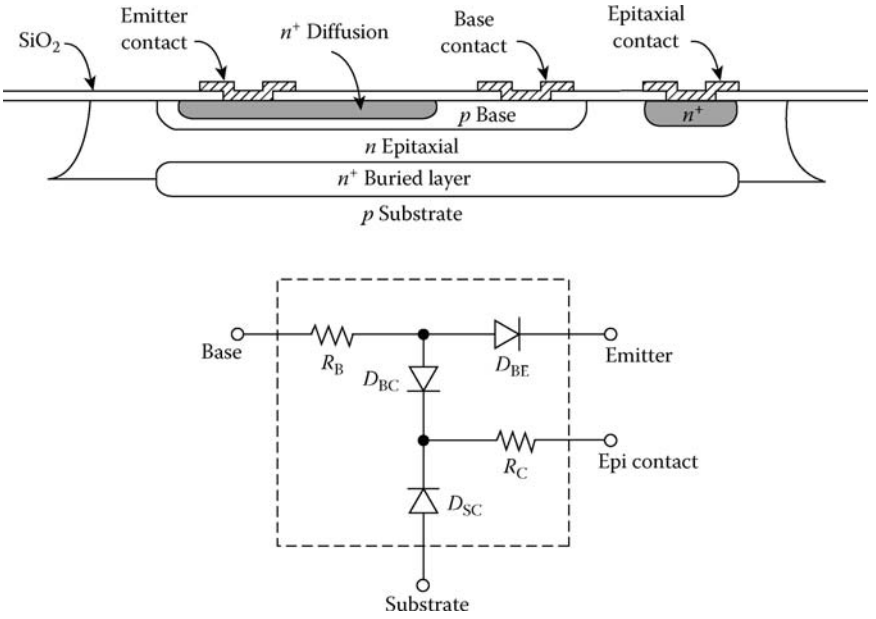


FIGURE 1.81 Base–emitter junction capacitor.

1.4.2.2 MOS Capacitors

MOS capacitors are preferable and commonly used in ICs since they are linear and not confined to a reverse-biased operating condition as in the junction capacitors. The structure of a MOS capacitor is shown in Figure 1.82, where by means of a local oxidation process a thin oxide layer is thermally grown on top of a heavily doped n^+ diffusion layer. The oxide layer has a typical thickness between 500 and 1500 Å (Å $= 10^{-10}$ m $= 10^{-4}$ μm) and functions as the insulating layer of the parallel-plate capacitor. The top plate is formed by overlapping the thin oxide area with a deposited layer of conductive metal. The bottom-plate diffusion layer is heavily doped for two reasons: to minimize the bottom-plate resistance and to minimize the depletion width at the oxide-semiconductor interface when the capacitor operates in the *depletion* and *inversion* modes [17]. By keeping the depletion width small, the effective capacitance is dominated by the parallel-plate oxide capacitance. The MOS capacitance is thus given by

$$C = \frac{\kappa_{ox}\varepsilon_0}{T} A \tag{1.254}$$

where

κ_{ox} is the relative dielectric constant of SiO_2 (2.7 to 4.2)
ε_0 is the permittivity constant
T is the oxide thickness
A is the area defined by the thin oxide layer

In practice, a thin layer of silicon nitride (Si_3N_4) is often deposited on the thin oxide layer and is used to minimize the charges inadvertently introduced in the oxide layer during oxidation and subsequent processing steps. These *oxide charges* are trapped within the oxide and can cause detrimental effect to the capacitor characteristic [17]. The silicon nitride assimilates an additional insulating layer and effectively

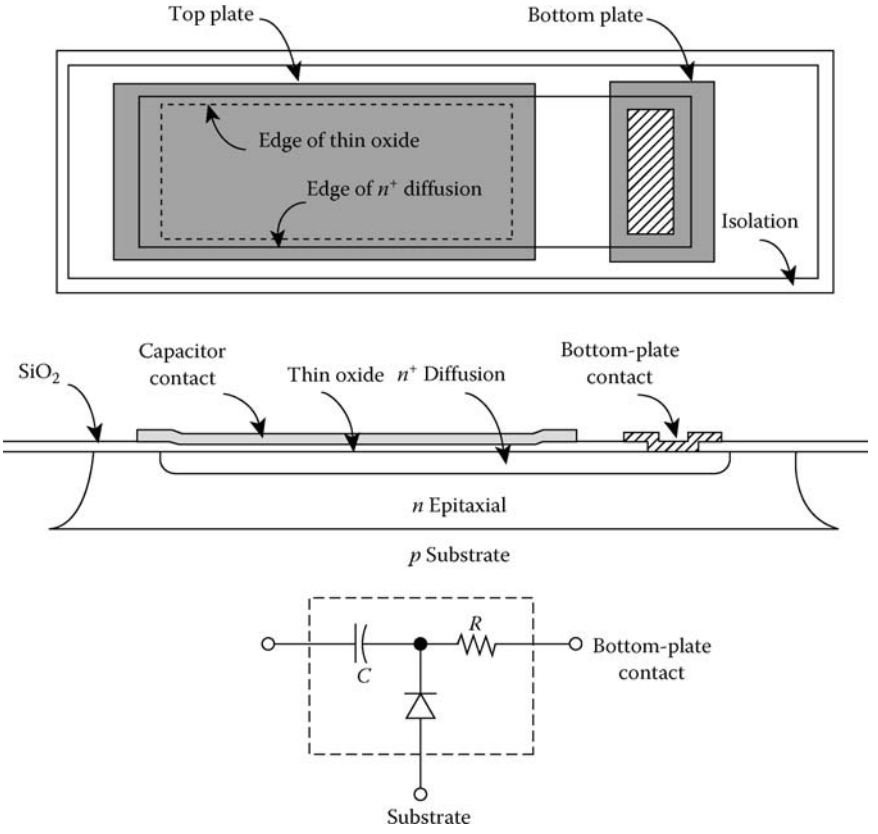


FIGURE 1.82 MOS capacitor.

creates an additional capacitor in series with the oxide capacitor. The capacitance for such a structure can be determined by an application of *Gauss's law*. It is given by

$$C = \frac{\varepsilon_o}{\left(\frac{T_{ni}}{\kappa_{ni}}\right) + \left(\frac{T_{ox}}{\kappa_{ox}}\right)} A \tag{1.255}$$

where

T_{ni} and T_{ox} are the thickness of the silicon nitride and oxide layers, respectively

κ_{ni} (2.7 to 4.2) and κ_{ni} (3.5 to 9) are the relative dielectric constant of oxide and silicon nitride, respectively

In the equivalent circuit model of Figure 1.82, the parasitic junction between the p-type substrate and the n-type bottom plate must always be reverse biased. The bottom-plate contact must be connected to a voltage relatively higher than the substrate voltage.

1.4.2.3 Polysilicon Capacitors

Polysilicon capacitors are conveniently available in MOSFET technology, where the gate of the MOSFET transistor is made of polysilicon material. Polysilicon capacitors also assimilate the parallel-plate capacitor. Figure 1.83 shows a typical structure of a polysilicon capacitor, where a thin oxide is deposited on top of a polysilicon layer and serves as an insulating layer between the top-plate metal layer and the

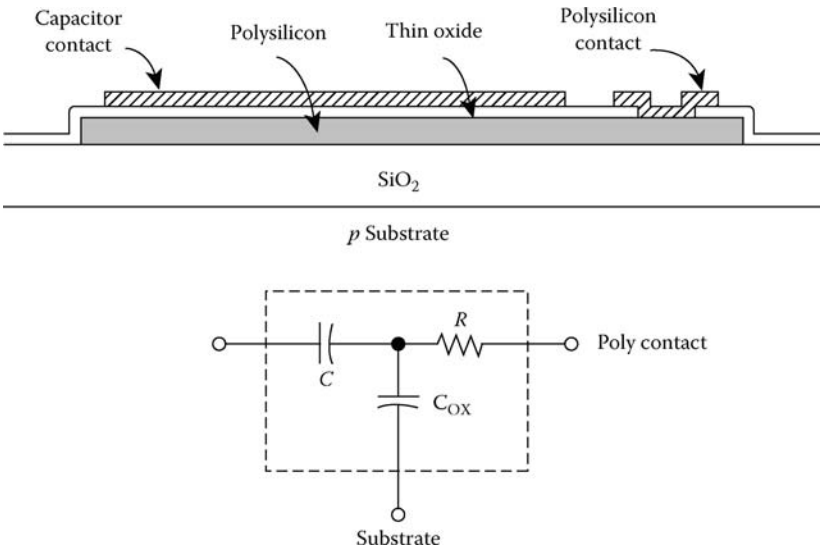


FIGURE 1.83 Polysilicon capacitor.

bottom-plate polysilicon layer. The polysilicon region is isolated from the substrate by a thick oxide layer that forms a parasitic parallel-plate capacitance between the polysilicon layer and the substrate. This parasitic capacitance must be accounted for in the equivalent circuit model. The capacitance of the polysilicon capacitor is determined by either Equation 1.254 or 1.255 depending on whether a thin silicon nitride is used in conjunction with the thin oxide.

1.4.3 Inductors

Planar inductors have been implemented using a variety of substrates such as standard PC boards, ceramic and sapphire hybrids, monolithic GaAs [24], and more recently monolithic silicon [18]. In the early development of silicon technology, planar inductors were investigated [26], but the prevailing lithographic limitations and relatively large inductance requirements (for low-frequency applications) resulted in excessive silicon area and poor performance. Reflected losses from the conductive silicon substrate were a major contribution to low inductor *Q*. Recent advances in silicon IC processing technology have achieved fabrication of metal width and metal spacing in the low micrometer range and thus allow many more inductor turns per unit area. Also, modern oxide-isolated processes with multilayer metal options allow thick oxides to help isolate the inductor from the silicon substrate. Practical applications of monolithic inductors in low-noise amplifiers, impedance matching amplifiers, filters and microwave oscillators in silicon technologies have been successfully demonstrated [19,20].

Monolithic inductors are especially useful in high-frequency applications where inductors of a few nano-Henrys of inductance are sufficient. Inductor structures in monolithic form include strip, loop, and spiral inductors. Rectangular and circular spiral inductors are by far the most commonly used structures.

1.4.3.1 Rectangular Spiral Inductors

The structure of a rectangular spiral inductor is shown in Figure 1.84, where the spiral loops are formed with the top metal layer M_2 and the connector bridge is formed with the bottom metal layer M_1. Using the top metal layer to form the spiral loops has the advantage of minimizing the parasitic metal-to-substrate

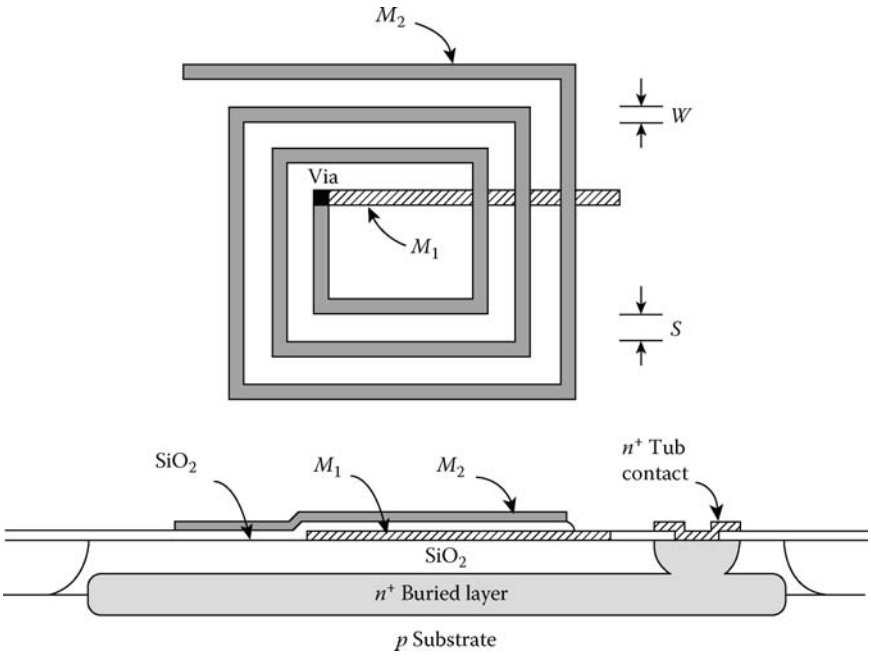


FIGURE 1.84 Rectangular spiral inductor.

capacitance. The metal width is denoted by W and the metal spacing is denoted by S. The total inductance is given by

$$L_T = \sum_{i=1}^{4N} L_S(i) + 2 \cdot \sum_{i=1}^{4N-1} \sum_{j=i+1}^{4N} L_M(ij) \tag{1.256}$$

where

N is the number of turns
$L_S(i)$ is the *self inductance* of the rectangular metal segment i
$L_M(ij)$ is the *mutual inductance* between metal segments i and j

The self-inductance is due to the magnetic flux surrounding each metal segment. The mutual inductance is due to the magnetic flux coupling around every two parallel metal segments and has a positive value if the currents applied to the metal conductors flow in the same direction and a negative value otherwise. Perpendicular metal segments have negligible mutual inductance.

The self-inductance and mutual inductance for straight rectangular conductors can be determined by the *geometric mean distance* method [10], in which the conductors are replaced by equivalent straight filaments whose basic inductive characteristics are well known.

Self-inductance. The self-inductance for the rectangular conductor of Figure 1.85 depends on the conductor length L, the conductor width W, and the conductor thickness T. The static self-inductance is given by [9,10].

$$L_S = 2L\left[\ln\left(\frac{2L}{GMD}\right) - 1.25 + \left(\frac{AMD}{L}\right) + \left(\frac{\mu_r}{4}\right)\zeta\right](nH) \tag{1.257}$$

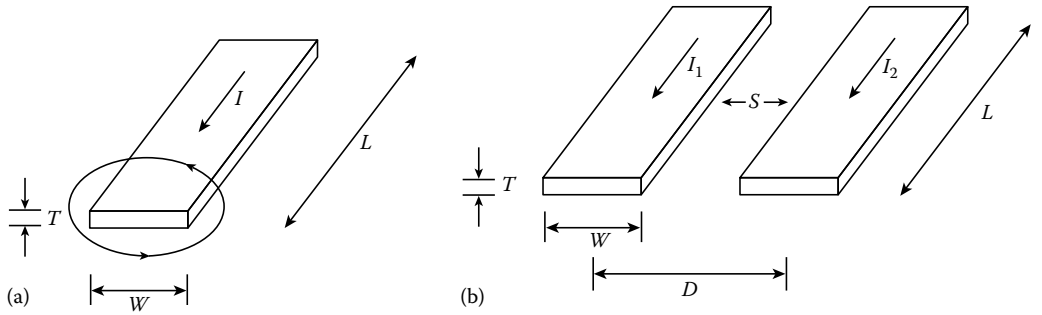


FIGURE 1.85 Calculation of (a) self-inductance and (b) mutual inductance for parallel rectangular conductors.

where

μ_r is the relative permeability constant of the conductor
GMD is the geometric mean distance
AMD is the arithmetic mean distance
ζ is a frequency-dependent parameter that equals 1 for direct and low-frequency alternating currents and approaches 0 for very high-frequency alternating currents

The AMD and GMD for the rectangular conductor of Figure 1.85 are

$$\text{AMD} = \left(\frac{W+T}{3}\right)$$
$$\text{GMD} = \begin{cases} 0.22313 \cdot (W+T) & T \to 0 \\ 0.22360 \cdot (W+T) & T = \text{W}/2 \\ 0.223525 \cdot (W+T) & T \to \text{W} \end{cases} \tag{1.258}$$

The rectangular dimensions L, W, and T are normalized to the centimeter in the preceding expressions.

Mutual inductance. The mutual inductance for the two parallel rectangular conductors of Figure 1.85 depends on the conductor length L, the conductor width W, and the conductor thickness T, and the distance D separating the conductor centers. The static mutual inductance is [10]

$$L_{\text{M}} = 2L\alpha(\text{nH}) \tag{1.259}$$

where

$$\alpha = \ln\left[\left(\frac{L}{\text{GMD}}\right) + \left[1 + \left(\frac{L}{\text{GMD}}\right)^2\right]^{1/2}\right] - \left[1 + \left(\frac{\text{GMD}}{L}\right)^2\right]^{1/2} + \left(\frac{\text{GMD}}{L}\right)$$

and

$$\text{GMD} = \exp(\ln D - \beta) \tag{1.260}$$

$$\beta = \begin{cases} \frac{1}{12}\left(\frac{D}{W}\right)^{-2} + \frac{1}{60}\left(\frac{D}{W}\right)^{-4} + \frac{1}{168}\left(\frac{D}{W}\right)^{-6} + \frac{1}{360}\left(\frac{D}{W}\right)^{-8} \\ + \frac{1}{660}\left(\frac{D}{W}\right)^{-10} + \cdots \\ 0.1137 \quad \text{for} \quad D = W \end{cases}$$

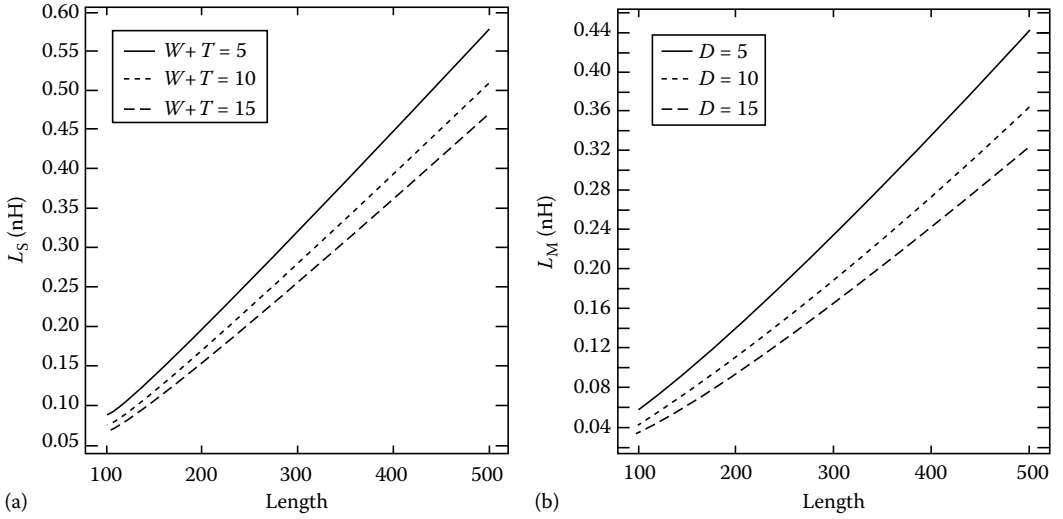


FIGURE 1.86 (a) Self-inductance as a function of width, thickness, and length for rectangular conductors. (b) Mutual inductance as a function of distance and length for rectangular conductors ($W = 5$, $T = 0$).

The GMD closed-form expression Equation 1.260 is valid for rectangular conductors with small thickness-to-width ratios T/W. As the thickness T approaches the width W, the GMD approaches the distance D and the GMD is no longer represented by the above closed-form expression. Figure 1.86 shows plots of the self inductance and the mutual inductance as expressed in Equations 1.257 and 1.259, respectively. The conductor dimensions are given in μm (μm $= 10^{-4}$ cm).

For the inductor structure of Figure 1.84 it is important to emphasize that since the spiral loops are of nonmagnetic metal material, the total inductance depends only on the geometry of the conductors and not on the current strength. At high-frequencies, especially those above the *self-resonant* frequency of the inductor, the *skin effect* due to current crowding toward the surface and the propagation delay as the current traverses the spiral must be fully accounted for [16,22]. The ground-plane effect due to the inductor image must also be considered regardless of the operation frequency.

An equivalent lumped model for the rectangular spiral inductor of Figure 1.84 is shown in Figure 1.87. This model consists of the total inductance L_T, the accumulated metal resistance R_S, the coupling

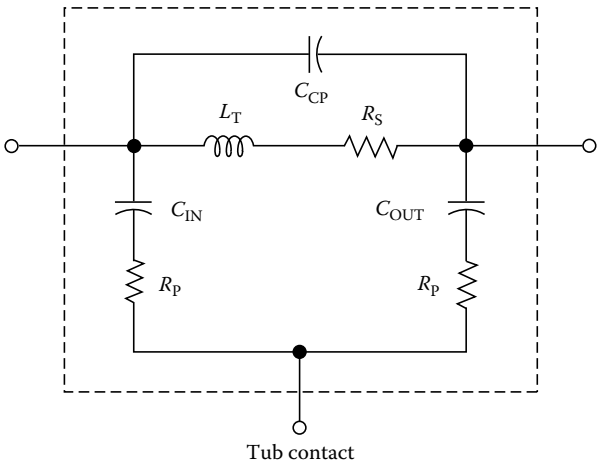


FIGURE 1.87 Electrical model for the spiral inductor.

capacitance C_{CP} between metal segments due to the electric fields in both the oxide region and the air region, the parasitic capacitances C_{IN} and C_{OUT} from the metal layers to the buried layer [2,11,15], and the buried-layer resistance R_P. Since the spiral structure of Figure 1.84 is not symmetrical, the parasitic capacitors C_{IN} and C_{OUT} are not the same, though the difference is relatively small. The self-resonant frequency can be approximated using the circuit model of Figure 1.87 with one side of the inductor being grounded. For simplicity, let $C_{IN} = C_{OUT} + C_P$ and neglect the relatively small coupling capacitor C_{CP}, the self-resonant frequency is given by

$$f_R = \frac{1}{2\pi} \frac{1}{\sqrt{L_T C_P}} \left[\frac{1 - R_S^2 \left(\frac{C_P}{L_T} \right)}{1 - R_P^2 \left(\frac{C_P}{L_T} \right)} \right]^{1/2} \tag{1.261}$$

Transformer structures. Transformers are often used in high-performance analog ICs that require conversions between single-ended signals and differential signals. In monolithic technology, transformers can be fabricated using the basic structure of the rectangular spiral inductor. Figure 1.88 shows a planar interdigitated spiral transformer that requires only two metal layers M_1 and M_2. The structure of Figure 1.89, on the other hand, requires three layers of metal for which the top metal layer M_3 is used for the upper spiral, the middle metal layer M_2 is used for the lower spiral, and the bottom metal layer M_1 is used for the two connector bridges. This structure can achieve a higher inductance per unit than that of Figure 1.88 due to a stronger magnetic coupling between the upper spiral and the lower spiral through a relatively thin oxide layer separating metal layers M_2 and M_3. An equivalent lumped model is

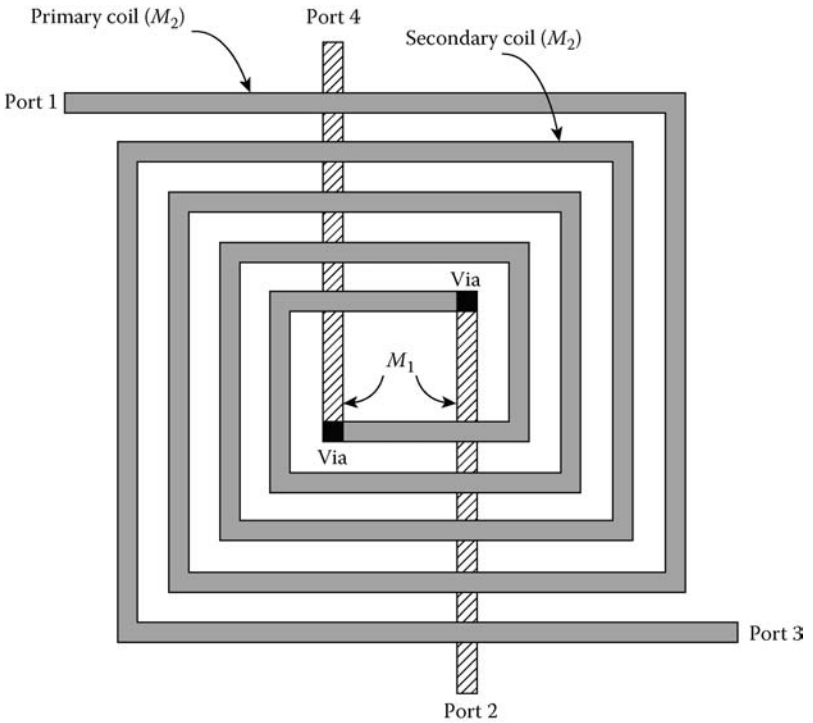


FIGURE 1.88 Rectangular spiral transformer I.

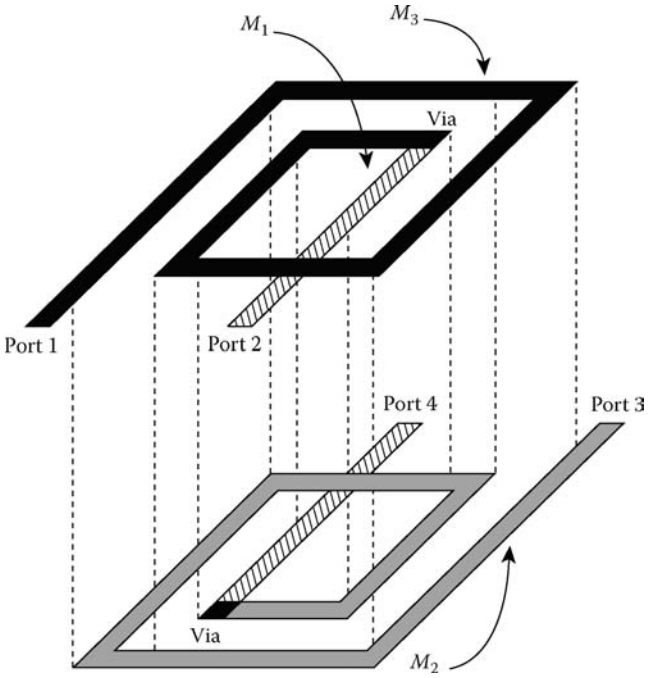


FIGURE 1.89 Rectangular spiral transformer II.

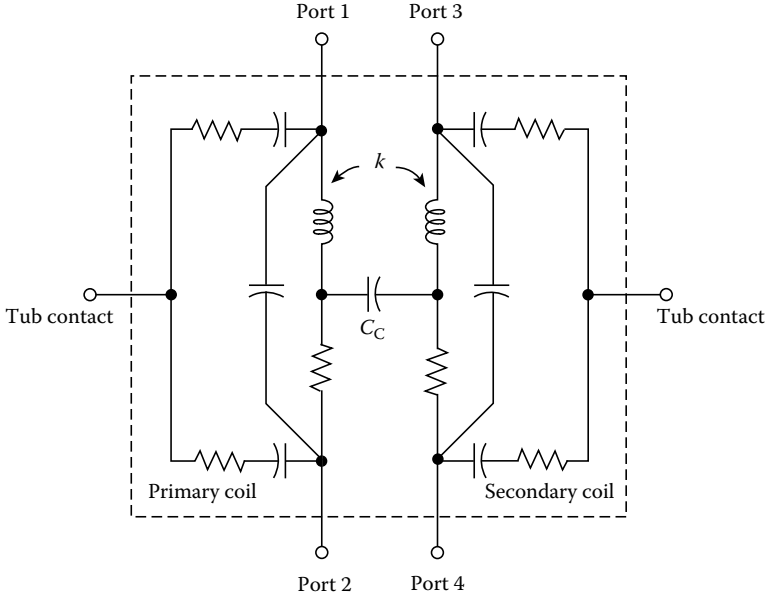


FIGURE 1.90 Electrical model for the spiral transformer.

shown in Figure 1.90. In addition to all the circuit elements of the two individual spiral inductors, there are also a magnetic coupling factor k and a coupling capacitance C_C between the primary and secondary coils.

1.4.3.2 Circular Spiral Inductors

The structure of a concentric circular spiral inductor is shown in Figure 1.91 where the circular loops share the same center point. The top metal layer M_2 is used for the circular conductors and the bottom metal layer M_1 is used for the connector bridge. The metal width is denoted by W and the spacing between two adjacent loops is denoted by S. The total inductance is given by

$$L_T = \sum_{i=1}^{N} L_S(i) + 2 \cdot \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} L_M(ij) \tag{1.262}$$

where

N is the number of circular turns
$L_S(i)$ is the *self-inductance* of the circular conductor i
$L_M(ij)$ is the *mutual inductance* between conductors i and j

Self-inductance. Consider the single circular conductor of Figure 1.92a that has a radius R and a width W. A current I applied to this conductor produces a magnetic flux encircled by the loop and another magnetic flux inside the conductor itself. The inductance associated with the former and the latter magnetic flux component is referred to as the *external* self-inductance and the *internal* self-inductance, respectively. The external self-inductance characterizing the change in the encircled magnetic flux to the change in current is [25].

$$L_S = \mu(2R - \delta)\left[\left(1 - \frac{k^2}{2}\right)K(k) - E(k)\right](\text{nH}) \tag{1.263}$$

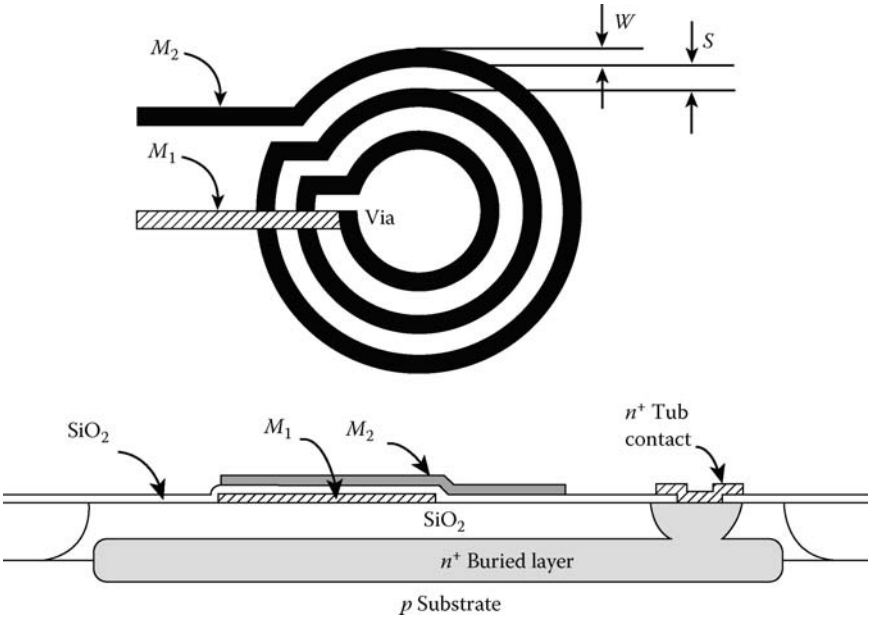


FIGURE 1.91 Concentric circular spiral inductor.

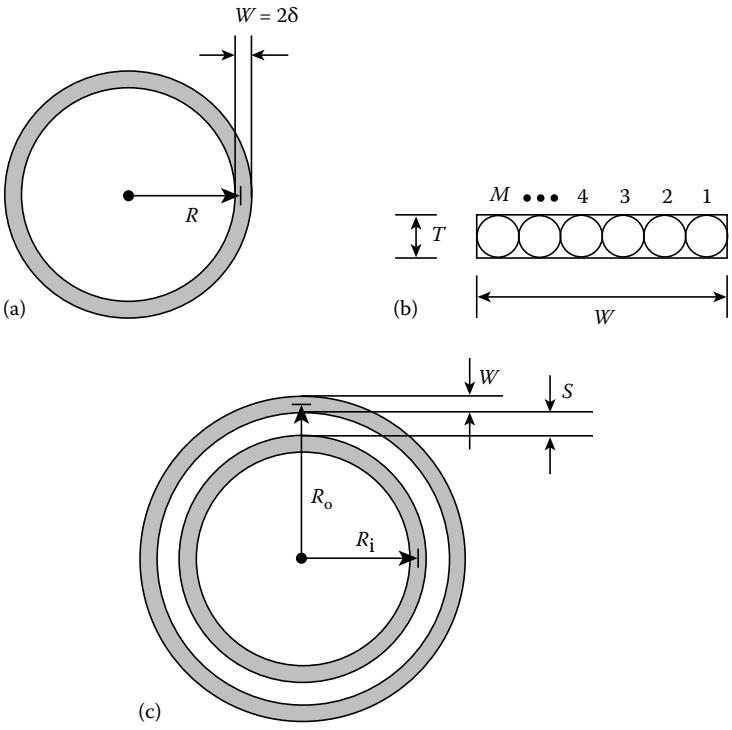


FIGURE 1.92 Calculation of self-inductance and mutual inductance for circular conductors. (a) External self-inductance; (b) internal self-inductance; and (c) mutual inductance.

where

$$k^2 = \frac{4R(R-\delta)}{(2R-\delta)^2} \tag{1.264}$$

and μ is the permeability of the conductor (equals 4π nH/cm for nonmagnetic conductors), and δ is one-half the conductor width W. $K(k)$ and $E(k)$ are the *complete elliptic integrals* of the first and second kind, respectively, and are given by

$$K(k) = \int_0^{\pi/2} \frac{d\phi}{\sqrt{1-k^2\sin^2\phi}}$$

$$E(k) = \int_0^{\pi/2} \sqrt{1-k^2\sin^2\phi}\, d\phi$$

The internal self-inductance is determined based on the concept of magnetic field energy. As shown in Figure 1.92b, the flat circular conductor is first approximated by an M number of round circular conductors [14] that are electrically in parallel and each conductor has a diameter equal to the thickness T of the flat conductor. The internal self-inductance of each round conductor is then determined as [25].

$$L = \frac{\mu}{8\pi}(\text{nH/cm})$$

The internal self-inductance of the flat conductor thus equals the parallel combination of these M components

$$L_S \approx \frac{\mu}{4}\left\{\sum_{i=1}^{M}[R - \delta + T(i - 0.5)]^{-1}\right\}^{-1}(\text{nH}) \tag{1.265}$$

where $R - \delta + T(i - 0.5)$ is the effective radius from the center of the loop to the center of the round conductor i. The typical contribution from the internal self-inductance of Equation 1.265 is less than 5% the contribution from the external self-inductance of Equation 1.263.

Mutual inductance. The mutual inductance of the two circular loops of Figure 1.92c depends on the inner radius R_i and the outer radius R_o. For any two adjacent loops of the circular spiral inductor, the outer radius is related to the inner radius by the simple relation $R_o = R_i + (W + S)$. The mutual inductance is determined based on the *Neumann's* line integral given as follows:

$$L_M = \frac{\mu}{4\pi}\int_C\int_C \frac{d\mathbf{l}_1 \cdot d\mathbf{l}_2}{D}$$

where

$d\mathbf{l}_1 \cdot d\mathbf{l}_2$ represents the dot product of the differential lengths
D is the distance separating the differential $\mathbf{l}_1$ vector and $\mathbf{l}_2$ vector

The static mutual inductance [25] is

$$L_M = \mu\sqrt{R_i R_o}\left[\left(\frac{2}{k} - k\right)K(k) - \frac{2}{k}E(k)\right](\text{nH}) \tag{1.266}$$

where

$$k^2 = \frac{4R_i R_o}{(R_i + R_o)^2} \tag{1.267}$$

Figure 1.93 shows plots of the external self-inductance and the mutual inductance as expressed in Equations 1.263 and 1.266, respectively. The conductor dimensions are given in μm.

As in the rectangular spiral inductor, the ground-plane effect and the retardation effect of the circular spiral inductor must be fully accounted for. The circuit model of Figure 1.87 can be used to characterize the electrical behavior of the circular inductor.

A comparison between the rectangular spiral of Figure 1.84 and the circular spiral of Figure 1.91 is shown in Figure 1.94, where the total inductance L_T is plotted against the turn number N. Both inductors have the same innermost dimension, the same conductor width, space, and thickness. The dimensions are given in μm, and the ground-plane effect and the retardation effect are not considered. For a given turn number, the rectangular spiral yields a higher inductance per semiconductor area than the circular spiral. Figure 1.95 shows a plot of the inductor Q vs. the total inductance of the same spiral inductors under consideration. Due to a higher inductance per length ratio, the Q of the circular inductor is higher than that of the rectangular inductor, about 10% for high inductance values.

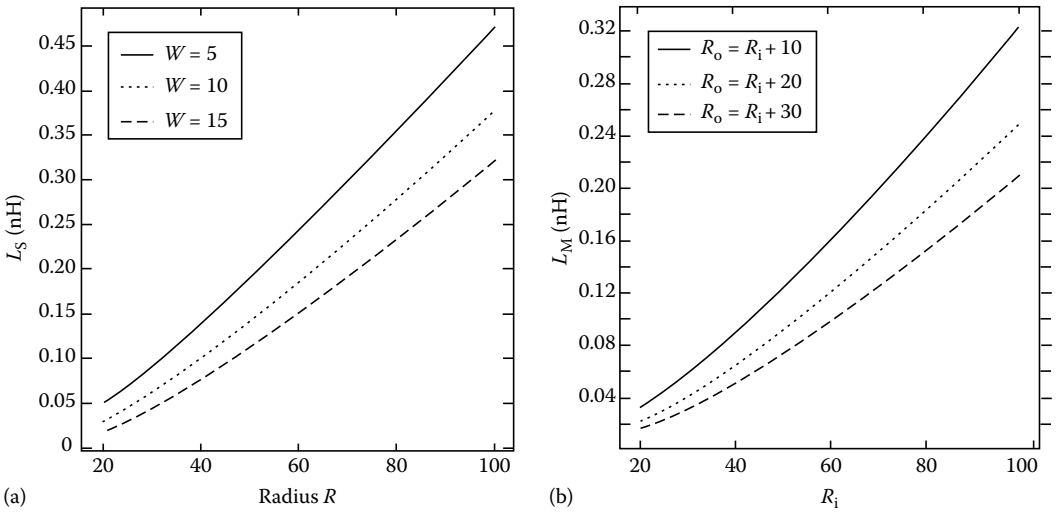


FIGURE 1.93 (a) External self-inductance as a function of radius and width for circular conductors. (b) Mutual inductance as a function of radii R_i and R_o for circular conductors.

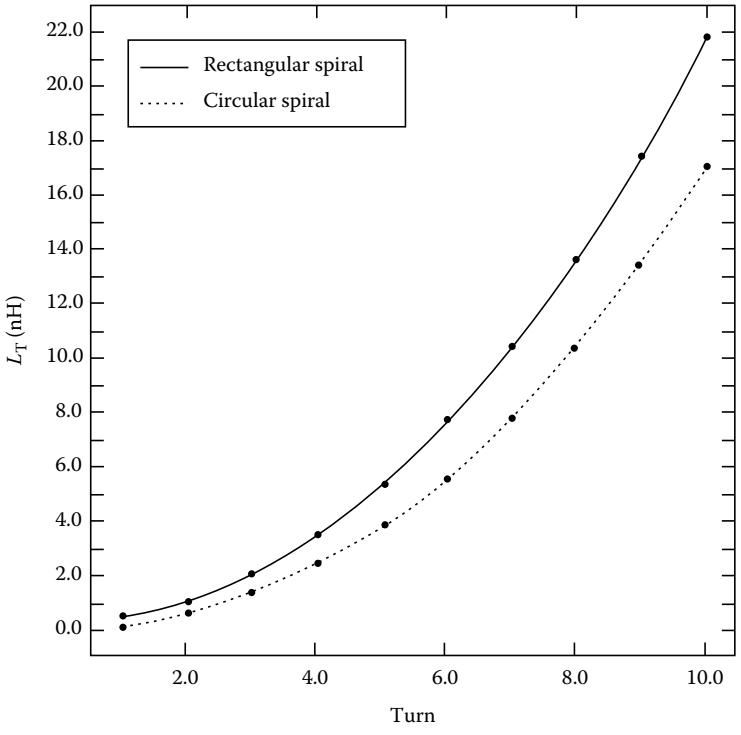


FIGURE 1.94 Total static inductance vs. turn number for the rectangular and circular inductors. The ground-plane effect is neglected. Innermost center dimension is 88 by 88, $W = 6$, $S = 3$, $T = 1.2$.

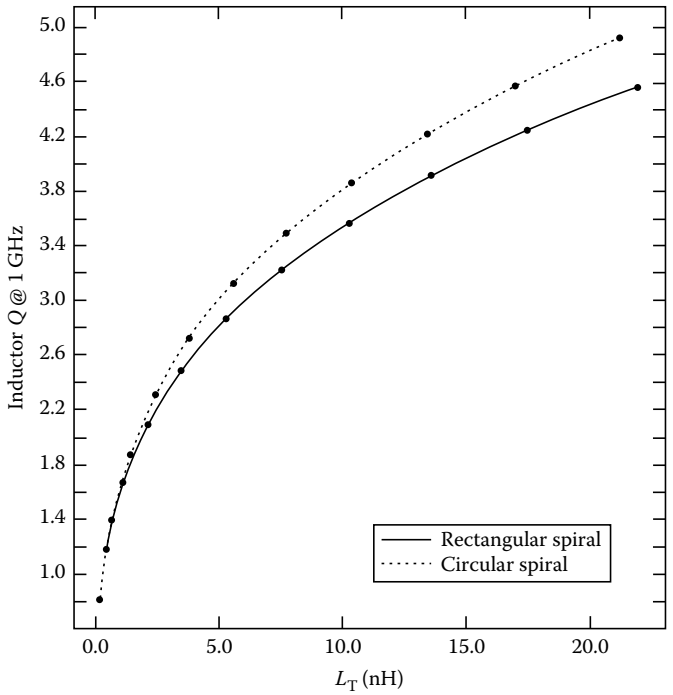


FIGURE 1.95 Inductor Q vs. total inductance for the rectangular and circular inductors. Metal sheet resistance is 25 mΩ/N. Innermost dimension is 88, $W = 6$, $S = 3$, $T = 1.2$.

References

1. I. Bahl and P. Bhartia, *Microwave Solid State Circuit Design*, New York: Wiley, 1988.
2. T. G. Bryant and J. A. Weiss, Parameters of microstrip transmission lines and of coupled pairs of microstrip lines, *IEEE Trans. Microwave Theory Tech.*, MTT-16, 1021–1027, 1968.
3. H. R. Camenzind, *Electronic Integrated Systems Design*, New York: Van Nostrand Reinhold, 1972.
4. E. M. Conwell, Properties of silicon and germanium, *Proc. IRE*, 46, 1281–1300, 1958.
5. R. Garg and I. J. Bahl, Characteristics of coupled microstrip lines, *IEEE Trans. Microwave Theory Tech.*, MTT-27, 700–705, 1979.
6. F. R. Gleason, Thin-film microelectronic inductors, in *Proc. Nat. Electron. Conf.*, 1964, pp. 197–198.
7. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 2nd ed., New York: Wiley, 1984.
8. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley, 1984.
9. H. M. Greenhouse, Design of planar rectangular microelectronic inductor, *IEEE Trans. Parts, Hybrids, Packaging*, PHP-10, 101–109, 1974.
10. F. W. Grover, *Inductance Calculations*, New York: Van Nostrand, 1946.
11. E. Hammerstad and O. Jensen, Accurate models for microstrip computer-aided design, *IEEE MTT-S Dig.*, 80, 407–409, 1980.
12. J. C. Irwin, Resistivity of bulk silicon and of diffused layers in silicon, *Bell Syst. Tech. J.*, 41, 387–410, 1962.
13. C. Jacoboni, C. Canali, G. Ottaviani, and A. A. Quaranta, A review of some charge transport properties of silicon, *Solid State Electron.*, 20, 77–89, 1977.

14. R. L. Kemke and G. A. Burdick, Spiral inductors for hybrid and microwave applications, in *Proc. Electron. Components Conf.*, 1974, pp. 152–161.
15. M. Kirschning and R. H. Jansen, Accurate wide-range design equations for the frequency-dependent characteristics of parallel-coupling microstrip lines, *IEEE Trans. Microwave Theory Tech.*, MTT-32, 83–90, 1984.
16. D. Krafcsik and D. Dawson, A close-form expression for representing the distributed nature of the spiral inductor, *IEEE MTT-S Dig.*, 86, 87–92, 1986.
17. R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, 2nd ed., New York: Wiley, 1986.
18. N. M. Nguyen and R. G. Meyer, Si IC-compatible inductors and LC passive filters, *IEEE J. Solid-State Circuits*, 25, 1028–1031, 1990.
19. N. M. Nguyen and R. G. Meyer, A Si bipolar monolithic RF bandpass amplifier, *IEEE J. Solid-State Circuits*, 27, 123–127, 1992.
20. N. M. Nguyen and R. G. Meyer, A 1.8-GHz monolithic LC voltage-controlled oscillator, *IEEE J. Solid-State Circuits*, 27, 444–450, 1992.
21. N. M. Nguyen and R. G. Meyer, Start-up and frequency stability in high-frequency oscillators, *IEEE J. Solid-State Circuits*, 27, 810–820, 1992.
22. M. Parisot, Y. Archambault, D. Pavlidis, and J. Magarshack, Highly accurate design of spiral inductors for MMIC's with small size and high cut-off frequency characteristics, *IEEE MTT-S Dig.*, 84, 106–110, 1984.
23. E. Pettenpaul, H. Kapusta, A. Weisgerber, H. Mampe, J. Luginsland, and I. Wolff, CAD models of lumped elements on GaAs up to 18 GHz, *IEEE Trans. Microwave Theory Tech.*, 36, 294–304, 1988.
24. R. A. Pucel, Design considerations for monolithic microwave circuits, *IEEE Trans. Microwave Theory Tech.*, MTT-29, 513–534, 1981.
25. S. Ramon, J. R. Whinnery, and T. V. Duzer, *Fields and Waves in Communication Electronics*, 2nd ed., New York: Wiley, 1984.
26. R. M. Warner, Jr., and J. N. Fordemwalt, *Integrated Circuits*, New York: McGraw-Hill, 1965.

1.5 Chip Parasitics in Analog Integrated Circuits

Martin A. Brooke

The parasitic elements in electronic devices and interconnect limit the performance of all ICs. No amount of improvement in device performance or circuit design can completely eliminate these effects. Thus, as circuit speeds increase, unaccounted for interconnect parasitics become a more and more common cause of analog IC design failure. Hence, the causes, characterization, and modeling of significant interconnect parasitics are essential knowledge for good analog IC design [1–4].

1.5.1 Interconnect Parasitics

The parasitics due to the wiring used to connect devices together on chip produce a host of problems. Unanticipated feedback through parasitic capacitances can cause unwanted oscillation. Mismatch due to differences in interconnect resistance contribute to unwanted offset voltages. For very-high-speed ICs, the inductance of interconnects is both a useful tool and a potential cause of yield problems.

Even the interactions between interconnect lines are both important and very difficult to model. So too are the distributed interactions of resistance, capacitance, and (in high-speed circuits) inductance that produce transmission line effects.

1.5.1.1 Parasitic Capacitance

Distributed capacitance of IC lines is perhaps the most important of all IC parasitics. It can lower the bandwidth of amplifiers, alter the frequency response of filters, and cause oscillations.

Physics. Every piece of IC interconnect has capacitance to the substrate. In the case of silicon circuitry, the substrate is conductive and connected to an ac ground, thus there is a capacitance to ground from every circuit node due to the interconnect. Figure 1.96 illustrates this substrate capacitance interconnect parasitic. The capacitance value will depend on the total area of the interconnect, and on the length of edge associated with the interconnect. This edge effect is due to the nonuniformity of the electric field at the interconnect edges. The nonuniformity of the electric field at edges is such that the capacitance value is larger for a given area of interconnect near the edge than elsewhere.

In addition to the substrate capacitance, all adjacent pieces of an interconnect will have capacitance between them. This capacitance is classified into two forms, overlap capacitance, and parallel line capacitance (also known as proximity capacitance). Overlap capacitance occurs when two pieces of interconnect cross each other, while parallel line capacitance occurs when two interconnect traces run close to each other for some distance.

When two lines cross each other, the properties of the overlapping region will determine that size of the overlap capacitance. The electric field through a cross section of two overlapping lines is illustrated in Figure 1.97. The electric field becomes nonuniform near the edges of the overlapping region, producing an edge-dependent capacitance term. The capacitance per unit area at the edge is always greater than elsewhere and, if the overlapping regions are small, the edge capacitance effect can be significant.

The size of parallel line capacitance depends on the distance for which the two lines run side by side and on the separation of the lines. Since parallel line capacitance occurs only at the edges of an interconnect, the electric field that produces it is very nonuniform. This particular nonuniformity, as illustrated in Figure 1.98, makes the capacitance much smaller for a given area of interconnect than either overlap or substrate capacitance. Thus, two lines must run parallel for some distance for this capacitance to be important. The nonuniformity of the electric field makes the dependence of the capacitance on line separation highly nonlinear, as a result the capacitance value decreases much more rapidly with separation than it would if it depended linearly on the line separation.

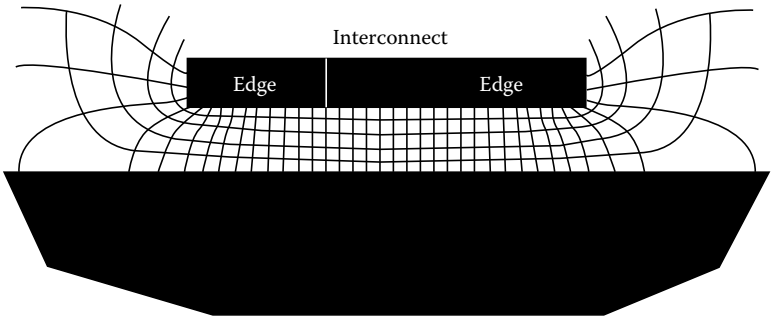


FIGURE 1.96 Substrate capacitance. The electric field distorts at the edges, making the capacitance larger there than elsewhere.

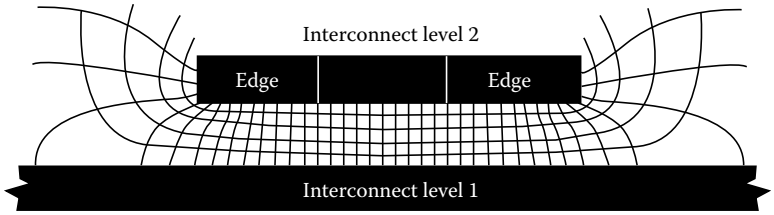


FIGURE 1.97 Overlap capacitance. The bottom interconnect level will have edges into and out of the page with distorted electric field similar to that shown for the top level of interconnect.

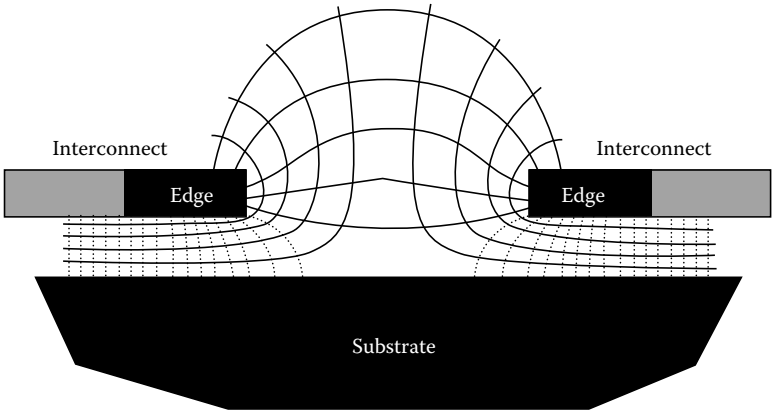


FIGURE 1.98 Parallel line capacitance. Only the solid field lines actually produce line-to-line capacitance, the dashed lines form substrate capacitance.

Modeling. In the absence of significant interconnect resistance effects, all of the parasitic capacitances can be modeled with enough accuracy for most analog circuit design applications by dissecting the interconnect into pieces with similar capacitance characteristics and adding up the capacitance of each piece to obtain a single capacitance term. For example, the dissected view of a piece of interconnect with substrate capacitance is shown in Figure 1.99. The interconnect has been dissected into squares that fall into three

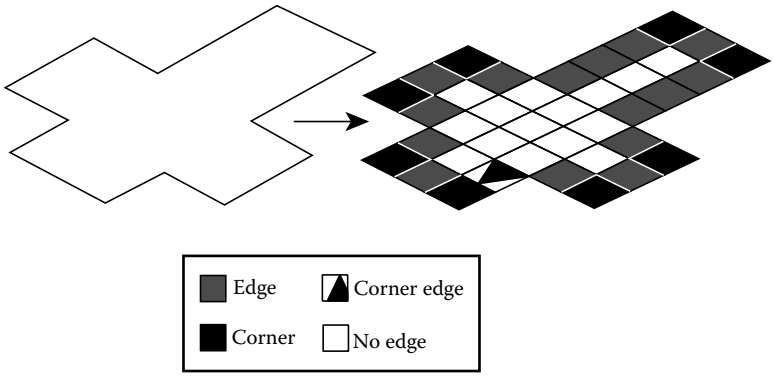


FIGURE 1.99 Determining substrate capacitance. The capacitance of each square in the dissected interconnect segment is summed.

classes: two types of edges, and one center type. The capacitance to the substrate for each of these squares in parallel and thus the total capacitance of the interconnect segment is simply the sum of the capacitance of each square. If the substrate capacitance contribution of each square has been previously measured or calculated, the calculation of the total interconnect segment substrate capacitance involves summing each type of squares capacitance multiplied by the number of squares of that type in the segment.

The accuracy of this modeling technique depends solely on the accuracy of the models used for each type of square. For example, in Figure 1.99, the accuracy could be improved by adding one more type of edge square to those that are modeled. One of these squares has been shaded differently in the figure and is called the corner edge square.

For the nonedge pieces of the interconnect the capacitance is approximately a parallel-plate capacitance and can be computed from Equation 1.268.

$$C = \frac{A \cdot \varepsilon_r \cdot \varepsilon_0}{t} \tag{1.268}$$

where

A is the area of the square or piece of interconnect
t is the thickness of the insulation layer beneath the interconnect
ε_r is the relative dielectric constant of the insulation material
ε_0 is the dielectric constant of free space

For silicon ICs insulated with silicon dioxide the parameters are given in Table 1.10.

The capacitance of edge interconnect pieces will always be larger than nonedge pieces. The amount by which the edge capacitance increases will depend on the ratio of the size of the piece of interconnect and the thickness of the insulation layer beneath the interconnect. If the interconnect width is significantly larger than the thickness of the insulation then edge effects are probably small and can be ignored. However, when thin lines are used in ICs the edge effects are usually significant. The factor by which the edge capacitance can increase over the parallel-plate approximation can easily be as high as 1.5 for thin lines.

The modeling of overlap capacitance is handled in the same fashion as substrate capacitance. The region where interconnect lines overlap is dissected into edges and nonedges and the value of capacitance for each type of square summed up to give a total capacitance between the two circuit nodes associated with each piece of interconnect that overlaps. The area of overlap between the two layers of interconnect can be used as A in Equation 1.268, while that separation between the layers can be used as t. The strong distortion of the electric fields will increase the actual value above this idealized computed value by a factor that depends on the thickness of the lines. This factor can be as high as 2 for thin lines.

Parallel line capacitance can also be handled in a manner similar to that used for substrate and overlap capacitance. However, we must now locate *pairs* of edge squares, one from each of the adjacent interconnect lines. In Figure 1.100, one possible pairing of the squares from adjacent pieces of interconnect is shown. The capacitance for each type of pair of squares is added together, weighted by the number of pairs of each type to get a single capacitance that connects the circuit nodes associated with each interconnect line.

The effect on the capacitance of the spacing between pairs must be either measured or computed for each possible spacing, and type of pair of squares. One approach to this is to use a table of measured or

TABLE 1.10 Parameters for Calculation of Substrate Capacitance in Silicon ICs Insulated with Silicon Dioxide

Parameter	Value
ε_r	3.9
ε_0	$8.854 \cdot 10^{-12}$ F/m
t	$1–5 \cdot 10^{-6}$ m

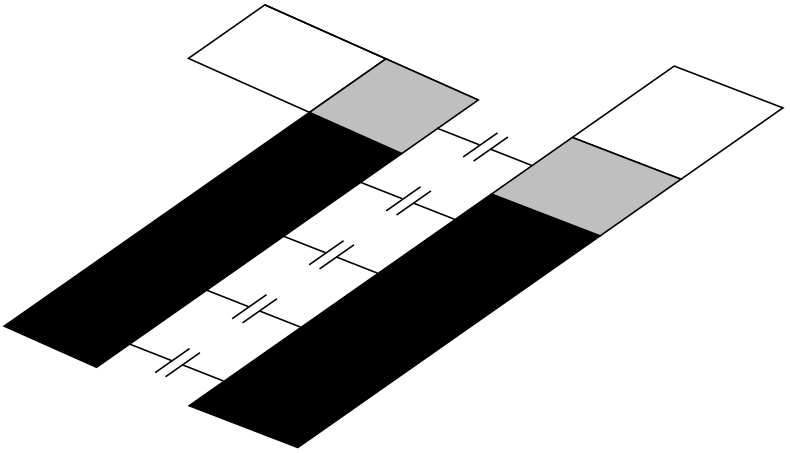


FIGURE 1.100 Determining parallel line capacitance. The differently shaded pairs of squares are different types and will each have a different capacitance between them.

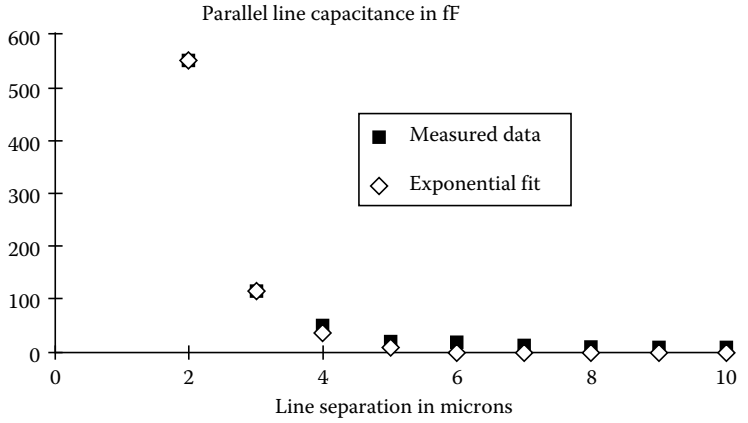


FIGURE 1.101 Parallel line capacitance measured from a silicon IC. The diamonds are an exponential fit to the data (using Equation 1.269). The fit is excellent at short separations when the capacitance is largest.

computed capacitances and separation distances. The measured parallel line capacitance between silicon IC lines for a variety of separations is presented in Figure 1.101. From the figure, we see that the capacitance value decreases exponentially with line separation. Thus an exponential fit to measured or simulated data is good choice for computing the capacitance [7,8].

Equation 1.269 can be used to predict the parallel line capacitance C for each type of pair of edge squares. L is the length of the edge of the squares, and the parameters C_c and S_d are computed or fit to measured coupling capacitance data like that in Figure 1.101.

$$C = C_c \cdot L \cdot e^{-(s/S_d)} \tag{1.269}$$

Effects on circuits. The effects that parasitic capacitances are likely to produce in circuits range from parametric variations, such as reduced bandwidth, to catastrophic failures, such as amplifier oscillation. Each type of parasitic capacitance produces a characteristic set of problems. Being aware of these typical problems will ease diagnosis of actual, or potential, parasitic capacitance problems.

Substrate capacitance usually causes lower than expected bandwidth in amplifiers and lowering of the poles in filters. The capacitance is always to ac ground and thus increases device and circuit capacitances to ground. Thus, circuit nodes that have a dominant effect on amplifier bandwidth, or filter poles, should be designed to have as little substrate capacitance as possible. Another, more subtle, parametric variation that can be caused by substrate capacitance is frequency-dependent mismatch. For example, if the parasitic capacitance to ground is different between the two inputs of a differential amplifier, then, for fast transient signals, the amplifier will appear unbalanced. This could limit the accuracy high-speed comparators, and is sometimes difficult to diagnose since the error only occurs at high speeds.

Overlap and parallel line capacitance can cause unwanted ac connections to be added to a circuit. These connections will produce crosstalk effects and can result in unstable amplifiers. The output interconnect and input interconnect of high-gain or high-frequency amplifiers must thus be kept far apart at all times. Care must be taken to watch for series capacitances of this type. For example, if the output and input interconnect of an amplifier both cross the power supply interconnect, unwanted feedback can result if the power supply line is not well ac grounded. This is a very common cause of IC amplifier oscillation. Because of the potential for crosstalk between parallel or crossing lines, great care should also be taken to keep weak (high-impedance) signal lines away from strong (low-impedance) signal lines.

1.5.1.2 Parasitic Resistance

For analog IC designers, the second most important interconnect parasitic is resistance. This unexpected resistance can cause both parametric problems, such as increased offset voltages, and catastrophic problems such as amplifier oscillation (for example, poorly sized power supply lines can cause resistive positive feedback paths in high gain amplifiers called ''ground loops''). To make matters worse, the resistivity of IC interconnect has been steadily increasing as the line widths of circuits have decreased.

Physics. Except for superconductors, all conductors have resistance. A length of interconnect used in an IC is no exception. The resistance of a straight section of interconnect is easily found by obtaining the resistance per square for the particular interconnect layer concerned, and then adding up the resistance of each of the series of squares that makes up the section. This procedure is illustrated in Figure 1.102.

For more complicated interconnect shapes the problem of determining the resistance between two points in the interconnect is also more complex. The simplest approach is to cut the interconnect up into rectangles and assume each rectangle has a resistance equal to the resistance per square of the interconnect material times the number of full and partial squares that will fit along the direction of current flow in the rectangle [5]. This scheme works whenever the direction of current flow is clear; however, for corners and intersections of interconnect the current flow is in fact quite complex. Figure 1.103 shows the kind of current flow that can occur in an interconnect section with complex geometry.

Modeling. To account for the effects of complex current flows the resistance of complex interconnect geometries must be determined by measurement or simulation. One simple empirical approach is to cut out sections of resistive material in the same shape as the interconnect shape to be modeled, and then

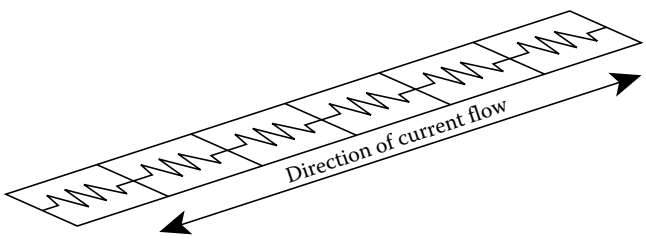


FIGURE 1.102 Determining the resistance of a length of interconnect. Each square has the same resistance regardless of size.

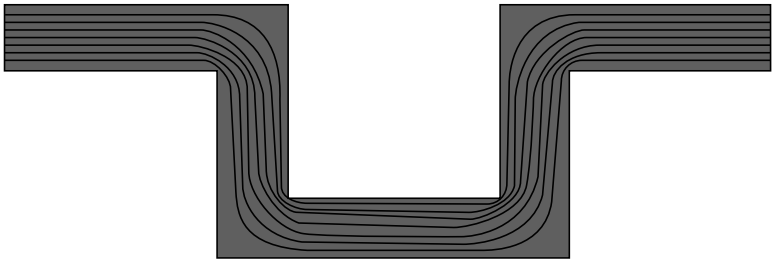


FIGURE 1.103 Current flow in a complex interconnect geometry.

measure the resistance. The resistance for other materials can be found by multiplying by the ratio of the respective resistances per square of the two materials.

Once the resistance has been found for a particular geometry it can be used for any linear scaling of that geometry. For most types of IC interconnect all complex geometries can be broken up into relatively few important subgeometries. If tables of the resistance of these subgeometries for various dimensions and connection patterns are obtained, the resistance of quite complex shapes can be accurately calculated by connecting the resistance of each subgeometry together and calculating the resistance of the connected resistances. This calculation can usually be performed quickly by replacing series and parallel connected resistor pairs with their equivalents. The process of breaking a complex geometry into subgeometries, constructing the equivalent connected resistance, and forming a single resistance for an interconnect section is illustrated in Figure 1.104.

Effects on circuits. The resistance of interconnect can have both parametric and catastrophic effects on circuit performance. Even small differences in the resistance on either input side of a differential amplifier can lead to increased offset voltage. Thus, when designing differential circuits care must be taken to make the interconnect identical on both input sides, as this ensures that the same resistance is present in both circuits.

The resistance of power supply interconnect can lead to both parametric errors in the voltages supplied and catastrophic failure due to oscillation. If power supply voltages are assumed to be identical in two parts of a circuit and, due to interconnect resistance, there is a voltage drop from one point to the next, designs that rely on the voltages being the same may fail. In high-gain and feedback circuits the resistance of the ground and power supply lines may become an unintentional positive feedback resistance which could lead to oscillation. Thus output and input stages for high-gain amplifiers will usually require separate ground and power supply interconnects. This ensures that no parasitic resistance is in a feedback path.

When using resistors provided in an IC process, the extra resistance provided by the interconnect may cause inaccuracies in resistor values. This would be most critical for small resistance values. The only solution in this case is to accurately compute the interconnect resistance. Since most resistance layers

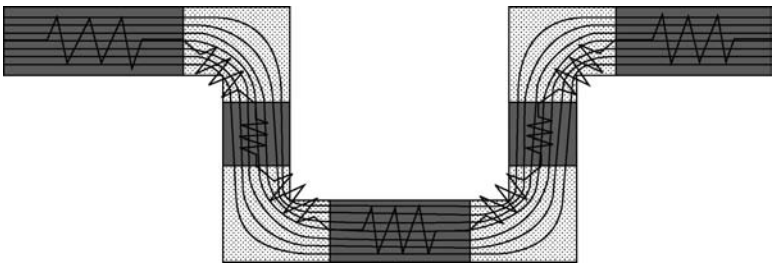


FIGURE 1.104 The process of breaking a complex geometry into subgeometries, constructing the equivalent connected resistance, and forming a single resistance for an interconnect section. In this example, only two subgeometries are used: a corner subgeometry and a basic rectangular subgeometry.

provided in analog IC processes are just a form of high resistivity interconnect, the methods described here for accurately computing the resistance of interconnect are also useful for predicting the resistance of resistors to be fabricated.

1.5.1.3 Parasitic Inductance

In high-speed ICs the inductance of long lines of interconnect becomes significant. In IC technologies that have an insulating substrate, such as gallium arsenide (GaAs) and silicon on insulator (SOI), reasonably high-performance inductive devices can be made from interconnect. In technologies with conductive substrates, resistive losses in the substrate restrict the application of interconnect inductance. High-frequency circuits are often tuned using interconnect inductance and capacitance (*LC*) to form a narrow bandpass filter or tank circuit, and *LC* transmission lines, or stubs, made from interconnect are useful for impedance matching. There is much similarity between this use of parasitic inductance and the design of microstripline-printed circuit boards. The major difference being that inductance does not become significant in IC interconnect until frequencies in the gigahertz are reached.

In order to make a good interconnect inductance, there are two requirements. First, there must not be any resistive material within range of the magnetic field of the inductance. If this occurs then induced currents flowing in the resistive material will make the inductor have high series resistance (low *Q* factor). This would make narrow bandwidth bandpass filters difficult to make using the inductance, and make transmission lines made from the interconnect lossy. The solution is to have an insulating substrate, or to remove the substrate from beneath the inductor.

The second requirement for large inductance is to form a coil or other device to concentrate the magnetic field lines. Within the confines of current IC manufacturing, spiral inductors, like that illustrated in Figure 1.105 are the most common method used to obtain useful inductances.

1.5.1.4 Transmission Line Behavior

Two types of transmission line behavior are important in ICs, *RC* transmission lines and *LC*/*RLC* transmission lines. For gigahertz operation inductive transmission lines are important. These can be lossy *RLC* transmission lines if a conductive substrate such as silicon is used, or nearly lossless *LC* transmission lines if an insulating substrate such as GaAs is used. The design of inductive transmission lines is very similar to designing microstripline-printed circuit boards. At lower frequencies of

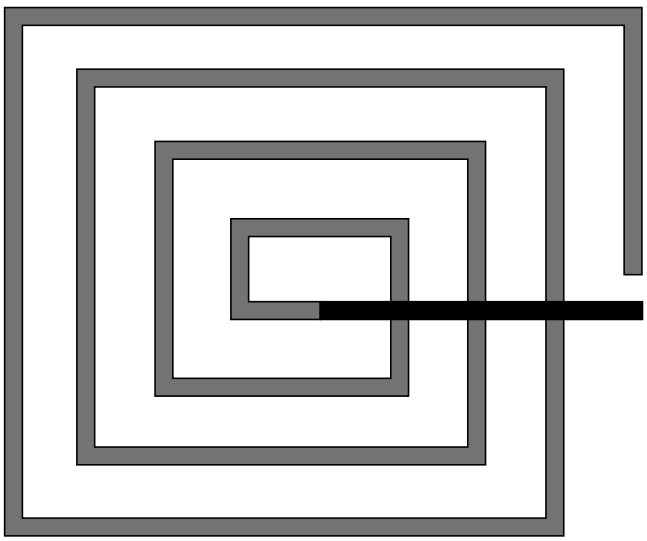


FIGURE 1.105 Spiral inductance used in insulated substrate ICs for gigahertz frequency operation.

10–1000 MHz resistive capacitive (*RC*) transmission lines are important for long low resistivity interconnect lines or short high resistivity lines.

RC transmission lines are of concern to analog circuit designers working in silicon ICs. When used correctly, an interconnect can behave as though it were purely capacitive in nature. However, when a higher resistivity interconnect layer, such as polysilicon or diffusion is used, the distributed resistance and capacitance can start to produce transmission line effects at relatively short distances. Similarly, for very long signal distribution lines or power supply lines, if they are not correctly sized, transmission line behavior ensues.

Physics. One method for modeling distributed transmission line interconnect effects is lumped equivalent modeling [6]. This method is useful for obtaining approximate models of complex geometries quickly, and is the basis of accurate numerical finite element simulation techniques. For analog circuit designers the conversion of interconnect layout sections into lumped equivalent models also provides an intuitive tool to understanding distributed transmission line interconnect behavior.

To be able to model a length of interconnect as a lumped *RC* equivalent, the error between the impedance of the interconnect when correctly treated as a transmission line, and when replaced with the lumped equivalent, must be kept low. If this error is *e*, then it can be shown that the maximum length of interconnect that can be modeled as a simple *RC T* or Π network is given in Equation 1.270. In the equation, *R* is the resistance per square of the particular type of interconnect used, *C* is the capacitance per unit area, and ω is the frequency of operation in radians per second.

$$D < \sqrt{\frac{3 \cdot e}{\omega \cdot R \cdot C}} \tag{1.270}$$

This length can be quite short. Consider the case of a polysilicon interconnect line in a 1.2 μm CMOS process that has a resistance per square of 40 Ω a capacitance per unit are of 0.1 fF/μm^2. For an error *e* of 10% the maximum line length of minimum width line that can be treated as a lumped T or Π network for various frequencies is given in Table 1.11. Longer interconnect lines than this must be cut up into lengths less than or equal to the length given by Equation 1.270.

Modeling. The accurate modeling of distributed transmission line effects in ICs is best performed with lumped equivalent circuits. These circuits can be accurately extracted by dissecting the interconnect geometry into lengths that are, at most, as long as the length given by Equation 1.270. These lengths are then modeled by either a *T* or Π lumped equivalent *RC* network. The extraction of the resistance and capacitance for these short interconnect sections can now follow the same procedures as were described in Sections 1.5.1.2 and 1.5.1.1. The resulting *RC* network is then an accurate transmission line model of the interconnect. Figure 1.106 shows an example of this process.

Effects on circuits. Several parametric and catastrophic problems can arise due to unmodeled transmission line behavior. Signal propagation delays in transmission lines are longer than predicted by a single lumped capacitance and resistance model of interconnect. Thus, ignoring the effects of transmission lines can result in slower circuits than expected. If the design of resistors for feedback networks

TABLE 1.11 The Maximum Length of Minimum Width Polysilicon Line That Can Be Modeled with a Single Lumped *RC T* or π Network and Remain 10% Accurate

Frequency (MHz)	Length (μm)
10	1262
100	399
1000	126

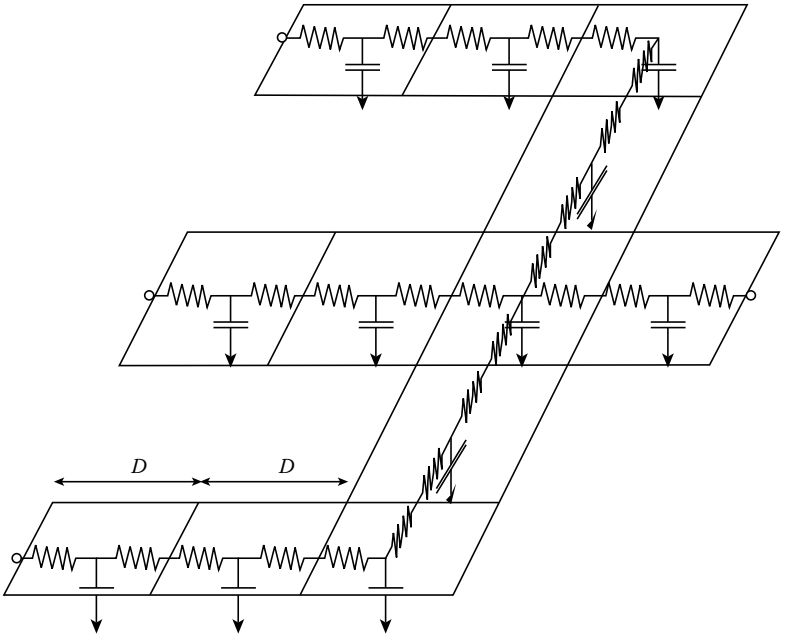


FIGURE 1.106 The extraction of an accurate *RC* transmission line model for resistive interconnect. The maximum allowable length D is computed from Equation 1.270.

results in long lengths of the resistive interconnect used to make the resistors, these resistors may in fact be *RC* transmission lines. The extra delay produced by the transmission line may well cause oscillation of the feedback loops using these resistors. The need for decoupling capacitors in digital and analog circuit power supplies is due to the *RC* transmission line behavior of the power supply interconnect. Correct modeling of the *RC* properties of the power distribution interconnect is needed to see whether fast power supply current surges will cause serious changes in the supply voltage or not.

1.5.1.5 Nonlinear Interconnect Parasitics

A number of types of interconnect can have nonlinear parasitics. These nonlinear effects are a challenge to model accurately because the effect can change with the operating conditions of the circuit. A conservative approach is to model the effects as constant at the worst likely value they can attain. This is adequate for predicting parameters, like circuit bandwidth, that need only exceed a specification value. If the specifications call for accurate prediction of parasitics then large nonlinear parasitics are generally undesirable and should be avoided.

Most nonlinear interconnect parasitics are associated with depletion or inversion of the semiconductor substrate. A diffusion interconnect is insulated from conducting substrates such as silicon by a reversed biased diode. This diode's depletion region width varies with the interconnect voltage and results in a voltage-dependent capacitance to the substrate. For example, the diffusion interconnect in Figure 1.107 has voltage-dependent capacitance to the substrate due to a depletion region. The capacitance value depends on the depletion region thickness, which depends on the voltage difference between the interconnect and the substrate.

$$C = C_0 \cdot \left(1 - \left(\frac{V_s}{\varphi_B}\right)\right)^M \tag{1.271}$$

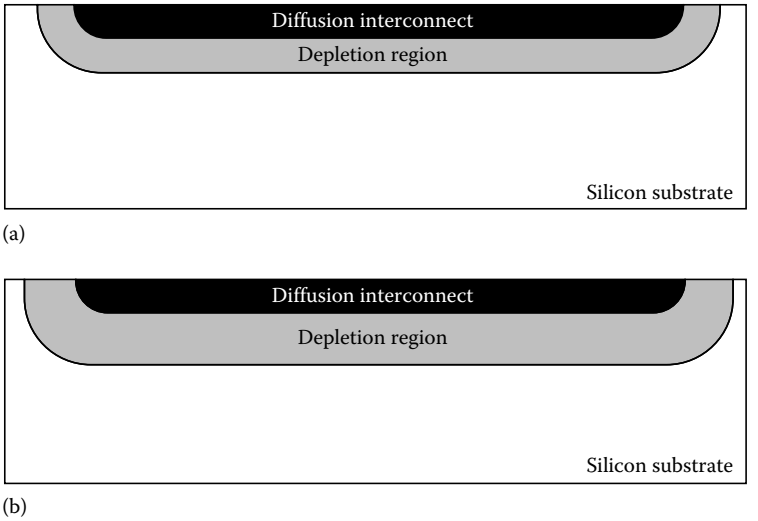


FIGURE 1.107 Diffusion interconnect has a voltage-dependent capacitance produced by the depletion region between the interconnect and the substrate. At low voltage difference between the interconnect and substrate (a), the capacitance is large. However, the capacitance decreases for larger voltage differences (b).

The typical equation for depletion capacitance is given in Equation 1.271. In this equation V_S is the voltage from the interconnect to the substrate, ϕ_B is the built-in potential of the semiconductor junction, M is the grading coefficient of the junction, while C_0 is the zero-bias capacitance of the junction. Since the capacitance is less than C_0 for reverse bias and the junction would not insulate for forward bias, we can assume that the capacitance is always less than C_0 and use C_0 as a conservative estimate of C. Because of the uncertainty in the exact structure of most semiconductor junctions φ_B and M are usually fit to measured capacitance versus voltage (CV) data.

Another common nonlinear parasitic occurs when metal interconnect placed over a conducting semiconductor substrate creates inversions at the semiconductor surface. This inversion layer increases the substrate capacitance of the interconnect and is voltage-dependent. To prevent this most silicon-IC manufacturers place an inversion-preventing implant on the surface of the substrate. The depletion between the substrate and n-type or p-type wells diffused into the substrate also creates a voltage-dependent capacitance. Thus use of the well as a high resistivity interconnect for making high value resistors will require consideration of a nonlinear capacitance to the substrate.

1.5.2 Pad and Packaging Parasitics

All signals and supply voltages that exit an IC must travel across the packaging interconnections. Just like the on-chip interconnect, the packaging interconnect has parasitic resistance, capacitance, and inductance. However, some of the packaging materials are significantly different in properties and dimension to those used in the IC, thus there are major differences in the importance of the various types of parasitics. Figure 1.108 is a typical packaged IC. The chief components of the packaging are the pads on the chip, the wire or bump bond used to connect the pad to the package, and then the package interconnect.

The pads used to attach wire bonds or bump bonds to ICs are often the largest features on an IC. The typical pad is 100 μm on a side and has a capacitance of 100 fF. In addition, protection diodes are often used on pads that will add a small nonlinear component to the pad capacitance.

The wire bonds that attach the pads to the package are typically very low resistivity and have negligible capacitance. Their major contribution to package parasitics is inductance. Typically, the

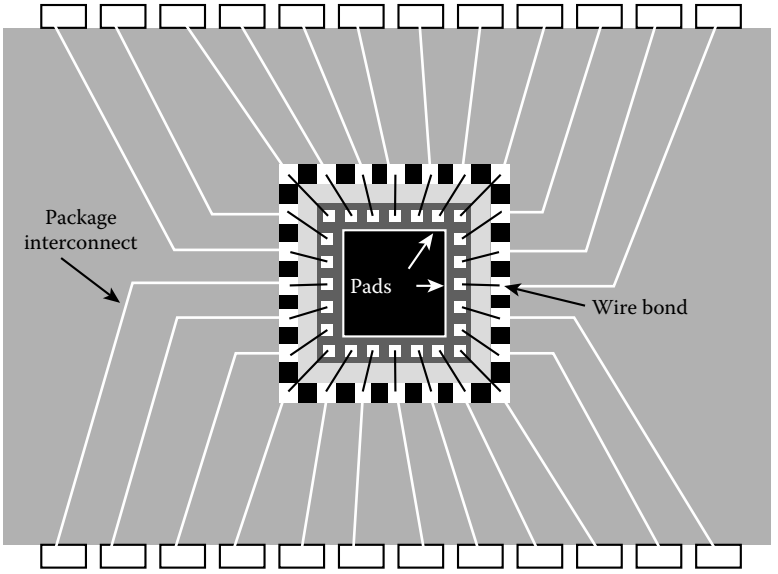


FIGURE 1.108 A packaged IC. The main sites of parasitics are the pad, bond, and package interconnect.

package interconnect inductance is greater than the wire bond inductance; however, when wire bonds are used to connect two ICs directly together, then the wire bond inductance is significant.

Often, the dominant component of package parasitics comes from the packaging interconnect itself. Depending on the package, there is inductance, capacitance to ground, and parallel line capacitance produced by this interconnect. Carefully made high-frequency packages do not exhibit much parallel line capacitance (at the expense of much capacitance to ground due to shielding), but in low-frequency packages with many connections this can become a problem.

Typical inductance and capacitance values for a high-speed package capable of output bandwidths around 5 GHz are incorporated into a circuit model for the package parasitics in Figure 1.109. When simulated with a variety of circuit source resistances (*RS*) this circuit reaches maximum bandwidth without peaking when the output resistance is 4 Ω. At lower output resistance, Figures 1.110 and 1.111 show that considerable peaking in the output frequency response occurs.

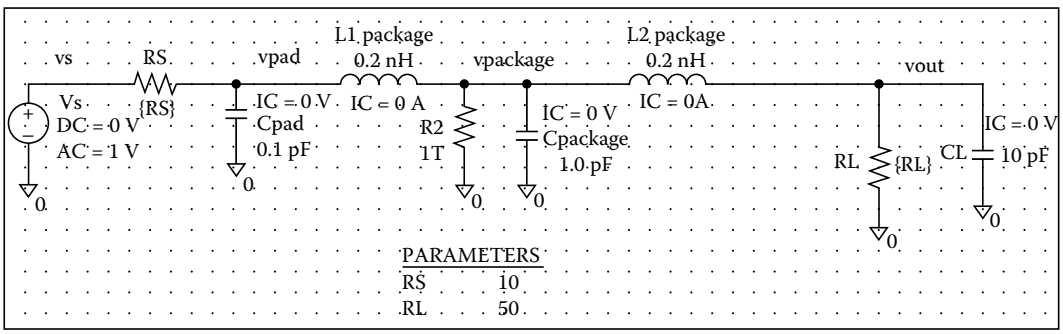


FIGURE 1.109 The circuit model of a high-frequency package output and associated parasitics. Cpad is the pad capacitance. L1package, Cpackage, R2, and L2package model the package interconnect. *RS* is the source resistance of the circuit. *RL* and *CL* are the external load.

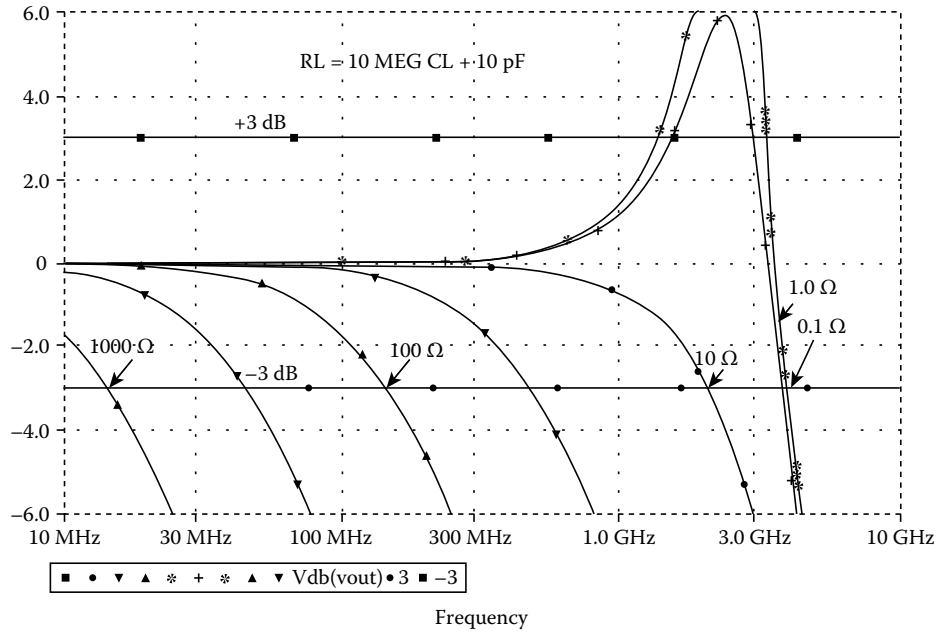


FIGURE 1.110 The PSPICE ac simulation of circuit in Figure 1.109 when the load resistance *RL* is 10 MΩ. This shows how the package inductance causes peaking for sufficiently low output resistance. In this case, peaking occurs for *RS* below 4 Ω and at about 2 GHz.

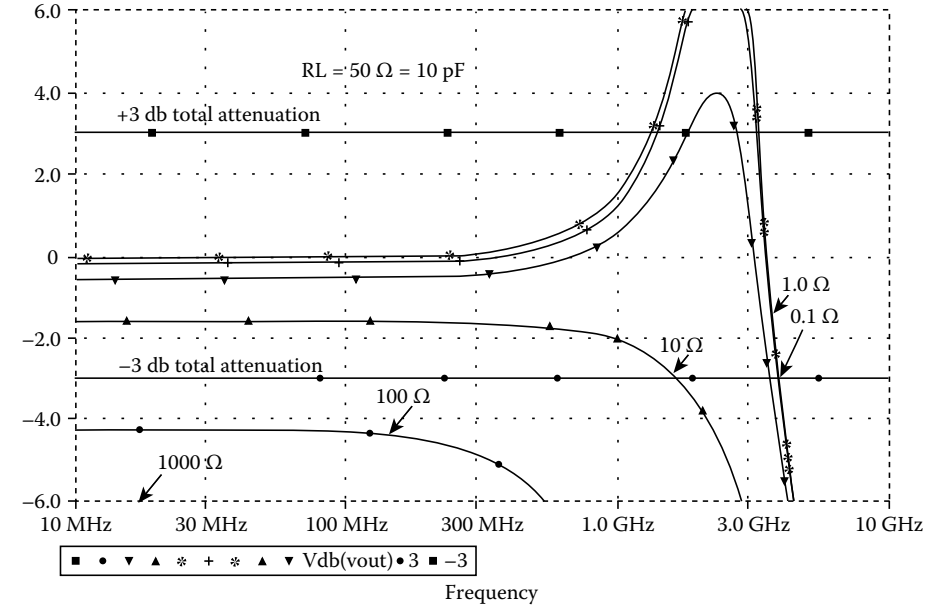


FIGURE 1.111 The PSPICE ac simulation of circuit in Figure 1.109 when the load is 50 Ω. The package inductance still causes peaking for *RS* below 4 Ω.

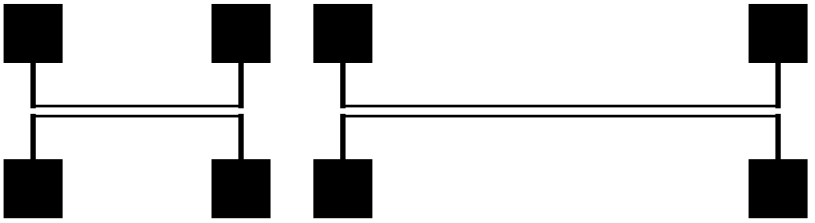


FIGURE 1.112 Test structures for measuring parallel line capacitance.

1.5.3 Parasitic Measurement

The major concern when measuring parasitics is to extract the individual parasitic values independently from measured data. This is normally achieved by exaggerating the effect that causes each individual parasitic in a special test structure, and then reproducing the structure with two or more different dimensions that will affect only the parasitic of interest. In this fashion, the effects of the other parasitics are minimized and can be subtracted from the desired parasitic in each measurement.

$$C_P = \frac{C_1 - C_2}{L_1 - L_2} \tag{1.272}$$

For example, to measure parallel line capacitance, the test structures in Figure 1.112 would be fabricated. These structures vary only in the length of the parallel lines. This means that if other parasitic capacitance ends up between the two signal lines used to measure the parasitic, then it will be a constant capacitance that can be subtracted from both measurements. The parallel line capacitance will vary in proportion to the variation of length between the two test structures. Thus the parallel line capacitance per unit length can be found from Equation 1.272. In this equation C_P is the parallel line capacitance per unit length, C_1 and C_2 are the capacitances measured from each test structure, and L_1 and L_2 are the length of the two parallel interconnect segments.

References

1. D. L. Carter and D. F. Guise, Effects of interconnections on submicron chip performance, *VLSI Design*, 4, 63–68, 1984.
2. H. B. Lunden, Detailed extraction of distributed networks and parasitics in IC designs, in *Proc. Euro. Conf. Circuit Theory, Design*, 1989, pp. 84–88.
3. R. A. Sainati and T. J. Moravec, Estimating high-speed circuit interconnect performance, *IEEE Trans. Circuits Syst*., 36, 533–541, April 1989.
4. D. S. Gao, A. T. Yang, and S. M. Kang, Modeling and simulation of interconnection delays and crosstalks in high-speed integrated circuits, *IEEE Trans, Circuits Syst*., 37, 1–8, Jan. 1990.
5. M. Horowitz and R. W. Dutton, Resistance extraction from mask layout data, *IEEE Trans. Computer-Aided Design Integrat. Circuits Syst*., 7, 1029–1037, Oct. 1988.
6. R. J. Antinone and G. W. Brown, The modeling of resistive interconnections for integrated circuits, *IEEE J. Solid-State Circuits*, SC-18, 200–203, April 1983.
7. A. E. Ruehli and P. A. Brennan, Capacitance models for integrated circuit, metallization wires, *IEEE J. Solid-State Circuits*, SC-10, 530–536, Dec. 1975.
8. S. Mori, I. Suwa, and J. Wilmore, Hierarchical capacitance extraction in an IC artwork verification system, in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1984, pp. 266–268.

2

Analog Circuit Cells

Kenneth V. Noren
University of Idaho

John Choma, Jr.
University of Southern California

J. Trujillo
University of Southern California

David G. Haigh
University College of London

Bill Redman-White
University of Southampton

Rahim Akbari-Dilmaghani
University College of London

Mohammed Ismail
Ohio State University

Shu-Chuan Huang
Ohio State University

Chung-Chih Hung
Tatung Institute of Technology

Trond Saether
Nordic VLSI A/S

2.1 Bipolar Biasing Circuits **2**-1
Common Bipolar Junction Transistor (BJT) Biasing Circuits
References **2**-10
2.2 Canonic Cells of Linear Bipolar Technology **2**-10
Introduction • Small-Signal Model • Single-Input–Single-Output Canonic Cells • Differential Amplifier
References **2**-54
2.3 MOSFET Biasing Circuits **2**-55
Introduction • Device Types and Models for Biasing • Voltage and Current Reference and Bias Circuits • Voltage and Current References Based on Less Usual Devices • Voltage References Based on N- and P-Doped Polysilicon Gate Threshold • Biasing of Simple Amplifiers and Other Circuits • Biasing of Circuits with Low Power Supply Voltage • Dynamic Biasing • Conclusions
References **2**-76
2.4 Canonical Cells of MOSFET Technology **2**-77
Matched Device Pairs • Unmatched Device Pairs • Composite Transistors • Super MOS Transistors • Basic Voltage Gain Cells • Conclusion
References **2**-98

2.1 Bipolar Biasing Circuits

Kenneth V. Noren

Establishing bias currents and voltages for building blocks comprising an overall design is fundamental to the design of bipolar integrated circuits. These building blocks include single-stage and differential amplifiers, output stages, etc. Biasing often has a direct relationship to electrical characteristics, such as gain, signal-swing, slew-rate, etc., of the individual building blocks and hence to the overall design. Biasing circuits include current sources, voltage references, and level-shifters. Most often, it is desirable that the integrated circuit design be robust and independent of a variety of external factors that can affect circuit performance. These factors include variations in process parameters, supply voltage, and temperature. Efforts to improve the performance of current sources and voltage references have led to many refinements and developments that have started from simple beginnings. This section presents some of

the fundamental current sources and voltage references used for biasing in bipolar integrated circuit technologies and refinements of these circuits that have evolved over time.

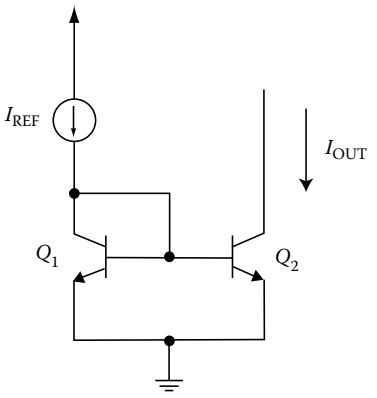


FIGURE 2.1 Simple current mirror.

2.1.1 Common Bipolar Junction Transistor (BJT) Biasing Circuits

2.1.1.1 Current Mirrors and Sources

The current mirror is a circuit that reproduces a reference current at one or more locations in larger circuit. A simple current mirror is depicted in Figure 2.1. Since $V_{BE1} = V_{BE2}$, $I_{OUT} \approx I_{REF}$, and the reference current I_{REF} is effectively mirrored to another location. In order to evaluate current mirrors and compare the properties of the many types of current mirrors to one another, we first define metrics for current mirrors and characteristics for an ideal current mirror. An ideal current mirror produces and output current that

1. Reproduces a reference current, exactly
2. Does not vary with loading (the output resistance [R_o] is infinite)
3. Is insensitive to process variations
4. Is insensitive to power supply variations
5. Is insensitive to temperature

2.1.1.1.1 Simple Current Mirror

The relationship between the output current and the reference current or the simple current mirror for matched transistors is

$$\frac{I_{OUT}}{I_{REF}} = \frac{1}{1 + (2/\beta)} \tag{2.1}$$

This equation does not include the effects of the early voltage, but does include the effects of nonzero base current often referred to as errors due to finite β. The error due to finite β results because I_{REF} must supply base current to Q_1 and Q_2. The key problem with this dependency of I_{OUT} on β is that β may vary from due to process variations, resulting in an I_{OUT} that varies due to process variations. Were it not for this dependency, I_{REF} could be adjusted to compensate for this and set I_{OUT} to a desired value. For $\beta \gg 2$, the fractional error is $-2/\beta$ percent.

A second error in I_{OUT} occurs due to finite output resistance. Performing a small-signal analysis for the circuit for Figure 2.1, it can be shown that the expression for R_o for the simple current source is equal to r_{o2}.

If the mirror in Figure 2.1 is configured with N output transistors and thus has N output currents, the transfer function becomes

$$\frac{I_{OUT}}{I_{REF}} = \frac{1}{1 + ((N+1)/\beta)} \tag{2.2}$$

2.1.1.1.2 Simple Current Mirror with Beta Helper

The β sensitivity of the simple current mirror can be improved by adding a third transistor to supply base current to Q_1 and Q_2 shown in Figure 2.2. Here, the base current of Q_1 and Q_2 is supplied by the emitter of Q_3 which draws the necessary current from I_{REF}, but reduced by a factor of $\beta + 1$. An analysis shows

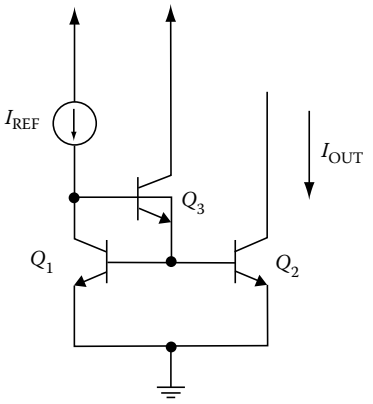


FIGURE 2.2 Simple current mirror with beta helper

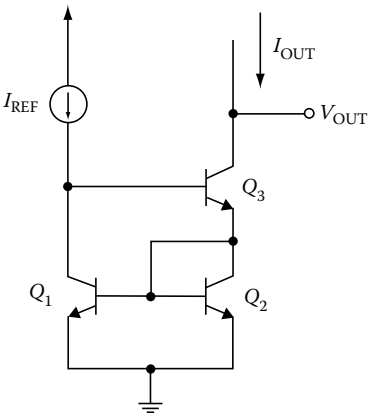


FIGURE 2.3 Wilson current mirror.

$$\frac{I_{OUT}}{I_{REF}} = \frac{1}{1 + (2/(\beta^2 + \beta))} \approx \frac{1}{1 + (2/\beta^2)} \tag{2.3}$$

This supports the argument that sensitivity to base current drain is reduced. It should be noted that the equation also depends on the betas of the transistors being matched as well. For $\beta \gg 2$, the fractional error is $-2/\beta^2$ percent. Thus, the fractional error is reduced by a factor of β. For this circuit $R_o = r_{o2}$, so there is no improvement in R_o.

2.1.1.1.3 Wilson Current Mirror

A current source that shows an improvement in R_o and has reduced β sensitivity is the Wilson current mirror shown in Figure 2.3 [1].

With this circuit, as with the simple current mirror with beta helper, the base current of Q_1 and Q_2 is supplied by emitter current of a third transistor, Q_3. A more rigorous analysis will show, neglecting the effects of the early voltage,

$$\frac{I_{OUT}}{I_{REF}} = \frac{1}{1 + (2/(\beta^2 + \beta))} \approx \frac{1}{1 + (2/\beta^2)} \tag{2.4}$$

For $\beta \gg 2$, the fractional error is $-2/\beta^2$. Thus, the fractional error is reduced by a factor of β.

The improvement in R_o is due to negative feedback present in the circuit due to the placement of Q_3. To see this, first consider the case for matched transistors and the effects of output voltage for all of the transistors, that an increase in V_{OUT} gives rise to an increase in output current. This, in turn, causes an increase in I_{C2}. Since Q_1 and Q_2 themselves form a simple current mirror, I_{C1} also increases which forces a decrease in I_{B3}, since I_{REF} is constant. This in turn reduces in I_{OUT}. A small-signal analysis to determine R_o shows

$$R_o \approx \frac{\beta}{2} r_o \tag{2.5}$$

R_o has been increased by a factor of $\beta/2$.

The Wilson current mirror can also be extended to N multiple outputs by placing additional transistors branches in parallel with Q_2 and Q_3. For this case, it can be shown that

$$\frac{I_{OUT}}{I_{REF}} = \frac{1}{N} \frac{1}{1 + (2/\beta^2)} \tag{2.6}$$

where N is the total number of output branches.

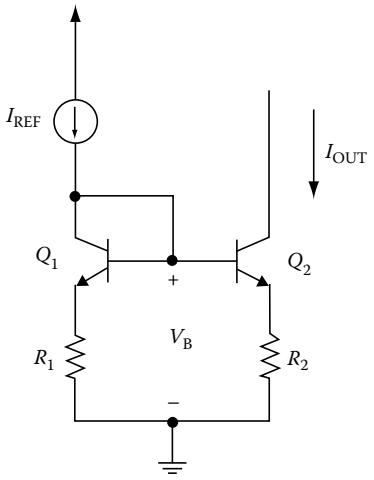


FIGURE 2.4 Simple current mirror with emitter degeneration.

2.1.1.1.4 Simple Current Mirror with Emitter Degeneration

Parameters and component values for fabricated devices usually exhibit deviation from some "nominal" value during the fabrication process. These variations may occur for across the die, from die-to-die, from wafer-to-wafer, or from lot-to-lot. An objective in designing circuits that are process insensitive is to minimize the effects of process variation. A good example of this is the simple current mirror with emitter degeneration shown in Figure 2.4. If I_{REF} and R_1 are such that the voltage drop across V_{BE1} is small in comparison, then the dominant voltage at V_B is approximately $I_{REF}R_1$. Likewise, if R_2 and I_{OUT} are such that the voltage drop across V_{BE2} can be neglected, then we have $I_{REF}R_1 \approx I_{OUT}R_2$ and

$$I_{OUT} = \frac{R_2}{R_1} I_{REF} \tag{2.7}$$

For the simple current mirror, neglecting β and considering the possibly of mismatched emitter areas, we have

$$I_{OUT} = \frac{A_2}{A_1} I_{REF} \tag{2.8}$$

Since resistors can be matched to ±0.1% and NPN matching for transistors can be as poor as ±1%, the current mirror with emitter degeneration is less susceptible to processing errors. The process insensitivity is made possible by having the relationship between I_{OUT} and I_{REF} dependent on resistor ratios.

2.1.1.1.5 Widlar Current Mirror

In bipolar integrated circuit design, it is sometimes desirable to create low currents levels, on the order of microamps, for example [2]. If either of the simple current mirrors or the Wilson current mirror is used, this has to be accomplished by creating a very small reference current and may require large values of resistors that may consume large amounts of area. The current mirror depicted in Figure 2.5 is capable of producing a small output current, from a nominal reference current and a reasonably sized resistor.

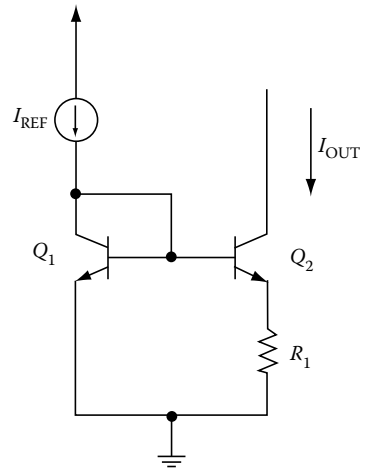


FIGURE 2.5 Widlar current mirror.

To analyze the circuit, recognize that I_{REF} is determines V_{BE1}. In the simple current mirror, all of V_{BE1} appears across V_{BE2}. In the Widlar current mirror, V_{BE1} is divided between the base–emitter junction of Q_2 and R_1, resulting in a smaller voltage for V_{BE2} than V_{BE1} and thus a smaller I_{OUT}. This suggests that with proper selection of R_1, the potential for generating very small currents exists. It can be shown, neglecting the effects of β, that

$$I_{OUT}R_1 = V_T \ln\left(\frac{I_{REF}}{I_{OUT}}\right) \tag{2.9}$$

where V_T is the thermal voltage.

For example, to create an I_{OUT} of 5 μA, from a reference of 1 mA, with $V_T = 26$ mV, we find $R_1 = 27$ kΩ. If the same 5 μA were desired from the simple current mirror, and we assume that the emitter areas of Q_1 and Q_2 are equal, we would need to generate a reference current of 5 μA. To do this, I_{REF} is replaced by a resistor, R_{REF}, tied to the positive supply voltage, V_{CC}, for example. Suppose for this example that V_{CC} is 5 V. Then, the voltage drop across R_{REF} is $V_{CC} - V_{BE}$. Taking $V_{BE} = 0.7$ V gives a value of $R_{REF} = (5 - 0.7)/5\mu = 860$ kΩ. This, of course, takes up much more chip real estate than for $R_1 = 27$ kΩ and is undesirable.

If the effects of β are included it is necessary to supply two base currents, though I_{B2} is less than I_{B1}. Equation 2.2 gives an upper bound for the error due to beta for the Widlar current mirror (the upper bound being the case where $I_{OUT} = I_{REF}$) and lower bound being $1/(1 + 1/\beta)$, supplying base current to only a single transistor. Thus, the errors due to base current drain are on the same order as that of the simple current mirror.

For the output resistance, R_1 provides negative feedback and thus increases the output resistance compared with that of the simple current mirror. It is identical to the increase in output resistance that results from emitter degeneration in a common-emitter amplifier. It can be verified that

$$R_o = (1 + g_m r_\pi R_1) r_o \tag{2.10}$$

2.1.1.1.6 Low-Bias Current Mirror

An alternative to the Widlar current source that also provides a low output current is the current source shown in Figure 2.6 [3,4]. Again, V_{BE1} is determined by I_{REF}. Applying Kirchhoff's voltage law,

$$V_{BE2} = V_{BE1} - I_{REF} R_1 \tag{2.11}$$

As with the Widlar current source voltage, the voltage V_{BE1} is divided between a resistor and the base emitter junction of Q_2, though in a less obvious manner. As a result, V_{BE2} must be smaller than V_{BE1} and effectively a fraction of I_{REF} is mirrored. A more exact equation that expresses the relationship between the output current and reference current can be derived from Equation 2.8, and this is

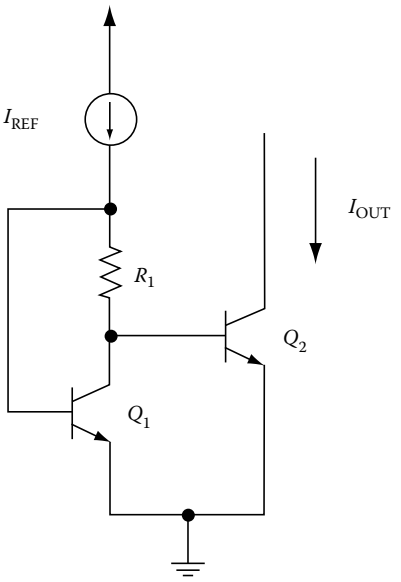


FIGURE 2.6 Current mirror for generating low-bias currents.

$$I_{OUT} = \frac{I_{REF}}{\exp(I_{REF} R_1 / V_T)} \tag{2.12}$$

or

$$R_1 = \frac{V_T \ln(I_{REF}/I_{OUT})}{I_{REF}} \tag{2.13}$$

In fact, this current source is capable of supplying even lower currents than the Widlar current source for a given I_{REF} and a lower bound for R_1. Consider the same example as was given for the Widlar current source. With $I_{REF} = 1$ mA, $I_{OUT} = 5$ μA, and $V_T = 26$ mV, we find that $R_1 = 137.75$ Ω. This is a substantial decrease in resistance from the Widlar current mirror example.

The price paid for this improvement is a reduced output resistance when compared to the Widlar current

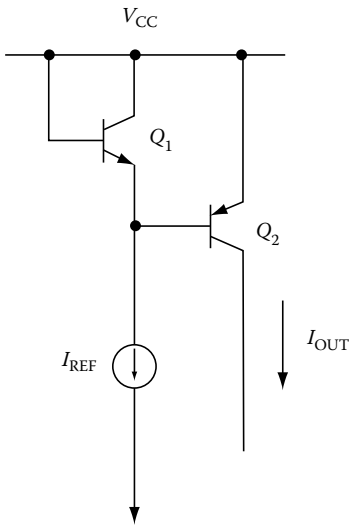


FIGURE 2.7 Current mirror for complementary bipolar design.

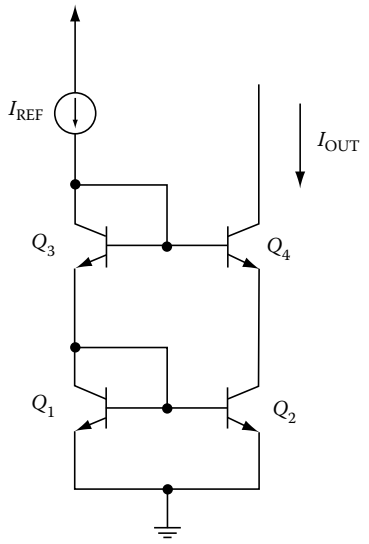


FIGURE 2.8 Cascode current mirror.

source. Negative feedback is not present in this topology and there are no improvements in output resistance when compared to the simple current mirror. For this current mirror, R_o is simply r_{o2}.

The basic current mirrors can be extended to complementary bipolar technology (CBT) as well. Figure 2.7 shows an example of a current mirror that can be found in a CBT. A problem that arises in the current mirrors used in bipolar technology results in the fact that PNP and NPN transistors have a different Gummel number [5]. This results in different base–emitter voltages (magnitudes) for equal collector currents. The effects of this can be deduced from Figure 2.7, where now ΔV_{BE} errors must be considered. Thus, care must be taken when biasing complementary bipolar designs. This issue of balancing and matching is a fundamental problem in this technology [5].

2.1.1.1.7 Cascode Current Mirror

The cascode current mirror is depicted in Figure 2.8. It derives its main advantage in an increased output resistance due to emitter degeneration as does the Widlar current source. In this case, r_{o2} replaces R_1 in the Widlar current source to provide the emitter degeneration. Though in theory the values of r_{o2} and R_1 may be on the same order, a large value for r_{o2} can be achieved using a transistor that takes up much less area than that of a resistor of the same value.

The complete expression for R_o is complicated. However, for the case where $I_{OUT} \approx I_{REF}$, the transistors are matched and $g_m r_o \gg 1$, $\beta \gg 2$, and $r_o \gg r_\pi$ (for any combination of transistors) the expression for the output resistance reduces to

$$R_o = \frac{\beta}{2} r_{o2} \tag{2.14}$$

2.1.1.1.8 V_{BE} Referenced Current Mirror

For many current mirrors, I_{REF} is determined by a resistor tied to the positive power supply. In the Widlar current mirror example, $I_{REF} = (V_{CC} - V_{BE})/R_{REF}$. The reference current is directly proportional to the supply voltage. In many situations, this is undesirable. One alternative is to replace V_{CC} by one of

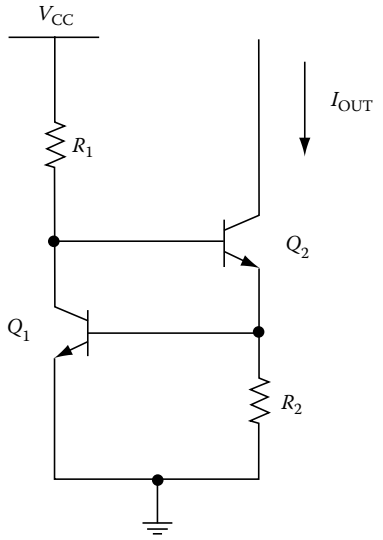


FIGURE 2.9 Current source that has reduced supply voltage dependency.

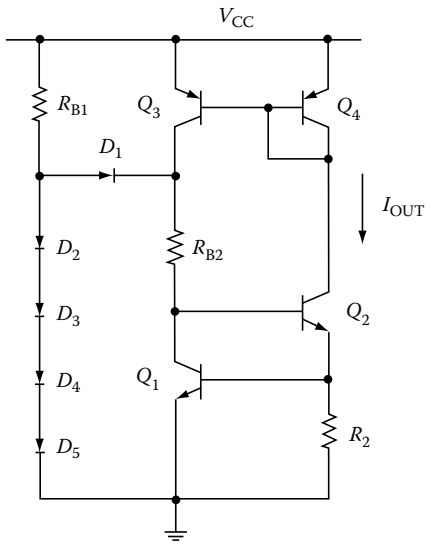


FIGURE 2.10 Self-biased V_{BE} referenced current source.

the many available circuits which provide a voltage reference that is independent of supply voltage. Another alternative is the V_{BE} referenced current mirror (Figure 2.9).

The basic principle of this current mirror is to establish a base–emitter voltage and to convert this voltage to a current using a resistor. Referring to Figure 2.9, the V_{BE} (V_{BE1}) drop is first established with V_{CC} and R_1. The voltage V_{BE1} and R_2 determines I_{OUT}. Neglecting finite betas for the transistors, this current is approximately equal to V_{BE1}/R_2. Since V_{BE1} is fairly constant, I_{OUT} is fairly constant. A complete derivation yields

$$I_{OUT} = \frac{V_T}{R_2} \ln\left(\frac{V_{CC}}{I_{S1}R_1}\right) \tag{2.15}$$

where I_{S1} is the saturation current of Q_1. Equation 2.13 shows the output current has a logarithmic variation with respect to V_{CC}, an improvement over the linear relationship found in other current mirrors.

2.1.1.1.9 Self-Biased V_{BE} Referenced Current Source

A self-biased V_{BE} referenced current source is depicted in Figure 2.10. Q_1, Q_2, Q_3, Q_4, and R_2 form the core of the current source. Q_1 and Q_2 form a V_{BE} referenced current mirror and the pair Q_3 and Q_4 form a simple current mirror. If current exists, then $I_{C1} = I_{C2}$, due to Q_3 and Q_4, and one valid solution to is $I_{C2} \approx V_{BE}/R_2$, independent of V_{CC}. However, a second valid solution is $I_{C2} = 0$ A (or practically, a value for I_{C2} on the order of leakage currents). For this reason, a start-up circuit is added to the circuit. D_{1-5}, R_{B1}, and R_{B2} form the start-up circuitry. If $I_{C1} = I_{C2} = 0$, the voltage at the cathode of D_1 is 0 V and D_1 turns on, injecting current into the core of the current source. Positive feedback in the circuit forces the current toward the condition where $I_{C2} \approx V_{BE}/R_2$. At some point, the voltage at the cathode of D_1 raises to a level that shuts D_1 off, thereby "disconnecting" the start-up circuitry from the current source core.

Once I_{C1} and I_{C2} have been established as a reference current for a larger circuit, the current can be mirrored to other parts of the circuit by placing transistors in parallel with Q_1 and Q_4, as shown with Q_5 and Q_6.

2.1.1.1.10 Self-Biased V_T Referenced Current Source

A second type of self-biased current source called self-biased V_T referenced current source is shown in Figure 2.11. In this circuit, Q_3 and Q_4 are current mirrors and force the condition that if current exists, then $I_{C1} = I_{C2}$. In general, the area of Q_2 (A_2) is set to be "N" times the area of Q_1 (A_1) and thus $I_{S2} = NI_{S1}$, where n is some integer. In practice, this is achieved by placing n transistors in parallel. This causes a difference in base–emitter voltages, $\Delta V_{BE} = V_{BE1} - V_{BE2} = nV_T \ln(N)$, and this difference is dropped across R_1. Thus,

$$I_{C1} = I_{C2} = \frac{nV_T \ln(N)}{R_1} \tag{2.16}$$

I_{C1} and I_{C2} can in turn be mirrored by placing transistors in parallel with Q_1 and Q_4 to form current mirrors.

2.1.1.2 Voltage References

Also fundamental to biasing in BJT circuits is the voltage reference. As with the current sources, we may define an ideal voltage source in order to have an adequate way of evaluating voltage references. An ideal voltage reference produces a voltage that

1. Does not vary with loading (zero output resistance)
2. Is insensitive to process variations
3. Is insensitive to power supply variations
4. Is insensitive to temperature

The simplest voltage reference is the zener diode reference, but it is well known that a zener diode exhibits temperature dependence and has a fairly high output resistance. Some of the modifications to a simple zener diode include placing a diode series with a zener, strings of diodes, and the common-collector stage. Figure 2.12 gives an example. Here it is assumed, but not always true, that V_{DZ} has a positive coefficient and V_{D1} has a negative temperature coefficient and provides some cancellation of the coefficients. Thus, this can produce a reference voltage that is insensitive to temperature. There are literally hundreds of deviations based on this fundamental principle.

A circuit that produces a voltage that is an arbitrary multiple of V_{BE} is the V_{BE} multiplier circuit shown in Figure 2.13. The circuit works on the principle that a current equal to V_{BE}/R_2 is generated and, neglecting base current, flows through R_1. Thus, the voltage across R_1 is $(V_{BE}/R_2)R_1$ and the total voltage can be written as

$$V_{REF} = \left(1 + \frac{R_2}{R_1}\right) V_{BE} \tag{2.17}$$

Many applications for biasing circuits demand that their performance remain constant through a wide range of temperatures. Thus, many circuits for temperature insensitive biasing have emerged. Ideally, a temperature insensitive output, voltage or current, would depend on a temperature insensitive element. Since all semiconductor components exhibit variation with temperature, most of the schemes for temperature independence involve some form of cancellation technique or compensation [6]. Instead of eliminating all of the sensitivity, the design techniques strive to minimize the errors. For example, a common solution is to place devices with positive temperature coefficients in series with devices with negative temperature coefficients, scaling some of these coefficients if necessary, to provide nearly zero sensitivity to temperature for an output is taken across the devices.

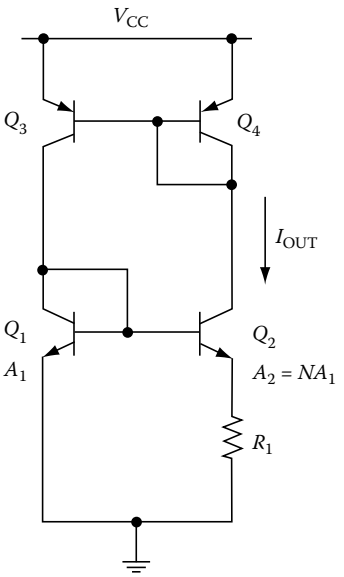


FIGURE 2.11 Self-biased VT referenced current source.

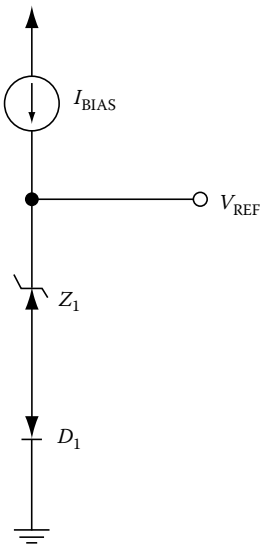


FIGURE 2.12 Zener-biased voltage reference.

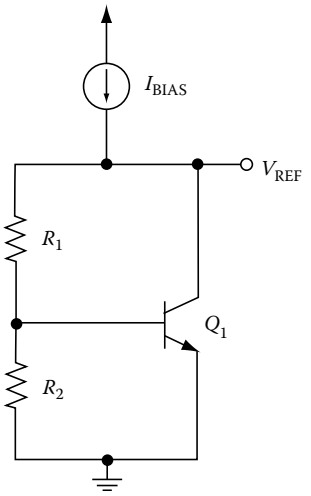


FIGURE 2.13 V_{BE} multiplier circuit.

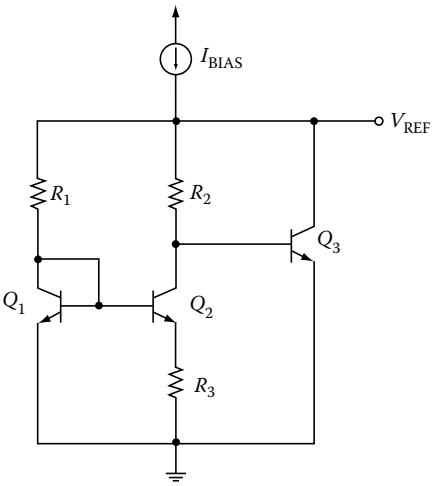


FIGURE 2.14 Simple bandgap reference.

A simple band-gap reference is depicted in Figure 2.14. Band-gap circuits also operate on a principle of cancellation of temperature coefficients. Generally, a voltage is developed which is a scaled value of V_T. This scaled value has a well-defined temperature coefficient which is the scaling constant times the temperature coefficient of V_T, which is positive. This voltage is added to a base–emitter voltage, which has a negative temperature coefficient. The scaling factor is chosen so that the sum of the temperature coefficients is zero. The output is then taken across the two voltages to produce a voltage with a temperature coefficient of approximately zero. Generally, the output voltage has the form

$$V_{REF} = V_{BE} + KV_T \tag{2.18}$$

If the temperature coefficient of V_{BE} is taken to be -2 mV/°C and the temperature coefficient of V_T is taken to be $k/q \approx +0.085$ mV/°C, this results in a value for K of about 23.52. This gives a value for V_{OUT} of about 0.7 V + 23.52(25.9 mV) = 1.3 V, close to the band-gap voltage of silicon, and gives rise to the nomenclature of band-gap references.

In the circuit in Figure 2.14, we assume that Q_1 and Q_2 operate at different current densities. This is done either by operating Q_1 and Q_2 at different collector current levels, with matched emitter areas, or by operating them at the same collector currents with emitter areas being mismatched. A voltage ΔV_{BE} is developed across R_3 and then current through R_3 and R_2, negecting the effects of β, is $\Delta V_{BE}/R_3$. This gives $V_{REF} = \Delta V_{BE} R_2/R_3 + V_{BE3}$. Since $\Delta V_{BE} = V_T \ln(J_1/J_2)$, this gives the constant K as being $R_2/R_3 \ln(J_1/J_2)$.

An improved band-gap reference is depicted in Figure 2.15. This reference forms a basic building block for several commercial voltage references. In this circuit, I_{C1} and I_{C2} are forced to be equal by the high-gain

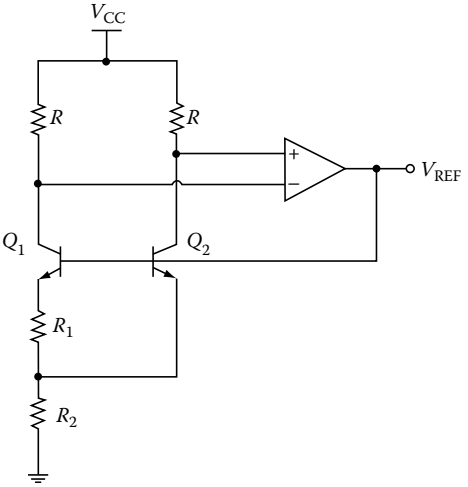


FIGURE 2.15 Improved bandgap reference.

amplifier operating with negative feedback. The current densities of Q_1 and Q_2 are made unequal by sizing the emitter areas of Q_1 and Q_2 differently. In this case, $A_1 = NA_2$. This means $\Delta V_{BE} = nV_T\ln(I_{S1}/I_{S2}) = V_T\ln(N)$. The voltage drop across R_2 is $\Delta V_{BE}R_2/R_1$. Finally, $V_{REF} = V_{BE} + V_T\ln(n)R_2/R_1$. Thus, in this case, $K = R_2/R_1\ln(n)$.

There is a wealth of information available in the literature on current sources and voltage references. Further depth into bias circuits behavior is provided in the accompanying references for this section.

References

1. G. R. Wilson, A monolithic junction FET-NPN operational amplifier. *IEEE J. Solid-State Circuits*, SC-3, 341–348, Dec. 1968.
2. R. J. Widlar, Some circuit design techniques for linear integrated circuits. *IEEE Trans. Circuit Theory*, CT-12, 586–590, Dec. 1965.
3. C. Kwok, Low-voltage peaking complementary current generator. *IEEE J. Solid-State Circuits*, SC-20, 816–818, Jun. 1985.
4. P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed., New York: Wiley, 2001, pp. 253–334.
5. C. Toumazou, F. J. Lidgey, and D. G. Haigh (ed.), *Analogue IC Design: The Current-Mode Approach*, London: Peter Peregrinus, 1990, Chapters 6 and 16.
6. A. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley, 1984, Chapter 4.

2.2 Canonic Cells of Linear Bipolar Technology

John Choma, Jr. and J. Trujillo

2.2.1 Introduction

The circuit configurations of linear signal processors realized in bipolar technology are as diverse as the system operating requirements that these circuits are designed to satisfy. Despite topological diversity, most practical open-loop linear bipolar circuits are derived from interconnections of surprisingly few basic subcircuits. These subcircuits include the diode-connected bipolar junction transistor (BJT), the common-emitter amplifier, the common-base amplifier, the common-emitter–common-base cascode, the emitter follower, the Darlington connection, and the balanced differential pair. Because these open-loop subcircuits underpin linear bipolar circuit technology, they are rightfully termed the ''canonic cells'' of linear bipolar circuit design.

By examining the low-frequency performance characteristics of the canonic cells of linear bipolar technology, this section achieves two objectives. First, the forward gain, the driving point input resistance, and the driving point output resistance are cataloged for each canonic circuit. This information produces Thévenin and Norton I/O port equivalent circuits that expedite the analysis and design of multistage electronics. Second, the forthcoming work establishes a basis for prudent circuit design in that all analytical results are studied by highlighting the attributes and uncovering the limitations of each cell. The understanding that resultantly accrues paves the way toward systematic design procedures that yield optimal circuit architectures capable of circumventing observed subcircuit shortcomings.

2.2.2 Small-Signal Model

The fundamental tool exploited in the analyses that follow is the low-frequency small-signal equivalent circuit of a BJT shown in Figure 2.16a. This equivalent circuit, which applies to NPN and PNP discrete component and monolithic transistors, is derived from the low-frequency, large-signal NPN BJT model

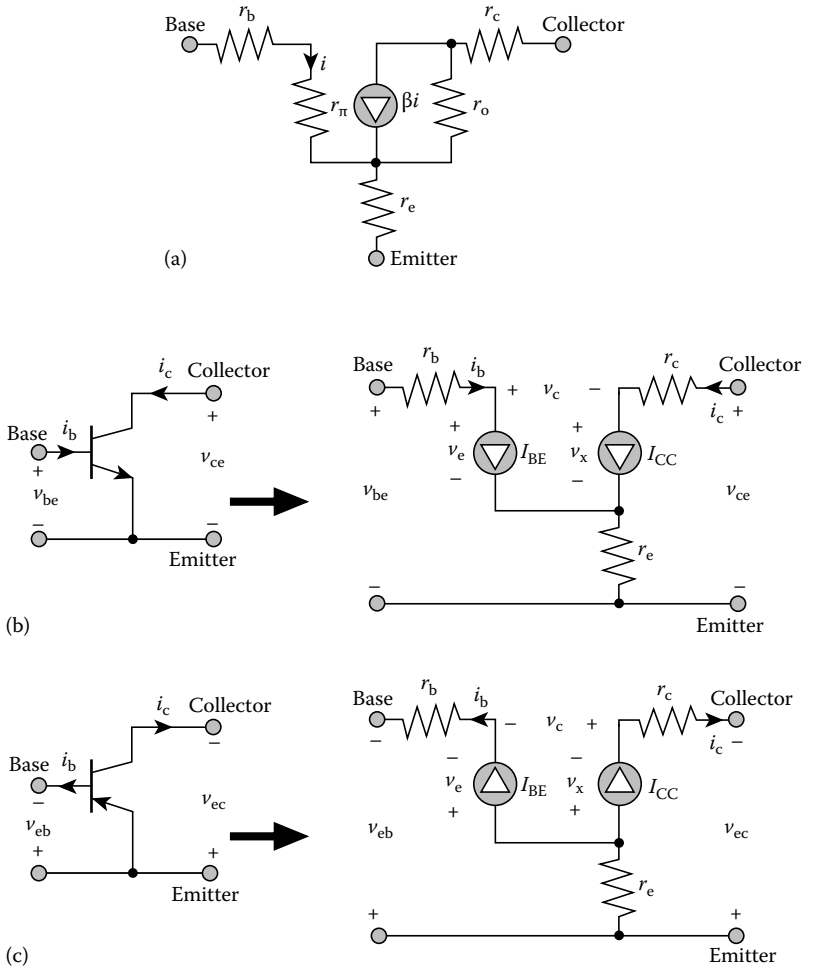


FIGURE 2.16 (a) Low-frequency, small-signal model of a BJT. (b) Low-frequency, large-signal model of an NPN BJT. (c) Low-frequency, large-signal model of a PNP BJT.

offered in Figure 2.16b [1,2]. As is depicted in Figure 2.16c, the PNP large-signal transistor model is topologically identical to its NPN counterpart. The only difference between the two models is a reversal in the direction of all controlled current sources and branch currents and a reversal in polarity of all assigned branch and port voltages.

The large-signal models in Figure 2.16b and c are simplified to reflect transistor biasing that assures nominally linear device operation for all values of applied input signal voltages. A necessary condition for linear operation is that the internal emitter–base junction voltage v_e be at least as large as the threshold voltage, say v_γ, of the junction for all time, that is,

$$v_e(t) \geq v_\gamma \quad \text{for all time } t \tag{2.19}$$

For silicon transistors, v_γ is typically in the neighborhood of 700–750 mV. A second condition underlying transistor operation in its linear regime is that the internal base–collector junction voltage v_c is never positive, that is,

$$v_c(t) \leq 0 \quad \text{for all time } t \tag{2.20}$$

In the models of Figure 2.16b and c, r_b represents the "effective base resistance" of a BJT, r_c is its net "internal collector resistance," and r_e is the net "internal emitter resistance." All three resistances, and particularly r_b, decrease monotonically with increasing quiescent base and collector currents, I_{BQ} and I_{CQ}, respectively [3,4]. The collector resistance also decreases with increase in the intrinsic collector–emitter voltage v_x. Large base, collector, and emitter resistances conduce reduced circuit gain, diminished gain-bandwidth product, and increased electrical noise. In view of these observations and in the interest of formulating a mathematically tractable analysis that produces conservative estimates of bipolar circuit performance, these resistances are usually interpreted as constants equal to their respective low-current, low-voltage values.

In a monolithic fabrication process, unacceptably large internal device resistances can be reduced by exploiting the fact that r_b, r_c, and r_e are inversely proportional to the emitter–base junction injection area. This area is a designable parameter chosen to ensure that the transistor in question conducts the proper density of collector current. Unfortunately, the engineering price potentially paid for a reduction of device resistances through increase in junction area is circuit response speed, since the capacitances associated with transistor junctions are directly proportional to device injection area.

The current I_{BE} in Figure 2.16b and c is given approximately by

$$I_{BE} = \frac{A_E J_S}{\beta_F} e^{v_e/n_f V_T} \tag{2.21}$$

where

A_E is the aforementioned emitter–base junction area
J_S is the density of transistor saturation current
β_F is the forward short-circuit current transfer ratio
n_f is the injection coefficient of the emitter–base junction
v_e is the internal junction voltage serving to forward bias the emitter–base junction

and

$$V_T = \frac{kT_j}{q} \tag{2.22}$$

is the Boltzmann voltage. In the last expression, k is Boltzmann's constant [1.38 (10^{-23}) J/K], T_j is the absolute temperature of the emitter–base junction, and q is the magnitude of electron charge [1.6 (10^{-19}) C].

The current I_{CC} is derived from [5]

$$I_{CC} = A_E J_S e^{v_e/n_f V_T} \left(1 - \frac{I_{CC}}{I_{KF}}\right)\left(1 + \frac{v_x}{V_{AF}}\right) \tag{2.23}$$

where

I_{KF}, which is proportional to A_E, is the "forward knee current" of the transistor [6]
V_{AF}, which is independent of A_E, is the "forward Early voltage" [7]

Note that the base current i_b is the current I_{BE}, while the collector current i_c is I_{CC}. Thus, the "static common-emitter current" gain (often referred to as the "DC beta"), h_{FE}, of a BJT is

$$h_{FE} = \frac{i_c}{i_b} = \frac{I_{CC}}{I_{BE}} = \beta_F \left(1 - \frac{i_c}{I_{KF}}\right)\left(1 + \frac{v_x}{V_{AF}}\right) \tag{2.24}$$

which is functionally dependent on both the collector current and the intrinsic collector–emitter voltage.

Unlike the base, collector, and emitter resistances, the resistance r_π in the small-signal model of Figure 2.16a is not an ohmic branch element. It is a mathematical resistance that arises from the Taylor series expansion of the current I_{BE} about the "quiescent-operating point," or "Q-point" of the transistor. In particular, r_π, which is known as the "emitter–base junction diffusion resistance," derives from

$$\frac{1}{r_\pi} = \left.\frac{\partial I_{BE}}{\partial v_e}\right|_Q \tag{2.25}$$

where it is understood that the indicated derivative is evaluated at the Q-point of the device. This Q-point is unambiguously defined by the "zero signal, or static," values of the base current I_{BQ}, the collector current I_{CQ}, and the internal collector–emitter voltage V_{XQ}. Using Equations 2.21 and 2.24, and the fact that $i_b + I_{BE}$,

$$r_\pi = \frac{h_{FE} n_f V_T}{I_{CQ}} \tag{2.26}$$

The inverse dependence of r_π on quiescent collector current renders r_π large at low collector current biases.

Similarly, r_o, the "forward Early resistance," derives from

$$\frac{1}{r_o} = \left.\frac{\partial I_{CC}}{\partial v_x}\right|_Q \tag{2.27}$$

It can be shown that

$$r_o = \frac{V_{XQ} + V_{AF}}{I_{CQ}\left(1 - \frac{I_{CQ}}{I_{KF}}\right)} \tag{2.28}$$

Like r_π, r_o is also large for low-level biasing.

Finally, the parameter β, which is the "low-frequency small-signal common-emitter short-circuit current gain" (often more simply referred to as the "AC beta") of the transistor, is

$$\beta = g_m r_\pi \tag{2.29}$$

where g_m, the "forward transconductance" of a BJT is

$$g_m = \left.\frac{\partial I_{CC}}{\partial v_e}\right|_Q \tag{2.30}$$

From Equations 2.23, 2.24, 2.26, and 2.29,

$$\beta = h_{FE}\left(1 - \frac{I_{CQ}}{I_{KF}}\right) \tag{2.31}$$

To the extent that $I_{CQ} \ll I_{KF}$, β is nominally independent of both Q-point collector current and emitter–base junction injection area.

2.2.3 Single-Input–Single-Output Canonic Cells

2.2.3.1 Diode-Connected Transistor

The simplest of the single-input–single-output, or "single-ended" canonic cells for linear bipolar circuits is the "diode-connected transistor" offered in Figure 2.17a. This transistor connection emulates the volt–ampere characteristics of a conventional PN junction diode. It can therefore be used in rectifier, voltage regulator, dc level shifting, and other applications that exploit conventional diodes. But unlike a conventional PN junction diode, the diode-connected transistor proves especially useful in current mirror biasing schemes. These and other similar circuits require that the base–emitter terminal voltage, v, of the diode track the base–emitter terminal voltage of a second, presumably identical transistor, over wide variations in junction-operating temperatures.

If the voltages dropped across the internal base, collector, and emitter resistances are small, the intrinsic emitter–base junction voltage, v_e, is approximately the indicated terminal voltage, v. Moreover, the intrinsic base–collector junction voltage, v_c, is essentially zero. It follows that for $v > v_\gamma$, the transistor in the diode connection operates in its linear regime.

In the subject diagram, the terminal voltage v is depicted as a superposition of a static voltage, V_Q, and a signal component, v_s. The resultant diode current, i, is a superposition of a quiescent current, I_Q, and a signal current, i_s. The quiescent components of diode voltage and current arise from static power supplied to the diode circuit to ensure that the diode-connected device operates in its linear regime. On the other hand, the signal components are established by a time-varying signal applied to the input port of the circuit in which the diode-connected transistor is embedded. In order to achieve reasonably linear processing of the applied input signal, the value of V_Q must be such as to ensure that $v = V_Q + v_s > v_\gamma$ for all values of the time-varying signal voltage v_s. Since v_s can be positive or negative at any instant of time, the requirement $V_Q + v_s > v_\gamma$ mandates that the amplitude of v_s be sufficiently small.

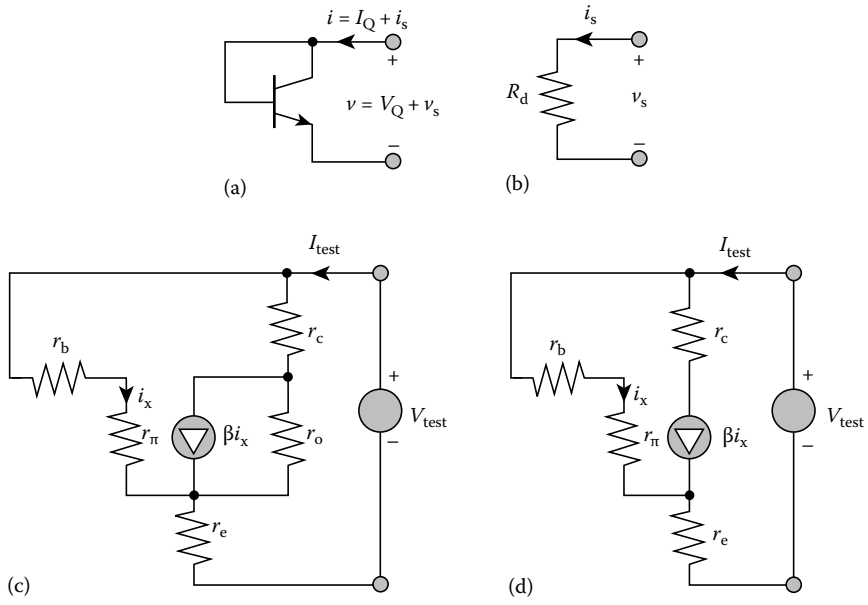


FIGURE 2.17 (a) Diode-connected BJT. (b) The small-signal equivalent circuit of the diode in (a). (c) Low-frequency, small-signal model of the diode-connected transistor in (a). The ratio V_{test}/I_{test} is the small-signal resistance R_d presented at the terminals of the diode-connected transistor. (d) The model in (c) approximated for the case of very large Early resistance.

The immediate impact of the small-signal condition corresponding to the linearity requirement $V_Q + v_s > v_\gamma$ is that the small-signal volt–ampere characteristics of the diode are linear. And since the diode is a two-terminal element, these characteristics can be modeled at low-signal frequencies by a simple resistance, say R_d, as suggested by the single-element macromodel offered in Figure 2.17b. The resistance in the latter figure can be determined by using the small-signal transistor model of Figure 2.16a to construct the small-signal equivalent circuit of the diode-connected transistor shown in Figure 2.17c. In this figure, the ratio of the test voltage, V_{test}, to the test current, I_{test}, is the desired resistance, R_d. A straightforward KVL analysis confirms that

$$R_d = \frac{V_{test}}{I_{test}} = r_e + \frac{(r_o + r_c) \| (r_b + r_\pi)}{1 + \frac{\beta r_o}{r_o + r_c + r_b + r_\pi}} \tag{2.32}$$

Typically, r_o is 25 kΩ or larger, r_c is smaller than 75 Ω, r_b is of the order of 100 Ω, and r_π is in the range of 1 kΩ for a minimal geometry device. It follows that $r_o \gg (r_c + r_b + r_\pi)$, and R_d can be approximated as

$$R_d \approx r_e + \frac{r_b + r_\pi}{\beta + 1} \tag{2.33}$$

Note that this terminal resistance is of the order of the low tens of ohms. For example, if $r_b = 100$ Ω, $r_\pi = 1.2$ kΩ, $r_e = 1$ Ω, and $\beta = 100$, $R_d = 13.9$ Ω. It is instructive to note that the approximation, $r_o \gg (r_c + r_b + r_\pi)$, collapses the model in Figure 2.17c to the structure in Figure 2.17d, from which Equation 2.33 follows immediately.

A variation of the diode scheme is the so-called V_{BE} "multiplier" depicted in Figure 2.18a. This circuit finds extensive use in regulator and level shifting applications that require either a series interconnection of more than one diode or a circuit branch voltage drop whose requisite value is a nonintegral multiple of the base–emitter terminal voltage of a single diode.

The circuit under consideration establishes a static terminal voltage, V_Q, whose value is a designable multiple of the static base–emitter terminal voltage, V_{BEQ}. To confirm this contention, observe that for static operating conditions, V_Q is

$$V_Q = R_Y(I_Q - I_{CQ}) + V_{BEQ} \tag{2.34}$$

where the voltage component V_{BEQ} of the net base–emitter terminal voltage v_{be} is

$$V_{BEQ} = R_X \left[I_Q - \left(\frac{h_{FE} + 1}{h_{FE}} \right) I_{CQ} \right] \tag{2.35}$$

The current, I_{CQ}, is the static component of the net collector current, i_c, and h_{FE} is the collector current to base current transfer ratio defined by Equation 2.24. An elimination of I_{CQ} from the foregoing two expressions leads to

$$V_Q = \left(1 + \frac{\alpha_{FE} R_Y}{R_X} \right) V_{BEQ} + \left(\frac{R_Y}{h_{FE} + 1} \right) I_Q \tag{2.36}$$

where

$$\alpha_{FE} = \frac{h_{FE}}{h_{FE} + 1} \tag{2.37}$$

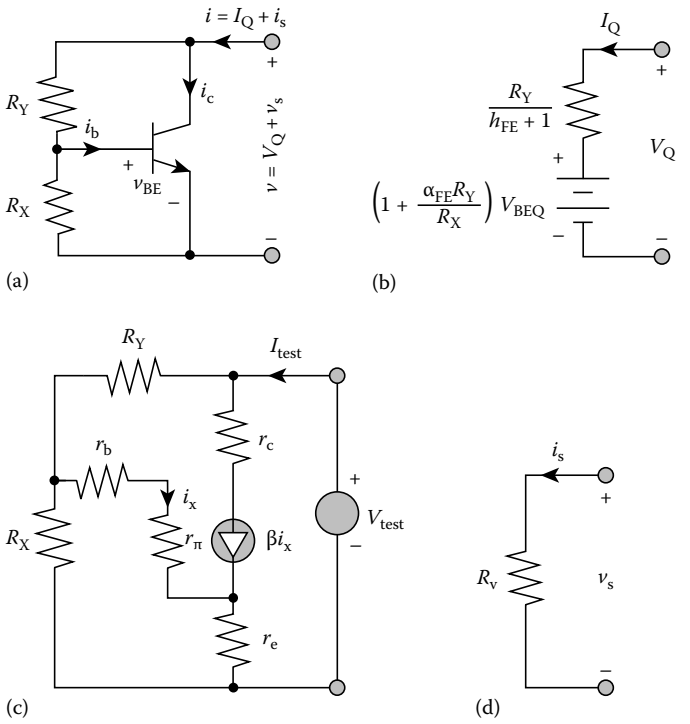


FIGURE 2.18 (a) Schematic diagram of V_{BE} multiplier. (b) DC macromodel of the multiplier in (a). (c) Low-frequency small-signal equivalent circuit of the Y_{BE} multiplier. (d) The small-signal equivalent resistance at the terminals of the V_{BE} multiplier.

is known as the ''static common-base current gain'' (often referred to as the ''DC alpha'') of a BJT. Equation 2.36 suggests that the static electrical behavior of the V_{BE} multiplier approximates a battery, whose voltage is controllable by the resistive ratio R_Y/R_X. The internal resistance of this effective battery is inversely dependent on $(h_{FE} + 1)$, and is therefore small. The macromodel in Figure 2.18b reflects the foregoing electrical interpretation. Note the for $R_Y = 0$ and R_X infinitely large, the circuit in Figure 2.18a collapses to the diode-connected transistor of Figure 2.17a, and V_Q understandably reduces to V_{BEQ}, the quiescent base–emitter terminal voltage of a diode-connected transistor.

For $V_Q + v_s > v_\gamma$, the transistor in the V_{BE} multiplier operates linearly. Accordingly, the pertinent small-signal terminal characteristics emulate a resistance, say R_V, which can be determined by applying the model of Figure 2.16a to the circuit in Figure 2.18a. The resultant equivalent circuit, simplified to reflect the realistic assumption of large r_o, is shown in Figure 2.18c while Figure 2.18d postulates the small-signal macromodel. An analysis of the circuit in Figure 2.18c reveals that

$$R_v \approx R_X \| R_d + R_Y \left[1 - \frac{\alpha R_X}{R_X + R_d}\right] \tag{2.38}$$

where

$$\alpha = \frac{\beta}{\beta + 1} \tag{2.39}$$

is the "low-frequency, small-signal, common-base, short-circuit current gain" (more simply referred to as the "ac alpha") of the transistor, and R_d is the resistance given by Equation 2.32. For $R_Y = 0$, $R_X = \infty$ reduces R_v to the expected result, $R_v \approx R_d$. Note further that for $\beta \gg 1$ (which makes $\alpha \approx 1$) and $R_d \gg R_X$, R_v is essentially the small-signal resistance presented at the terminals of a diode-connected transistor.

2.2.3.2 Common-Emitter Amplifier

The most commonly used single-ended canonic gain cell is the "common-emitter amplifier," whose NPN and PNP AC schematic diagrams are shown in Figure 2.19a and b, respectively. The AC schematic diagram delineates only the signal paths of a circuit. Thus, the biasing subcircuits required for linear operation of the transistors are not shown, thereby affording topological and analytical simplification. This simplification is accomplished without loss of engineering generality, for the results produced by an analysis of the AC schematic diagram reveal all salient performance traits of the common-emitter configuration.

The common-emitter amplifier is distinguished by the facts that signal is applied to the base of the transistor, and the resultant response is extracted as either the voltage, V_{OS}, or the current, I_{OS}, at the collector port. The effective load resistance terminating the collector to ground is indicated as R_{LT}, while the signal source is represented as a traditional Thévenin equivalent circuit. Alternatively, a Norton representation of the input source can be used, with the understanding that the Norton equivalent signal current, say I_{ST}, is simply the ratio of the Thévenin signal voltage, V_{ST}, to the Thévenin source resistance, R_{ST}.

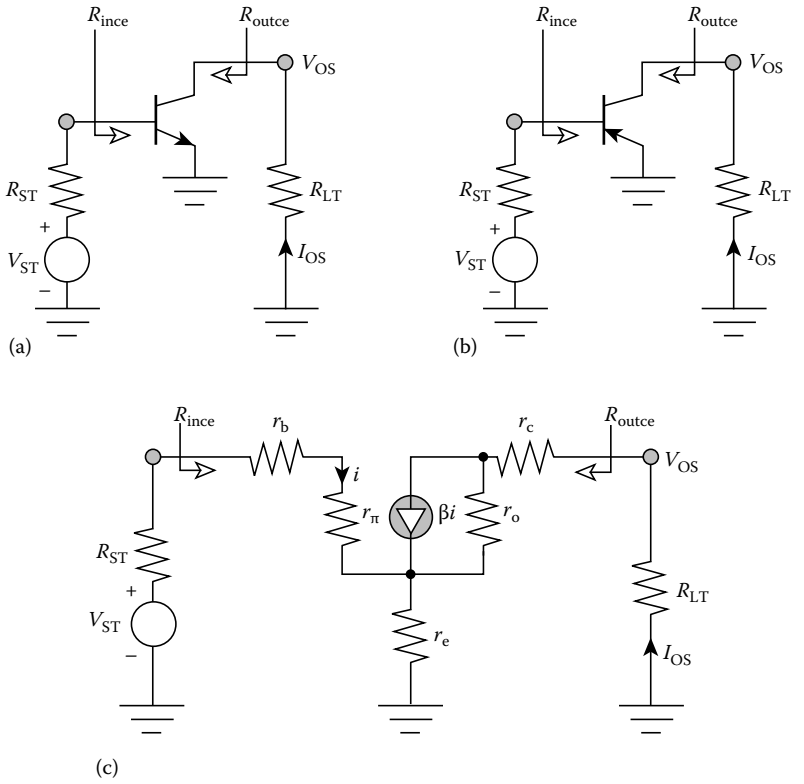


FIGURE 2.19 (a) AC schematic diagram of an NPN common-emitter amplifier. (b) AC schematic diagram of a PNP common-emitter amplifier. (c) Small-signal, low-frequency equivalent circuit of the common-emitter amplifier.

The common-emitter amplifier is capable of large magnitudes of voltage and current gains, moderately large input resistance, and very large driving point output resistance. An analytical confirmation of these contentions begins by drawing the small-signal equivalent circuit of the amplifier. This structure is given in Figure 2.19c and is valid for either the NPN or the PNP versions of the amplifier. An analysis of the small-signal model yields a voltage gain,

$A_{vce} = V_{OS}/V_{ST}$, of

$$A_{vce} = -\left\{\frac{(\beta - (r_e/r_o))(r_o/(r_o + r_c + r_e + R_{LT}))R_{LT}}{R_{ST} + r_b + r_\pi + ((\beta r_o/(r_o + r_c + R_{LT})) + 1)[r_e \| (r_o + r_c + R_{LT})]}\right\} \tag{2.40}$$

This relationship can be simplified by exploiting the fact that the internal resistance r_e of a transistor is small. Thus, $\beta \gg r_e/r_o$ and $r_e \gg (r_o + r_c + R_{LT})$, thereby implying

$$A_{vce} \approx -\frac{\beta_{eff} R_{LT}}{R_{ST} + r_b + r_\pi + (\beta_{eff} + 1)r_e} \tag{2.41}$$

where

$$\beta_{eff} \triangleq \beta\left[\frac{r_o}{r_o + r_c + R_{LT}}\right] \tag{2.42}$$

is an attenuated version of the AC beta for the utilized transistor. This effective beta approximates β itself, since $r_o \gg (r_c + R_{LT})$ is typical.

In concert with earlier arguments, Equation 2.41 confirms a diminished magnitude of gain for large internal device resistances. Note also that phase inversion, as inferred by the negative sign in either Equation 2.40 or 2.41 prevails between the Thévenin source voltage, V_{ST}, and the voltage signal response, V_{OS}. Finally, observe that large magnitudes of voltage gain are possible in the common-emitter orientation when β_{eff} is sufficiently large.

The driving point input resistance, R_{ince}, of the common-emitter amplifier can be determined as the ratio V_x/I_x for the test structure depicted in Figure 2.20a. It is easily shown that

$$R_{ince} = r_b + r_\pi + (\beta_{eff} + 1)[r_e \| (r_o + r_c + R_{LT})] \tag{2.43}$$

Since $r_e \ll (r_o + r_c + R_{LT})$, Equation 2.43 collapses to

$$R_{ince} \approx r_b + r_\pi + (\beta_{eff} + 1)r_e \tag{2.44}$$

Similarly, the driving point output resistance, R_{outce}, is derived as the V_x/I_x ratio of the equivalent circuit offered in Figure 2.20b. In particular,

$$R_{outce} = r_c + r_e \| (r_\pi + r_b + R_{ST}) + \left(\frac{\beta r_e}{r_e + r_\pi + r_b + R_{ST}} + 1\right) r_o \tag{2.45}$$

Since the model resistance r_π varies inversely with collector bias current, R_{ince} is moderately large when the common-emitter transistor is biased at low currents. On the other hand, R_{outce} is very large since Equation 2.45 confirms $R_{outce} > r_o$.

When the foregoing results are simplified to reflect the practical special case of a very large forward Early resistance, r_o, the cumbersome small-signal equivalent circuit of Figure 2.19c reduces to a ''small-signal macromodel'' useful for design-oriented circuit analysis of multistage amplifiers. To this end, note that a large r_o produces a driving point common-emitter input resistance that is independent of the

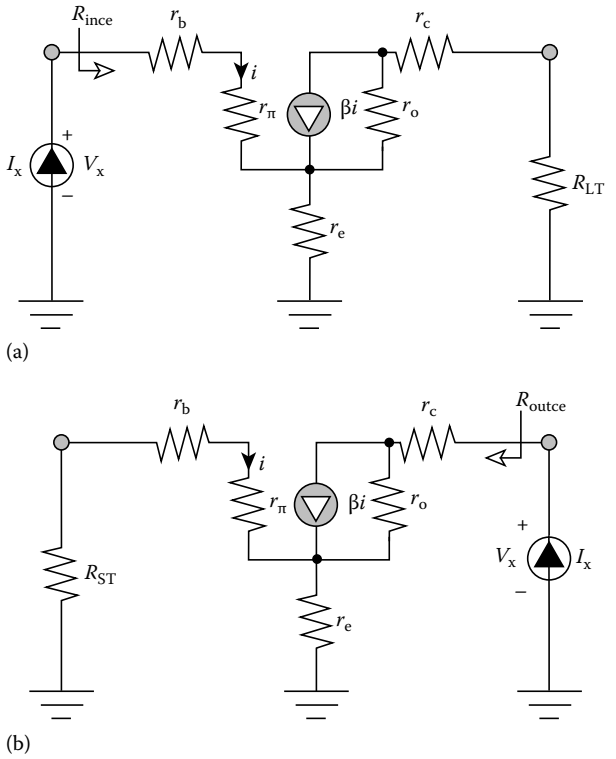


FIGURE 2.20 (a) Small-signal test structure used to determine the driving point input resistance of the common-emitter amplifier. (b) Small-signal test structure used to determine the driving point output resistance of the common-emitter amplifier.

terminating load resistance. Such independence implies no internal feedback from the output to input ports. It follows that the small-signal volt–ampere characteristics at the input port of a common-emitter amplifier can be modeled approximately by a simple resistance of value, R_{ince}, as defined by Equation 2.44. On the other hand, the large driving point output resistance R_{outce} suggests that a prudent output port model of a common-emitter stage is a Norton equivalent circuit. The Norton, or short-circuit, output current is proportional to the applied input signal voltage, V_{ST}, as depicted in Figure 2.21a. Alternatively, it can be expressed as a proportionality of the Norton input signal current, I_{ST}, as suggested in Figure 2.21b. In the former figure, the Norton current is

$$G_{fce}V_{ST} = \lim_{R_{LT}\to 0} I_{OS} = \lim_{R_{LT}\to 0}\left(-\frac{V_{OS}}{R_{LT}}\right) = \lim_{R_{LT}\to 0}\left(-\frac{A_{vce}V_{ST}}{R_{LT}}\right) \tag{2.46}$$

Subject to the assumption of large r_o,

$$G_{fce} = \lim_{R_{LT}\to 0}\left(-\frac{A_{vce}}{R_{LT}}\right) \approx \frac{\beta}{R_{ST} + r_b + r_\pi + (\beta+1)r_e} \tag{2.47}$$

Recalling Equation 2.29, this effective forward transconductance of the amplifier can be expressed in terms of the transconductance, g_m, of the transistor utilized in the amplifier. Specifically,

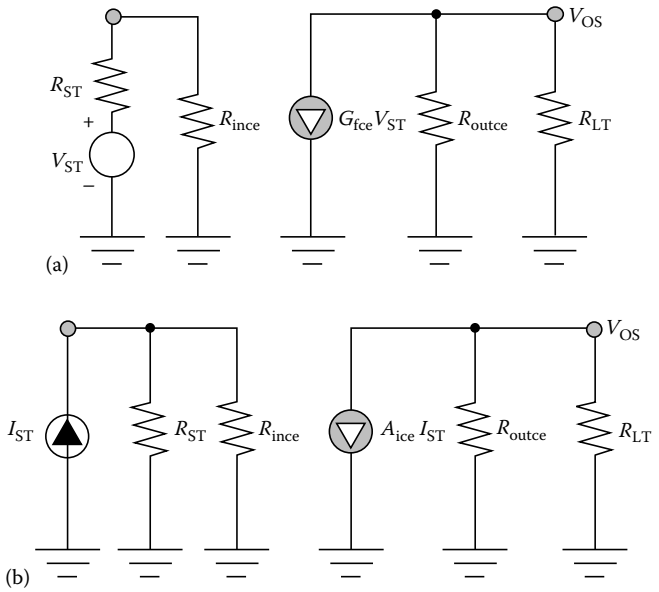


FIGURE 2.21 (a) Small-signal macromodel of a common-emitter amplifier in which the Norton output port circuit uses a voltage-controlled current source. (b) Small-signal macromodel of a common-emitter amplifier in which the Norton output port circuit uses a current-controlled current source.

$$G_{fce} \approx \frac{g_m}{1 = g_m r_e + ((r_e + r_b + R_{ST})/r_\pi)} \tag{2.48}$$

In the macromodel of Figure 2.21a, R_{outce} is very large by virtue of large r_o. Accordingly, the parallel combination of R_{outce} and R_{LT} is essentially R_{LT}, thereby implying an approximate common-emitter voltage gain of

$$A_{vce} \approx -G_{fce}R_{LT} \tag{2.49}$$

For the alternative macromodel in Figure 2.21b, the Norton current is

$$A_{ice}I_{ST} = A_{ice}\left(\frac{V_{ST}}{R_{ST}}\right) = \lim_{R_{LT}\to 0} I_{OS} = G_{fce}V_{ST} \tag{2.50}$$

Since $V_{ST} = R_{ST}I_{ST}$, it follows that A_{ice} is, for large r_o,

$$A_{ice} = G_{fce}R_{ST} \approx \beta\left[\frac{R_{ST}}{R_{ST} + r_b + r_\pi + (\beta + 1)r_e}\right] \tag{2.51}$$

Note that the Norton current proportionality A_{ice}, which is, in fact, the approximate ratio of the indicated output current, I_{OS}, to the Norton source current, I_{ST}, in the common-emitter configuration, is always smaller than β.

Example 2.1

Transistor Q_1 in the amplifier depicted in Figure 2.22a is fundamentally a common-emitter configuration since input signal is applied to its base terminal and the output voltage signal response is extracted at its

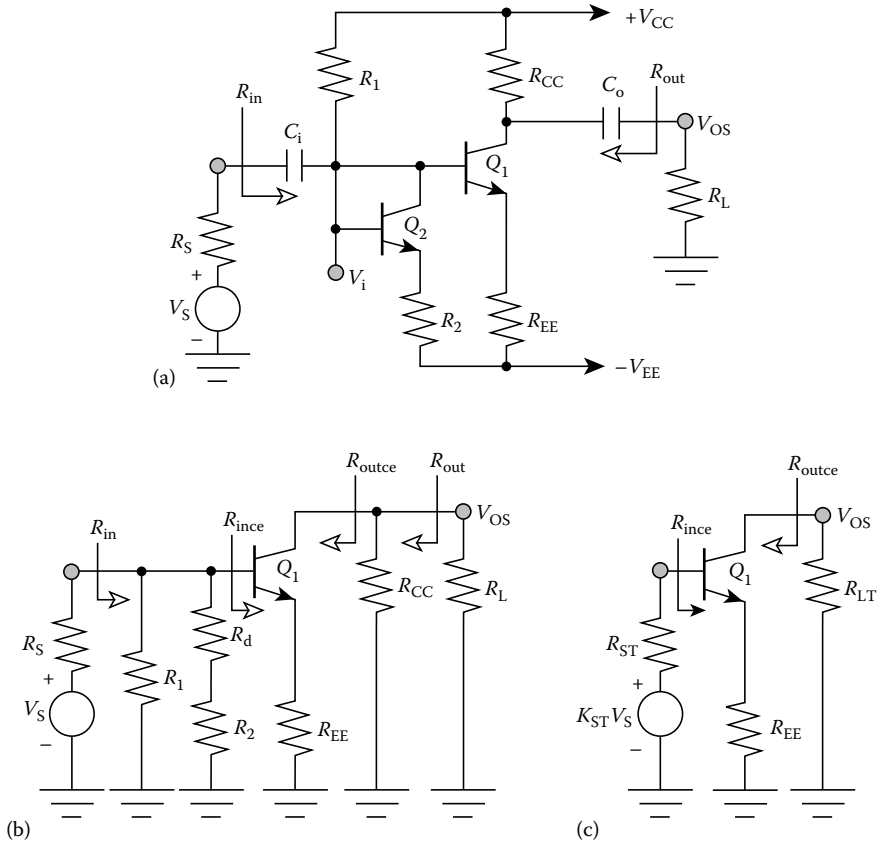


FIGURE 2.22 (a) Common-emitter amplifier with capacitively coupled input and output signal ports. (b) AC schematic diagram of the amplifier in (a). (c) Simplified AC schematic diagram of the amplifier in (a).

collector. The amplifier uses coupling capacitors C_i and C_o at its input and output ports. The input coupling capacitor, C_i, blocks the flow of static current in the source signal branch consisting of the series interconnection of the voltage, V_S, and the Thévenin source resistance, R_S. Accordingly, C_i, precludes R_S from affecting the biasing of both transistors used in the amplifier. Similarly, the output coupling capacitor, C_o, blocks the flow of static current in the external load resistance, R_L. Thus, both C_i and C_o can be viewed as open circuits for dc considerations. But simultaneously, these capacitors can be rendered transparent for AC situations by choosing them sufficiently large so that they emulate short circuits at the lowest frequency, say f_l, of signal-processing interest. In this problem, it is tacitly assumed that C_i and C_o, which are perfect DC open circuits, behave as good approximations of AC short circuits.

The subject amplifier utilizes a diode-connected transistor (Q_2) for temperature compensation of the static collector current conducted by transistor Q_1. For simplicity, assume that these two transistors have identical small-signal parameters of $r_b = 90\ \Omega$, $r_c = 55\ \Omega$, $r_\pi = 970\ \Omega$, $r_o = 42\ k\Omega$, and $\beta = 115$. Let the indicated circuit parameters be $R_1 = 2.2\ k\Omega$, $R_2 = 1.3\ k\Omega$, $R_{EE} = 75\ \Omega$, $R_{CC} = 3.9\ k\Omega$, $R_L = 1.0\ k\Omega$, and $R_S = 300\ \Omega$. Assuming that these circuit variables ensure linear operation of both devices, determine the small-signal voltage gain, $A_v = V_{OS}/V_S$, the driving point input resistance, R_{in}, and the driving point output resistance, R_{out}, of the amplifier. Finally, calculate the requisite minimum values of the input and output coupling capacitors, C_i and C_o, such that the lowest frequency f_l of interest is 500 Hz.

Solution

1. The first step of the solution process entails drawing the AC schematic diagram of the subject amplifier. By casting this diagram in the form of the canonic cell shown in Figure 2.19a, the gain and resistance expressions provided above can be exploited directly to assess the small-signal performance of the circuit at hand. Such a solution tack maximizes design-oriented understanding by avoiding the algebraic tedium implicit to an analysis of the entire small-signal equivalent circuit of the amplifier.

   To the foregoing end, observe that transistor Q_2 operates in its linear regime as a diode. It can therefore be viewed as the two terminal resistance R_d, given by Equation 2.32 or 2.33. Since the Early resistance is large, the latter expression can be used to arrive at $R_d = 10.64\ \Omega$.

   Since the power supply voltage, V_{EE}, is presumed ideal in the sense that it contains no signal component, the resultant series combination of R_d and R_2 returns base of transistor Q_1 to ground, as depicted in Figure 2.22b. Similarly, R_1 appears in shunt with the series interconnection of R_d and R_2, since V_{CC}, like V_{EE}, is also presumed to be an ideal constant (zero signal component) source of voltage. The AC schematic diagram of the input port is completed by noting that the AC short-circuit nature of the coupling capacitance C_i effectively connects the Thévenin representation of the signal source directly between the base of Q_1 and ground.

   At the output port of the amplifier, R_{CC} connects between the collector and ground since, as already exploited, V_{CC} is an AC short circuit. Moreover, the external load resistance, R_L, shunts R_{CC}, as shown in Figure 2.22b, because C_o behaves as an AC short circuit. The AC schematic diagram is completed by inserting the emitter degeneration resistance R_{EE} as a series element between ground and the emitter of transistor Q_1.
2. The diagram in Figure 2.22b can be straightforwardly collapsed to the simplified topology of Figure 2.22c. In the latter circuit, the effective load resistance, R_{LT}, is

$$R_{LT} = R_{CC} \| R_L = 795.9\ \Omega$$

   At the input port, the Thévenin resistance seen by the base of Q_1 is

$$R_{ST} = R_1 \| (R_d + R_2) \| R_S = 219.7\ \Omega$$

   while the corresponding Thévenin signal voltage can be expressed as $K_{ST}V_S$, where the voltage divider K_{ST} is

$$K_{ST} = \frac{R_1 \| (R_d + R_2)}{R_1 \| (R_d + R_2) + R_S} = 0.733$$

   The implication of this calculation is that, insofar as the active transistor Q_1 is concerned, the biasing resistances R_1 and R_2, cause a loss of more than 25% of the applied input signal.
3. The resultant AC schematic diagram in Figure 2.22c is virtually identical to the canonic topology in Figure 2.19a. Indeed, if the circuit resistance R_{EE} is absorbed into Q_1, where it appears in series with the internal emitter resistance r_e, the diagram is identical to the AC schematic diagram of the canonic common-emitter cell. Since $(r_e + R_{EE}) = 76.5\ \Omega$ is better than 560 times smaller than the resistance sum, $(r_o + r_c + R_{LT}) = 42.85\ \text{k}\Omega$, and $\beta = 115$ is more than 63,000 times larger than the resistance ratio $(r_e + R_{EE})/r_o = 0.00182$, the simplified expression in Equation 2.41 can be used to evaluate the voltage gain $V_{OS}/K_{ST}V_S$ in Figure 2.22c. From Equation 2.42, the effective small-signal beta is $\beta_{eff} = 112.7$. Then, with r_e replaced by $(r_e + R_{EE}) = 76.5\ \Omega$. Equation 2.41 gives

$$A_{vce} = \frac{V_{OS}}{K_{ST}V_S} = -8.99\ \text{V/V}$$

   It follows that the actual voltage gain of the amplifier in Figure 2.22a is

$$A_v = \frac{V_{OS}}{V_S} = K_{ST}A_{vce} = -6.59 \text{ V/V}$$

A better design, in the sense of achieving an adequate desensitization of circuit transfer characteristics with respect to parametric uncertainties, entails the use of a slightly larger emitter degeneration resistance R_{EE} selected to ensure that $(\beta_{eff}+1)(r_e+R_{EE}) \approx \beta_{eff} R_{EE} \gg (R_{ST}+r_b+r_\pi)$. For such a design, Equation 2.41 produces

$$A_v \approx -\frac{K_{ST}R_{LT}}{R_{EE}}$$

which is nominally independent of transistor parameters.

4. With r_e replaced by $(r_e+R_{EE}) = 76.5\ \Omega$, Equation 2.44 gives for the input resistance seen looking into the base of transistor Q_1 in Figure 2.22c, $R_{ince} = 9.76\ \text{k}\Omega$. It follows that the driving point input resistance seen by the source circuit in Figure 2.22b is

$$R_{in} = R_1 \| (R_d + R_2) \| R_{ince} = 757.6\ \Omega$$

5. The output resistance, R_{outce}, seen looking into the collector of transistor Q_1 in the diagram of Figure 2.22c is derived from Equation 2.45. With r_e replaced by $(r_e+R_{EE}) = 76.5\ \Omega$, $R_{outce} = 2.49\ \text{M}\Omega$. The resultant driving point output resistance seen by the load circuit in Figure 2.22b is

$$R_{out} = R_{CC} \| R_{outce} = 3894\ \Omega$$

The circuit output resistance is only a scant 6 Ω smaller than the collector biasing resistance R_{CC}, owning to the large value of R_{outce}. In turn, the value of the latter resistance is dominated by the last term on the right-hand side of Equation 2.45, which is proportional to the large forward Early resistance, r_o.

6. The input coupling capacitance, C_i, can be calculated with the help of the input port macromodel of Figure 2.23a. In this model, the subcircuit to the right of C_i in Figure 2.22a is replaced by its Thévenin equivalent circuit, which consists of the driving point input resistance, R_{in}, calculated previously. The voltage transfer function of this input port is

$$\frac{V_i(j\omega)}{V_S(j\omega)} = \left(\frac{R_{in}}{R_{in}+R_S}\right)\left[\frac{j\omega(R_{in}+R_S)C_i}{1+j\omega(R_{in}+R_S)C_i}\right]$$

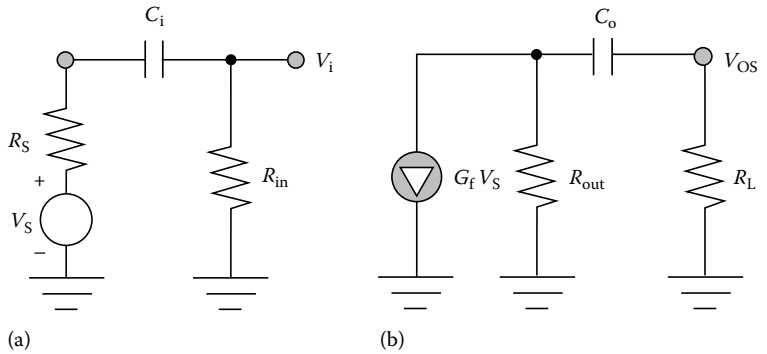


FIGURE 2.23 (a) Input port macromodel used in the calculation of the input coupling capacitor C_i. (b) Output port macromodel used to calculate the output coupling capacitor C_o.

An inspection of the foregoing relationship confirms that the dynamical effect of C_i on the voltage transfer function response is minimized if $\omega(R_{in} + R_s)C_i \gg 1$. At $\omega = 2\pi f_l$, the value of C_i that makes the left-hand side of this inequality equal to one is $C_i = 0.30$ μF. Observe that at $\omega_l(R_{in} + R_S)C_i = 1$,

$$\left|\frac{V_i(j\omega_1)}{V_S(j\omega_1)}\right| = \left(\frac{R_{in}}{R_{in} + R_S}\right)\left|\frac{j}{1+j}\right| = \frac{1}{\sqrt{2}}\left(\frac{R_{in}}{R_{in} + R_S}\right)$$

that is the magnitude of the input port voltage transfer function is a factor of the square root of two, or 3 dB, below the transfer function value realized at signal frequencies that are significantly higher than f_l. If this 3 dB attenuation is acceptable, $C_i = 0.30$ μF is appropriate to the design requirement.

7. The output coupling capacitance, C_o, is calculated analogously by exploiting the macromodel concepts overviewed in Figure 2.21a. To this end, the output port macromodel is offered in Figure 2.23b, where the effective forward transconductance, G_f, is such that $-G_f(R_{out}||R_L) = A_V$, as calculated in step 3. The voltage gain is seen to be

$$A_v(j\omega) = \frac{V_{OS}(j\omega)}{V_S(j\omega)} = -[G_f(R_{out}\|R_L)]\left[\frac{j\omega(R_{out} + R_L)C_o}{1 + j\omega(R_{out} + R_L)C_o}\right]$$

which has an algebraic form that is similar to the foregoing transfer relationship for the amplifier input port Thus, C_o is

$$C_o \geq \frac{1}{2\pi f_l(R_{out} + R_L)} = 0.065\ \mu F$$

Since the 0.30 μF input capacitor and the 0.065 μF output capacitor establish identical input and output port left-half-plane poles at the same frequency (f_1), the resultant attenuation at f_1 is actually larger than 3 dB. If this enhanced attenuation is unacceptable, the smaller of the two coupling capacitances can be made larger by a factor of three or so, thereby translating the associated pole frequency downward by a factor of three. In the present case, a plausible value of C_o is $C_o \geq (3)(0.065\ \mu F) = 0.2\ \mu F$.

It should be noted that the requisite two coupling capacitances are orders of magnitude too large for monolithic realization. Accordingly, if the subject amplifier is an integrated circuit, C_i and C_o are necessarily off-chip elements.

Example 2.2

When very large magnitudes of voltage gains are required, the output port of a common-emitter configuration can be terminated in an active load, as opposed to the passive resistive load encountered in the preceding example. Consider, for example, the complementary, NPN–PNP transistor amplifier whose schematic diagram appears in Figure 2.24a. The subcircuit containing transistors Q_1 and Q_2 is identical to that of the amplifier in Figure 2.22a. Indeed, for the purpose of this example, let the resistances, R_1, R_2, R_{EE}, and R_S, as well as small-signal parameters of transistors Q_1 and Q_2, remain at the values respectively stipulated for them in the preceding example. In the present diagram, the PNP transistor Q_3, along with its peripheral biasing resistances R_3, R_4, and R_5, supplants the resistance R_{CC} in the previously addressed common-emitter unit. Since no signal is applied to the base of Q_3, the subcircuit consisting of Q_3, R_3, R_4, and R_5 serves only to supply the appropriate biasing current to the collector of transistor Q_1. To the extent that this static current is invariant with temperature and the voltage signal response, V_{os}, established at the collector of Q_1, the Q_1 load circuit functions as a nominally constant current source. As a result, the effective load resistance, indicated as R_L in the subject figure,

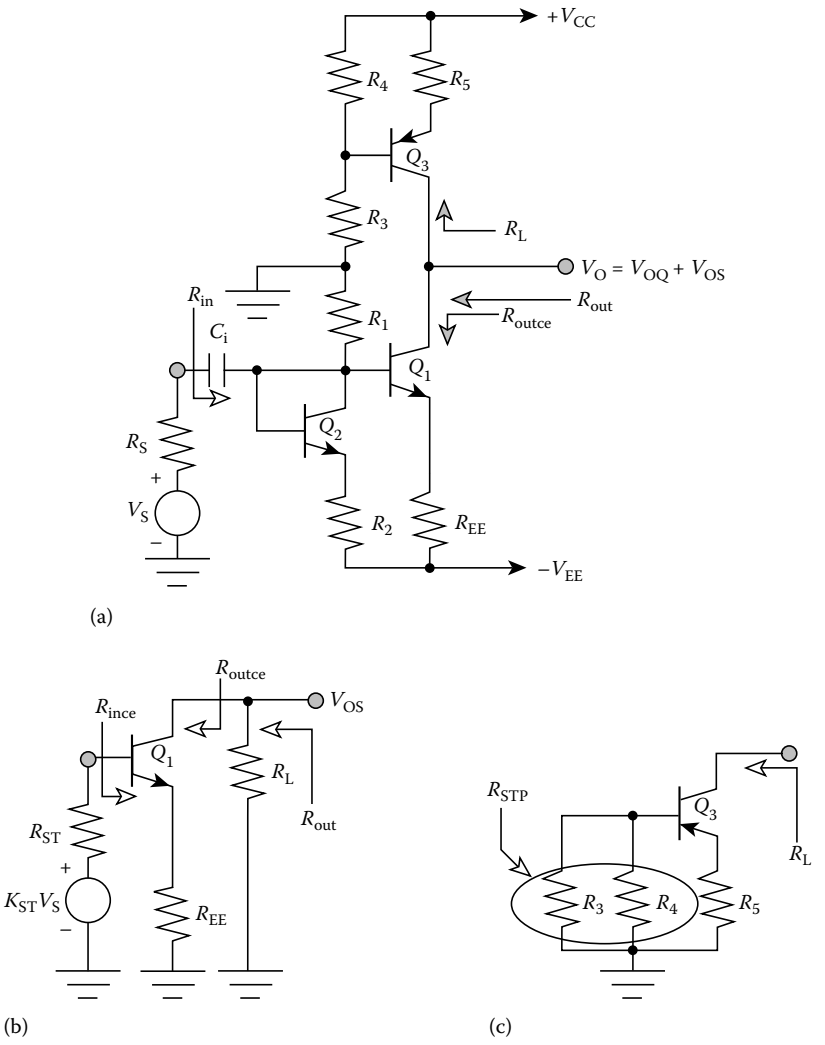


FIGURE 2.24 (a) Common-emitter amplifier with active current source load. (b) AC schematic diagram of the common-emitter unit. (c) AC schematic diagram of the active PNP transistor load.

seen by the collector of Q_1 is very large. In view of the absence of an external load appended to the output port, the resultant voltage gain of the amplifier is commensurately large.

Let $R_3 = 1.8$ kΩ, $R_4 = 3.3$ kΩ, and $R_5 = 100$ Ω. Moreover, let the small-signal parameters of the PNP transistor be $r_{bp} = 40$ Ω, $r_{cp} = 70$ Ω, $r_{ep} = 9$ Ω, $r_{\pi p} = 1100$ Ω, $r_{op} = 30$ kΩ, and $\beta_p = 60$. Assuming linear operation of all devices, determine the small-signal voltage gain $A_v = V_{OS}/V_S$, the driving point input resistance R_{in}, and the driving point output resistance R_{out} of the amplifier. As in the preceding example, the input coupling capacitance C_i can be presumed to act as an ac short circuit for the signal frequencies of interest.

Solution

1. The ac schematic diagram of the amplifier in Figure 2.24a is given in Figure 2.24b, where the PNP transistor load subcircuit is represented as an effective two terminal load resistance, R_L.

This representation is rendered possible by the fact that no signal is applied to the PNP load, which therefore acts only to supply biasing current to the collector of transistor Q_1. In the diagram, K_{ST}, R_{ST}, and R_{EE} (which effectively appears in series with r_e, the internal emitter resistance of Q_1) remain the same as in Example 2.1, namely, $K_{ST} = 0.733$, $R_{ST} = 219.7\ \Omega$, and $R_{EE} = 75\ \Omega$.

2. The AC schematic diagram of the Q_3 subcircuit alone appears in Figure 2.24c. A comparison of this figure with that shown in Figure 2.24b suggests that the subject diagram represents a PNP common-emitter amplifier under zero signal conditions. In particular, the Thévenin source resistance seen by the base of the PNP unit is $R_{STP} = R_3||R_4 = 1165\ \Omega$, while the emitter degeneration resistance of this subcircuit is $R_5 = 100\ \Omega$. It follows that the effective AC load resistance R_L terminating the collector port of Q_1 is the driving point output resistance of a common-emitter stage. With $r_c \triangleq r_{cp} = 70\ \Omega$, $r_b \triangleq r_{bp} = 40\ \Omega$, $r_e \triangleq (r_{ep} + R_5 = 109\ \Omega$, $r_\pi \triangleq r_{\pi p} = 1.1\ k\Omega$, $R_{ST} \triangleq R_{STP} = 1165\ \Omega$, $r_o \triangleq r_{op} = 30\ k\Omega$, and $\beta \triangleq \beta_p = 60$, (Equation 2.45) yields $R_{outce} \triangleq R_L = 111.5\ k\Omega$.
3. The voltage gain, input resistance, and output resistance of the actively loaded common-emitter amplifier can now be computed. For $r_c \triangleq r_{cn} = 55\ \Omega$, $r_b \triangleq r_{bn} = 90\ \Omega$, $r_e \triangleq (r_{en} + R_{EE} = 76.5\ \Omega$, $r_\pi \triangleq r_{\pi n} = 970\ k\Omega$, $R_{ST} \triangleq R_{STN} = 219.7\ \Omega$, $r_o \triangleq r_{on} = 42\ k\Omega$, $R_{LT} \triangleq R_L = 111.5\ k\omega$, and $\beta \triangleq \beta_n = 115$, Equation 2.42 gives an effective NPN transistor beta of $\beta_{eff} = 31.46$, and Equation 2.41 yields a voltage gain of $A_{vce} = V_{OS}/K_{ST}V_{ST} = -931.9$. It follows that the small-signal voltage gain of the stage at hand is $A_v = V_{OS}/V_S = K_{ST}A_{vce} = -682.6\ V/V$. It should be noted that this voltage gain is the ratio of only the signal component, V_{OS}, of the net output voltage, V_O (which contains a quiescent component of V_{OQ}) to the source signal voltage, V_S.
4. From Equation 2.43, the driving point input resistance seen looking into the base of transistor Q_1 in Figure 2.22b is $R_{ince} = 3.54\ k\Omega$. Then,

$$R_{in} = R_1||(R_d + R_2)||R_{ince} = 666.7\ \Omega$$

This input resistance differs slightly from the corresponding calculation in the preceding example owing to the reduction in the effective forward AC beta caused by the large active load resistance.

5. The resistance R_{outce} seen looking into the collector of transistor Q_1 remains the same as calculated in Example 2.1, namely, $R_{outce} = 2.49\ M\Omega$. It follows that the driving point output resistance of the amplifier under investigation is

$$R_{out} = R_L||R_{outce} = 106.7\ k\Omega$$

This large output resistance means that the actively loaded common-emitter configuration is a relatively poor voltage amplifier. In particular, an output buffer is mandated to couple virtually any practical external load resistance to the amplifier output port. In addition to reducing the output resistance, such a properly designed and implemented output buffer can reliably establish and stabilize the quiescent output voltage, V_{OQ}.

2.2.3.3 Common-Base Amplifier

The second of the canonic linear bipolar gain cells in the ''common-base amplifier,'' whose NPN and PNP AC schematic diagrams and corresponding small-signal equivalent circuit appear in Figure 2.25a and b, respectively. As it is confirmed below, the input resistance, R_{incb}, of this stage is very small and the output resistance, R_{outcb}, is very large. Accordingly, the common-base unit comprises a relatively poor voltage amplifier in the sense that its voltage gain, though potentially large, is a sensitive function of both the Thévenin source resistance R_{ST} and the Thévenin load resistance R_{LT}.

Although the common-base amplifier is not well suited for general voltage gain applications, it is an excellent ''current buffer,'' which is ideally characterized by zero input resistance, infinitely large output resistance, and unity current gain. When used for current buffering purposes, the common-base amplifier

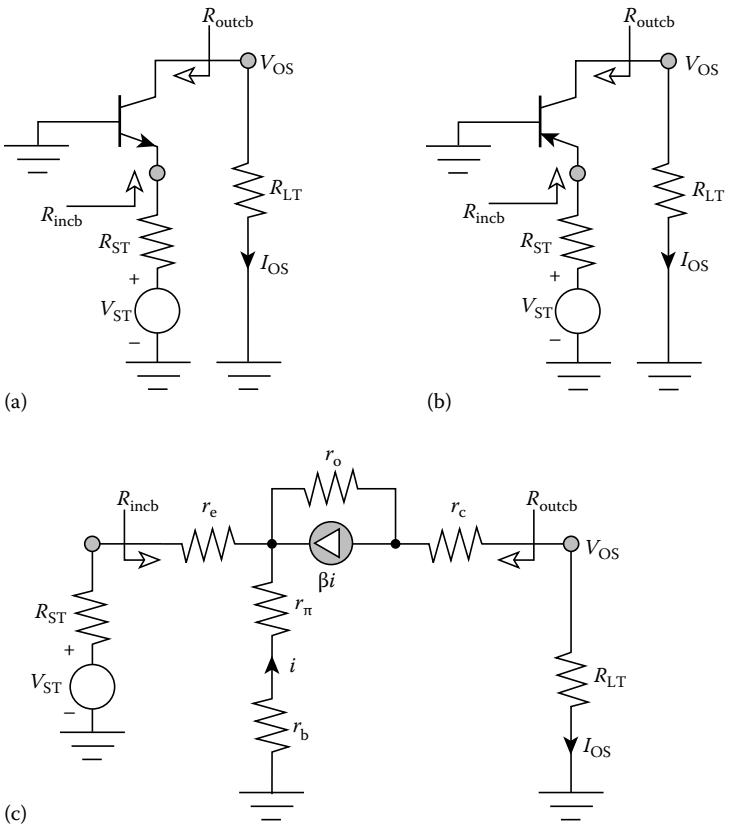


FIGURE 2.25 (a) AC schematic diagram of an NPN common-base amplifier. (b) AC schematic diagram of a PNP common-base amplifier. (c) Small-signal, low-frequency equivalent circuit of the common-base amplifier.

rarely appears as a stand alone single-stage amplifier, since signal excitations, particularly at the input and output ports of an electronic system are invariably formatted as voltages. Instead, it is invariably used in conjunction with an input voltage to current converter and/or an output current to voltage converter to achieve desired system performance characteristics.

The small-signal analysis of the common-base stage is considerably simplified if the assumption of large r_o is exploited at the outset. To this end, the equivalent circuit shown in Figure 2.25b reduces to the structure of Figure 2.26a. In the latter equivalent circuit, observe a signal emitter current i_{es} that relates to the indicated signal base current i in accordance with the Kirchhoff's current law constraint

$$i_{es} = -(\beta + 1)i \tag{2.52}$$

The signal component of the output current I_{OS} is therefore expressible as

$$I_{OS} = -\beta_i = \left(\frac{\beta}{\beta + 1}\right) i_{es} = \alpha i_{es} \tag{2.53}$$

where Equations 2.52 and 2.53 are used. The last result suggests the alternative model in Figure 2.26b, which is slightly more convenient version of the model in Figure 2.26a in that the current-controlled current source, αi_{es}, is dependent on the signal input port current i_{es}, as opposed to the signal current, i, that flows in the grounded base lead.

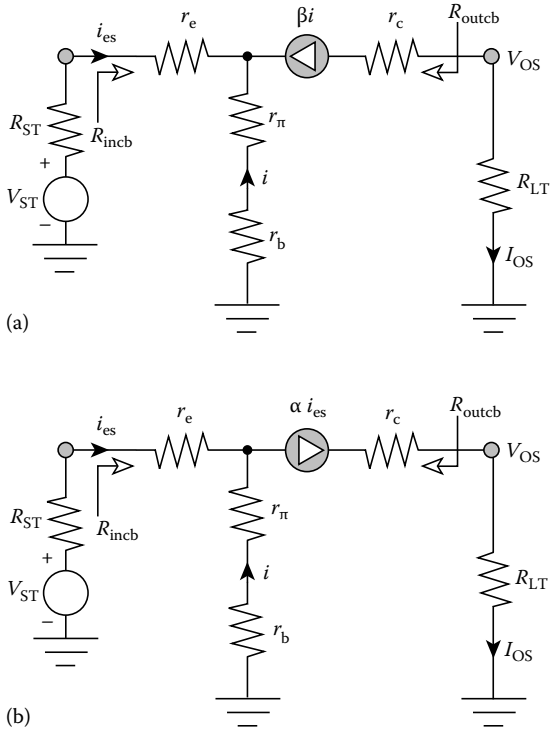


FIGURE 2.26 (a) The equivalent circuit of Figure 2.25c, simplified for the case of very large Early resistance r_o. (b) Modification of the circuit in (a) in which the current-controlled current source is rendered dependent on the input signal current i_{es}.

By inspection of the equivalent circuit in Figure 2.26b, the small-signal voltage gain A_{vcb} of the common-base cell is

$$A_{vcb} = \frac{\alpha R_{LT}}{R_{ST} + r_e + (1-\alpha)(r_\pi + r_b)} = \frac{\alpha R_{LT}}{R_{ST} + R_d} \tag{2.54}$$

where Equation 2.39 is used once again and R_d is the diode resistance defined by Equation 2.33. In contrast to the common-emitter cell, the common-base stage has no voltage gain phase inversion. But like the common-emitter configuration, the common-base voltage gain is directly proportional to the effective load resistance. It is also almost inversely proportional to the effective source resistance, given that the diode resistance R_d is small.

Although the voltage gain is vulnerable to uncertainties in the terminating load and source resistances, the common-base current gain, A_{icb}, is virtually independent of R_{LT} and R_{ST}. This contention follows from the fact that A_{icb}, which is the ratio of I_{OS} to the Norton equivalent source current, V_{ST}/R_{ST}, is

$$A_{icb} = \left(\frac{R_{ST}}{R_{LT}}\right) A_{vcb} = \frac{\alpha R_{ST}}{R_{ST} + R_d} \tag{2.55}$$

which is independent of R_{LT} (to the extent that the Early resistance r_o can indeed be ignored). Since the signal in a current-drive amplifier is likely to have a large source resistance, $R_{ST} \gg R_d$, which

implies $A_{icb} \approx \alpha$, independent of R_{LT} and R_{ST}. Note that this approximate current gain is essentially unity, since α as introduced by Equation 2.39 approaches one for the typically encountered circumstances of large β.

The input and output resistances of the common-base amplifier follow immediately from an analysis of the model in Figure 2.26b. In particular, the driving point input resistance, R_{incb}, is

$$R_{incb} = \left(\frac{V_{ST}}{i_{es}}\right)\Bigg|_{R_{ST}=0} = r_e + (1-\alpha)(r_\pi + r_b) = R_d \tag{2.56}$$

where the numerical value is of the order of only a few tens of ohms. On the other hand, the driving point output resistance, R_{outcb}, is infinitely large since $V_{ST} = 0$ constrains i_{es}, and thus αi_{es}, to zero. In turn, $\alpha i_{es}, = 0$ means that R_{LT} in Figure 2.26b faces an open circuit, whence $R_{outcb} = \infty$.

To the extent that the common-base amplifier is excited from a signal current source and that the forward Early resistance of the utilized transistor is very large, the common-base amplifier is seen to have almost unity current gain, very low input resistance, and infinitely large output resistance. Its transfer characteristics therefore approximate those of an ideal current buffer. Of course, the finite nature of the forward Early resistance renders the observable driving point output resistance of a common-base cell large, but nonetheless finite. The actual output resistance can be determined as the V_x to I_x ratio in the ac schematic diagram of Figure 2.27a. The requisite analysis is algebraically cumbersome owing to the presence of r_o in shunt with the current-controlled current source in the equivalent circuit of Figure 2.25c. Fortunately, however, an actual circuit analysis can be circumvented by a proper interpretation of cognate common-emitter results formulated earlier.

In order to demonstrate the foregoing contention, consider Figure 2.27b, which depicts the AC schematic diagram for determining the driving point output resistance R_{outce} of a common-emitter amplifier. The only difference between the two AC schematic diagrams in Figure 2.27 is the topological placement of the effective source resistance, R_{ST}. In the common-base stage, this source resistance is in series with the emitter of a transistor with a base that is grounded. On the other hand, R_{ST} appears in the common-emitter configuration as an element in series with the base of a transistor whose emitter is ground. It follows that Equation 2.45 can be used to deduce an expression for R_{outcb}, provided that in Equation 2.45 R_{ST} is set to zero and r_e is replaced by $(r_e + R_{ST})$.

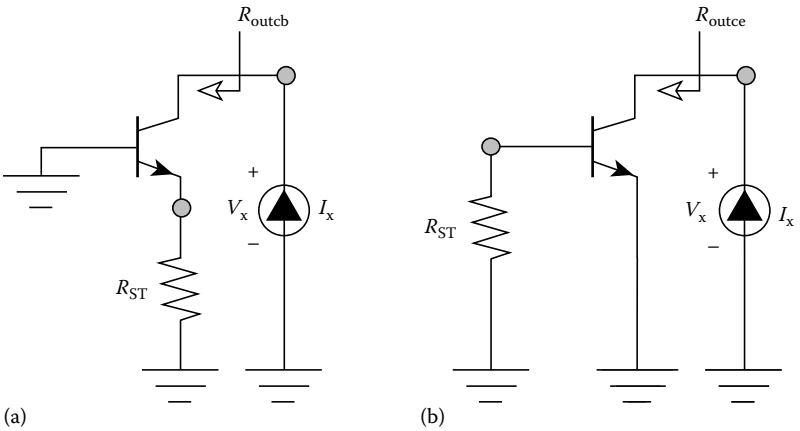


FIGURE 2.27 (a) AC schematic diagram appropriate to the computation of the driving point output resistance of a common-base amplifier. (b) AC schematic diagram pertinent to computing the driving point output resistance of a common-emitter amplifier.

The result is

$$R_{\text{outcb}} = r_c + (r_e + R_{ST}) \| (r_\pi + r_b) + \left[\frac{\beta(r_e + R_{ST})}{r_e + R_{ST} + r_\pi + r_b} + 1 \right] r_o \tag{2.57}$$

For large R_{ST} and large r_o, Equation 2.47 reduces to

$$R_{\text{outcb}} \approx (\beta + 1) r_o \tag{2.58}$$

which is an extremely large output resistance.

The common-base stage is generally used in conjunction with a common-emitter amplifier to form the "common-emitter-common-base cascode," whose schematic diagram is shown in Figure 2.28. In this application, the common-emitter stage formed by transistor Q_1, the emitter degeneration resistance, R_{EE}, and the biasing elements, R_1 and R_2, serves as a transconductor that converts the input signal voltage, V_S, to a collector current whose signal component is i_{1s}. Note that such conversion is encouraged by the fact that the effective load resistance, R_{Leff}, terminating the collector of Q_1, is the presumably low input resistance of the common-base stage formed by transistor Q_2 and the biasing resistances, R_3 and R_4. Since the current gain of a common-base stage is essentially unity, Q_2 translates the signal current in its emitter to an almost identical signal current flowing through the collector load resistance, R_L. The latter element acts as a current to voltage convert to establish the signal component, V_{OS}, of the net output voltage V_O.

The analysis of the common-emitter–common-base cascode begins by representing the collector port of the common-emitter configuration by its Norton equivalent circuit. Assuming that the input coupling

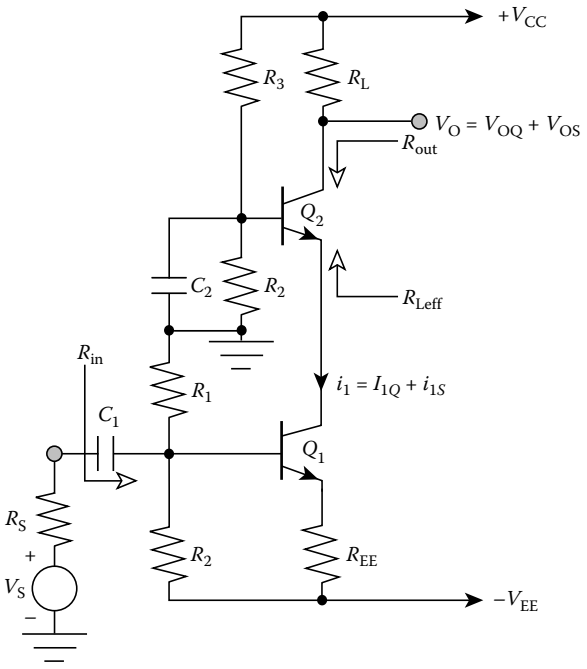


FIGURE 2.28 Schematic diagram of a common-emitter–common-base cascode. The common-emitter stage formed by transistor Q_1 and its peripheral elements acts as a voltage to current converter. Transistor Q_2 and its associated biasing elements function as a current amplifier, while the load resistance, R_L, acts as a current to voltage converter.

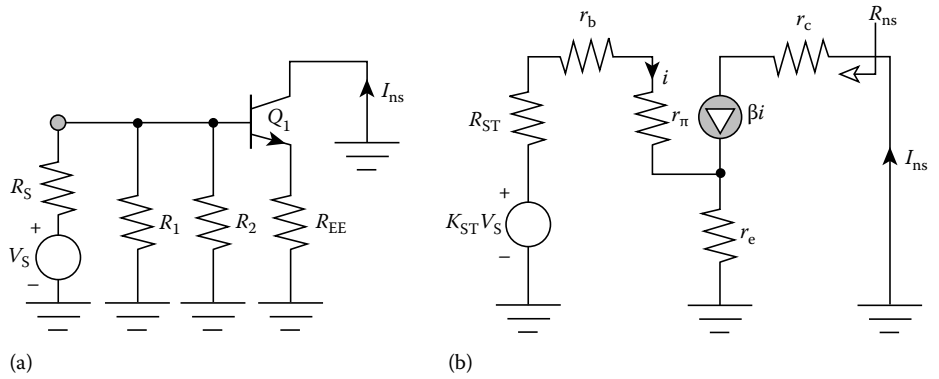


FIGURE 2.29 (a) AC schematic diagram used to calculate the Norton equivalent output circuit of the common-emitter subcircuit in the cascode configuration of Figure 2.28. (b) Small-signal model of the AC circuit in (a).

capacitor, C_1, is sufficiently large to enable its replacement by a short circuit over the signal frequency range of interest, the pertinent AC schematic diagram is the circuit in Figure 2.29a, where I_{ns} symbolizes the Norton, or short-circuit signal current conducted by the collector of transistor Q_1. The corresponding small-signal equivalent circuit appears in Figure 2.29b, where the Early resistance is tacitly ignored, the effective source resistance, R_{ST}, seen by the base of Q_1 is

$$R_{ST} = R_S \| R_1 \| R_2 \tag{2.59}$$

and the voltage divider K_{ST} is

$$K_{ST} = \frac{R_1 \| R_2}{R_1 \| R_2 + R_S} \tag{2.60}$$

Using the model in Figure 2.29b, it is a simple matter to show that

$$I_{ns} = \beta i = G_{ns} V_S \tag{2.61}$$

where G_{ns}, which can be termed the "Norton transconductance" of the common-emitter stage, is

$$G_{ns} = \frac{\beta K_{ST}}{R_{ST} + r_b + r_\pi + (\beta + 1) r_e} \tag{2.62}$$

The Norton output resistance R_{ns} is infinitely large by virtue of the assumption of infinitely large Early resistance.

The foregoing results permit drawing the AC schematic diagram of the common-base component of the common-emitter–common-base cascode in the topological form depicted in Figure 2.30a. From the corresponding small-signal equivalent circuit in Figure 2.30b, which assumes that the capacitor, C_2, behaves as an AC short circuit and which once again ignores transistor Early resistance, the voltage gain, say A_v, follows immediately as

$$A_v = \frac{V_{OS}}{V_S} = -\alpha G_{ns} R_L = -\frac{\alpha \beta K_{ST} R_L}{R_{ST} + r_b + r_\pi + (\beta + 1) r_e} \tag{2.63}$$

The driving point output resistance R_{out}, like the Norton output resistance of the common-emitter stage, is infinitely large. In fact, R_{out} is a good approximation of infinity. Its numerical value

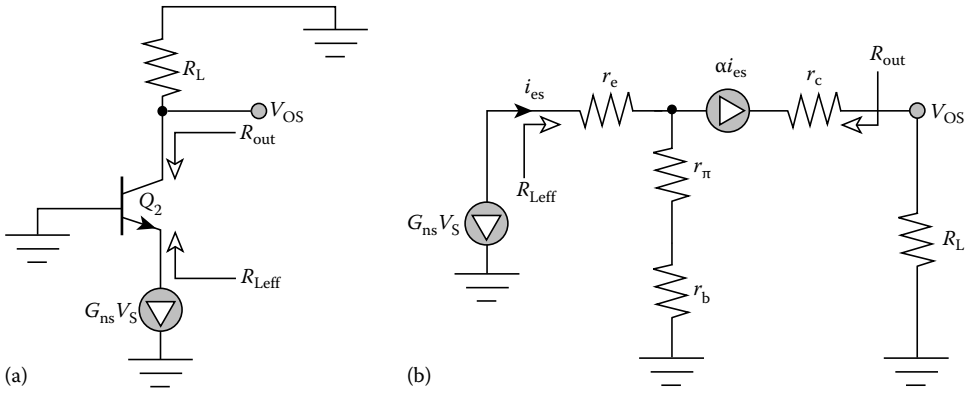


FIGURE 2.30 (a) The effective AC schematic diagram of the common-base component of the common-emitter–common-base cascode of Figure 2.28. (b) Small-signal model of the AC circuit in (a).

approaches $(\beta + 1)r_o$, since the terminating resistance, R_{ns}, seen in the emitter circuit of transistor Q_2 is of the order of r_o.

Equation 2.63 is similar in form to Equation 2.41 which defines the voltage gain of a simple common-emitter amplifier. A careful comparison of the two subject relationships suggests that the voltage gain of the common-emitter–common-base cascode of Figure 2.28 is equivalent to the voltage gain achieved by a simple common-emitter stage, whose output port is loaded in an effective load resistance of αR_L. Although α is close to unity, but nonetheless always less than one, an effective load of αR_L implies that the voltage gain of the cascode is slightly less than that achieved by the common-emitter stage alone, provided, of course, that the transistors utilized in both configurations have identical small-signal parameters. A question therefore arises as to the prudence of incorporating common-base signal processing in conjunction with a common-emitter unit stage.

In fact, no practical purpose is served by a common-emitter–common-base cascode if the load resistance R_L driven by the amplifier is very small. But if the load resistance imposed on the output port of a simple common-emitter amplifier is large, as it is when the load itself is realized actively, as per Example 2.2, the effective transistor beta defined by Equation 2.42 is appreciably smaller than the actual small-signal beta. The result is a degraded common-emitter amplifier voltage gain. In this situation, the insertion of a common-base stage between the output port of the common-emitter amplifier and R_L, as diagrammed in Figure 2.28, increases the degraded gain of the common-emitter amplifier alone by restoring the effective beta of the common-emitter transistor to a value that approximates the actual small-signal beta of the transistor. This observation follows from the fact that the effective load resistance, R_{Leff}, seen by the collector of the common-emitter transistor in the cascode topology is of the order of only a small diode resistance. It follows from Equation 2.42 that β_{eff} for $R_{LT} = R_{Leff}$ is likely to significantly larger than the value of β_{eff} that derives from the load condition, $R_{LT} = R_L$.

The reason for using common-base circuit technology in conjunction with a common-emitter amplifier is circuit broadbanding. In particular, a carefully designed common-emitter–common-base cascode configuration displays a 3 dB bandwidth and a gain-bandwidth product that are significantly larger than the bandwidth and gain-bandwidth product afforded by a common-emitter stage of comparable gain. The primary reason underlying this laudable attribute is the low effective load resistance presented to the collector of the common-emitter stage by the emitter of the common-base structure. This low resistance attenuates the magnitude of the phase-inverted voltage gain of the common-emitter circuit, thereby reducing the deleterious effects of Miller multiplication of the base-collector junction depletion capacitance implicit to the common-emitter transistor [8].

2.2.3.4 Common-Collector Amplifier

While the common-base configuration functions as a current buffer, the ''common-collector amplifier,'' or ''emitter follower,'' with AC schematic diagrams that appear in Figure 2.31a and b operates as a voltage buffer. It offers high input resistance, low output resistance, a voltage gain approaching unity, and a moderately large current gain. The small-signal model of the emitter follower is the circuit in Figure 2.31c.

Assuming infinitely large Early resistance, r_o, a straightforward analysis of the subject model reveals an emitter-follower voltage gain A_{vcc} of

$$A_{vcc} = \frac{V_{OS}}{V_S} = \frac{R_{LT}}{R_{LT} + R_d + (1-\alpha)R_{ST}} \tag{2.64}$$

where R_d is the diode resistance given by Equation 2.33. Observe that A_{vcc} is a positive less than unity number. The indicated gain approaches one for $R_{LT} \gg R_d + (1-\alpha)R_{ST}$. Although the voltage gain is less than one, the corresponding current gain, which is simply the voltage gain scaled by a factor of (R_{ST}/R_{LT}), can be substantially larger than one.

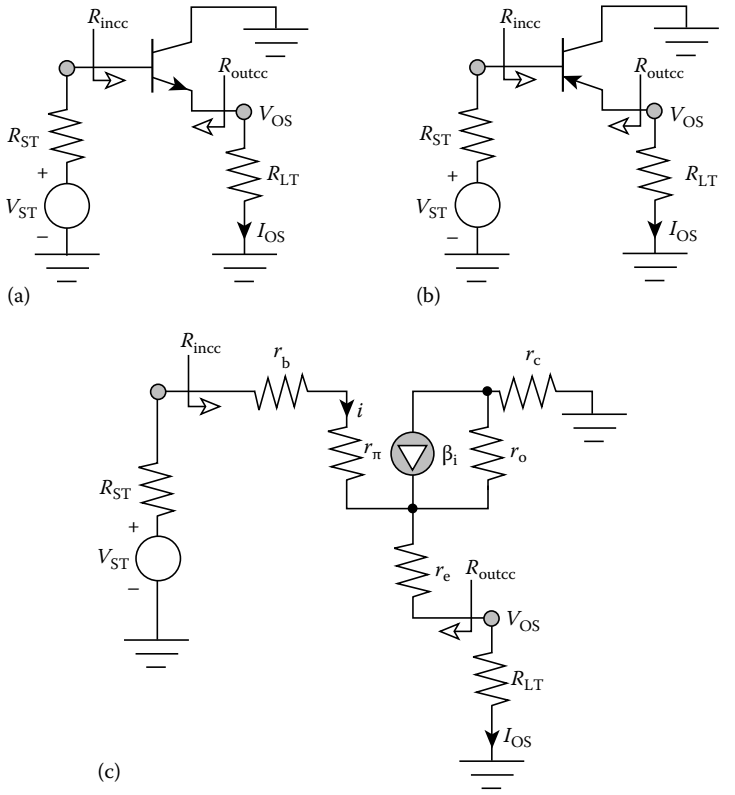


FIGURE 2.31 (a) AC schematic diagram of an NPN common-collector amplifier. (b) AC schematic diagram of PNP common-collector amplifier. (c) Small-signal low-frequency equivalent circuit of the common-collector amplifier, assuming that the Early resistance is sufficiently large to ignore.

It is simple to confirm that the driving point input resistance R_{incc} and the driving point output resistance R_{outcc} of the emitter follower are, respectively,

$$R_{incc} = r_b + r_\pi + (\beta + 1)(r_e + R_{LT}) = (\beta + 1)(R_d + R_{LT}) \tag{2.65}$$

and

$$R_{outcc} = r_e + \frac{r_\pi + r_b + R_{ST}}{(\beta + 1)} = R_d + (1 - \alpha)R_{ST} \tag{2.66}$$

It is interesting and instructive to note that the driving point input resistance, R_{incc}, of an emitter follower is of the same form as the driving point input resistance, R_{incc}, of a common-emitter stage. Indeed, if R_{LT} in the emitter follower is zero, $R_{incc} + R_{ince}$. This result is reasonable in view of the fact that for both the emitter-follower and common-emitter configurations, the input resistance is, as suggested by the test circuits in Figure 2.32a and b, the Thévenin resistance presented to the source circuit by the base of the subject transistor. Moreover, the common-collector output resistance R_{outcc} mirrors the driving point input resistance R_{incb} for the common-base amplifier. In fact, $R_{incb} + R_{outcc}$ if the base of a common-base amplifier is terminated to ground in a resistance of value R_{ST}. Once again, the latter observation is

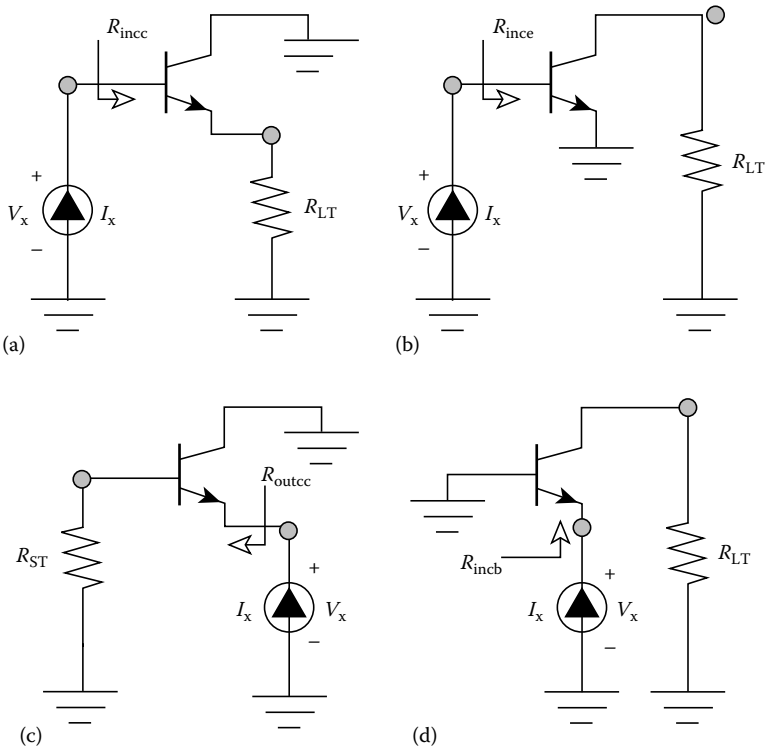


FIGURE 2.32 (a) Test circuit for determining the driving point input resistance R_{incc} of a common-collector amplifier. (b) Test circuit for determining the driving point input resistance R_{ince} of a common-emitter amplifier. Note that $R_{incc} = R_{ince}$ if R_{LT} in the common-collector unit is zero. (c) Test circuit for determining the driving point output resistance R_{outcc} of a common-collector amplifier. (d) Test circuit for determining the driving point input resistance R_{incb} of a common-base amplifier. Note that $R_{outcc} = R_{incb}$ if R_{ST} in the common-collector unit is zero.

intuitively correct, for, as depicted in Figure 2.32c and d, the emitter of the transistor comprises the input terminal for the common-base amplifier and the output terminal of an emitter follower.

An inspection of Equation 2.64 confirms a common-collector voltage gain that tends toward unity for progressively larger Thévenin load resistances, R_{LT}. Correspondingly, the driving point input resistance of an emitter follower increases dramatically with increasing R_{LT}. These observations are often pragmatically exploited by supplanting the passive load in the schematic diagram of Figure 2.31a with an active load, as suggested in Figure 2.33a. Since the effective AC load resistance must be large to achieve a near unity voltage gain, this active load must function as a sink of nominally constant current. To this end, the active load in question is shown conducting a net current that consists of a static current component I_{CS} and a signal current component I_{OS}, where the constant current sink nature of the load implies $I_{CS} \gg I_{OS}$.

The subject active load can be represented by its Norton equivalent circuit, as diagrammed in Figure 2.33b, where I_{CS} is depicted as a constant current source and I_{OS} is made to flow through a resistance, R_{cs}. The latter branch element represents the dynamic resistance presented to the emitter-follower output port by the two terminal active termination. Note that $R_{cs} = \infty$ yields $I_{os} = 0$, which implies an active load that behaves as an ideal constant current sink. The corresponding AC schematic

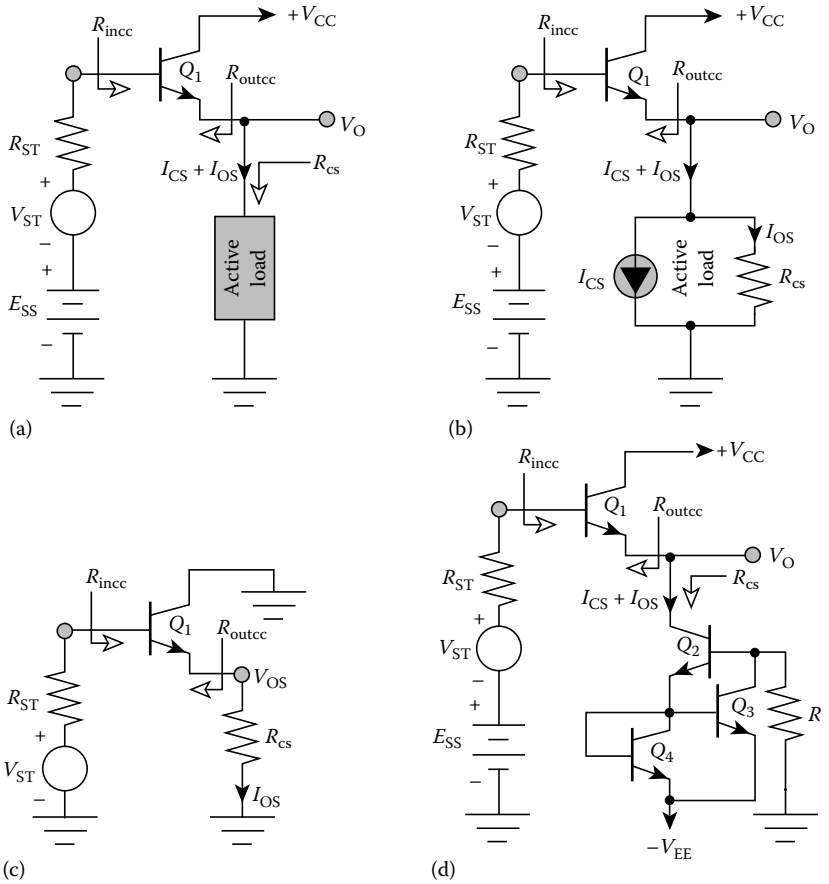


FIGURE 2.33 (a) Emitter follower with active load that conducts a static current I_{CS} and a signal component I_{OS}. (b) The Norton equivalent circuit of the active load. (c) AC schematic diagram of the actively loaded emitter follower. (d) Wilson current mirror realization of the active load.

diagram, which is offered in Figure 2.33c, is derived from Figure 2.33b by setting I_{CS} to zero, since I_{CS} itself is a constant current that is devoid of any signal component. Additionally, the biasing sources, V_{CC}, V_{EE}, and E_{SS} are presumed ideal and are therefore set to zero as well.

The AC schematic diagram in Figure 2.33c is identical to that in Figure 2.31a, subject to the proviso that $R_{LT} = R_{cs}$. But since R_{cs} is presumably a large resistance, substituting $R_{LT} = R_{cs}$ into Equation 2.66 to evaluate the voltage gain of the actively loaded emitter follower is at least theoretically inappropriate because the subject gain equation is premised on the assumption of a large Early resistance, r_o; specifically Equation 2.64 reflects the assumption $R_{LT} \ll r_o$. A voltage gain expression, more accurate than Equation 2.64, derives from an analysis of the model in Figure 2.31c. If r_o is included in this analysis, but if r_e and r_c are sufficiently small to justify their neglect, the result for $R_{LT} = R_{cs}$ is

$$A_{vcc} = \frac{V_{OS}}{V_S} \approx \frac{r_o \| R_{cs}}{(r_o \| R_{cs}) + R_d + (1 - \alpha) R_{ST}} \tag{2.67}$$

where the algebraic form collapses to Equation 2.64 if $R_{cs} \ll r_o$. Similarly, the revised expression for the driving point input resistance is

$$R_{incc} \approx r_b + r_\pi + (\beta + 1)(r_o \| R_{cs}) \tag{2.68}$$

The output resistance R_{outcc} remains as stipulated by Equation 2.66.

The active load appearing in Figure 2.33a can be realized as any one of a variety of NPN current sources [9]. Figure 2.33d offers an examples of such a realization in the form of the Wilson current mirror formed of transistors Q_2, Q_3, and Q_4, and the current setting resistor, R [10]. This subcircuit establishes an extremely high dynamic resistance between the collector of transistor Q_2 and the signal ground. In particular, if β_2 and r_{o2} symbolize the AC beta and forward Early resistance, respectively, of transistor Q_2, it can be shown that

$$R_{cs} \approx \frac{\beta_2 r_{o2.}}{2} \tag{2.69}$$

Note further that the static current, I_{CS}, conducted by the Wilson mirror flows through the emitter lead of the emitter-follower transistor Q_1. Thus, the biasing stability of Q_1 is determined by the thermal sensitivity of the static current that flows in the Wilson subcircuit.

Example 2.3

In order to dramatize the voltage buffering property of an emitter follower, return to the amplifier addressed analytically in Example 2.1 and drawn schematically in Figure 2.22a. Let the two coupling capacitors remain large enough to approximate them as AC short circuits over the signal frequency range of interest, and let the small-signal parameters of the two transistors remain at $r_b = 90\ \Omega$, $r_c = 55\ \Omega$, $r_e = 1.5\ \Omega$, $r_\pi = 970\ \Omega$, $r_o = 42\ k\Omega$, and $\beta = 115$. The circuit parameters also remain the same; namely, $R_1 = 2.2\ k\Omega$, $R_2 = 1.3\ k\Omega$, $R_{EE} = 75\ \Omega$, $R_{CC} = 3.9\ k\Omega$, and $R_S = 300\ \Omega$. But instead of $R_L = 1.0\ k\Omega$, consider an external load termination of $R_L = 300$. Reevaluate the small-signal voltage gain, $A_v = V_{OS}/V_S$, for the subject amplifier. Compare this result to the voltage gain achieved when an emitter follower is inserted between the collector of transistor Q_1 and the 300 Ω load termination, as depicted in Figure 2.34a. Assume that the small-signal parameters of the emitter-follower transistor Q_3 and those of the two transistors that comprise the diode-compensated current sink load of the follower are identical to the model parameters of transistors Q_1 and Q_2.

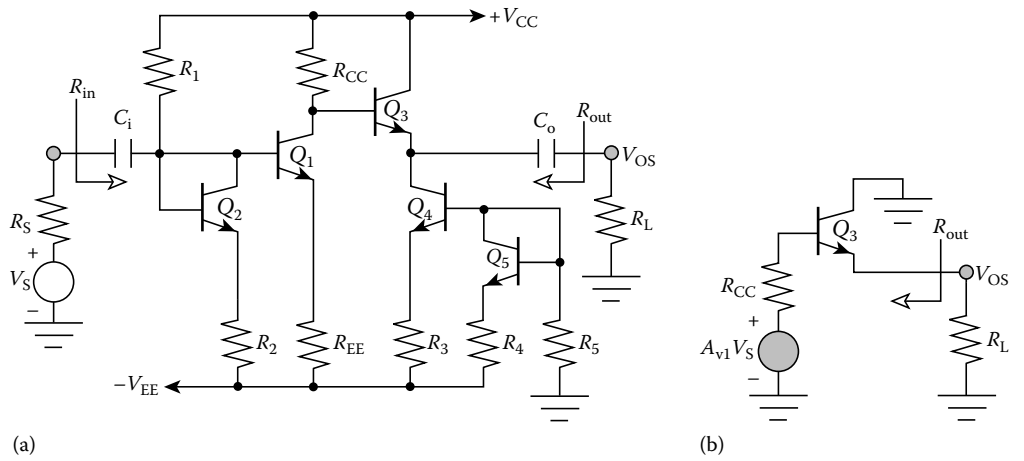


FIGURE 2.34 (a) The amplifier of Figure 2.22, but with an emitter-follower buffer inserted between the gain stage and the terminating load resistance. (b) AC schematic diagram of the amplifier in (a).

Solution

1. With reference to Figure 2.22 and Example 2.1, the Thévenin load resistance R_{LT} is now

$$R_{LT} = R_{CC} \| R_L = 278.6\ \Omega$$

The parameters R_{ST} and K_{ST} in Figure 2.22c remain unchanged at the previously computed values of $R_{ST} = 219.7\ \Omega$ and $K_{ST} = 0.733$. Then, ignoring the effects of the finite, but large, Early resistance r_o, the voltage gain is

$$A_v = \frac{V_{OS}}{V_S} \approx -\frac{\beta K_{ST} R_{LT}}{R_{ST} + r_b + r_\pi + (\beta + 1)(r_e + R_{EE})}$$

whence $A_v = -2.31$ V/V.

2. Consider now the amplifier modification shown in Figure 2.34a. Transistor Q_3 functions as an emitter follower to buffer the terminating load resistance R_L effectively seen by the gain stage formed of transistor Q_1 and its peripheral elements. Transistors Q_3 and Q_4 form a diode-compensated current sink that comprises the active load presented to the emitter-follower output port under static operational conditions. To the extent that r_o can be tacitly ignored, this current sink comprises an infinitely large dynamic resistance. Accordingly, the AC schematic diagram seen to the right of the collector of transistor Q_1 is the structure identified in Figure 2.34b.
3. The source circuit that drives the base of transistor Q_3 in Figure 2.34b is the Thévenin equivalent circuit established at the collector of transistor Q_1 in Figure 2.34a. The signal voltage associated with this source circuit is the open circuit voltage developed at the Q_1 collector; that is, it is the voltage at the Q_1 collector with the load formed of transistor Q_3 and its peripheral elements removed. Since the circuit to the left of the base of transistor Q_3 is a linear network, this Thévenin voltage is necessarily proportional to the input signal V_S. The indicated constant of proportionality in Figure 2.34b, A_{v1}, can rightfully be termed the open-circuit voltage gain of the first stage of the subject amplifier. This is to say that A_{v1} is A_v, as determined in step 1 above, but with R_L removed and therefore, R_{LT} set equal to R_{CC}. It follows that

$$A_{v1} = -\frac{\beta K_{ST} R_{CC}}{R_{ST} + r_b + r_\pi + (\beta + 1)(r_e + R_{EE})}$$

or $A_{v1} = -32.38$ V/V. Since r_o is taken to be infinitely large, the resistance seen looking into the collector of transistor Q_1 is likewise infinitely large. As a result, the Thévenin resistance associated with the source circuit in the AC diagram of Figure 2.34b is $R_{CC} = 3.9$ kΩ.

4. Recalling Equation 2.64, the voltage gain of the circuit in Figure 2.22b is

$$A_{vcc} = \frac{V_{OS}}{A_{v1} V_S} = \frac{R_L}{R_L + R_d + (1 - \alpha) R_{CC}} = 0.871 \text{ V/V}$$

The resultant overall circuit gain is

$$A_v = \frac{V_{OS}}{V_S} = A_{v1} A_{vcc}$$

or $A_v = -28.21$ V/V. Recalling the results of step 1 of this computational procedure, the effect of the emitter follower is to boost the gain magnitude of the original configuration by a factor of about 12.2.

5. From Equation 2.66, the driving point output resistance of the buffered amplifier is

$$R_{out} = R_d + (1 - \alpha) R_{CC}$$

or $R_{out} = 44.3$ Ω. Note that for the original nonbuffered case, the output resistance is $R_{CC} = 3.9$ kΩ.

2.2.3.5 Darlington Connection

In the Darlington connection, whose basic schematic diagram is abstracted in Figure 2.35a, the emitter of one transistor Q_1 is incident with the base of a second transistor Q_2 and the two transistor collector leads are connected. The output signal is extracted as either the current flowing in the collector of transistor Q_2 or the voltage developed at the emitter of Q_2. In the former case, the indicated Darlington connection functions as a transconductance amplifier. In the latter case, an output signal voltage at the emitter of Q_2 renders the connection functional as a voltage follower, or buffer. In both applications, the small-signal driving point input resistance, R_{ind}, seen looking into the base of transistor Q_1 is large. On the other hand, the driving point output resistance, R_{outed}, seen at the emitter is small and virtually independent of the source resistance R_S. The output resistance, R_{outcd}, presented to the node at which the two transistor collectors are incident is large. At the expense of forward transconductance, R_{outcd} can be enhanced by returning the collector of Q_1 to the $+V_{CC}$ bus, instead of to the collector of transistor Q_2. For a nonzero collector load resistance, R_{LC}, this alternate connection, which is diagrammed in Figure 2.35b eliminates Miller multiplication of the base–collector junction capacitance of transistor Q_1, thereby resulting in an improved transconductance frequency response.

A fundamental problem that plagues both of the foregoing Darlington connections is the fact that the static emitter current conducted by Q_1 is identical to the static base current drawn by Q_2. Accordingly, the emitter current of Q_1 is likely to be much smaller than the biasing current commensurate with optimal gain-bandwidth product in this device. Moreover, this emitter current cannot be predicted accurately since it is inversely proportional to the Q_2 static beta, whose numerical value is an unavoidable uncertainty. This poor biasing translates into an unreliable delineation of the static and small-signal parameters for Q_1. In turn, potentially significant uncertainties shroud the forward transfer and driving point resistance characteristics of the Darlington configuration.

To remedy the situation at hand, an additional current path, usually directed to signal ground, is provided at the junction of the Q_1 emitter and the Q_2 base, as suggested in Figure 2.35c and d.

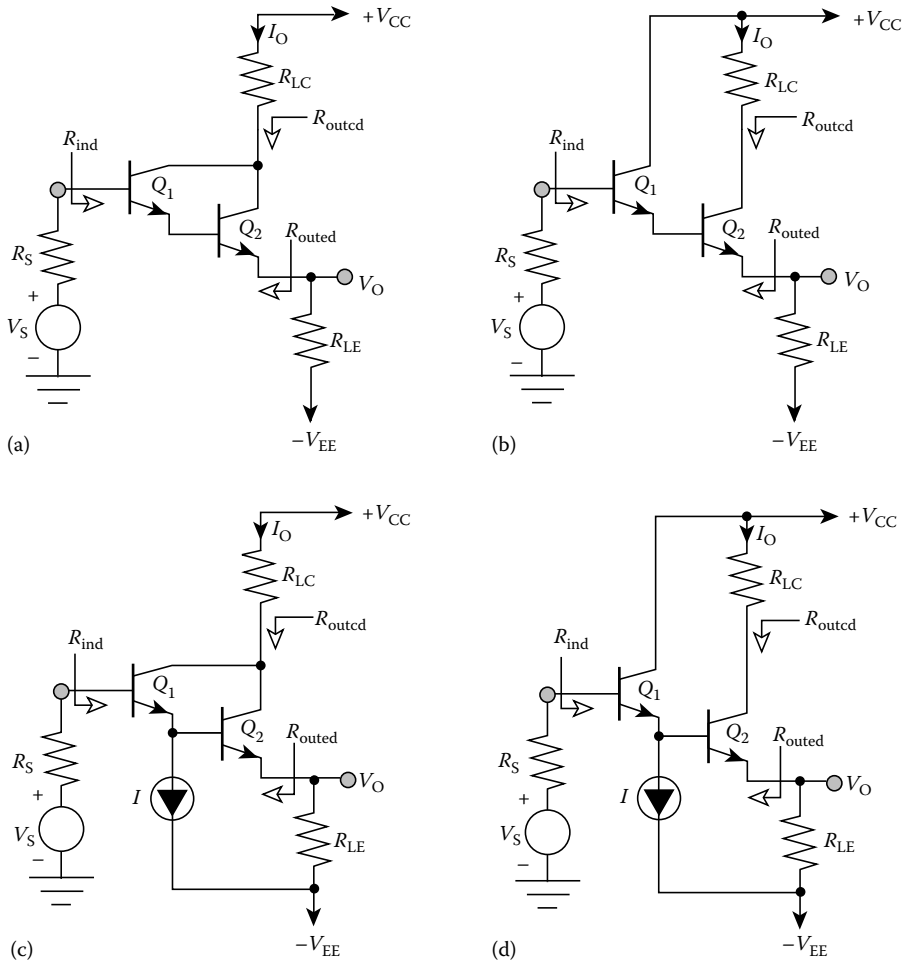


FIGURE 2.35 (a) The basic Darlington connection. (b) Alternative Darlington connection for wide-band transconductance response. (c) Darlington connection with input transistor current compensation. (d) Alternative Darlington connection with input transistor current compensation.

The appended current path can be a simple two terminal resistance, although care must be exercised to ensure that this resistance is sufficiently large to avoid seriously compromising the large driving point input resistance afforded by the basic Darlington connection in either of the preceding diagrams. Since large resistance and realistic biasing currents may prove to be conflicting design requirements, the appended current path is often an active current sink, such as the Wilson mirror load explored earlier in conjunction with the common-collector amplifier. Note in the latter two diagrams that the current, indicated as I, conducted by the appended passive or active current path is essentially the emitter current of transistor Q_1, provided that I is much larger than the base currents of Q_2.

The small-signal BJT equivalent circuit of Figure 2.16a can be used to deduce the transfer and driving point resistance characteristics of any of the Darlington connections depicted in Figure 2.35. An analysis is provided herewith for only the configuration in Figure 2.35c, since this topology is the most commonly encountered Darlington circuit and the others are amenable to very straightforward analyses. To this end, the model for the subject structure is offered in Figure 2.36, where it is assumed that both transistors are biased so that their corresponding small-signal parameters are nominally identical. Moreover, the Early

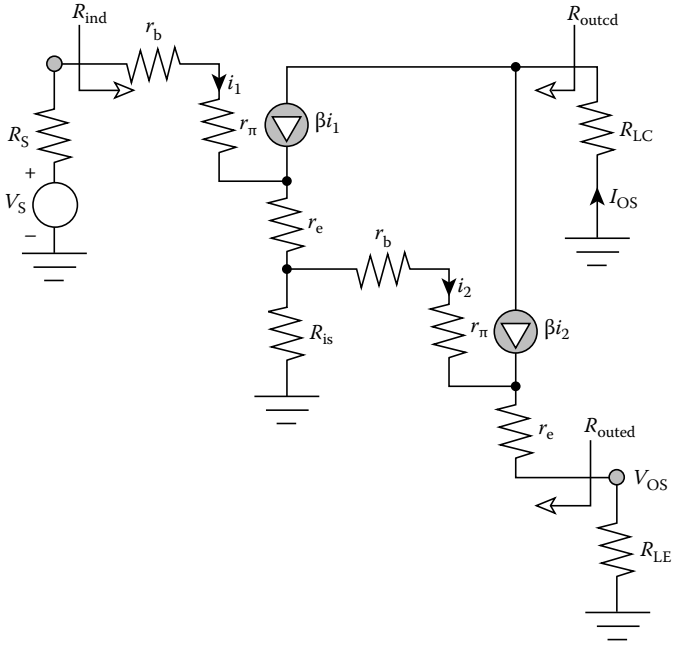


FIGURE 2.36 Small-signal equivalent circuit of the Darlington connection in Figure 2.35c. The Early resistance is ignored, and both transistors are presumed to have identical corresponding small-signal parameters. The resistance, R_{is}, represents the terminal AC resistance associated with the appended current path conducting current I in Figure 2.35c.

resistance of each transistor is presumed to be sufficiently large to warrant its neglect, and a resistance, R_{is}, is included to account for the terminal resistance of the appended current path discussed above. Letting

$$k_{is} \triangleq \frac{R_{is}}{R_{is} + (\beta + 1)(R_d + R_{LE})} \tag{2.70}$$

denote the small-signal current divider between the appended current path and the base circuit of transistor Q_2, it can be shown that the driving point input resistance R_{ind} is

$$R_{ind} = (\beta + 1)[R_d + (\beta + 1)k_{is}(R_d + R_{LE})] \tag{2.71}$$

For large ac beta,

$$R_{ind} \approx (\beta + 1)^2 k_{is}(R_d + R_{LE}) \tag{2.72}$$

which is maximal for $k_{is} \approx 1$. From Equation 2.70, the latter constraint mandates that the appended current path be designed so that its small-signal terminal resistance satisfies the inequality $R_{is} \gg (\beta + 1)(R_d + R_{LE})$.

The voltage gain, A_{vd}, from the signal source to the emitter port is

$$A_{vd} = \frac{V_{OS}}{V_S} = \frac{(\beta + 1)^2 k_{is} R_{LE}}{R_S + (\beta + 1)R_d + (\beta + 1)^2 k_{is}(R_d + R_{LE})} \tag{2.73}$$

where V_{OS} is the signal component of the net output voltage V_O. Equation 2.53 reduces to

$$A_{vd} \approx \frac{R_{LE}}{R_d + R_{LE}} \tag{2.74}$$

for large ac beta. The corresponding driving point output resistance, R_{outed}, is

$$R_{outed} = R_d + \frac{R_d}{(\beta + 1)k_{is}} + \frac{R_S}{(\beta + 1)^2 k_{is}} \approx R_d \tag{2.75}$$

At the collector port, the driving point output resistance, R_{outcd}, is infinitely large to the extent that the Early resistance r_o of both transistors can be ignored. For finite r_o, this resistance is of the order of, and slightly larger than, $(r_o/2)$. Finally, the model in Figure 2.36 yields a forward transconductance, G_{fd}, from the signal source to the collector port of

$$G_{fd} = \frac{I_{OS}}{V_S} = \frac{\beta[1 + (\beta + 1)k_{is}]}{R_S + (\beta + 1)R_d + (\beta + 1)^2 k_{is}(R_d + R_{LE})} \tag{2.76}$$

where I_{OS} is the signal component of the net output current I_O. Equation 2.76 collapses to

$$G_{fd} \approx \frac{\alpha}{R_d + R_{LE}} \tag{2.77}$$

for large AC beta.

2.2.4 Differential Amplifier

The ''differential amplifier'' is a four-port network, as suggested in Figure 2.37a. Source signals represented by the voltages, V_{S1} and V_{S2}, which have Thévenin resistances of R_{S1} and R_{S2}, respectively, are applied to the two amplifier input ports. The two output ports are terminated in three load resistances. Two of these loads, R_{L1} and R_{L2}, are ''single-ended terminations'' in that they provide a signal path to ground from each of the two output terminals. A third load resistance, R_{LL}, is differentially connected between the two output terminals. In response to the two applied source signals, two ''single-ended output'' voltages, V_{O1} and V_{O2}, are generated across R_{L1} and R_{L2}, and a ''differential output voltage,'' V_{DO}, is established across R_{LL}. This third output response is the difference between V_{O1} and V_{O2}; that is,

$$V_{DO} = V_{O1} - V_{O2} \tag{2.78}$$

The salient features of a differential amplifier are unmasked by the concepts of ''differential- and common-mode'' excitation and response. To this end, let the ''differential input source voltage,'' V_{DI}, be defined as

$$V_{DI} \triangleq V_{S1} - V_{S2} \tag{2.79}$$

and let the ''common-mode input voltage,'' V_{CI}, be

$$V_{CI} \triangleq \frac{1}{2}(V_{S1} + V_{S2}) \tag{2.80}$$

The differential input voltage is seen as the difference between the two applied source excitations. On the other hand, the common-mode input voltage is the arithmetic average of the two source voltages.

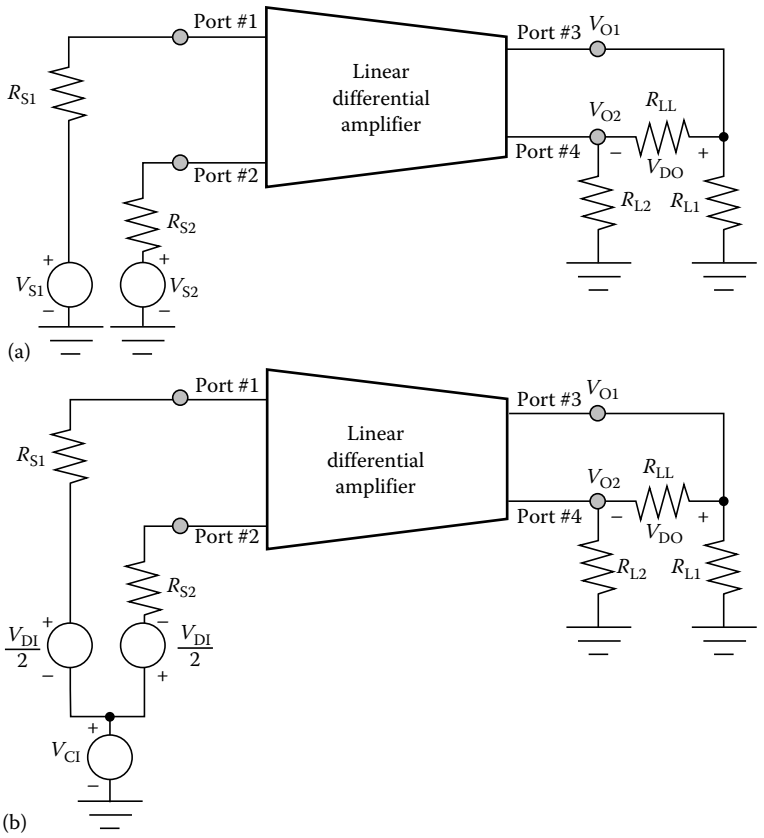


FIGURE 2.37 (a) System-level diagram of a differential amplifier. (b) System-level diagram that depicts the electrical implications of the common-mode and the differential-mode input source voltages.

When solved for V_{S1} and V_{S2}, Equations 2.79 and 2.80 give

$$V_{S1} = V_{CI} + \frac{1}{2}V_{DI} \tag{2.81a}$$

$$V_{S2} = V_{CI} - \frac{1}{2}V_{DI} \tag{2.81b}$$

The preceding two expressions allow the diagram of Figure 2.37a to be drawn in the form shown in Figure 2.37b. This alternative representation underscores the fact that the Thévenin voltage applied to either input port is the superposition of a common-mode source voltage and a component proportional to the differential-mode source voltage. The common-mode component raises both of the open-circuit input terminals to a voltage that lies above ground by an amount, V_{CI}. Superimposed with V_{CI} at the open circuit terminals of port 1 is a differential-mode voltage, $V_{DI}/2$. Simultaneously, a voltage of $-V_{DI}/2$ superimposes with V_{CI} at the open-circuit terminals of port 2.

The fact that two general source excitations applied to a four-port system can be separated into a voltage component that appears only differentially across the two system input ports and a single-ended common-mode voltage component that is simultaneously incident with both of the system input ports makes it possible to achieve signal discrimination in a differential circuit. In particular, a differential amplifier can be designed so that it amplifies the differential component of two source signals while

rejecting (in the sense of amplifying with near zero gain) their common-mode component. Signal discrimination is useful whenever an electronic system must process low-level electrical signals that are contaminated by spurious inputs, such as the voltage ramifications of electromagnetic interference or the biasing perturbations induced by temperature. If the two input ports of a differential amplifier are geometrically proximate and have matched driving point input impedances, these spurious excitations impact the two input ports identically. The undesired inputs are therefore common-mode excitations that can be rejected by a differential amplifier that is well designed in the sense of producing output port responses that are sensitive to only differential inputs.

If the differential amplifier in Figure 2.37 is linear, superposition theory gives

$$V_{O1} = A_{11}V_{S1} + A_{12}V_{S2} \tag{2.82a}$$

$$V_{O2} = A_{21}V_{S1} + A_{22}V_{S2} \tag{2.82b}$$

where the A_{ij} are constants, independent of V_{S1} and V_{S2}. When Equation 2.81a and b are inserted into the last two relationships, the single-ended output voltages are expressible as

$$V_{O1} = (A_{11} + A_{12})V_{CI} + (A_{11} - A_{12})\frac{V_{DI}}{2} \tag{2.83a}$$

$$V_{O2} = (A_{22} + A_{21})V_{CI} - (A_{22} - A_{21})\frac{V_{DI}}{2} \tag{2.83b}$$

It follows that the differential output voltage is

$$V_{DO} = (A_{11} - A_{22} + A_{12} - A_{21})V_{CI} + (A_{11} + A_{22} - A_{12} - A_{21})\frac{V_{DI}}{2} \tag{2.84}$$

Since the common-mode output voltage is

$$V_{CO} \triangleq \frac{1}{2}(V_{O1} + V_{O2}) \tag{2.85}$$

Equation 2.83a and b yield

$$V_{CO} = \left(\frac{A_{11} + A_{22} + A_{12} + A_{21}}{2}\right)V_{CI} + \left(\frac{A_{11} - A_{22} - A_{12} + A_{21}}{2}\right)\frac{V_{DI}}{2} \tag{2.86}$$

The ability of a differential amplifier to process differential excitations is measured by the ''differential-mode voltage gain,'' A_D. This performance index is defined as the ratio of the differential output voltage to the differential input voltage, under the condition of zero common-mode input voltage. From Equation 2.84,

$$A_D \triangleq \left[\frac{V_{DO}}{V_{DI}}\right]\Bigg|_{V_{CI}=0} = \frac{A_{11} + A_{22} - A_{12} - A_{21}}{2} \tag{2.87}$$

On the other hand, the ''common-mode voltage gain,'' A_C, is a measure of the common-mode signal rejection characteristics of a differential amplifier. It is the ratio of the common-mode output voltage to the common-mode input voltage, under the condition of zero differential input voltage. Using Equation 2.86,

$$A_C \triangleq \left[\frac{V_{CO}}{V_{CI}}\right]\Bigg|_{V_{DI}=0} = \frac{A_{11} + A_{22} + A_{12} + A_{21}}{2} \tag{2.88}$$

A measure of the degree to which a differential amplifier rejects common-mode excitation is the "common-mode rejection ratio ρ," which is the ratio of the differential-mode voltage gain to the common-mode voltage gain. From Equations 2.87 and 2.88

$$\rho \triangleq \frac{A_D}{A_C} = \frac{A_{11} + A_{22} - A_{12} - A_{21}}{A_{11} + A_{22} + A_{12} + A_{21}} \tag{2.89}$$

A common-mode gain of zero indicates that no common-mode output results from the application of common-mode input signals. Therefore, a practical design goal is the realization of a differential amplifier that has the largest possible magnitude of common-mode rejection ratio.

2.2.4.1 Balanced Differential Amplifier

Most differential amplifiers are "balanced." Two operating requirements are satisfied by balanced differential systems. First, with zero common-mode input voltage, the two single-ended output voltages are mutually phase inverted, but otherwise identical. By Equation 2.83a and b, the balance requirement implies the parametric constraint.

$$A_{11} - A_{12} = A_{22} - A_{21} \tag{2.90}$$

Second, equal single-ended output voltages result when the differential-mode input voltage is zero. Using Equation 2.83a and b once again, this stipulation requires

$$A_{11} + A_{12} = A_{22} + A_{21} \tag{2.91}$$

Equations 2.90 and 2.91 combine to deliver the balanced operating requirement

$$\left.\begin{aligned} A_{11} &= A_{22} \\ A_{12} &= A_{21} \end{aligned}\right\} \tag{2.92}$$

From Equations 2.87 through 2.89, the differential-mode voltage gain, the common-mode voltage gain, and the common-mode rejection ratios of a balanced differential amplifier are

$$A_D = A_{11} - A_{12} \tag{2.93}$$

$$A_C = A_{11} + A_{12} \tag{2.94}$$

$$\rho = \frac{A_{11} - A_{12}}{A_{11} + A_{12}} \tag{2.95}$$

Moreover, the single-ended output voltages in Equation 2.83a and b become

$$V_{O1} = A_C V_{CI} + A_D \frac{V_{DI}}{2} = \frac{A_D}{2}\left(1 + \frac{2V_{CI}}{\rho V_{DI}}\right) V_{DI} \tag{2.96a}$$

$$V_{O2} = A_C V_{CI} - A_D \frac{V_{DI}}{2} = -\frac{A_D}{2}\left(1 - \frac{2V_{CI}}{\rho V_{DI}}\right) V_{DI} \tag{2.96b}$$

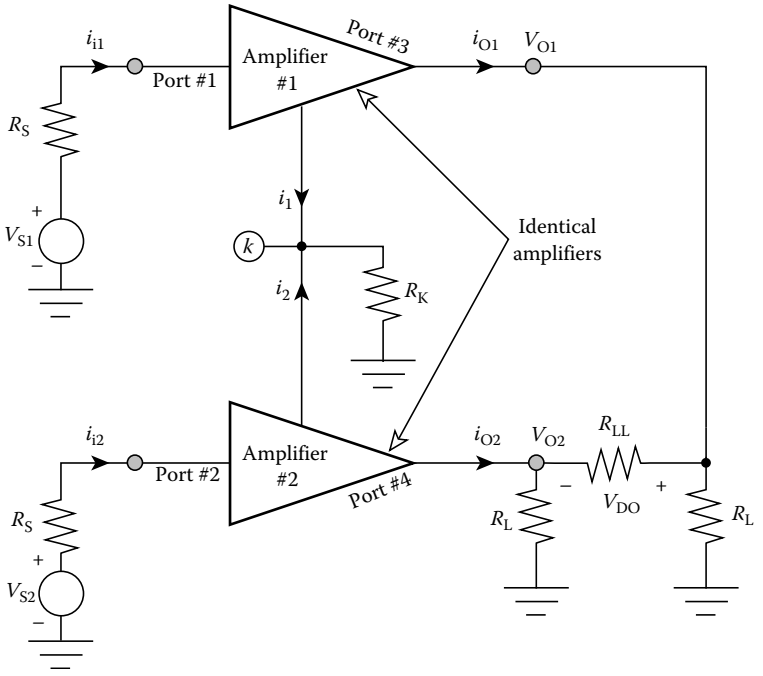


FIGURE 2.38 Generalized system diagram of a balanced differential system. The topology is an AC schematic diagram in that requisite biasing subcircuits of either amplifier are not shown.

which give rise to a differential response of

$$V_{DO} = V_{O1} - V_{O2} = A_D V_{DI} \tag{2.97}$$

Equation 2.96a and b shows that a balanced differential amplifier having a very large common-mode rejection ratio produces single-ended outputs that are nominally phase-inverted versions of one another and approximately independent of the common-mode input voltage. On the other hand, the differential output voltage of a balanced system is independent of the common-mode input signal, regardless of the value of the common-mode rejection ratio.

Figure 2.38 depicts the most straightforward way to implement balance in a differential configuration. In this abstraction, two identical single-ended amplifiers, such as those discussed in earlier subsections, are interconnected to establish signal flow paths between single-ended input and single-ended output ports. This topology boasts integrated circuit practicality, since it exploits the inherent ability of a mature monolithic fabrication process to produce well-matched equivalent components. The two single-ended amplifiers in the subject figure are topologically identical, and they incorporate matched active devices that are biased at the same quiescent-operating points. Thus, amplifiers 1 and 2 have small-signal two-port equivalent circuits that are reflective of one another. In order to ensure balanced operation, the single-ended output ports of each amplifier are terminated to ground in equal load resistances R_L. Similarly, the Thévenin source resistances are equivalent. Observe that balance implies that the upper half of the circuit in Figure 2.38 is a mirror image of the lower half of the system schematic diagram. This interpretation begets the common reference to a balanced differential amplifier as a ''differential pair.''

The balance condition entails the following engineering constraints:

1. Under the case of differential-mode excitation, which implies $V_{S1} = -V_{S2} = V_{DI}/2$ and hence, $V_{CI} = 0$, the currents indicated in Figure 2.38 are such that $i_{i1} = -i_{i2}$, $i_{01} = -i_{02}$, and $i_1 = -i_2$.

Since the resistance, R_K, conducts a current equal to the sum of i_1 and i_2, $i_1 = -i_2$ clamps node k to signal ground potential for exclusively differential-mode inputs. Moreover, Equation 2.96a and b confirm $V_{01} = -V_{02} = A_D\ V_{DI}/2$, which produces a signal current through the differential load resistance R_{LL} of

$$\frac{V_{O1} - V_{O2}}{R_{LL}} = \frac{V_{O1}}{R_{LL}/2}$$

Since the single-ended response voltage, V_{O1}, is referred to signal ground, the midpoint of the differential load resistance is effectively grounded for differential inputs.

The foregoing disclosures imply the circuit diagram of Figure 2.39a, which is the so-called "differential-mode half-circuit equivalent" [11,12] of the differential amplifier. For a balanced differential pair driven by exclusively differential inputs, the branch currents, branch voltages, and node voltages computed from an analysis of the structure in Figure 2.39a are precisely the negative of the corresponding circuit variables in the remaining half of the system.

2. Under the case of common-mode inputs, which implies $V_{SI} = V_{S2} = V_{CI}$ and hence, $V_{DI} = 0$, the currents delineated in Figure 2.38 satisfy the constraints, $i_{i1} = i_{i2}$, $i_{01} = i_{02}$, and $i_1 = i_2$. Since the resistance, R_K, conducts a current equal to the sum of i_1 and i_2, $i_1 = i_2$ establishes a voltage at node k of

$$R_K(i_1 + i_2) = (2R_K)i_1$$

that is, the signal voltage developed at node k corresponds to an amplifier 1 current of i_1 flowing through a resistance whose value is twice R_K. Additionally, $V_{O1} = V_{O2} = V_{CO}$, which means that no signal current flows through R_{LL} under exclusively common-mode excitation.

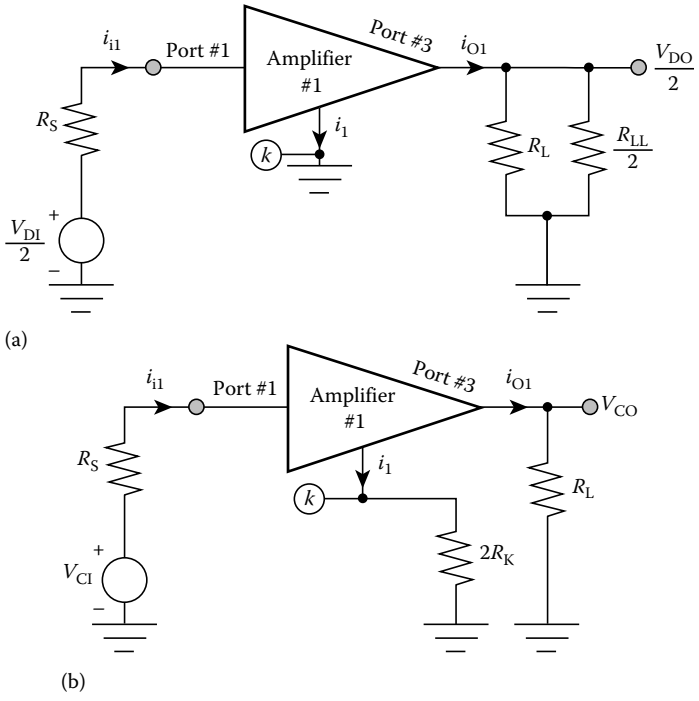


FIGURE 2.39 (a) Differential-mode half-circuit equivalent of the balanced differential amplifier. (b) Common-mode half-circuit equivalent of the balanced differential amplifier.

It follows that the "common-mode half-circuit equivalent" of the differential amplifier is as drawn in Figure 2.39b. For a balanced differential pair driven by exclusively common-mode input signals, the branch currents, branch voltages, and nodal voltages computed for the structure in Figure 2.39b are identical to the corresponding circuit variables in the other half of the circuit.

2.2.4.2 Thévenin Equivalent I/O Circuits

Because the two input ports of a differential amplifier electrically interact with one another, the Thévenin equivalent circuit seen by the two signal sources in Figure 2.37a is itself a two-port network. The branch elements of the Thévenin model are defined in terms of a "differential-mode input resistance," R_{DI}, and a "common-mode input resistance," R_{CI}.

Consider the test circuit of Figure 2.40a, which is configured to formulate the Thévenin equivalent input circuit. This structure is analogous to that of Figure 2.37a, except that the linear differential unit is presumed balanced. Furthermore, the original source circuits are supplanted by the test voltages, V_{tl} and V_{t2}, which establish the input port currents I_{t1} and I_{t2}. Because no signals other than V_{tl} and V_{t2} are applied to the differential pair, the Thévenin equivalent circuit seen between ports 1 and 2 is a resistance, say R_{XI}. Similarly, a second resistance, R_{XX}, is introduced to terminate port 1 to ground. System balance implies that a resistance of the same value terminates the second input port. The hypothesized Thévenin equivalent input circuit is given in Figure 2.40b.

The application of Kirchhoff's current and voltage laws to the model is Figure 2.40b produces

$$I_{t1} - \frac{V_{t1}}{R_{XX}} = \frac{V_{t2}}{R_{XX}} - I_{t2} \tag{2.98a}$$

$$V_{t1} - V_{t2} = R_{XI}\left(I_{t1} - \frac{V_{t1}}{R_{XX}}\right) \tag{2.98b}$$

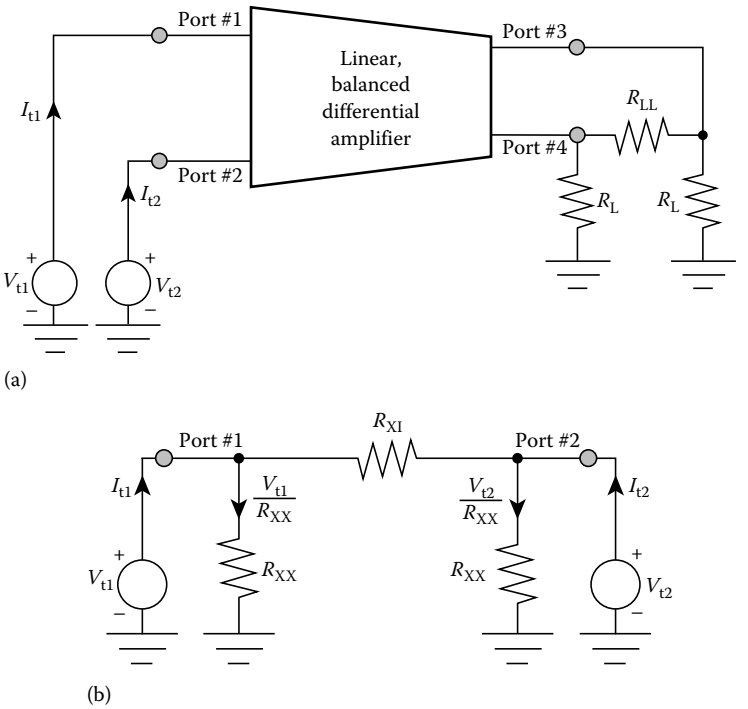


FIGURE 2.40 (a) Test circuit used to evaluate the Thévenin input equivalent circuit of a balanced differential amplifier. Note the connection of balanced loads at ports 3 and 4. (b) Hypothesized Thévenin equivalent input circuit.

If the test voltages, V_{t1} and V_{t2}, are decomposed into their differential (V_{Dt}) and common mode (V_{Ct}) components in accordance with

$$V_{t1} = V_{Ct} + \frac{V_{Dt}}{2} \tag{2.99a}$$

$$V_{t2} = V_{Ct} - \frac{V_{Dt}}{2} \tag{2.99b}$$

Equation 2.98a and b lead to

$$I_{t1} = \frac{V_{Ct}}{R_{XX}} + \frac{V_{Dt}}{R_{XI} \| (2R_{XX})} \tag{2.100a}$$

$$I_{t2} = \frac{V_{Ct}}{R_{XX}} - \frac{V_{Dt}}{R_{XI} \| (2R_{XX})} \tag{2.100b}$$

Equation 2.100a and b implicitly define the common-mode and differential-mode components of the currents I_{t1} and I_{t2} resulting from the test voltages V_{t1} and V_{t2}. Accordingly, the common-mode driving point input resistance, R_{CI}, is

$$R_{CI} = R_{XX} \tag{2.101}$$

On the other hand, the differential-mode driving point input resistance, R_{DI}, is

$$R_{DI} = R_{XI} \| (2R_{CI}) \tag{2.102}$$

where use has been made of Equations 2.100a, b, and 2.101. Note that the model resistance, R_{XI}, is related to the differential input resistance, R_{DI}, of the amplifier by

$$R_{XI} = \frac{2R_{CI}R_{DI}}{2R_{CI} - R_{DI}} \tag{2.103}$$

The test circuits for measuring the common-mode and the differential-mode driving point input resistances derive directly from Equation 2.100a and b. For a differential input test voltage, V_{Dt} of zero, V_{t1} and V_{t2} are identical to the common-mode input test voltage V_{Ct}, whence

$$I_{t1} = \frac{V_{t1}}{R_{XX}} = \frac{V_{t1}}{R_{CI}} \equiv I_{t2} \tag{2.104}$$

It follows that the circuit of Figure 2.41a is appropriate to the measurement (or calculation) of R_{CI} as the Ohm's law ratio of the common-mode test voltage to the resultant common-mode test current. Observe that half circuit analysis measures apply, wherein R_{CI} is the resistance seen looking into port 1, with port 3 terminated to ground in the resistance R_L. For differential testing, $V_{t1} = -V_{t2} = V_{Dt}/2$, whence

$$I_{t1} = \frac{2V_{t1}}{R_{XI} \| (2R_{CI})} = \frac{2V_{t1}}{R_{DI}} = -I_{t2} \tag{2.105}$$

The pertinent test cell is shown in Figure 2.41b. For half-circuit analysis, care should be exercised to recognize that the ratio of the test voltage, V_{t1}, to the corresponding test current, I_{t1}, is one-half of the driving point differential input resistance, R_{DI}. In addition, the subcircuit connecting port 2 to port 4

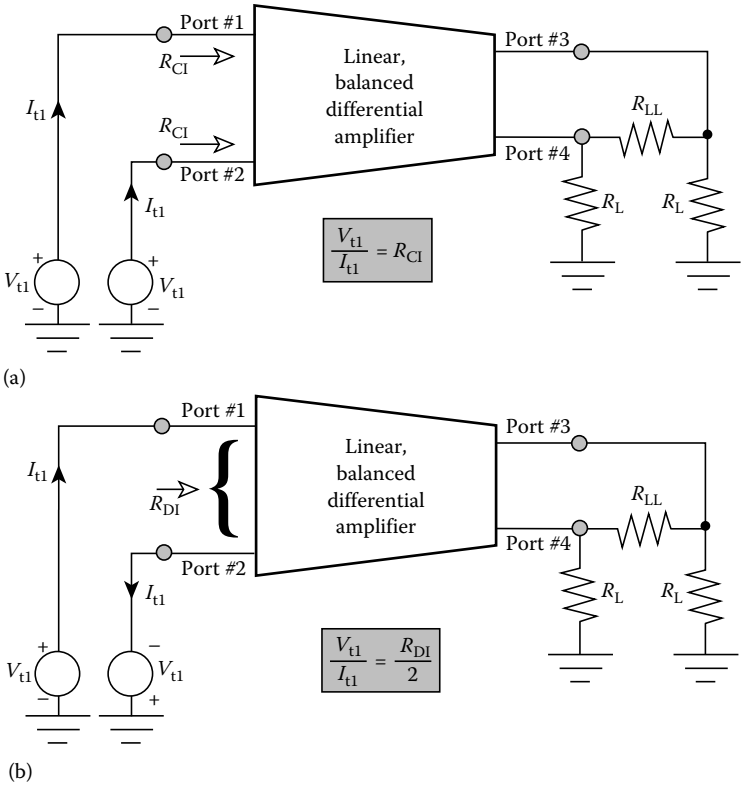


FIGURE 2.41 (a) Test circuit used to evaluate the driving point common-mode input resistance of a linear, balanced differential pair. (b) Test circuit used to evaluate the driving point differential-mode input resistance of a linear, balanced differential pair.

must be removed, and port 3 must be terminated to ground by the shunt interconnection of the resistances, R_L and $R_{LL}/2$.

Just as a Thévenin model can be constructed for the input ports of a balanced differential pair, a Thévenin equivalent circuit can be developed for the output ports. Under zero input conditions, this output model, which is presented in Figure 2.42, is topologically identical to the equivalent circuit in Figure 2.40b. In a fashion that reflects the computation of the resistance parameters for the input equivalent circuit, Figure 2.43a is the test circuit for evaluating the driving point common-mode output resistance, R_{CO}. Figure 2.43b is the test structure for calculating the driving point differential-mode output resistance, R_{DO}. Following Equation 2.103, R_{XO} in Figure 2.42 is given by

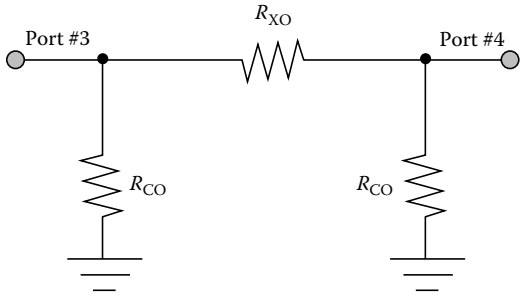


FIGURE 2.42 Thévenin equivalent output circuit of the balanced differential amplifier for the case of zero input signal excitation.

$$R_{XO} = \frac{2R_{CO}R_{DO}}{2R_{CO} - R_{DO}} \tag{2.106}$$

Two electrically interactive output ports are included in the differential system of Figure 2.37a. Thus, two Thévenin voltage V_{th1} and V_{th2},

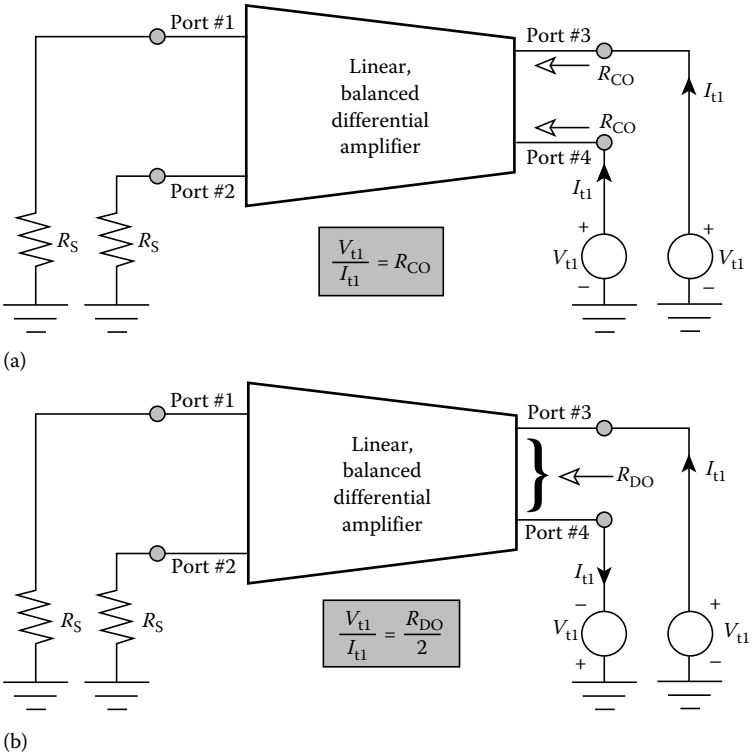


FIGURE 2.43 (a) Test circuit used to evaluate the driving point common-mode output resistance of a linear, balanced differential pair. (b) Test circuit used to evaluate the driving point differential-mode output resistance of a linear, balanced differential pair.

each of which is linearly dependent on the differential-mode and the common-mode components of the applied source signals must be evaluated. These Thévenin output responses derive from open-circuited load conditions, as indicated in Figure 2.44a. With $R_{S1} = R_{S2} = \Delta R_S$, balance prevails, and V_{th1} and V_{th2} are characterized by differential and common-mode components, analogous to the characterization of the terminated outputs, V_{O1} and V_{O2}.

The Thévenin voltages in question derive from an analysis of the equivalent circuit in Figure 2.44b, which represents the model of Figure 2.42 modified to account for nonzero source excitation. The proportionality constants, k_c and k_d, are related to the previously determined common-mode and differential-mode voltage gains. It is a simple matter to confirm that

$$V_{thl},\ V_{th2} = k_c V_{CI} \pm \left(\frac{k_d R_{XO}}{2R_{CO} + R_{XO}}\right)\frac{V_{DI}}{2} \tag{2.107}$$

The first term on the right-hand side of this relationship is the open-circuit common-mode output voltage, while the second term is the open-circuit differential-mode output voltage. It follows that k_c represents the open-circuit common-mode voltage gain, A_{CO}; that is

$$k_c = \lim_{\substack{R_L \to \infty \\ R_{LL} \to \infty}} (A_C) \triangleq A_{CO} \tag{2.108}$$

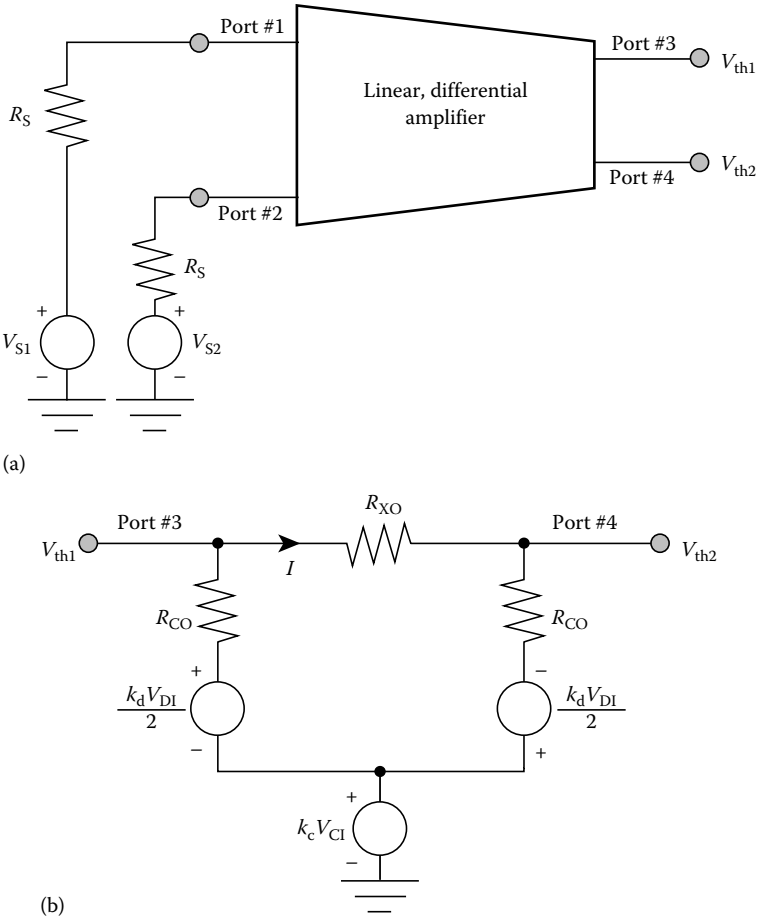


FIGURE 2.44 (a) System schematic diagram used to define the Thévenin voltages at the output ports of a balanced differential amplifier. (b) Thévenin equivalent circuit for the output ports of a balanced differential pair.

On the other hand, the open-circuit differential-mode gain, A_{DO}, is

$$\frac{k_d\ R_{XO}}{2R_{CO} + R_{XO}} = \lim_{\substack{R_L \to \infty \\ R_{LL} \to \infty}} (A_D) \triangleq A_{DO} \tag{2.109}$$

Using Equation 2.53, the Thévenin model parameter, k_d, in the last expression can be cast as

$$k_d = \left(\frac{2R_{CO}}{R_{DO}}\right) A_{DO} \tag{2.110}$$

Figure 2.45 summarizes the foregoing modeling results [13].

Example 2.4

Consider the balanced circuit of Figure 2.46 which is operated as a single-ended input–single-ended output amplifier. The input voltage signal, which is capacitively coupled to the base of transistor Q_1, is

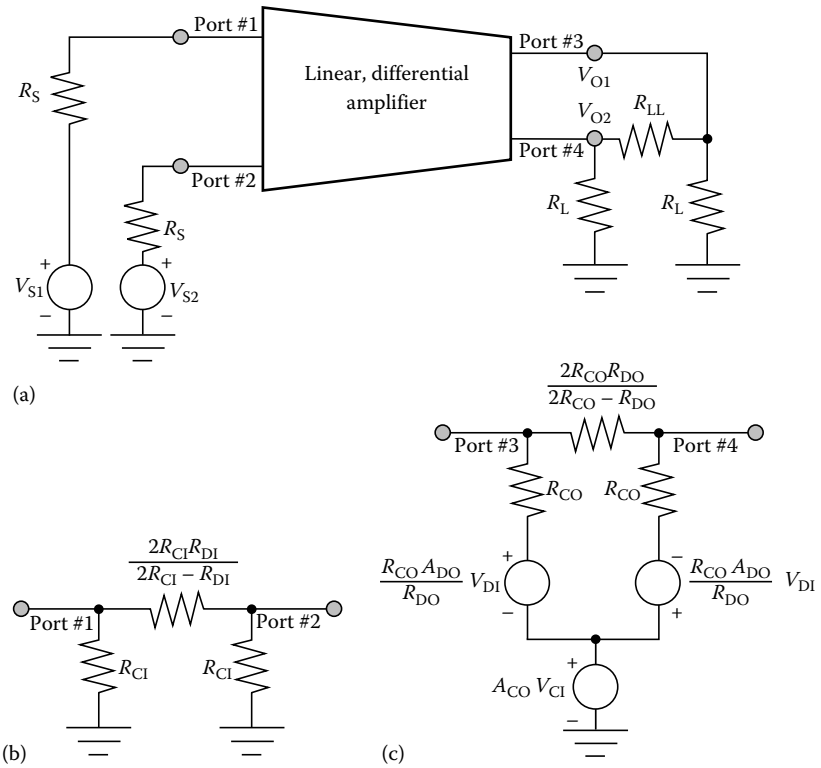


FIGURE 2.45 (a) System schematic diagram of a linear, balanced differential amplifier. (b) Thévenin equivalent input circuit. (c) Thévenin equivalent output circuit. The parameters, A_{DO} and A_{CO}, represent the open-circuit values of the differential- and common-mode voltage gains, respectively, of the balanced pair in (a).

represented as a Thévenin equivalent circuit consisting of the source voltage, V_S, in series with a source resistance, R_S. In order to preserve electrical balance, the base of transistor Q_2 is capacitively returned to ground through a resistance whose value is also equal to R_S. The capacitors can be presumed to act as AC short circuits over the signal frequency range of interest. For the parameters delineated in the inset to Figure 2.46, a computer-aided circuit simulation of the subject amplifier indicates that both transistors have the small-signal parameters $r_b = 33.5\ \Omega$, $r_\pi = 1.22\ k\Omega$, and $\beta = 81.1$. Determine the small-signal voltage gain $A_v = V_{OS}/V_S$, driving point input resistance R_{in}, and driving point output resistance R_{out}.

Solution

1. The AC schematic diagram of the differential-mode half circuit of the balanced amplifier in Figure 2.46 is shown in Figure 2.47a. In concert with earlier arguments, note that the junction of the two emitter degeneration resistances, R_{EE}, is grounded, as are the mid-point of the resistance, R_{LL}, and the node at which R_1, R_2, and the two resistances labeled R are incident. Using the bipolar model of Figure 2.16a, with r_o and r_e ignored, the voltage gain of this structure is the differential-mode voltage gain of the differential pair. Moreover, the driving point input resistance of the circuit at hand is one-half of the differential input resistance of the original pair, while its driving point output resistance is one-half of the differential-mode output resistance. Analysis confirms

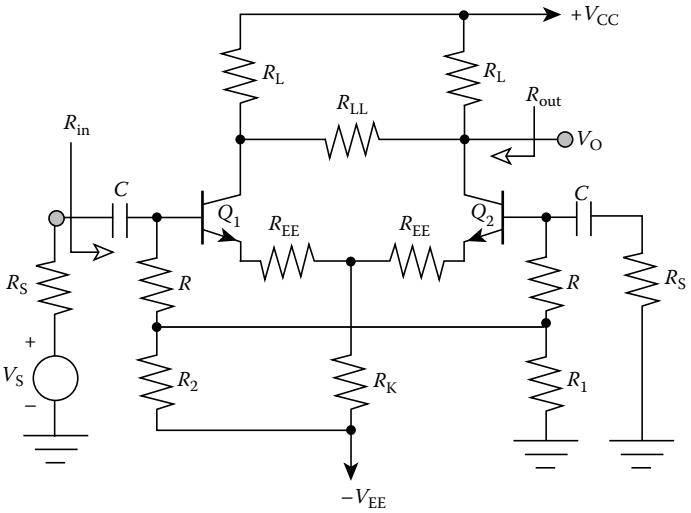


FIGURE 2.46 A balanced bipolar differential amplifier used in a single-ended input–single-ended output mode.

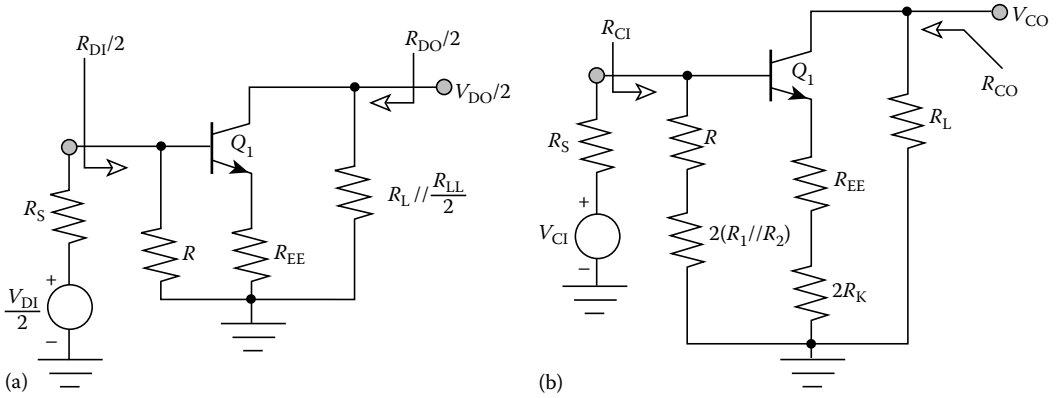


FIGURE 2.47 (a) Differential-mode half-circuit ac equivalent schematic of the differential amplifier shown in Figure 2.46. (b) Common-mode half-circuit ac equivalent schematic of the differential amplifier shown in Figure 2.46.

$$A_D = \frac{V_{DO}/2}{V_{DI}/2} = -\frac{\beta\left(\frac{R}{R+R_S}\right)\left(R_L \| \frac{R_{LL}}{2}\right)}{(R_S \| R) + r_b + r_\pi + (\beta+1)R_{EE}}$$

$$\frac{R_{DI}}{2} = R \| [r_b + r_\pi + (\beta+1)R_{EE}]$$

$$\frac{R_{DO}}{2} = R_L \| \frac{R_{LL}}{2}$$

Numerically, $A_D = -10.09$, $R_{DI} = 5971\ \Omega$, and $R_{DO} = 1875\ \Omega$.

2. The AC schematic diagram of the pertinent common-mode half circuit is given in Figure 2.47b. The input signal voltage is now the common-mode input voltage, V_{CI}, which produces the common-mode output response, V_{CO}. Using the bipolar model of Figure 2.16a, with r_o and r_e ignored, it is easily shown that

$$A_C = \frac{V_{CO}}{V_{CI}} = -\frac{\beta\left(\frac{R+2(R_1\|R_2)}{R+2(R_1\|R_2)+R_S}\right)R_L}{\{R_S\|[R+2(R_1\|R_2)]\}+r_b+r_\pi+(\beta+1)(R_{EE}+2R_K)}$$

$$R_{CI} = [R+2(R_1\|R_2)]\|[r_b+r_\pi+(\beta+1)(R_{EE}+2R_K)]$$

$$R_{CO} = R_L$$

   Numerically, $A_C = -923.3\ (10^{-3})$, $R_{CI} = 5493\ \Omega$, and $R_{CO} = 1500\ \Omega$. The common-mode rejection ratio $p = A_D/A_C$ 10.93 is small owing to the relatively small value of the resistance R_K.
3. As the output voltage is extracted at the collector of transistor Q_2, Equation 2.96b is the applicable equation for determining the output signal voltage, V_{OS}. With only a single source voltage, V_S, applied, the differential input voltage is V_S, and the common-mode input voltage is $V_S/2$. It follows that

$$V_{OS} = A_C V_{CI} - \left(\frac{A_D}{2}\right)V_{DI} = \left(\frac{A_C - A_D}{2}\right)V_S$$

   whence a voltage gain of

$$A_v = \frac{V_{OS}}{V_S} = \frac{A_C - A_D}{2}$$

   These analyses give $A_v = 4.583$.
4. In order to evaluate the driving point input and output resistances, the parameters, R_{XI} and R_{XO} must be calculated. From Equations 2.103 and 2.106, $R_{XI} = 13.08\ \text{k}\Omega$, and $R_{XO} = 5.0\ \Omega$.

The two-port model for calculating the driving point input resistance R_{in} is given in Figure 2.45b. Recall that a circuit resistance, whose value is numerically equal to the internal signal source resistance, R_S, is connected between ground and the node to which the base of transistor Q_2 is incident. By inspection,

$$R_{in} = R_{CI}\|[R_{XI}+(R_{CI}\|R_S)]$$

or $R_{in} = 3.87\ \text{k}\Omega$.

The output port model that emulates the driving point output resistance is given in Figure 2.45c. This model is analogous to that of Figure 2.45a, except that no external loads are connected between signal ground and the node to which the collector of transistor Q_1 is incident. Clearly,

$$R_{out} = R_{CO}\|(R_{XO}+R_{CO})$$

which produces $R_{out} = 1219\ \Omega$.

References

1. J. J. Ebers and J. L. Moll, Large-signal behavior of junction transistors, *Proc. IRE*, 42, 1761–1772, Dec. 1954.
2. H. K. Gummel and H. C. Poon, An integral charge-control model of bipolar transistors, *Bell System Tech. J.*, 49, 115–120, May–June 1970.

3. H. N. Ghosh, A distributed model of the junction transistor and its application in the prediction of the emitter-base diode characteristic, base impedance, and pulse response of the device, *IEEE Trans. Electron Devices*, ED-12, 513–531, Oct. 1965.
4. J. R. Hauser, The effects of distributed base potential on emitter current injection density and effective base resistance for stripe transistor geometries, *IEEE Trans. Electron Devices*, ED-11, 238–242, May 1965.
5. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, New York: Wiley, 1977, pp. 16–19.
6. C. T. Kirk, A theory of transistor cut-off frequency (f_T) at high current densities, *IEEE Trans. Electron Devices*, ED-9, 164–174, Mar. 1962.
7. J. M. Early, Effects of space-charge layer widening in junction transistors, *Proc. IRE*, 46, 1141–1152, Nov. 1952.
8. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, New York: Holt, Rinehart & Winston, 1987, pp. 52–57, 639–642.
9. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley Interscience, 1984, pp. 170–182.
10. G. R. Wilson, A monolithic junction FET-NPN operational amplifier, *IEEE J. Solid-State Circuits*, SC-3, 341–348, Dec. 1968.
11. E. J. Angelo, *Electronic Circuits*, New York: McGraw-Hill, 1970, Chapter 4.
12. A. B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley Interscience, 1984, pp. 217–224.
13. S. A. Witherspoon and J. Choma, Jr., The analysis of balanced linear differential circuits, *IEEE Trans. Educ.*, 38, 40–50, Feb. 1995.

2.3 MOSFET Biasing Circuits

David G. Haigh, Bill Redman-White, and Rahim Akbari-Dilmaghani

2.3.1 Introduction

CMOS technology is finding a very wide range of applications in analog and analog-digital mixed-mode circuit implementations in addition to its traditional role in digital circuits. In mixed-mode circuits compatibility of analog circuits with digital very large scale integration (VLSI) is important, particularly in cost-sensitive areas and situations where low power consumption is required. Such CMOS circuits require a range of biasing circuits and it is this topic that is the main subject of this section, although the subject is mentioned elsewhere in this text, where particular designs are covered.

The requirements for biasing circuits in CMOS circuit design can be divided into the requirements for voltage and for current sources. These sources can be further subdivided into two additional categories, high precision and noncritical. High-precision voltage or current sources are essential components in data converters, both analog-to-digital and digital-to-analog and the precision required depends on the overall target precision of the data converter. For a large number of bits, the precision required could be very great indeed and would need to be maintained over a specified temperature and supply voltage range and in the presence of on-chip, chip-to-chip and wafer-to-wafer component parameter variations. Precision sources are also required in other applications such as dc pedestals for video signals in video systems.

Noncritical voltage sources are generally required for setting up an internal analog ground or for biasing the gates of FETs in common-gate configuration, as in a cascode FET. In these cases, the sensitivity of overall circuit performance parameters to the bias voltage would generally not be high and moderate precision circuit techniques would be acceptable. The main considerations would be to

maintain correct operation, especially in terms of signal headroom, overall variations in process, power supply, and temperature conditions. The requirement for current sources for biasing CMOS analog and mixed-mode circuits is very considerable. The reason for this is that most circuits, such as an operational amplifier, consist of several stages, each of which requires biasing. In discrete circuits, the tendency is to use resistors for biasing, for reasons of cost and the poor sample-to-sample tolerances on discrete active device parameters. In integrated circuit implementation, on the other hand, relatively well-matched devices on the same chip are available and the use of current source biasing minimizes gain loss due to loading effects. Furthermore, using a multioutput current mirror to supply different parts of the circuit allows stabilization of the current against temperature and power supply voltage variations to be performed at one location only (on or off the chip) and the stabilized current can be distributed throughout the chip or subcircuit using current mirror circuits. This also produces significant immunity to localized power supply fluctuation and noise. Full or partial stabilization of bias currents with operating and environmental changes is desirable in order to minimize the range of operating current for which design must be specified, allowing higher design performance targets to be achieved.

In many cases, CMOS circuits have their power supplies derived from an off-chip bipolar regulator (with its own internal bandgap). In these situations, a reasonable voltage reference can sometimes be obtained from the power supply voltage via a potential divider. A voltage obtained in this way can be applied to an external low tolerance resistor to obtain a current reference of moderate precision. The value of realizing a reference on chip is that the cost of the external reference can be avoided. For example, in battery supplied equipment, a large degree of supply voltage immunity is required in the presence of a widely varying battery voltage. It is possible to use an on-chip voltage regulator or an on-chip, switched-mode power supply. In CMOS technology, high-precision voltage references are difficult to design, although a reference with good power supply rejection is possible. BiCMOS technology overcomes many of the problems experienced with CMOS technology since well-controlled bipolar devices for very high-precision references are available on-chip.

Many references to biasing appear in this text under the heading of the circuit concerned and only some general guidelines and principles, together with some example circuits will be given here. We begin this section on CMOS biasing circuits by considering the devices available for biasing in a CMOS-integrated circuit including parasitically realized bipolar junction devices. Some useful simplified models of these devices and a brief examination of the variability of the relevant model parameters will be presented. We then consider different types of references and biasing circuits. Since voltage and current references are closely interrelated, they are dealt with concurrently. The material is presented according to a gradually increasing level of sophistication and achievable precision, starting with simple circuits with only minor supply voltage, temperature, and process independence and leading to fully curvature-compensated bandgap references. This is followed by a consideration of references based on less usual devices that may not be available or usable in every process but that offer potentially attractive solutions. We then illustrate the application of some biasing techniques in the context of simple operational amplifier circuits. Finally, we consider the biasing of amplifiers for very low supply voltages, where rail-to-rail optimized performance is required, and dynamic biasing techniques. The topic of CMOS biasing circuits is sufficiently large to warrant an entire book and we include a list of references that will help provide the reader with more detailed information.

2.3.2 Device Types and Models for Biasing

2.3.2.1 Devices

The principal devices of CMOS technology are the enhancement-mode N-channel and P-channel MOSFET, which are shown schematically in Figure 2.48a and b for N-well and P-well technologies, respectively. Currently, N-well technology is more widely available than P-well. Depletion-mode devices

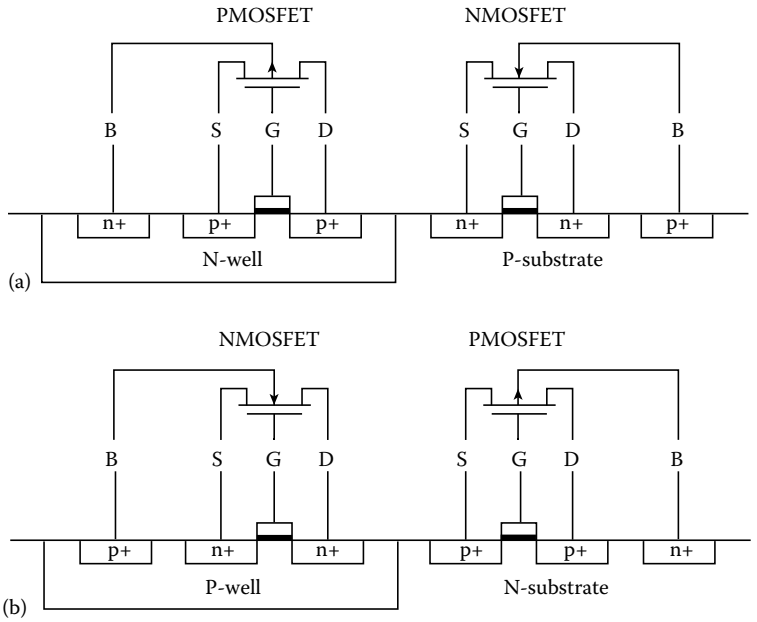


FIGURE 2.48 Realization of NMOS and PMOS FETs in CMOS technology. (a) N-well process. (b) P-well process.

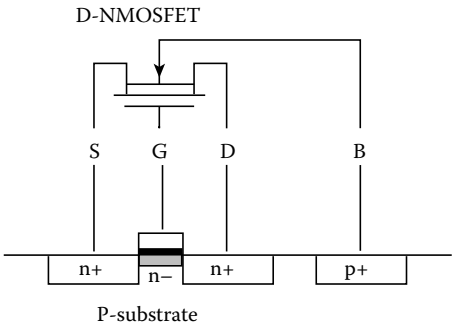


FIGURE 2.49 Realization of NMOS depletion-mode FET in N-well CMOS technology.

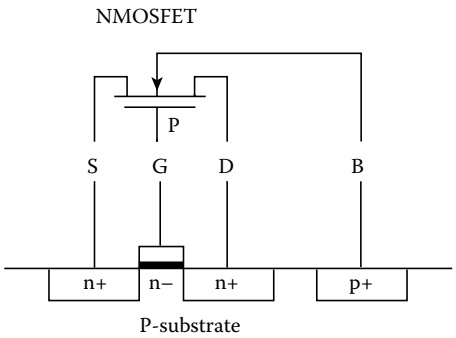


FIGURE 2.50 NMOS enhancement-mode FET with P-doped polysilicon gate.

are not routinely provided in CMOS (in contrast with NMOS), but they are available in some processes [23] and they can be used to realize reference circuits [4,5,23]. They are produced by an additional implementation under the gate region (N-type for NMOS and P-type for PMOS). The NMOS depletion-mode symbol is shown schematically in Figure 2.49.

In most processes, the MOSFET gate material is highly N-doped polysilicon. In some processes [8], P-doped gates are available, and these can be used to realize reference circuits. The symbol of an N-channel enhancement MOSFET with P-type doped base is shown in Figure 2.50.

In order to realize temperature and process desensitized biasing of CMOS circuits, the special properties of BJTs are advantageous. BJT devices can be realized in CMOS technology as parasitic devices with certain restrictions. Two classes of such device are available, namely, vertical and lateral. For the vertical device [13], the restrictions are that an N-well process can realize PNP devices with the collectors connected to the substrate (most negative supply rail) and that a P-well process can realize

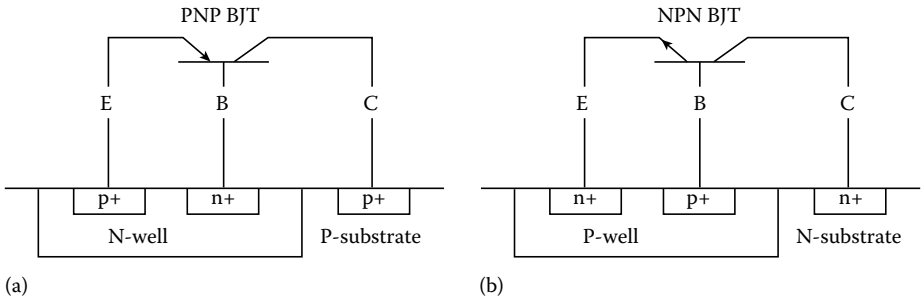


FIGURE 2.51 Realization of vertical BJT devices in CMOS technology. (a) N-well process. (b) P-well process.

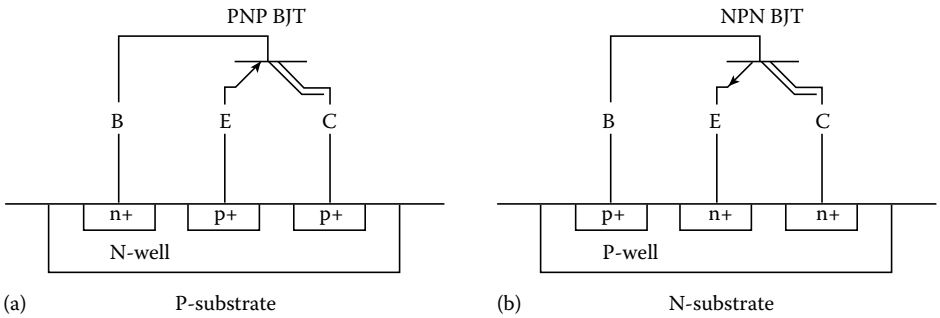


FIGURE 2.52 Realization of lateral BJT devices in CMOS technology. (a) N-well process. (b) P-well process.

NPN devices with the collectors connected to the substrate (most positive supply rail). The realization of these vertical devices in the case of N-well and P-well processes is shown schematically in Figure 2.51a and b, respectively. The devices can have typical current gains of around 100 and may have high leakage, and certain precautions have to be taken in the layout. One problem is that the control on the parameters of these devices in production is minimal. Nevertheless, such devices are adequate to realize moderate-to-high precision bias and reference circuits.

The restriction on the collector connections of the vertical BJT devices in CMOS technology is effectively removed in lateral BJT devices [16]. The realization of these devices in the case of N-well and P-well processes is illustrated in Figure 2.52a and b, respectively. It should be noted that β_∂ for lateral bipolar transistors is not related to α_∂ in the usual way due to substrate currents. For both vertical and lateral parasitic bipolar devices, it is often the case that the foundry will not provide detailed characterization and models, and also that the parameters of the devices will not be well controlled.

Apart from MOSFETs and BJTs, the remaining component needed to realize bias circuits are resistors and in some cases capacitors. In the case of precision bias circuits, the realized variable (voltage or current) would be dependent on resistor ratios rather than on absolute values. On-chip resistors may be realized using polysilicon, or as N- or P-well diffusion, as illustrated in Figure 2.53. Diffused resistors are sensitive to substrate potential and thus they are not suitable for precision potential dividers. In some cases, voltages or currents on a chip that are required to have high stability are referred to an off-chip highly stable and accurate discrete resistor. In some advanced processes, film resistors (usually nichrome) are available and they have excellent temperature stability, a wide range of values and can sometimes be laser-trimmed.

Capacitors, both on-chip and external, are used for decoupling bias and reference voltages and are especially valuable where low noise is critically important. On-chip capacitors also provide the basic components used for dynamic biasing.

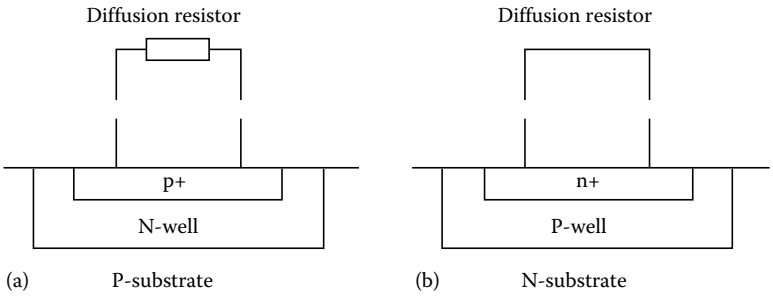


FIGURE 2.53 Realization of diffusion resistors in CMOS technology. (a) N-well process. (b) P-well process.

2.3.2.2 Device Models and Parameter Variability

The MOSFETs in Figure 2.48 may be very approximately described by [13,15]

$$I_d = \beta(V_{gs} - V_t)^2(1 + \lambda V_{ds}) \tag{2.111}$$

where

$$\beta = \mu C_{ox} W/L \tag{2.112}$$

and

$$V_t = V_{to} + \gamma\left[(2\phi_{fb} - V_{bs})^{0.5} - 2\phi_{fb}^{0.5}\right] \tag{2.113}$$

with

$$\phi_{fb} = \left(\frac{kT}{q}\right)\ln\left(\frac{N_{sub}}{n_i}\right) \tag{2.114}$$

and

$$\mu \propto T^{-\eta} \tag{2.115}$$

The parameters in Equations 2.111 through 2.115 are defined in Table 2.1. As a result of Equations 2.111 through 2.115, MOSFET parameters show considerable temperature dependence. In particular, threshold voltage varies with temperature according to

$$\frac{\partial V_t}{\partial T} = \frac{2}{T}\left(\phi_F - \frac{E_{go}}{2q}\right)\left(1 + \frac{\gamma}{2(2\phi_{fb} - V_{bs})}\right) \tag{2.116}$$

which amounts typically to about −2 mV/°C (Ego is the bandgap of silicon at 0 K) [15]. Transconductance varies according to

$$\frac{\partial g_m}{\partial T} = \frac{g_m}{2}\left(-\frac{\eta}{T} + \frac{1}{I_d}\frac{\partial I_d}{\partial T}\right) \tag{2.117}$$

The temperature and process dependencies of MOS devices have led to the exploitation of some properties of combinations of less usual MOS devices. It has been observed that the difference between

TABLE 2.1 Device Parameters

Symbols	Parameters
I_d	Drain current
β	Transconductance parameter
V_{gs}	Gate–source voltage
V_t	Threshold voltage
λ	Output conductance parameter
V_{ds}	Drain–source voltage
μ	Mobility
C_{ox}	Oxide capacitance per unit area
W	Gate width
L	Gate length
V_{to}	Zero substrate bias threshold voltage
γ	Body factor
ϕ_{fb}	Fermi potential
V_{bs}	Substrate–source voltage
K	Boltzmann's constant
T	Absolute temperature in K
Q	Electronic charge
N_{sub}	Substrate doping density
n_i	Intrinsic carrier density
$\acute{\eta}$	Mobility temperature coefficient
V_{FB}	Flat-band voltage
Q_{SS}	Surface charge per unit area
ϕ_P	Bulk potential
Q_d	Charge per unit area in inversion layer
ϕ_{bi}	Channel to substrate built-in potential
Q_i	Implanted charge per unit area
C_{impl}	Capacitance defined by implanted channel depth
ϕ_G	Bandgap voltage for silicon
ϕ_{GO}	Bandgap voltage for silicon at 0 K

the threshold voltages of an enhancement- and depletion-mode FET pair is relatively temperature independent. The threshold voltages for enhancement- and depletion-mode MOSFETs may be written

$$V_{t(enh)} = V_{FB} - \frac{Q_{SS}}{C_{ox}} + 2|\phi_P| + \frac{|Q_d|}{C_{ox}} \tag{2.118}$$

$$V_{t(depl)} = V_{FB} - \frac{Q_{SS}}{C_{ox}} + \phi_{bi} + (|Q_d| - |Q_i|)\left(\frac{1}{C_{ox}} + \frac{1}{C_{impl}}\right) \tag{2.119}$$

where the meaning of the parameters is given in Table 2.1 [4]. Many of the parameters in Equations 2.118 and 2.119 show considerable temperature dependence. The difference between the threshold voltages is given by

$$\begin{aligned} V_{t(diff)} &= V_{t(enh)} - V_{t(depl)} \\ &= 2|\phi_P| - \phi_{bi} - |Q_i|\left(\frac{1}{C_{ox}} + \frac{1}{C_{impl}}\right) \end{aligned} \tag{2.120}$$

assuming that $1/C_{ox} \gg 1/C_{impl}$, which is true in practice. In practice, it is also the case that $2|\phi_{P|} \approx \phi_{bi}$. Since the implanted charge Q_i is controllable and independent of temperature to first order [4], the threshold voltage difference exhibits temperature independence to first order.

The difference between the threshold voltages of two N-channel FETs with polysilicon gates of opposite doping polarity (P and N) also shows relative insensitivity to temperature variations [8]. To a first approximation, the threshold voltage difference is given by

$$\Delta V_G = \phi_G = 1.12 \text{ V (room temperature)} \tag{2.121}$$

which is the bandgap voltage for silicon [8]. A more detailed analysis [1] gives

$$\Delta V_G(T) = \phi_{GO} - \frac{\alpha T^2}{T + \beta} \tag{2.122}$$

where $\alpha = 7.02 \times 10^{-4}$ V/K, $\beta = 1109$ K and the meaning of the remaining parameters is given in Table 2.1. In practice, the degree of temperature independence obtained provides useful reference circuits [8]. The exploitation of both the enhancement–depletion FET threshold difference and the N–P-doped polysilicon gate threshold voltage difference for the design of references will be described later.

The strong temperature dependence of conventional MOS device parameters means that for stable biasing circuits, MOS devices are mainly useful where the critical variable depends on a ratio of parameters of similar devices. Even in this case, the matching is not as good as for bipolar devices. The matching of the gate–source voltages of two similar devices with nominally identical drain currents is inversely proportional to the square root of the gate area and is typically of the order of 10 mV, which limits the minimum offset voltage of a CMOS op-amp. Since op-amps form key components in many voltage and current reference circuits, this is a serious limitation.

In contrast to the rather complex dependence of MOS device parameters with temperature, the situation in the case of BJT devices is relatively straightforward [15,17,26]. The BJT may be described by

$$I_c = I_s e^{V_{be} q/kT} \tag{2.123}$$

where the additional parameters are defined in Table 2.1. Equation 2.123 may alternatively be written

$$V_{be} = \frac{kT}{q} \ln \frac{I_c}{I_s} \tag{2.124}$$

For two devices with an emitter area ratio of A

$$A = \frac{I_{s1}}{I_{s2}} \tag{2.125}$$

We have

$$\begin{aligned} \Delta V_{be} &= V_{be1} - V_{be2} \\ &= \frac{kT}{q} \ln \frac{I_{c1}}{I_{c2}} \frac{1}{A} \\ &= V_T \ln \frac{1}{A} \quad (\text{for } I_{c1} = I_{c2}) \end{aligned} \tag{2.126}$$

Thus, the difference between the V_{be}s of two BJTs with different current densities is proportional to the thermal voltage V_T, which is proportional to absolute temperature (PTAT). The positive temperature coefficient of V_T can be effectively used to cancel the negative temperature coefficient of V_{be} [13]. This is referred to as the bandgap principle.

Resistors are also key elements in MOS biasing circuits and they may be realized using diffusion, polysilicon and, in some advanced processes, using film techniques. Polysilicon and diffused resistors suffer from a high temperature coefficient that is positive for diffusion. The resistivity of gate polysilicon is typically rather low at about 20 Ω/square and its initial value tolerance is quite high. Film resistors have a very low temperature coefficient.

2.3.3 Voltage and Current Reference and Bias Circuits

2.3.3.1 Supply-Voltage-Referenced Voltage and Current References

When the supply voltages to a chip are well-regulated off-chip, then a voltage reference, acceptable in some cases, can be realized by a simple potential divider from the power supply voltage, as shown in Figure 2.54. An external decoupling capacitor may be used if needed and the voltage dividing elements may be resistors (Figure 2.54a) or MOSFETs (Figure 2.54b). If the power supply voltages are well-controlled off-chip, using an external regulator circuit, then a simple current reference can be realized by the arrangement in Figure 2.55, where the reference current is defined by applying a well-defined fraction of the controlled supply voltage to a well-controlled external resistor.

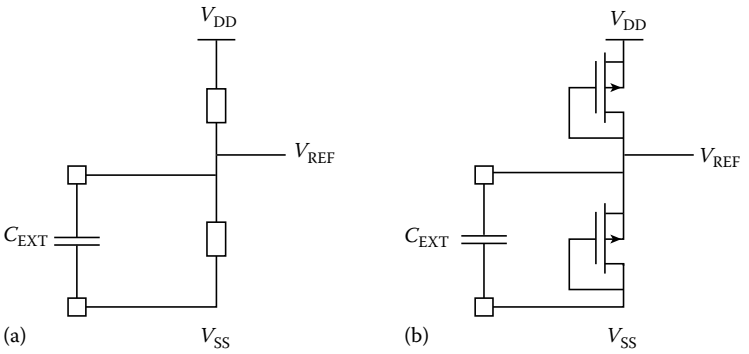


FIGURE 2.54 Voltage reference obtained via potential divider from regulated power supply: (a) using resistors; (b) using MOSFETs.

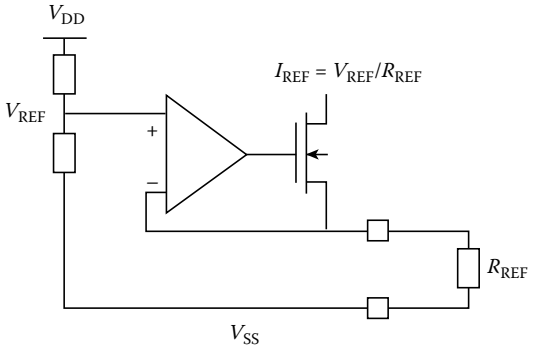


FIGURE 2.55 Current source realized using potential divider.

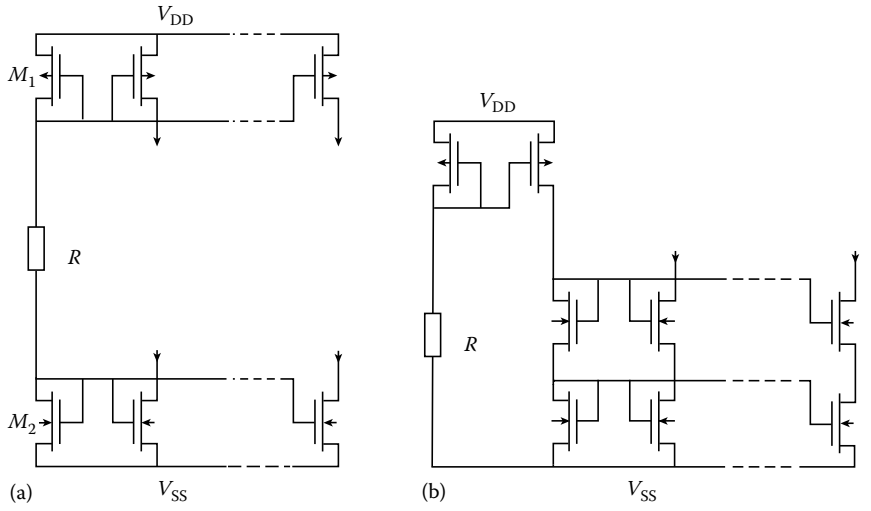


FIGURE 2.56 Resistor/current mirror bias circuits: (a) simple; (b) cascode.

A very simple biasing circuit proving multiple current sources and sinks, as required in CMOS analog signal-processing circuits, is shown in Figure 2.56a [13]. Assuming large FET W/L ratios, the voltage across the resistor is approximately $V_{dd} - V_{ss} - 2V_t$. The current in R is mirrored in the output MOSFETs where the W/L ratios may be chosen to provide required current magnitudes. The resistor R may have to be realized off-chip as a precision film component with narrow tolerance and small temperature coefficient. In practice, the voltage across the diode-connected MOSFETs M_1 and M_2 will be greater than V_t and have some dependence on MOSFET β as well as V_t.

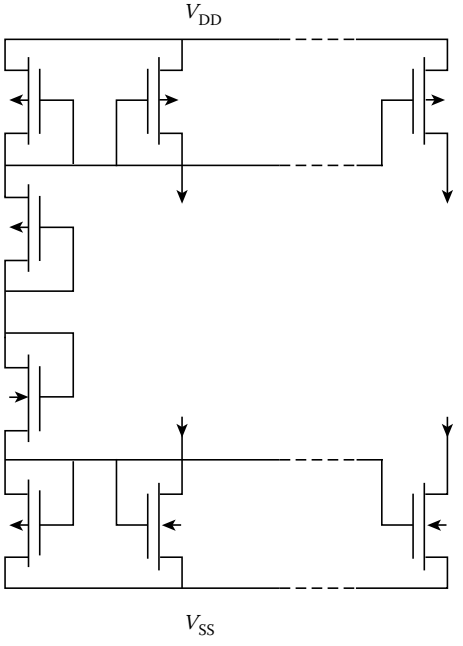


FIGURE 2.57 FET/current mirror bias circuit.

For the circuit in Figure 2.56a, the source conductance is equal to the MOSFET g_{ds}, which might not be sufficiently low for some applications. This disadvantage can be overcome by introducing cascode FETs as in Figure 2.56b. Nevertheless, the supply voltage dependence of the circuit in Figure 2.56 remains.

For uncritical applications where the current source is to be realized entirely on-chip, the resistor R in Figure 2.56a may be replaced by a chain of diode-connected P- and N-channel MOSFETs, as shown in Figure 2.57 [13]. The number of these devices and their W/L ratios may be chosen according to the supply voltage and the value of the current to be realized. Since the diode-connected MOSFETs are effectively realizing the resistor in the basic current source of Figure 2.56a, the power supply voltage dependence of the current remains. In addition, the effective resistance realized depends on MOSFET β and V_t, both of which have large tolerances and high temperature coefficients.

2.3.3.2 MOSFET Threshold Voltage-Based References

A current reference with reduced power supply voltage dependence is shown in Figure 2.58 [13]. By choosing W/L for M_1 to be large, the gate–source voltage of M_1 can be made close to the device threshold voltage V_t. Since the gate–source voltage of M_1 appears across the resistor R, the current in R is approximately V_t/R, which is ideally independent of power supply voltage. The W/L ratios of MOSFETs M_3 and M_4 are chosen to define a fixed ratio for the currents in M_1 and R. The combination of MOSFETs M_2, M_3, and M_4 constitute a positive feedback loop and it is important to choose the W/L ratios so that the loop gain is less than unity to avoid oscillation. Many reference circuits have a stable state with all currents zero. In such cases, it is necessary to provide a start-up circuit [13] to prevent the reference circuit locking into an undesired operating point. A source follower can be introduced at the gate of M_2 such that in this condition a current is injected into the circuit. The added components must be such that in the normal operating point, they are switched off and therefore do not influence operation. In practice, the currents realized by the circuit in Figure 2.58 will have some supply voltage dependence due to channel length modulation in the MOSFETs. This effect can be reduced by introducing cascode devices appropriately. Although the currents realized can be made substantially independent of supply voltage, the dependence on resistance R remains. For high precision and temperature independence, R may need to be realized as an off-chip film resistor. It must be borne in mind that the device threshold voltage on which the current depends is rather variable (typically 0.5–0.8 V) and also rather temperature dependent. Solution of this problem requires the introduction of alternative techniques based on BJT or unconventional CMOS devices, which will be discussed.

An alternative circuit to that in Figure 2.58 is shown in Figure 2.59 [25]. This circuit regulates the MOSFET drain currents with the result that MOSFET transconductance is proportional to $1/R$. This circuit also relies on positive feedback and care must be taken in the design to avoid instability. It has been shown in Ref. [21] that a practical stable design results from the choice $(W/L)_4 = (W/L)_3$ and $(W/L)_2 = 4(W/L)_1$, giving $g_m = 1/R$. In processes where R can be realized as a film resistor on-chip, this circuit can stabilize transconductances to within 3% over a 100°C temperature range [25].

2.3.3.3 BJT V_{be}-Based References

The problem that MOSFET threshold voltage is not very well controlled from chip sample to chip sample leads to the idea of using the V_{be} of a parasitic bipolar transistor [13]. Such a V_{be}-referenced circuit is

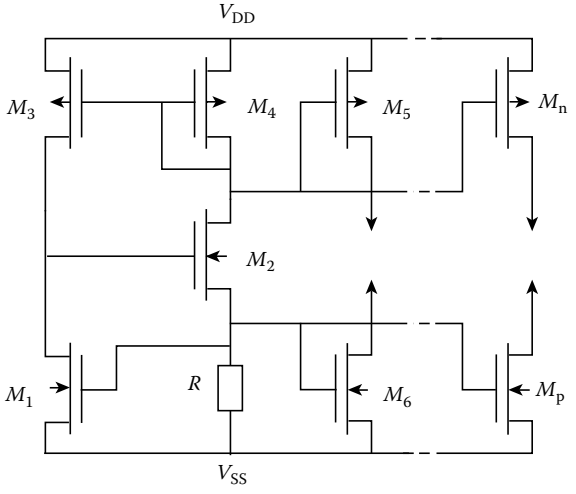


FIGURE 2.58 V_t-referenced current bias circuit.

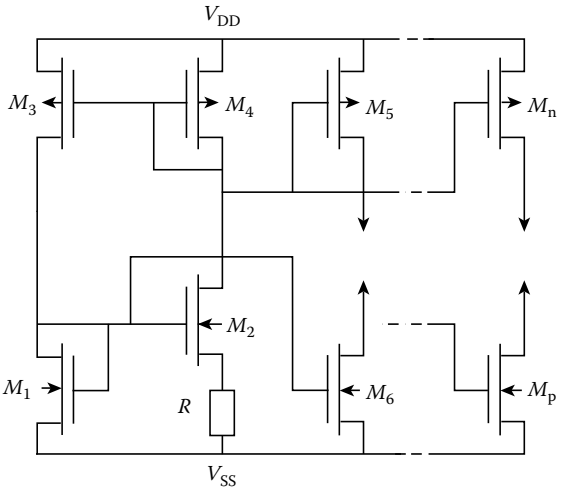


FIGURE 2.59 g_m-R tracking current reference circuit.

shown in Figure 2.60 for the case of an N-well process, where the BJT is PNP [13]. The W/L ratios for M_1 and M_2 are made large so that the V_{be} of T_1 appears substantially across R. In order to achieve high precision and temperature independence, R would need to be an off-chip film resistor. However, the V_{be} has a process-dependent tolerance of about 5% and a dependence with temperature of about -2 mV/°C.

2.3.3.4 BJT V_T-Based References

The temperature dependence of V_{be} in the circuit of Figure 2.60 can be overcome in the V_T-based circuit of Figure 2.61 [13]. The emitter areas of the BJTs Q_1 and Q_2 are scaled in the ratio 1:n and the MOS current mirrors force the emitter currents to be equal. The difference between the V_{be} of Q_1 and Q_2 given by Equation 2.121 appears across the resistor R, hence defining the current. The positive temperature coefficient of V_T can be used to counteract the positive temperature coefficient of the resistor R to obtain

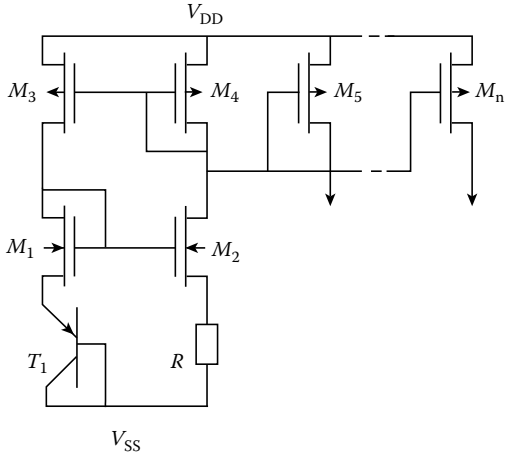


FIGURE 2.60 V_{be}-referenced current bias circuit.

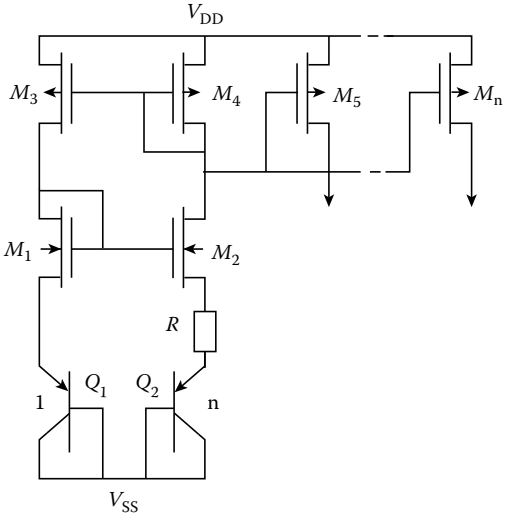


FIGURE 2.61 V_T-referenced current bias circuit.

a stable current. In the circuit of Figure 2.61, M_1 and M_2 must have large W/L to minimize the effect of MOSFET process variability. Also, cascoding of the current mirrors may be required to reduce the effect of channel length modulation.

2.3.3.5 Bandgap References

Precision voltage sources are key requirements for the realization of precision data converters and have received much attention [2,3,7,10,12–14,20,22]. The requirements for high precision and very low temperature and supply voltage dependence have led to the development of the bandgap principle [2,3,7,10,13]. The bandgap principle was originally developed for bipolar technology and a typical architecture is shown in Figure 2.62a. As described previously, the difference between the V_{be}s of two BJTs with different current densities is proportional to the thermal voltage V_T and is PTAT. In Figure 2.62a, the difference between the V_{be}s appears across R_3 and in scaled form across R_2. Thus the output voltage is equal to the V_{be} of Q_1 plus the scaled version of ΔV_{be}. Thus R_2/R_3 may be chosen so that the opposite temperature coefficients of V_{be} and ΔV_{be} cancel. The ratio R_1/R_2 determines the ratio of the currents in Q_1 and Q_2. The circuit in Figure 2.62a is incompatible with implementation using CMOS technology with vertical parasitic bipolar devices because the collectors are not grounded. This can be overcome using the architecture in Figure 2.62b. However, there is the further problem that the offset voltage of the operational amplifier is multiplied by the internal gain of the feedback loop and added to the output. Offset is worse for CMOS than for bipolar operational amplifiers. This problem has been overcome in various ways. In Refs. [12,22], use is made of a discrete time offset compensated differential amplifier, which can have very low offset. Another approach is to make use of lateral bipolar devices, which do not suffer from the topological restrictions of their vertical counterparts [16]. Thus the architecture of Figure 2.62a or an equivalent topology may be implemented.

A typical example of a current reference based on a bandgap voltage reference is shown in Figure 2.63 [13]. The current in the resistor xR is V_T-referenced as in Figure 2.61 and therefore has a negative temperature coefficient (the BJTs are vertical parasitic devices). This current is converted to a voltage and weighted by the resistor xR before being added to the V_{be} of Q_3, which has a negative temperature coefficient. The parameter x is chosen to obtain an overall zero temperature coefficient for the output current, which is given by V_{REF}/R_0. Clearly, R_0 needs to be a high-precision resistor and could be external to the chip. The current mirrors need to be very well matched as any offset is amplified. The operational amplifier needs to have a low offset voltage since this is added to the reference voltage. Further current mirroring may be used to change the sign of the current or to increase the permissible range of the output voltage, referred to as compliance.

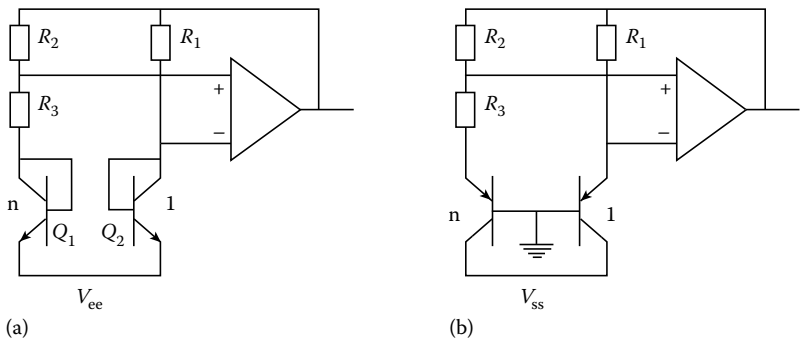


FIGURE 2.62 Basic bandgap circuits. (a) Classical bandgap circuit. (b) Modified form with grounded-collector PNP transistors, assuming an N-well process.

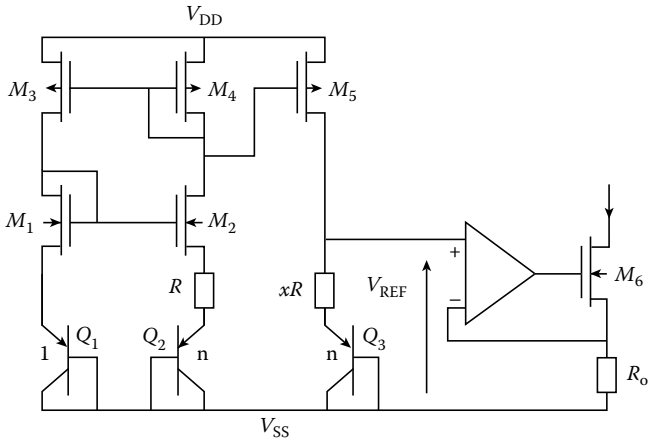


FIGURE 2.63 Bandgap current bias circuit.

2.3.3.6 Curvature-Compensated Bandgap References

The bandgap reference principle can provide a zero temperature coefficient at a single temperature, leaving a temperature dependence that is dominated by a second-order temperature dependence. Very sophisticated techniques have been developed [12,24] to eliminate this second-order dependence to leave a typically much smaller third-order dependence. This technique is referred to as curvature compensation. An example of a curvature-compensated current reference [24] is shown in Figure 2.64. This circuit can achieve precisions of the order of 5 ppm/°C for supply voltages over 5–15 V.

2.3.3.7 Discrete Time Bandgap References

The voltage reference in Ref. [12] provides curvature compensation and achieves a drift of the order of 13 ppm/°C over the commercial temperature range. The design is based on a comprehensive analysis of nonideal effects in the basic bandgap circuit including finite β and base resistance of the bipolar devices, operational amplifier offset, and bias current variation. This leads to a system involving a very low offset switched capacitor differential amplifier and a system of injecting a differential pair of currents into the

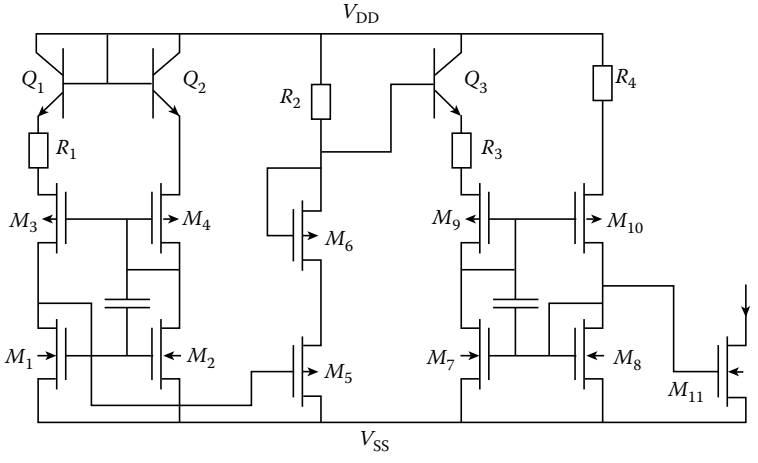


FIGURE 2.64 Curvature-compensated current bias circuit.

emitters of the bipolar devices to provide curvature compensation. The offset cancellation of the switched capacitor differential amplifier is accompanied by techniques for cancellation of the effect of base currents and base resistance in the bipolar devices. Base currents can sometimes be a severe problem due to the available current gains of parasitic bipolar devices. The design is fully compatible with a digital IC process and achieves an equivalent precision of 12 b. Room temperature trims are necessary for a zero temperature coefficient and for curvature compensation. Although low-frequency power supply rejection is good, it falls with increasing frequency.

In Ref. [22], a floating voltage reference for signal-processing applications with a good power supply rejection ratio of at least 85 dB maintained up to 500 MHz is realized. Over a temperature range of -40 to $+85$°C, voltage dependence is 40 ppm/°C and supply voltage dependence $\pm 5\%$. The circuit has the important advantage that trimming is not required.

2.3.4 Voltage and Current References Based on Less Usual Devices

2.3.4.1 Use of Device in Subthreshold Region

An alternative approach to current reference making use of MOSFETs in the subthreshold region is reported in Ref. [19]. The principle of the approach is illustrated in Figure 2.65a, where for thermal stabilization the voltage source is required to be PTAT. The PTAT voltage source is realized as a cascade of 5 of the PTAT voltage sources shown in Figure 2.65b, which rely on the subthreshold mode operation of the devices. In practice, cascoding of the current mirrors and current sources is required and a start-up circuit is needed. A current accuracy of 3% with temperature stability of 3% over 0–80°C can be achieved with this approach [19].

2.3.4.2 Voltage Reference Circuits Using Lateral Bipolar Devices

The circuit diagram of a bandgap voltage reference making use of lateral bipolar devices is shown in Figure 2.66 [16]. The circuit is designed to be insensitive to low β and α of the bipolar devices. It is also insensitive to offsets and mismatch. A single trim at room temperature is required and a high power supply rejection ratio, at least at low frequencies, is obtained. The output voltage is stable to within 2 mV over a wide temperature range.

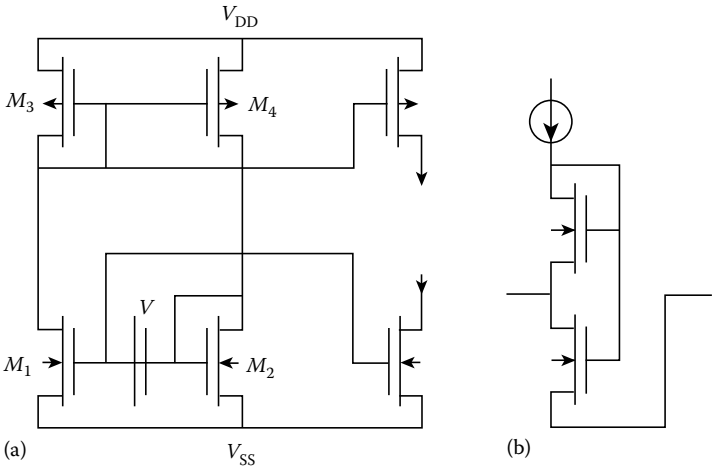


FIGURE 2.65 MOS current bias circuit based on weak inversion operation: (a) basic circuit; (b) voltage source cell.

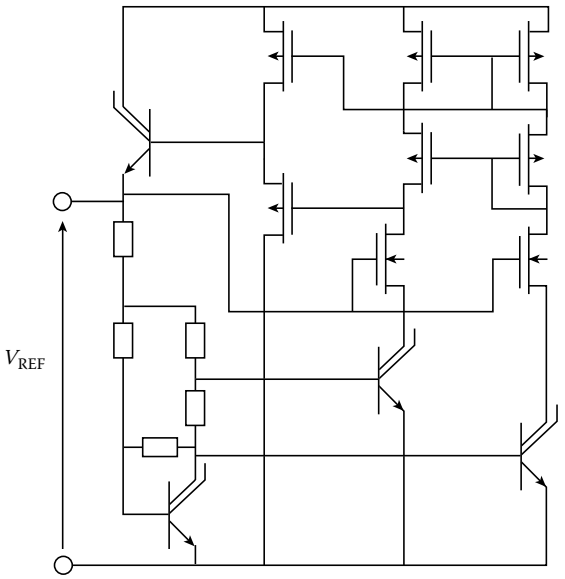


FIGURE 2.66 Voltage reference using lateral bipolar devices.

2.3.4.3 Voltage References Based on Enhancement and Depletion-Mode Threshold Voltage Difference

The topology restrictions and imperfections of the BJT devices available in CMOS technology have led to the development of alternative techniques for designing references without needing bipolar devices. In one technique, the fact that the difference between the threshold voltage of depletion-mode and enhancement-mode devices is relatively temperature independent has been exploited [4,5,23].

The section on device models and parameter variability demonstrated that the difference in threshold voltages of an enhancement- and depletion-mode MOSFET is relatively insensitive to temperature. Since the threshold voltage of the depletion-mode device is negative, this approach leads to a reference voltage of the order of 2 V, which is higher than the bandgap voltage, and this can be an advantage. A basic scheme for exploiting this principle for a voltage reference is shown in Figure 2.67. The op amp adjusts the gate voltage of the enhancement-mode FET to keep the drain voltages the same and the resistor values can be used to adjust the ratio of the currents in the two FETs. The operational amplifier may be implemented at device level [4]. The gate voltage of the depletion-mode FET may be connected to the output of a buffer amplifier whose output voltage may be adjusted using polysilicon fuses to typically 3.15 ± 0.02 V [5].

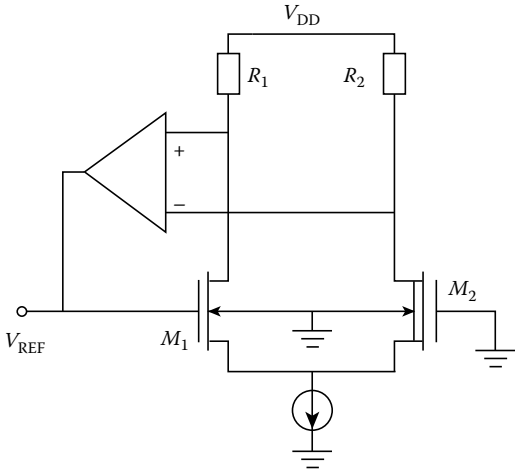


FIGURE 2.67 Basic reference based on enhancement–depletion threshold difference.

Higher reference voltages may be obtained by replicating the enhancement- and depletion-mode MOSFETs. In Figure 2.68, the reference voltage is the difference between the threshold voltages of the three enhancement-mode MOSFETs M_1–M_3 and the three depletion-mode MOSFETs

M_4–M_6 [23]. M_7–M_{10} are providing the necessary bias currents. In Ref. [23], the variation of V_{REF} with temperature is 1.5 mV/°C, which is useful for many biasing situations and the reference voltage is of the order of 3 V in spite of low threshold voltage devices.

2.3.5 Voltage References Based on N- and P-Doped Polysilicon Gate Threshold

2.3.5.1 Voltage Difference

In CMOS technology, the gate material is usually polysilicon with N-type doping. In some processes, selective doping to provide P-type doping of the polysilicon gate is also available and the presence of both types of doping has been exploited for reference circuit design [8].

The basic principle of a voltage reference based on the difference between the threshold voltages of N- and P-doped polysilicon gate MOSFETs is illustrated in Figure 2.69. M_1 has a P-doped gate and a higher threshold voltage than M_2. M_1 and M_2 are in different P-wells but have the same effective dimensions and bias currents.

A full transistor-level implementation of the basic circuit in Figure 2.69 is shown in Figure 2.70 [8]. M_1 and M_2 are the reference MOSFETs. M_3 has a very long channel and its current is the same as that in M_1 by virtue of the current mirror M_4: M_5. Thus the current in M_1 adjusts itself to the crosspoint of the characteristics of M_1 and M_3. M_7, M_8, and M_9 ensure that the currents in M_1 and M_2 are identical. M_6 is a start-up device. When the power supply is switched on, M_6 comes on but within 1 ms is switched off by the reverse leakage resistance of the polysilicon diode D. In Ref. [8], M_1 and M_2 can have a W/L of 100/20 μm; supply voltage sensitivity of $<10^{-3}$ is obtained for V_{DD} between 2 and 9 V [8]. Digital tuning using polysilicon fuses to reference voltages other than the polysilicon gate work function difference can be obtained and a further level of temperature compensation applied [8].

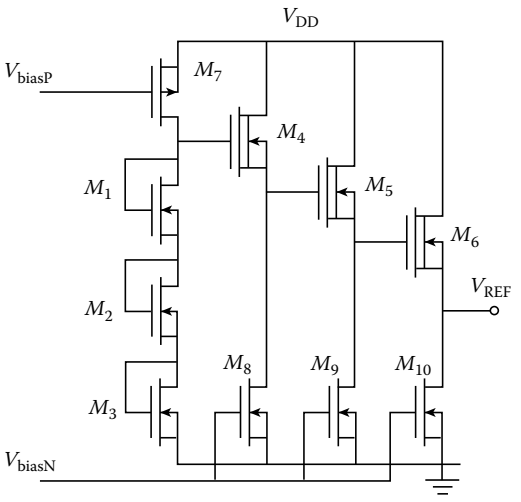


FIGURE 2.68 High-output enhancement–depletion threshold difference reference.

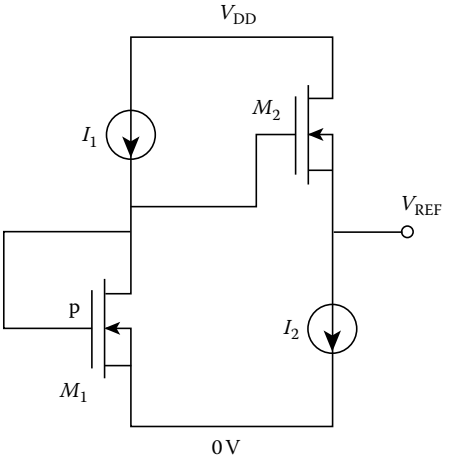


FIGURE 2.69 Basic reference based on polysilicon work function difference.

2.3.6 Biasing of Simple Amplifiers and Other Circuits

2.3.6.1 Simple Amplifiers

In traditional two-stage amplifier design, the bias for the whole circuit is easily set up from one reference current and no critical voltage differences have to be set up. It is only necessary to ensure that the operating currents, and hence the transconductance, of each device have a required value. A typical example of the biasing of a two-stage amplifier is shown in Figure 2.71 [13]. The ratio of currents between

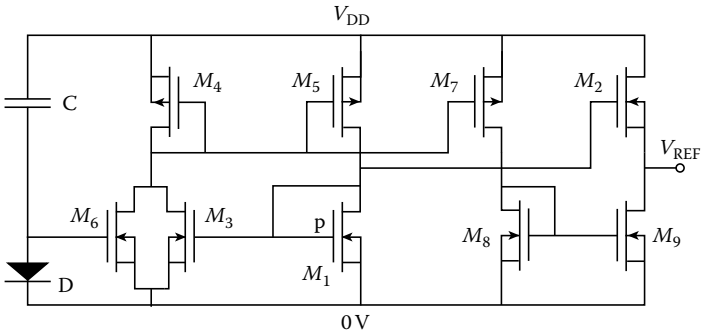


FIGURE 2.70 Example of reference based on polysilicon work function difference.

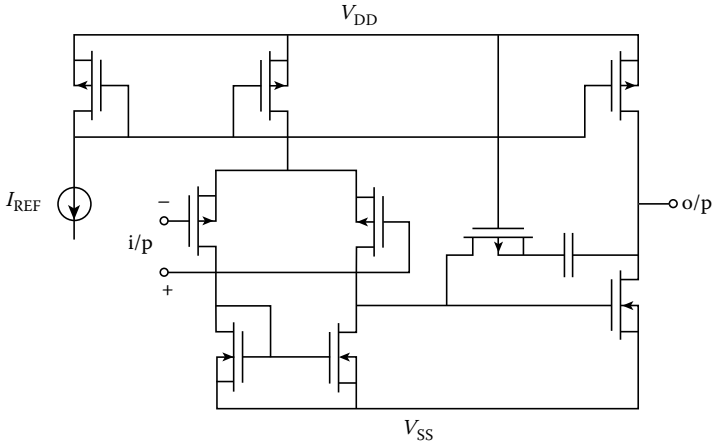


FIGURE 2.71 Two-stage amplifier.

the first- and second-stage controls the separation of the poles and also the systematic offset. The internal biasing circuit consists simply of a set of current mirrors.

2.3.6.2 Cascode Amplifiers

In cascode amplifiers [13], the idea is to raise the amplifier output impedance in order to increase the gain. An important requirement, especially in a low supply voltage environment, is to obtain maximum output voltage swing, or compliance, in the cascode amplifier. This requirement makes the biasing of the cascode devices critical.

A simple amplifier designed for cascode loads is shown in Figure 2.72. Only a single current mirror from the main current reference is needed to control all the bias currents. For reasonable low-frequency gains of say >60 dB, the output impedance must be made high while keeping component parameters practical. Use of very long channel output FETs is undesirable because of poor bandwidth and the chip area requirement. Therefore, the cascode technique, as shown in Figure 2.73, is the ideal solution. This raises output impedance by approximately $g_{m(M11)}r_{ds(M11)}$ and $g_{m(M12)}r_{ds(M12)}$ on each side. However, the maximum output voltage swing, or compliance, of the circuit has now been reduced by at least the saturation voltages of M_{11} and M_{12}. The cascode devices must be biased so that the voltage across M_8 and M_9 is just above V_{dsat}.

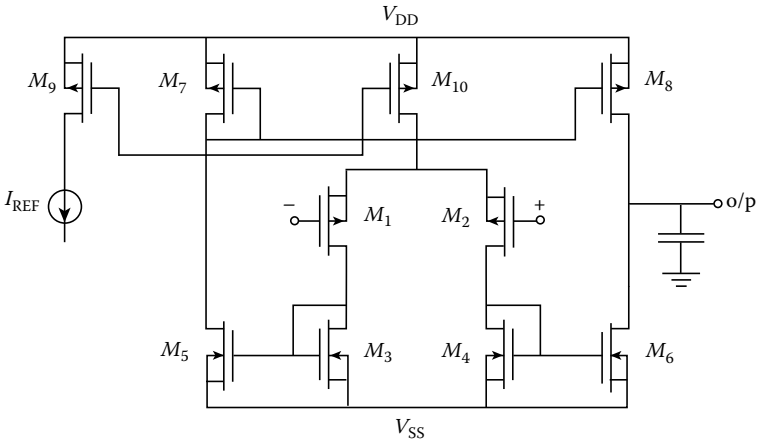


FIGURE 2.72 Simple single-stage amplifier.

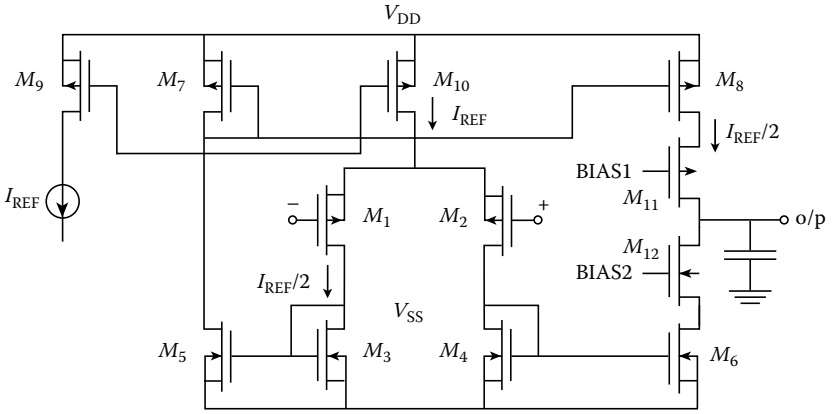


FIGURE 2.73 Simple single-stage amplifier with cascoded output FETs.

The usual way of achieving this is to arrange that a current is passed through an FET with a scaled width so that one obtains a voltage $V_{TN} + V_{Dsat(M6)} + V_{Dsat(M12)}$ for the N-channel side and $V_{TP} + V_{Dsat(M8)} + V_{Dsat(M11)}$ for the P-channel side. If the saturation voltage of the driver FET (M_6) and the cascode FET (M_{12}) are the same, then this requires a bias FET with a current density four times higher. In reality, the body effect in M_{12} and tolerance considerations mean that this factor will have to be somewhat higher than 4 [13].

Hence, there is a requirement for more replicas of the incoming reference current. The whole scheme develops to the configuration shown in Figure 2.74, where the circuit is represented in its simplest ideal form. Note that all FET scaling is applied to device widths; the lengths are the same throughout. In practice, this circuit would have a large offset due to the unequal drain voltages in the various current mirrors, and balancing dummy FETs would be needed.

2.3.6.3 Folded Cascode Amplifiers

A common problem in modern CMOS design is the basing of folded cascode amplifier stages [13]. Folded cascode amplifiers are much used to get reasonable common-mode and output range in low power supply voltage situations. A typical example of a folded cascode amplifier is shown in Figure 2.75.

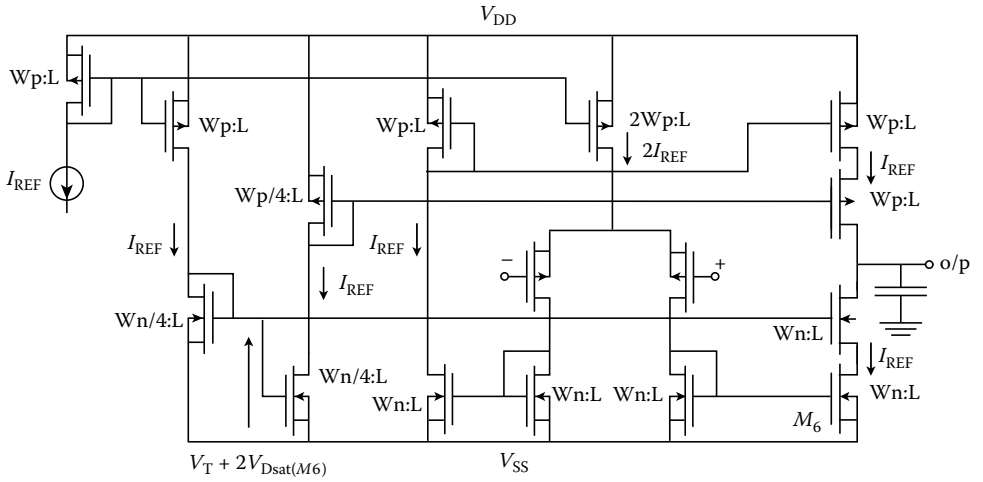


FIGURE 2.74 Cascode simple single-stage amplifier with biasing circuits.

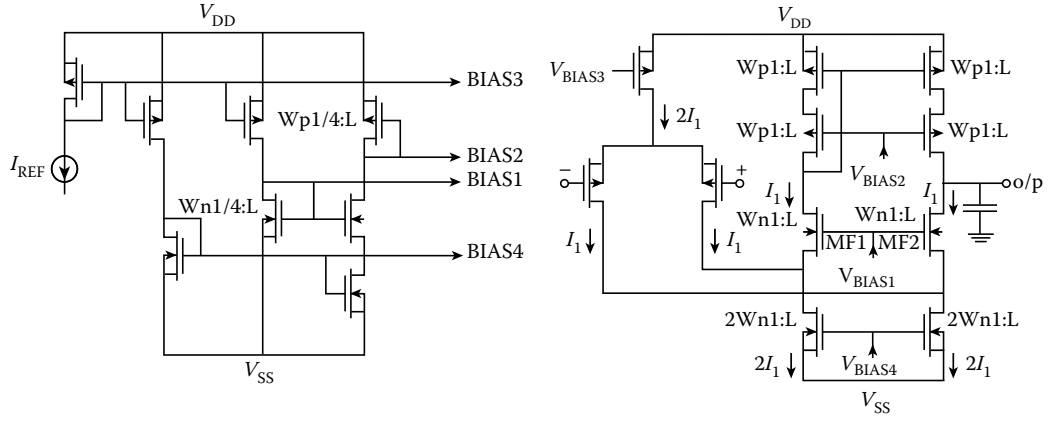


FIGURE 2.75 Folded cascode amplifier with biasing circuits.

The biasing of the folded cascode architecture is basically similar to that of a nonfolded cascode provided that correct current densities are maintained in the FETs. This is important because the folding current source (*MF*1, *MF*2) and the cascode will have different current levels and the bias must allow for this and set the cascode FET to the minimum safe operating bias. A ratio of 1:4 in width for the same current density gives the ideal bias for equal saturation voltages. Differing values of saturation voltage must be summed and the bias FET scaled accordingly.

2.3.6.4 Current Mirrors

The folded cascode amplifier of Figure 2.75 includes the current mirror of Figure 2.76 [18]. This circuit is a high compliance current mirror featuring low input voltage and low minimum output voltage. The cascode devices embedded in it are biased from an FET of width ratio running at the same quiescent current as the mirror. A width ratio of 1:4 is predicted by simple theory for equal saturation voltages in the mirror.

2.3.7 Biasing of Circuits with Low Power Supply Voltage

The topic of low-voltage analog MOS circuits is an important one because of the requirement for battery operation and also for compatibility with advanced digital IC processes with low power supply voltages. In order to maintain a reasonably high dynamic range in low supply voltage analog circuits, it is essential that the circuits operate with signal swings that are a very large fraction of the total supply voltage. Since operational amplifiers are key components in analog circuits, this has led to the design of operational amplifiers with "rail-to-rail" input common-mode voltage and output voltage capability. The design of such an input stage requires new approaches to biasing and we shall give brief details of an example of one such circuit [27].

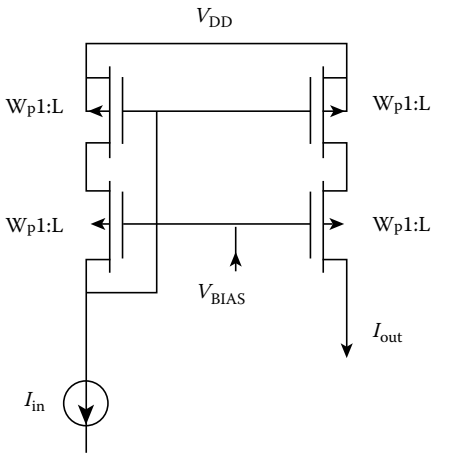


FIGURE 2.76 High-compliance current mirror.

A conventional differential pair of N- or P-channel MOSFETs would not provide sufficient input voltage common-mode range. This is because the input FET pair and the FET realizing the tail current source would tend to come out of saturation at one extreme of input common-mode voltage (negative for NMOS, positive for PMOS). This is overcome by combining an NMOS and PMOS differential pair as shown in Figure 2.77. However, this circuit has the disadvantage that the effective transconductance varies widely with common-mode input voltage since in the middle range, both differential pairs are conducting but at the extremes, only one differential pair is conducting. This produces a common-mode voltage dependence of amplifier dynamic performance and makes it difficult to optimize the dynamic performance for all input conditions.

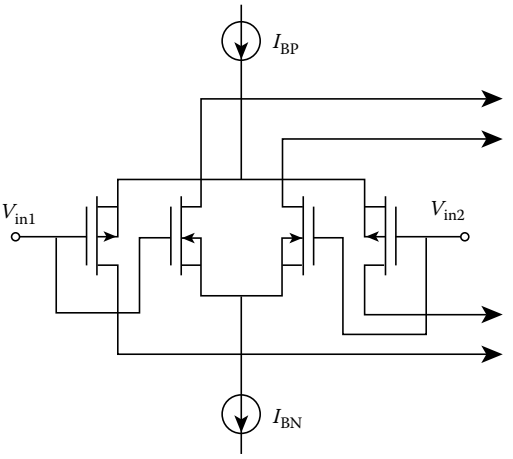


FIGURE 2.77 Op-amp input stage for rail-to-rail operation.

An elegant solution to this problem is reported in Ref. [27]. Assuming that the drain current of a MOSFET can be described by a square-law relationship, then transconductance is proportional to the square root of bias current. Since the overall effective transconductance of the input stage in Figure 2.77 is the sum of the effective transconductance of each pair, it follows that the condition when the overall transconductance is independent of common-mode input voltage is

$$\sqrt{I_{BN}} + \sqrt{I_{BP}} = \text{Constant} \tag{2.127}$$

Bias currents satisfying this relationship can be implemented using the MOS translinear circuit principle [27]. This principle applies to circuits where the gate–source ports of MOSFETs form a closed loop. Assuming that the devices are describable by a square-law drain–current relationship, the sum of the square roots of the drain currents of the MOSFETs whose ports are connected in a clockwise fashion equals the sum of the square roots of the drain currents of the counterclockwise-connected MOSFETs.

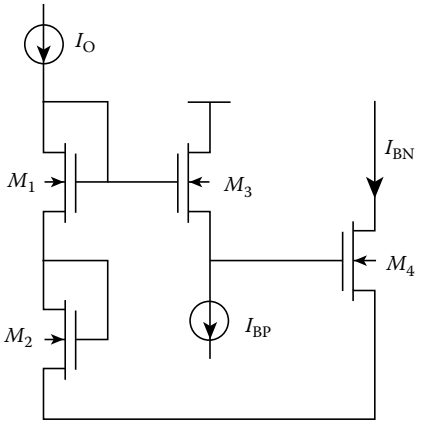


FIGURE 2.78 Bias circuit based on MOS translinear principle.

The application of the basic idea to implement bias currents according to Equation 2.127 is shown in Figure 2.78. Since the clockwise-connected MOSFETs M_1 and M_2 have a constant drain current I_o, the translinear principle implies that the drain current in M_3 and M_4 satisfy Equation 2.127. The development of the schematic bias circuit in Figure 2.78, the input stage in Figure 2.77, and a class *AB* output stage into a fully operational amplifier is described in Ref. [27]. The circuit operates with a minimum power supply of 2.5 V.

2.3.8 Dynamic Biasing

Dynamic biasing is a technique that is applicable to amplifiers in sampled data systems, such as switched capacitor filters and data converters [6,9]. Such amplifiers are required to meet two key requirements. These are fast settling time in order to allow high switching rates and high gain in order to obtain precision performance [13]. Fast settling time is obtained for maximum effective device transconductance and device transconductance g_m is approximately proportional to, $\sqrt{I_B}$, where I_B is bias current. Gain is given by g_m/g_o, where g_o is output conductance. Since g_o is proportional to bias current, gain is "inversely" proportional to $\sqrt{I_B}$. Thus maximum settling time requires a high bias current and maximum gain requires a low bias current. The dynamic bias technique reconciles these two requirements.

Figure 2.79 shows a typical switched capacitor integrator, which is a basic building block for implementing high-order switched capacitor systems. The switches are controlled by two-phase nonoverlapping clock signals ϕ and $\bar{\phi}$. The operational amplifier would generally have a first stage comprising a differential MOSFET pair with constant current source bias. The equivalent of this with dynamic biasing is shown in Figure 2.80, where the constant current source has been replaced by the combination of capacitor *C* and switches S1 and S2. We refer to the integrator of Figure 2.79. During phase ϕ, the capacitor C_1 is being charged up to the input voltage and the amplifier is inactive. Meanwhile, in the dynamically biased amplifier of Figure 2.80, the capacitor *C* is being discharged. In phase $\bar{\phi}$, capacitor C_1 in Figure 2.79 is connected to the input of the amplifier, where the output voltage is required to change to absorb the incoming charge. At the same time, capacitor *C* in Figure 2.80 is connected to the differential pair and immediately starts to conduct a high current. The high current through the amplifier MOSFETs provides a high effective amplifier slew rate and fast initial settling time, although the gain of the amplifier during this initial part of the clock phase is low. As time progresses, capacitor *C* becomes charged and the current in the amplifier MOSFETs reduces. This increases the gain of the amplifier leading to a high precision of the amplifier output voltage.

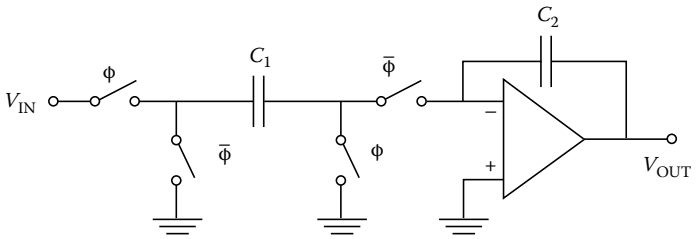


FIGURE 2.79 Typical switched capacitor integrator.

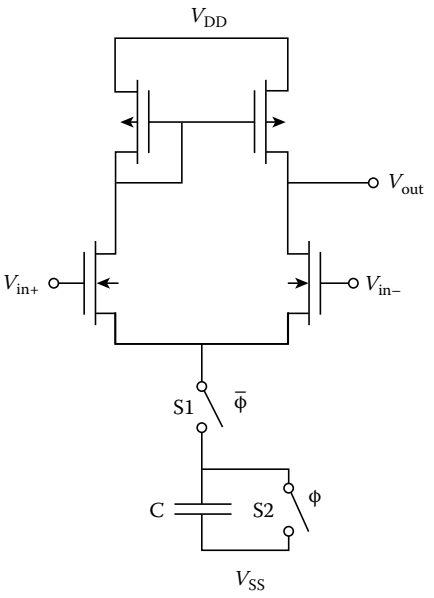


FIGURE 2.80 Simple op-amp with dynamic biasing.

Eventually, the amplifier current falls to zero with the output voltage at this required level. If, as would usually be the case, it is required to sample the amplifier output voltage in both phases, then the dynamic current source comprising capacitor *C* and the two switches S1 and S2 in Figure 2.80 would need to be duplicated with opposite switch phasing. This technique considerably increases the gain available from an amplifier since the effective gain depends on the low bias current condition and is very high. Dynamic biasing may, however, be easily applied to both stages of a two-stage amplifier if required [9]. Also, efficient schemes are available for the dynamic biasing of several amplifiers in a circuit.

Dynamic biasing is well worth considering in sampled date applications, such as switched capacitor filters and data converters. It can maximally exploit a given low power consumption to obtain good dynamic circuit performance. A variant of this approach [11] is adaptive biasing in which the input differential signal is sensed and the bias current is increased for large differential input signals to speed up the slewing response.

2.3.9 Conclusions

The task of designing voltage and current references and bias circuits is an important one. The requirements are very diverse, ranging from high precision, as required in data converters, to moderate, as required in general biasing situations. In these sections, space has been sufficient only to discuss some outstanding work in the area and some of the main principles. It is hoped that the reader will consult the references for more detailed information.

References

1. S. M. Sze, *Physics of Semiconductor Devices*, New York: Wiley Interscience, 1969.
2. R. J. Widlar, New developments in IC voltage regulators, *IEEE J. Solid-State Circuits*, SC-6, 2–7, Feb. 1971.
3. Y. P. Tsividis, A CMOS voltage reference, *IEEE J. Solid-State Circuits*, SC-13, 774–778, Dec. 1978.
4. R. A. Blauschild, P. A. Tucci, R. S. Muller, and R. G. Meyer, A new NMOS temperature-stable voltage reference, *IEEE J. Solid-State Circuits*, SC-13, 767–773, Dec. 1978.
5. M. E. Hoff, J. Huggins, and B. M. Warren, An NMOS telephone Codec for transmission and switching applications, *IEEE J. Solid-State Circuits*, SC-14, 47–50, Feb. 1979.
6. M. A. Copelend and J. M. Rabaey, Dynamic amplifier for MOS technology, *Electron. Lett.*, 15, 301, 302, May 1979.
7. E. A. Vittoz and O. Neyroud, A low voltage CMOS bandgap reference, *IEEE J. Solid-State Circuits*, SC-14, 573–577, Jun. 1979.
8. H. I. Oguey and B. Gerber, MOS voltage reference based on polysilicon gate work function difference, *IEEE J. Solid-State Circuits*, SC-15, 264–269, Jun. 1980.
9. B. J. Hosticka, Dynamic CMOS amplifiers, *IEEE J. Solid-State Circuits*, SC-15, 887–894, Oct. 1980.
10. R. Ye and T. Tsividis, Bandgap voltage reference sources in CMOS technology, *Electron. Lett.*, 18(1), 24, 25, Jan. 7, 1982.

11. M. C. Degruwe, J. Rijmenants, E. A. Vittoz, and H. J. de Man, Adaptive biasing CMOS amplifiers, *IEEE J. Solid-State Circuits*, SC-17, 552–528, Oct. 1980.
12. B.-S. Song and P. R. Gray, A precision curvature-compensated CMOS bandgap reference, *IEEE J. Solid-State Circuits*, SC-18, 634–643, Dec. 1983.
13. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, New York: Wiley, 1984, pp. 730–737.
14. J. Michejda and S. K. Kim, A precision CMOS bandgap reference, *IEEE J. Solid-State Circuits*, SC-19, 1014–1021, Dec. 1984.
15. B. J. Hosticka, K.-G. Dalsab, D. Krey, and G. Zimmer, Behavior of analog MOS integrated circuits at high temperatures, *IEEE J. Solid-State Circuits*, SC-20, 871–874, Aug. 1985.
16. M. G. K. R. Degrauwe, O. N. Leuthold, E. A. Vittoz, H. J. Oguey, and A. Descombes, CMOS voltage reference using lateral bipolar transistors, *IEEE J. Solid-State Circuits*, SC-20, 1151–1157, Dec. 1985.
17. S. L. Lin and C. A. T. Salama, A $V_{be}(T)$ model with application to bandgap reference design, *IEEE J. Solid-State Circuits*, SC-20, 1283–1285, Dec. 1985.
18. A. J. J. Boudewijns, Amplifier arrangement, U.S. Patent 4,893,090, Granted Jan. 8, 1990 (submitted Sept. 1988).
19. W. M. Sansen, F. O. Eynde, and M. Steyaert, A CMOS temperature-compensated current reference, *IEEE J. Solid-State Circuits*, SC-23, 821–824, Jun. 1988.
20. M. Ferro, F. Salerno, and R. Castello, A floating CMOS bandgap voltage reference for differential applications, *IEEE J. Solid-State Circuits*, SC-24, 690–697, Jun. 1989.
21. J. M. Steininger, Understanding wideband MOS transistors, *IEEE Circuits Devices Mag.*, 6, 26–31, May 1990.
22. G. Nicollini and D. Senderowicz, A CMOS bandgap reference for differential signal processing, *IEEE J. Solid-State Circuits*, SC-21, 41–50, Jan. 1991.
23. K. Ishibashi and K. Sasaki, A voltage down converter with submicroampere standby current for low power static RAMs, *IEEE J. Solid-State Circuits*, SC-27, 920–925, June 1992.
24. C.-Y. Wu and S.-Y. Chin, High precision curvature-compensated CMOS band-gap voltage and current references, *J. Analog Integrat. Circuits Signal Process.*, 2(3), 207–215, Sept. 1992.
25. S. D. Willingham and K. W. Martin, A BiCMOS low distortion 8 MHz low-pass filter, *IEEE J. Solid-State Circuits*, SC-28, 1234–1245, Dec. 1993.
26. J. Choma, Jr., Temperature stable voltage controlled current source, *IEEE Trans. Circuits Syst. I*, 41, 405–411, May 1994.
27. J. H. Botma, R. Jiegerink, S. L. J. Gierkink, and R. F. Wassenaar, Rail-to-rail constant Gm input stage and class AB output stage for low-voltage CMOS op amps, *Analog Integrat. Circuits Signal Process.*, 6(2), 121–133, Sept. 1994.

2.4 Canonical Cells of MOSFET Technology

Mohammed Ismail, Shu-Chuan Huang, Chung-Chih Hung, and Trond Saether

Analog integrated circuits have long been designed in technologies other than CMOS. But modern analog and mixed-signal VLSI applications in areas such as telecommunications, smart sensors, battery-operated consumer electronics, and artificial neural computation require CMOS analog design solutions. In recent years, analog CMOS circuit design has shown signs of dramatic change. Field programmable analog arrays and modular analog VLSI circuits [1] are representatives of emerging analog design philosophies leading to a whole new generation of analog circuit and layout design methodologies.

This section discusses basic cells used in contemporary CMOS analog integrated circuits. The performance of a CMOS (bipolar) circuit can often be improved further by incorporating a limited

number of bipolar (CMOS) transistors on the same substrate. The resulting circuits are called BiCMOS circuits. BiCMOS circuits that are predominantly CMOS will also be discussed. First, we discuss primitive analog cells. These cells may or may not require device matching for proper operation. Second, we introduce modern and simple circuit techniques to mitigate nonideal effects and significantly improve circuit performance, and finally, we discuss basic voltage amplifier circuits. The presented cells will help in the systematic design of analog integrated circuits and could constitute an efficient analog VLSI cell library. Throughout this section, MOS transistors are assumed to be biased in strong inversion.

2.4.1 Matched Device Pairs

Figure 2.81 shows basic MOS transistors pairs [2] operating in the saturation region, where only NMOS transistors pairs are shown. Figure 2.81a shows a differential pair with no direct connection between the two transistors. The resultant differential pair is characterized by the difference in the drain currents (using the simple square-law equation); that is

$$\begin{aligned} I_{a1} - I_{a2} &= \frac{K}{2}(V_{G1} - V_{S1} - V_T)^2 - \frac{K}{2}(V_{G2} - V_{S2} - V_T)^2 \\ &= \frac{K}{2}[(V_{G1} - V_{G2}) - (V_{S1} - V_{S2})] \\ &\quad \times [(V_{G1} + V_{G2}) - (V_{S1} + V_{S2}) - 2V_T] \end{aligned} \tag{2.128}$$

where $K(=\mu C_{ox}W/L)$ and V_T are the transconductor parameter and the threshold voltage of the transistor, respectively. Figure 2.81b is a common-source or source-coupled differential pair, a special case of circuit (a) with $V_{S1} = V_{S2} = V_S$, and the differential current is

$$I_{b1} - I_{b2} = \frac{K}{2}(V_{G1} - V_{G2})[(V_{G1} + V_{G2}) - 2V_S - 2V_T] \tag{2.129}$$

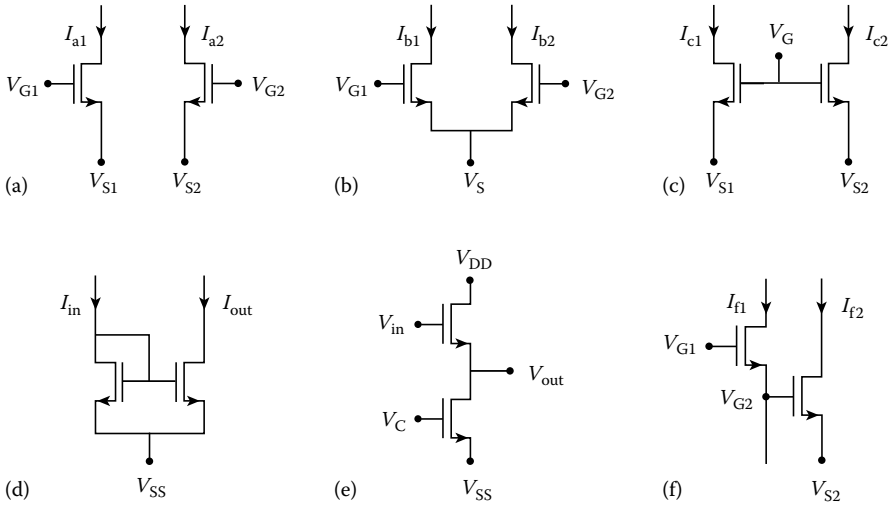


FIGURE 2.81 Matched primitive cells operating in the saturation region. (a) A differential pair with no direct connection between the two transistors, (b) is a common source or source coupled differential pair, (c) a common-gate differential pair, (d) a well-known simple current mirror, (e) a voltage follower, and (f) a rearranged transistor pair.

Figure 2.81c is a common-gate differential pair with $V_{G1} = V_{G2} = V_G$ in circuit (a), and the differential current is obtained as

$$I_{c1} - I_{c2} = -\frac{K}{2}(V_{S1} - V_{S2})[2V_G - (V_{S1} + V_{S2}) - 2V_T] \tag{2.130}$$

Differential pairs are essential building blocks of circuits such as op-amps, differential difference amplifiers and operational transconductance amplifiers. Several linear *V–I* converters built by these cells have been developed.

Current mirrors are usually used as loads for amplifier stages. Moreover, current mirrors are essential building blocks in modern current-mode analog integrated circuits. Figure 2.81d shows a well-known simple current mirror. Ideally, the input current I_{in} is equal to the output current I_{out} for matched transistors. In practice, a nonunity I_{out} to I_{in} ratio occurs due to finite output resistance resulting from channel length modulation effects. The output resistance can be increased by Wilson or cascode current mirrors at the expense of a limited output swing, which is not desired in low-voltage applications. A regulated current mirror can improve both the output resistance and swing but increase circuit complexity. Detail analysis and comparison are discussed in Ref. [3].

A voltage follower is shown in Figure 2.81e. Since the same current flows in both transistors, their gate–source voltages are the same. That is,

$$V_{in} - V_{out} = V_C - V_{SS} \tag{2.131}$$

and therefore

$$V_{out} = V_{in} - V_C + V_{SS} \tag{2.132}$$

Alternatively, a transistor pair can be arranged as the circuit shown in Figure 2.81f, which is used as a basic cell for composite MOSFET (COMFET) circuits [4]. The differential current is given by

$$\begin{aligned} I_{f1} - I_{f2} &= \frac{K}{2}(V_{G1} - V_{G2} - V_T)^2 - \frac{K}{2}(V_{G2} - V_{S2} - V_T)^2 \\ &= \frac{K}{2}(V_{G1} - V_{S2} - 2V_T)(V_{G1} - 2V_{G2} + V_{S2}) \end{aligned} \tag{2.133}$$

With proper biasing, linear *V–I* conversion can be achieved by this transistor cell.

Transistor pairs operating in the triode region are found mostly in simulating linear transconductors and resistors, for example, those in MOSFET-C filters. Figure 2.82 shows three popular examples, where the nonlinear terms in the drain current equations are canceled. A simple drain current equation in the triode region is

$$I_D = K\left[(V_G - V_T)(V_D - V_S) - \frac{1}{2}\left(V_D^2 - V_S^2\right)\right] \tag{2.134}$$

$$= \frac{K}{2}(V_G - V_S - V_T)^2 - \frac{K}{2}(V_G - V_D - V_T)^2 \tag{2.135}$$

Equation 2.135 gives another form of the triode current equation. In some cases, circuit analysis can be performed more easily with this form than using Equation 2.134. The resulting current equations of the equivalent ''MOS resistors'' are now obtained.

For the circuit shown in Figure 2.82a, a two-transistor transconductor, the current difference is given by [5]

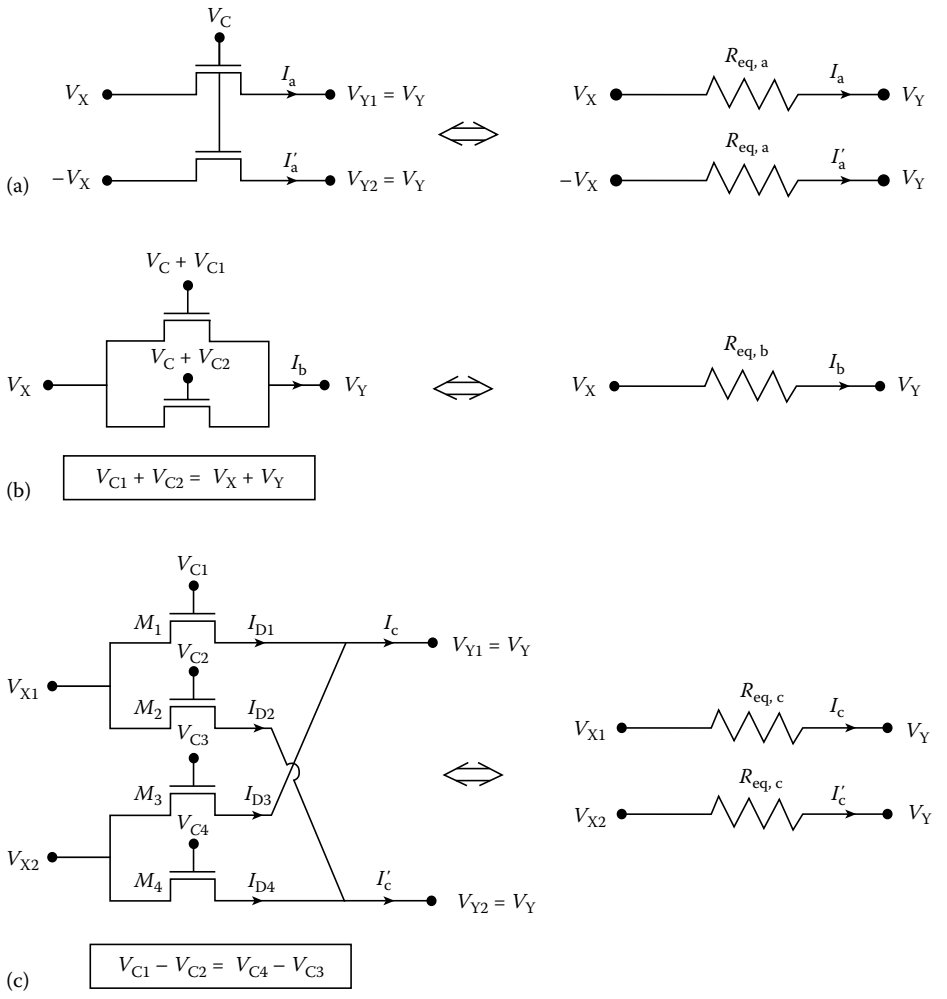


FIGURE 2.82 Matched primitive cells operating in the triode region. Three popular examples: (a) a two-transistor transconductor, (b) a realized floating resistor, and (c) a four-transistor transconductor.

$$\begin{aligned} I_a - I'_a &= K\left[(V_C - V_T)(V_X - V_Y) - \frac{1}{2}\left(V_X^2 - V_Y^2\right)\right] \\ &\quad - K\left[(V_C - V_T)(-V_X - V_Y) - \frac{1}{2}\left(V_X^2 - V_Y^2\right)\right] \\ &= 2K(V_C - V_T)V_X \end{aligned} \tag{2.136}$$

and the equivalent resistance is obtained as

$$R_{eq,a} = \frac{2V_X}{I_a - I'_a} = \frac{1}{K(V_C - V_T)} \tag{2.137}$$

The circuit shown in Figure 2.82b realizes a floating resistor [5], where

$$
\begin{aligned}
I_b &= K\left[(V_C + V_{C1} - V_T)(V_X - V_Y) - \frac{1}{2}\left(V_X^2 - V_Y^2\right)\right] \\
&\quad + K\left[(V_C + V_{C2} - V_T)(V_X - V_Y) - \frac{1}{2}\left(V_X^2 - V_Y^2\right)\right] \\
&= K\left[(2V_C - 2V_T + V_X + V_Y)(V_X - V_Y) - \left(V_X^2 - V_Y^2\right)\right] \\
&= 2K(V_C - V_T)(V_X - V_Y)
\end{aligned} \tag{2.138}
$$

and the equivalent resistance is

$$
R_{eq,b} = \frac{V_X - V_Y}{I_b} = \frac{1}{2K(V_C - V_T)} \tag{2.139}
$$

An implementation with $V_{C1} = V_{C2} = (V_X + V_Y)/2$ has been described in Ref. [6].

The circuit shown in Figure 2.82c, a four-transistor transconductor [7], gives the following current equation:

$$
\begin{aligned}
I_c - I_c' &= K\left[(V_{C1} - V_T)(V_{X1} - V_Y) - \frac{1}{2}\left(V_{X1}^2 - V_Y^2\right)\right] \\
&\quad - K\left[(V_{C2} - V_T)(V_{X1} - V_Y) - \frac{1}{2}\left(V_{X1}^2 - V_Y^2\right)\right] \\
&\quad + K\left[(V_{C3} - V_T)(V_{X2} - V_Y) - \frac{1}{2}\left(V_{X2}^2 - V_Y^2\right)\right] \\
&\quad - K\left[(V_{C4} - V_T)(V_{X2} - V_Y) - \frac{1}{2}\left(V_{X2}^2 - V_Y^2\right)\right]
\end{aligned} \tag{2.140}
$$

$$
\begin{aligned}
&= K(V_{C1} - V_{C2})(V_{X1} - V_{X2}) \\
&= K(V_{C4} - V_{C3})(V_{X1} - V_{X2})
\end{aligned} \tag{2.141}
$$

and

$$
R_{eq,c} = \frac{V_{X1} - V_{X2}}{I_c - I_c'} = \frac{1}{K(V_{C1} - V_{C2})} = \frac{1}{K(V_{C4} - V_{C3})} \tag{2.142}
$$

Note that I_c and I_c' are taken at the V_Y nodes. It is very interesting to know that nonlinearity cancellation is also achieved with the four transistors operating in the saturation region [2].

Figure 2.82a and c are usually used together with op-amps to simulate resistors, where the virtual short property of op-amps makes $V_{Y1} = V_{Y2}$.

2.4.2 Unmatched Device Pairs

Figure 2.83 shows primitive cells that do not require matching, unless specified. Figure 2.83a and b are parallel and series composite NMOS transistors, respectively, which are very useful in laying out very wide or long transistors, respectively. The equivalent device transconductance parameter K_{eq} is calculated as follows:

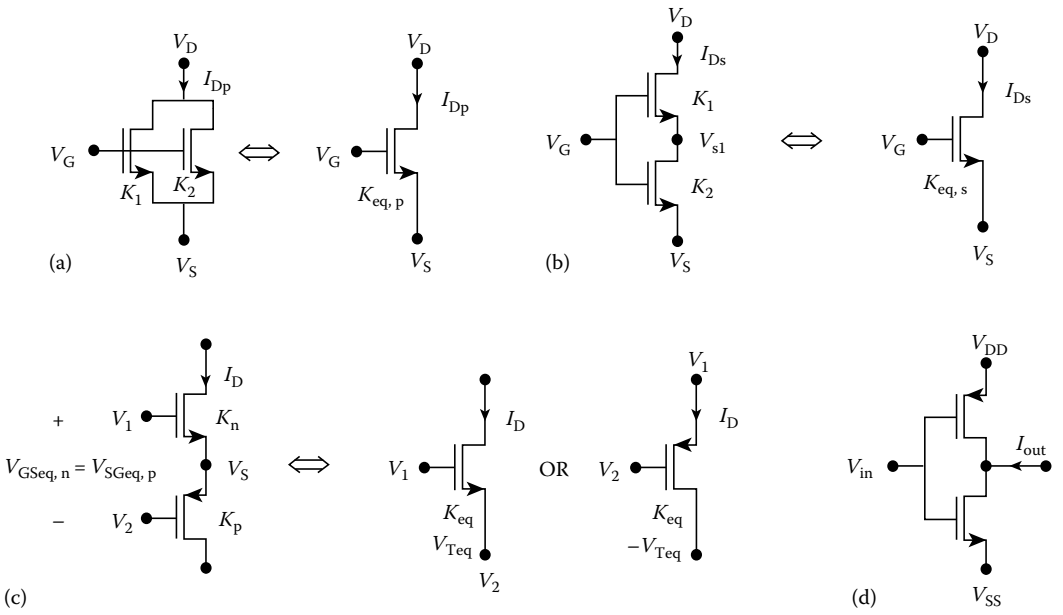


FIGURE 2.83 Unmatched primitive cells. (a) A parallel composite NMOS transistor, (b) a series composite NMOS transistor, (c) a CMOS composite transistor, and (d) a CMOS inverter.

For the parallel composite transistor, the drain current is written as

$$\begin{aligned} I_{Dp} &= \frac{K_{eq,p}}{2}(V_G - V_S - V_T)^2 \\ &= \frac{K_1}{2}(V_G - V_S - V_T)^2 + \frac{K_2}{2}(V_G - V_S - V_T)^2 \end{aligned} \tag{2.143}$$

That is,

$$K_{eq,p} = K_1 + K_2 \tag{2.144}$$

or alternatively

$$\left(\frac{W}{L}\right)_{eq,p} = \left(\frac{W}{L}\right)_1 + \left(\frac{W}{L}\right)_2 \tag{2.145}$$

Using the same channel length L, it can be simplified as

$$W_{eq,p} = W_1 + W_2 \tag{2.146}$$

As a result, a wider transistor can be realized by parallel connection of two or more narrower transistors.

For the series composite transistor, note that the lower transistor is always operating in the triode region due to the requirement $V_G - V_{S1} > V_T$, to turn on the upper transistor. The resultant drain current is given by

$$I_{Ds} = \frac{K_{eq,s}}{2}(V_G - V_S - V_T)^2 \tag{2.147}$$

$$= \frac{K_1}{2}(V_G - V_{S1} - V_T)^2 \tag{2.148}$$

$$= \frac{K_2}{2}\left[(V_G - V_S - V_T)^2 - (V_G - V_{S1} - V_T)^2\right] \tag{2.149}$$

From the preceding equations, we have

$$(V_G - V_S - V_T)^2 = \frac{2I_{Ds}}{K_{eq,s}} \tag{2.150}$$

$$(V_G - V_{S1} - V_T)^2 = \frac{2I_{Ds}}{K_1} \tag{2.151}$$

Substituting the preceding equations into Equation 2.149, we obtain

$$I_{Ds} = \frac{K_2}{2}\left(\frac{2I_{Ds}}{K_{eq,s}} - \frac{2I_{Ds}}{K_1}\right) \tag{2.152}$$

That is,

$$\frac{1}{K_{eq,s}} = \frac{1}{K_1} + \frac{1}{K_2} \tag{2.153}$$

or

$$\left(\frac{L}{W}\right)_{eq,s} = \left(\frac{L}{W}\right)_1 + \left(\frac{L}{W}\right)_2 \tag{2.154}$$

Similarly, with fixed channel width W, the above equation is simply obtained as

$$L_{eq,s} = L_1 + L_2 \tag{2.155}$$

which indicates that the equivalent transistor can be used to realize a long-channel device with shorter channel ones.

Figure 2.83c is a CMOS composite transistor, which can be seen as equivalent to either an NMOS or a PMOS transistor operating in the saturation region. In contrast, the composite transistors shown in Figure 2.83a and b can operate in both saturation and triode regions. The main advantage of the equivalent composite transistor shown in Figure 2.83c is that both their equivalent gate and source nodes have high input impedances, which is desired in some circuits. The equivalent K and V_T are obtained by the equations of the gate–source voltages.

$$\begin{aligned} V_1 - V_2 &= V_{GSn} + V_{SGp} \\ &= \sqrt{\frac{2I_D}{K_n}} + \sqrt{\frac{2I_D}{K_p}} + V_{Tn} - V_{Tp} \\ &= \sqrt{\frac{2I_D}{K_{eq}}} + V_{Teq} \end{aligned} \tag{2.156}$$

which give

$$\frac{1}{\sqrt{K_{eq}}} = \frac{1}{\sqrt{K_n}} + \frac{1}{\sqrt{K_p}} \tag{2.157}$$

and

$$V_{Teq} = V_{Tn} - V_{Tp} \tag{2.158}$$

Finally, Figure 2.83d shows a CMOS inverter, which could be used as a transconductor [8]. Its output current is

$$\begin{aligned} I_{out} &= \frac{K_n}{2}(V_{in} - V_{SS} - V_{Tn})^2 - \frac{K_p}{2}(V_{DD} - V_{in} + V_{Tp})^2 \\ &= a(V_{in} - V_{Tn})^2 + bV_{in} + c \end{aligned} \tag{2.159}$$

where

$$\begin{aligned} a &= \frac{1}{2}(K_n - K_p) \\ b &= -K_n V_{SS} + K_P(V_{DD} - V_{Tn} + V_{Tp}) \\ c &= \frac{K_n}{2}\left(2V_{SS}V_{Tn} + V_{SS}^2\right) + \frac{K_p}{2}\left[V_{Tn}^2 - (V_{DD} + V_{Tp})^2\right] \end{aligned}$$

2.4.3 Composite Transistors

The body effect of a transistor is due to nonzero source to bulk voltage (V_{SB}), which widens the depletion region between the source and bulk and therefore increases the absolute value of its threshold voltage. The threshold voltage (referred to the source) is dependent on V_{SB} and is given by

$$\begin{aligned} V_{Tn} &= V_{Tno} + \gamma\left(\sqrt{2|\phi_F| + V_{SB}} - \sqrt{2|\phi_F|}\right) \quad \text{for NMOS} \\ V_{Tp} &= V_{Tpo} - \gamma\left(\sqrt{2|\phi_F| + V_{SB}} - \sqrt{2|\phi_F|}\right) \quad \text{for PMOS} \end{aligned}$$

where $2|\phi_F|$ is the potential required for strong inversion and γ is the body effect parameter.

Usually, bulk regions of an NMOS transistor and a PMOS transistor are tied to the most negative voltage (V_{SS}) and the most positive voltage (V_{DD}) respectively to turn off the parasitic diodes associated with source-bulk and drain-bulk. In some cases, bulk regions are directly connected to transistor sources ($V_{SB} = 0$) to eliminate the body effect; for example, in the follower of Figure 2.81e, the bulk must be connected to the source in each transistor to ensure equal threshold voltages. This is achieved by putting each device in a separated well, which must be the P-well for NMOS devices. However, separate wells require large layout areas. Besides, unless twin-tub processes are used, only one type of transistor (either NMOS or PMOS depending on the process) can be connected this way.

Due to the body effect, the equivalent threshold voltage of a CMOS composite transistor would be large (two threshold voltages plus extra voltage resulting from the body effect), which would render it unsuitable for low-voltage applications. The equivalent threshold voltage could be reduced by replacing one of the MOS transistors with a BJT, as shown in Figure 2.84 [9]. For the stacked composite BiCMOS transistors, the equivalent threshold voltage is given by $V_{Teq} \approx |V_T| + 0.7$ V, where V_T is the threshold

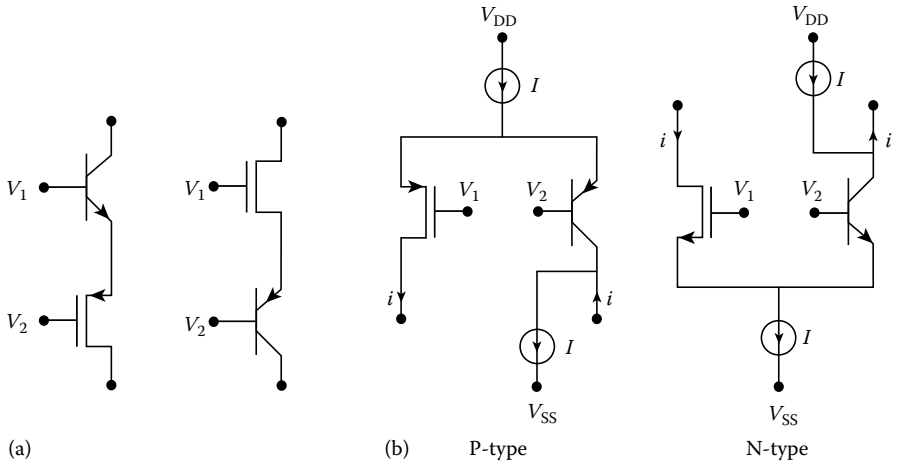


FIGURE 2.84 BiCMOS composite transistors: (a) stacked version and (b) folded version.

voltage of the NMOS or PMOS transistor, and 0.7 V is the BJT turn-on voltage V_{BE}, which is not subject to body effects. It can be further reduced by the folded arrangement as shown in Figure 2.84b, where $V_{Teq} \approx |V_T| - 0.7$ V. An all-MOS-folded composite transistor can be implemented in a similar manner as shown in Figure 2.85 [10], where $K_2 \gg K_1$. As a result,

$$K_{eq} \approx K_1$$

$$V_{Teq} = V_T - V_{GS2} \tag{2.160}$$

$$\approx -\sqrt{\frac{2I}{K_2}} \tag{2.161}$$

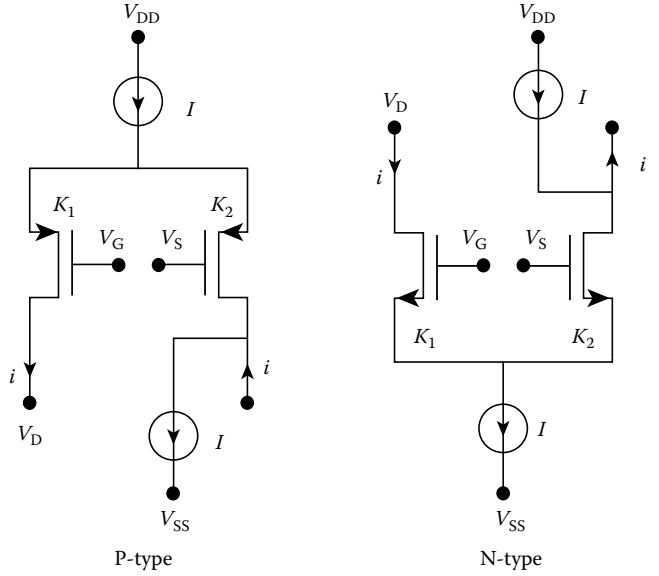


FIGURE 2.85 MOS-folded composite transistors.

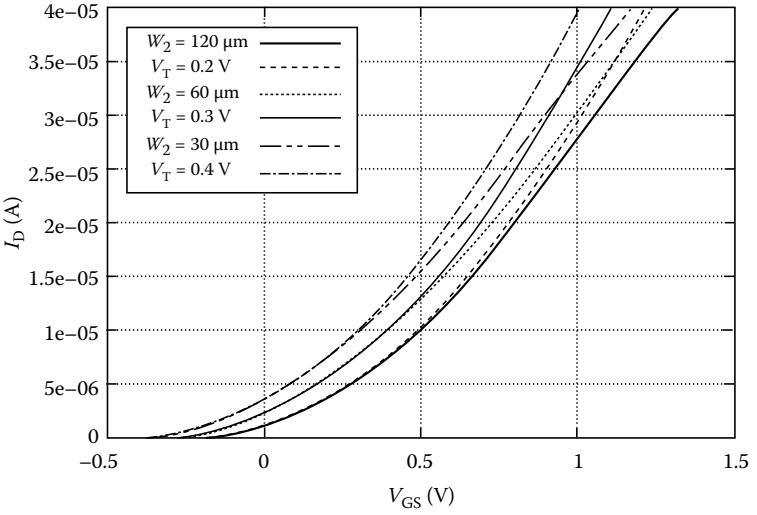


FIGURE 2.86 I_D curves for a folded N-type transistor with various W_2 and a single transistor with various V_T.

where $V_{GS2} \approx \sqrt{(2I)/K_2} + V_T$. Simulation program with integrated circuit emphasis (SPICE) simulation results for the N-type folded transistors ($W_1 = L_1 = L_2 = 3$ μm) operating in the saturation region ($V_{DG} = 0$) with various W_2 are compared with a single NMOS depletion transistor ($W = L = 3$ μm) with various V_T shown in Figure 2.86. It can be seen that a smaller K_2 results in a smaller V_{Teq} (more negative), but with a larger K_2 the composite transistor behaves more like a single transistor having a smaller $|V_{Teq}|$, which could be useful in low-voltage applications.

Forcing $V_{SB} = 0$ to eliminate the body effect is usable only for MOS circuits conducting currents in a single direction. For the circuits shown in Figure 2.82, since resistors operate bidirectionally, the bulk regions of each transistor must be connected to the rail to assure that parasitic diodes are turned off when currents flow in either direction. In fact, the transistors are operating in the triode region symmetrically between drain and source and biased at $V_{DS} \approx 0$. It would, however, result in nonzero V_{SB} and increase the threshold voltage, which increases the equivalent resistance, in Figure 2.82a and b, but introduces nonlinearities. To overcome this problem, one may configure two transistors into one composite transistor as shown in Figure 2.87a, where the bulks of the transistors are interconnected to node V_{S1}. Due to symmetry, this composite transistor can be operated in either direction. Its physical cross section is shown in Figure 2.87b, where the diodes represent the p–n junctions composed by bulk and source/drain nodes. This configuration is equivalent to Figure 2.87c, and one can find that the parasitic diode connected between V_{S1} and V_S would turn on when $V_{S1} - V_S$ is larger than the turn-on voltage of the parasitic diode. However, this is undesired, but fortunately the diode current is restricted by the drain current of M_2. This effect can be illustrated more clearly through the comparison of transistors with various bulk connections, as shown in Figure 2.88, where Figure 2.88d and e are composite transistors (same as Figures 2.87 and Figure 2.83b, respectively), which simulate single transistors. With $V_{DG} = 0$ (diode connection), $V_S = 0$ and $V_{SS} = -5$ V, Figure 2.89 gives simulation results of drain currents for the circuits, shown in (a) and (c)–(e), where the transistor sizes for the circuits, shown in (a)–(c), (d), and (e) are 20/3 μm, 30/3 μm, and 36/3 μm, respectively. One can observe that the curve (c) shown in Figure 2.89 completely departs from curve (a), due to the body effect. Curve (e) fits perfectly to curve (a). Although behaving slightly differently from curve (a), circuit (d) approximates a single transistor as well.

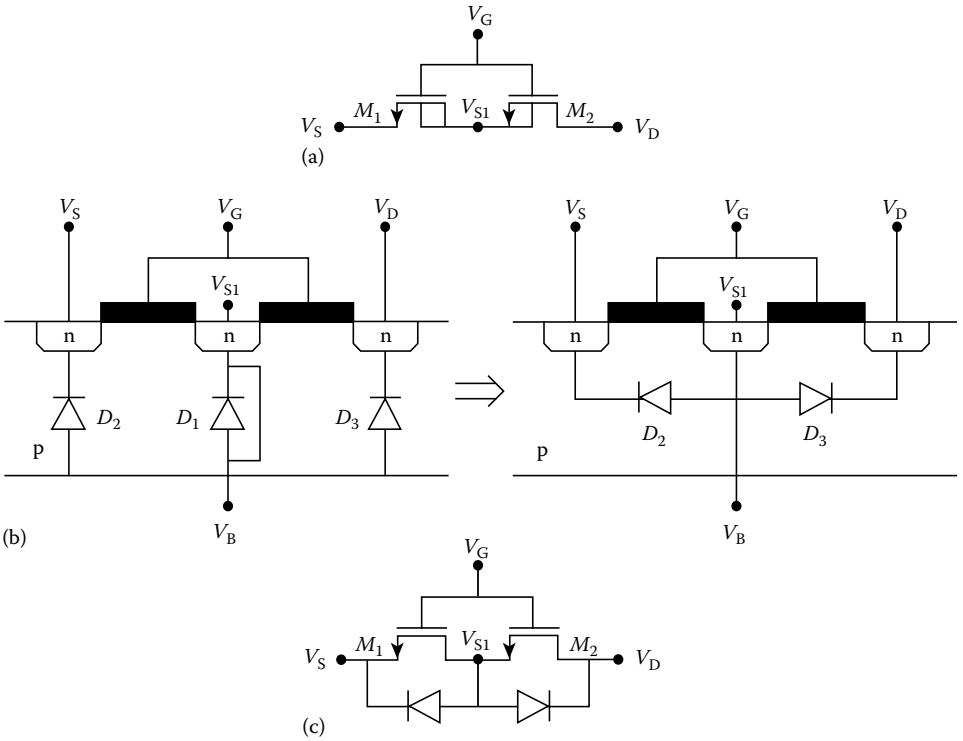


FIGURE 2.87 (a) Composite bidirectional transistor with reduced body effect. (b) Cross-sectional view of the physical device, where the short connection across D_1 is, in effect, placing D_2 and D_3 back-to-back between V_D and V_S and forming a parasitic symmetrical bipolar device. (c) Equivalent circuit of (a). (From Huang, S.-C., Systematic design solutions for analog VLSI circuits, PhD dissertation, Department of Electrical Engineering, Ohio State University, Columbus, OH, 1994. With permission.)

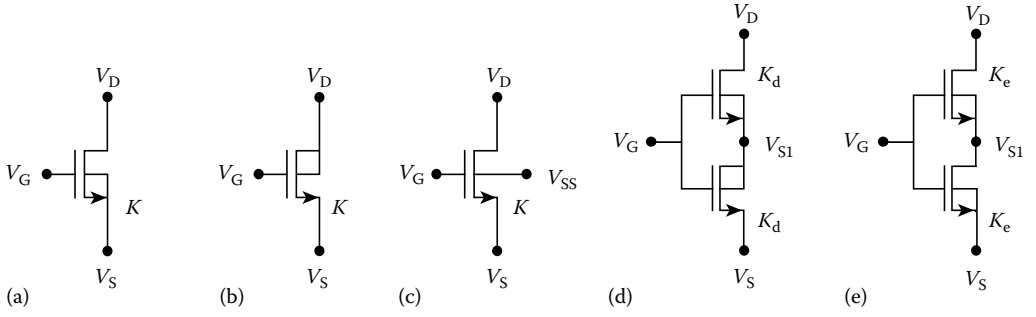


FIGURE 2.88 Transistors with various bulk connections.

Figure 2.90 shows the drain current for the circuit labeled (b), whose current is much larger than those of the rest of the circuits shown in Figure 2.88. Since the bulk of the circuit shown in Figure 2.88b is connected to its drain and the parasitic diode between the drain and bulk is on, the current is dominated by the diode current due to its exponential nature. By using it as in Figure 2.88d, the diode current is limited to the current level of an MOS transistor. This can be seen from Figure 2.91, showing V_{S1} of the

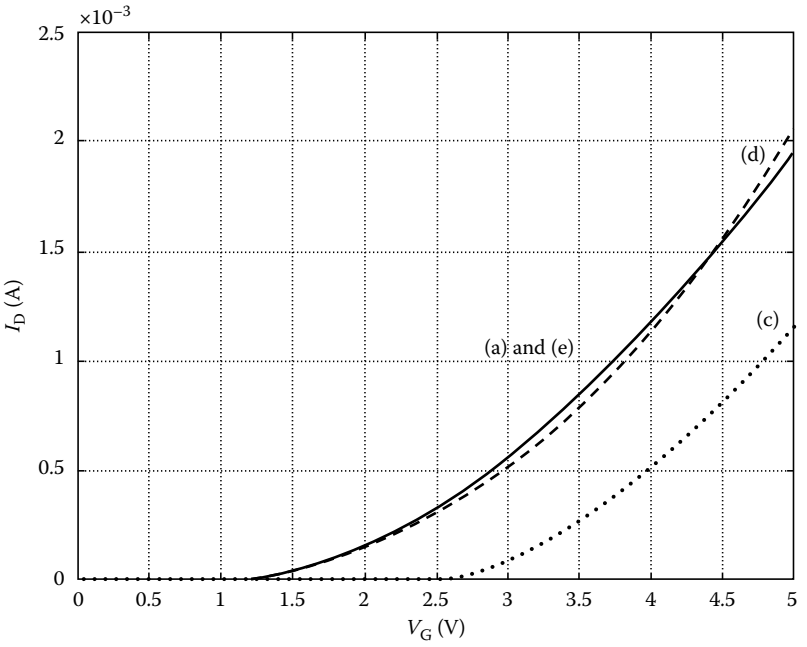


FIGURE 2.89 I_D curves for transistors with various bulk connections.

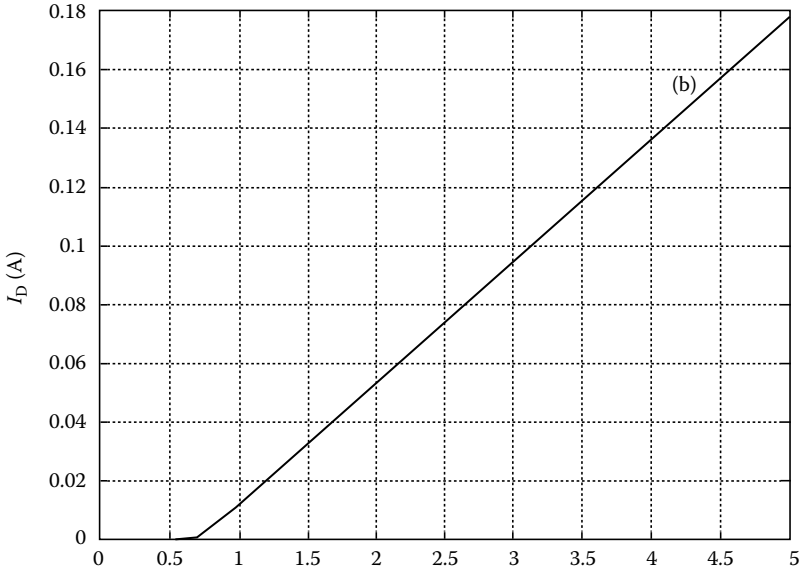


FIGURE 2.90 I_D curve for transistor with bulk connected to drain.

circuit (d) saturated at a voltage ∼0.7 V, which is close to the turn-on voltage of a p–n junction diode. The transistor sizes of the circuits labeled (d) and (e) are adjusted to achieve the same K value of the single transistor (a). According to Equation 2.153, $1/K_{eq,s} = 1/K_1 + 1/K_2$. That is, to achieve $K_{eq,s} = K$, K_e $(= K_1 = K_2)$ is given by

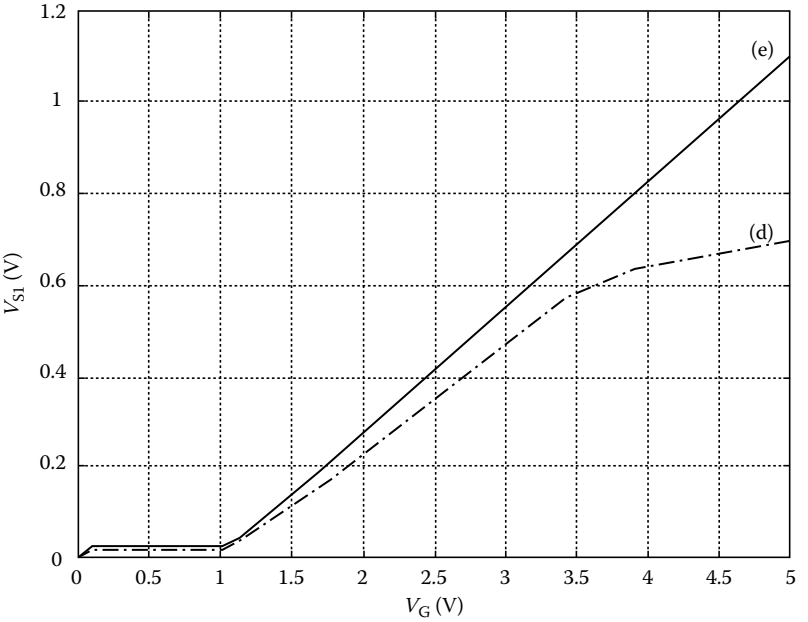


FIGURE 2.91 V_{S1} curves for composite transistors, labeled (d) and (e).

$$K_e = 2K \tag{2.162}$$

For circuit (d), K_d is obtained by rewriting Equation 2.149, which follows that

$$\begin{aligned} I_{Ds} &= \frac{K}{2}(V_G - V_S - V_T)^2 \\ &= \frac{K_d}{2}(V_G - V_{S1} - V_T)^2 \\ &= \frac{K_d}{2}\left[(V_G - V_S - V_T)^2 - (V_G - V_{S1} - V_T)^2\right] + I_s e^{(V_{S1}-V_S)/U_T} \end{aligned}$$

where I_s is the leakage current of the diode and U_T is the thermal voltage. Therefore,

$$\begin{aligned} I_{Ds} &= \frac{K_d}{2}\left(\frac{2I_{Ds}}{K} - \frac{2I_{Ds}}{K_d}\right) + I_s e^{(V_{S1}-V_S)/U_T} \\ &= I_{Ds}\left(\frac{K_d}{K} - 1\right) + I_s e^{(V_{S1}-V_S)/U_T} \end{aligned} \tag{2.163}$$

Divided by I_{Ds}, the preceding equation becomes

$$\begin{aligned} 1 &= \frac{K_d}{K} - 1 + \frac{I_s}{I_{Ds}} e^{(V_{S1}-V_S)/U_T} \\ 2K &= K_d + K\frac{I_s}{I_{Ds}} e^{(V_{S1}-V_S)/U_T} \end{aligned} \tag{2.164}$$

and hence

$$K_d < 2K \tag{2.165}$$

due to the parasitic diode. A SPICE level 2 model is used in the simulation and its higher order effects result in using a device size of 36/3 μm($K_e \neq 2K$), instead of 40/3 μm.

2.4.4 Super MOS Transistors

The channel length modulation effect models the channel shortening effect in the saturation region due to the increase in the depletion width near the drain when increasing V_{DS}. It is modeled as

$$I_D = \frac{K}{2}(V_{GS} - V_T)^2(1 + \lambda V_{DS}) \tag{2.166}$$

where λ is the channel length modulation parameter. The effect results in a finite output impedance of a transistor, since the output impedance is given by

$$r_o = \left(\frac{\partial I_D}{\partial V_{DS}}\right)^{-1} = \frac{1}{\lambda \frac{K}{2}(V_{GS} - V_T)^2} \simeq \frac{1}{\lambda I_D} \tag{2.167}$$

As mentioned previously, this effect would cause inaccuracy in the single current mirror shown in Figure 2.81d, and can be mitigated by using cascode, improved Wilson, or regulated current mirrors as shown in Figure 2.92 [3]. These are based on the gain-boosting principle as shown in Figure 2.93 [11], where for the cascode stage in (a) (used in the cascode and the Wilson current mirrors) the output impedance is given by

$$r_{o,a} = (g_{m2}r_{o2} + 1)r_{o1} + r_{o2} \tag{2.168}$$

where g_{mi} and r_{oi} are, respectively, the small-signal transconductance and output impedance for transistor M_i. An addition gain stage A_{dd}, as in Figure 2.93b (implemented in the regulated current mirror by M_{add}) increases the output impedance almost by a factor of $(A_{dd} + 1)$ and gives

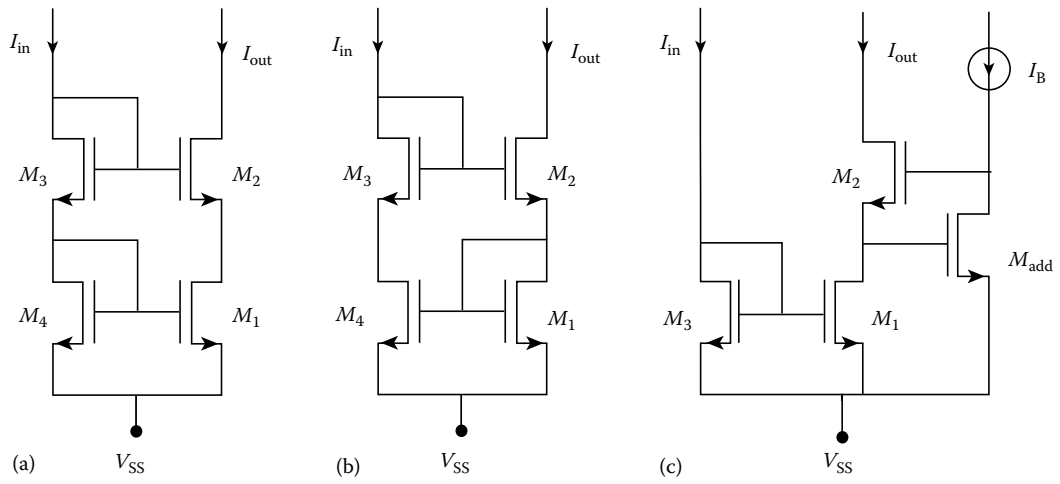


FIGURE 2.92 (a) Cascode, (b) improved Wilson, and (c) regulated current mirrors.

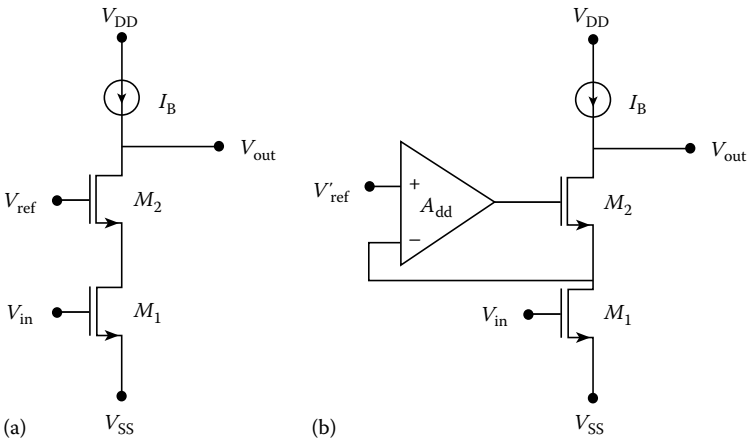


FIGURE 2.93 (a) Cascode stage and (b) cascode stage with an additional gain stage.

$$r_{o,b} = [g_{m2}r_{o2}(A_{dd} + 1) + 1]r_{o1} + r_{o2} \tag{2.169}$$

As a result, composite transistors with high output impedances can be obtained as shown in Figure 2.94 [10]. Figure 2.94a is directly obtained from the regulated current mirror, where M_{N1} and M_{N2} are cascoded and M_{P2} and M_{N4} compose an additional gain stage. The drain-source voltage of M_{N1} biased by I_1 is given by

$$V'_{DS,a} = \sqrt{\frac{2I_1}{K_4}} + V_T \tag{2.170}$$

Figure 2.94b, a modified version of (a), employs the biasing technique in Ref. [12], biasing V_{DS} for a triode-mode V–I converter. The resultant $V'_{Ds,b}$ is given by

$$V'_{DS,b} = \sqrt{\frac{2I_1}{K_4}} - \sqrt{\frac{2I_2}{K_5}} \tag{2.171}$$

Therefore, unlike the circuit shown in Figure 2.93a, where $V'_{Ds,a}$ is larger than V_T, M_{N1} can be biased at the edge of saturation by properly choosing currents I_1 and I_2 or K_4 and K_5 In addition, V_F provides a low impedance node for folded cascode configurations. Figure 2.94c, also called the super-MOS transistor [11], uses a similar concept. The CMOS cascode gain stage, composed by M_{P2}, M_{P4}, M_{N4}, and M_{N6}, gives a higher gain than the previous two circuits. Since M_{N8} in the series composite transistor (constituted by M_{N7} and M_{N8}) is always operating in the triode region, V_{DS} of M_{N8} can be very small, and M_{N1} can also be biased at the edge of saturation. However, due to its circuit complexity, the input range for V_{GS} is limited. The drain currents of the circuits shown in Figure 2.94b versus V_{DS} with various V_{GS} are compared to those obtained from a single transistor and are given in Figure 2.95. It can be seen that the output impedance of the composite transistor is significantly larger than that of a single one. The use of super MOS transistors in the design of high-gain operational amplifiers is discussed in Ref. [11].

2.4.5 Basic Voltage Gain Cells

In this subsection, we discuss simple voltage amplifier circuits implemented in NMOS, CMOS, and BiCMOS technologies.

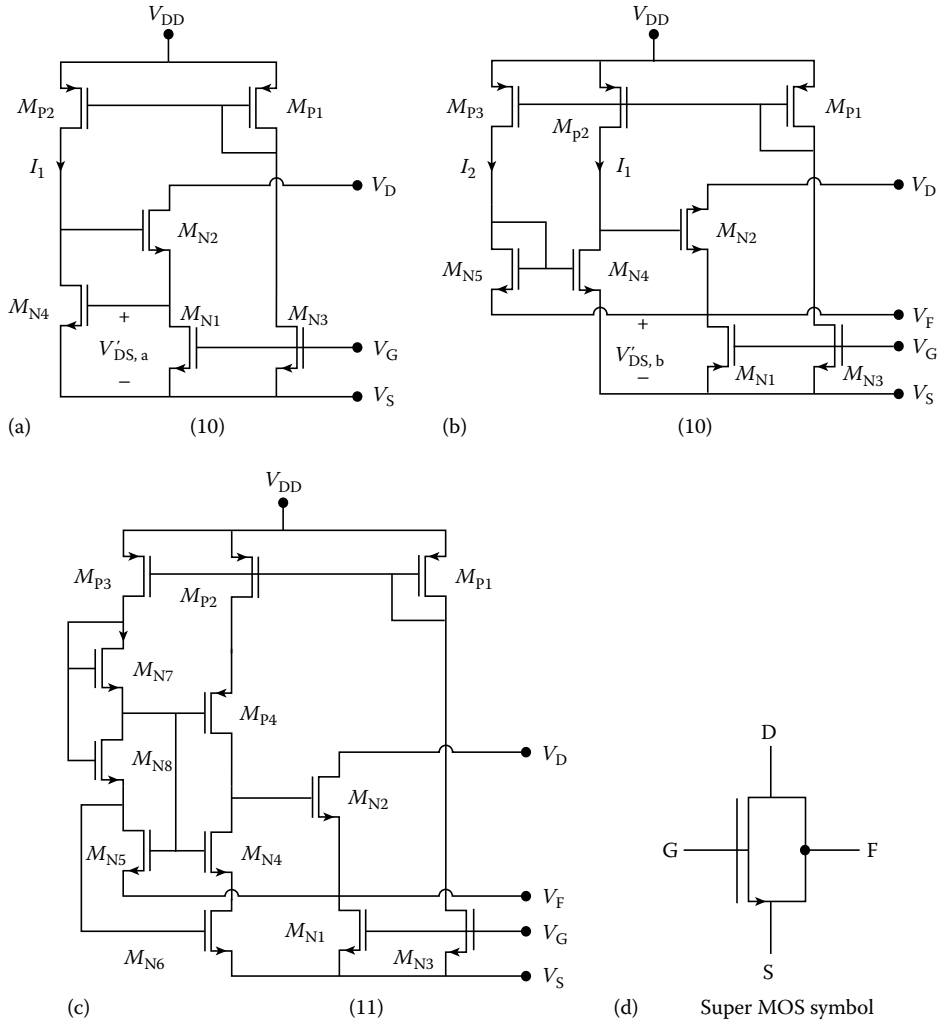


FIGURE 2.94 Composite super NMOS transistors. (a, b) (From Huang, S.-C., Systematic design solutions for analog VLSI circuits, PhD dissertation, Department of Electrical Engineering, Ohio State University, 1994.) (c) (From Bult, K. and Geelen, G.J., *J. Analog Integrat. Circuits Signal Process.*, 1, 119, 1991.)

2.4.5.1 NMOS Amplifier

Figure 2.96a shows an enhancement common-source NMOS amplifier with an enhancement load. M_1 is the driving (amplifying) transistor and the diode-connected transistor M_2 is the load device. The large-signal transfer characteristics of the amplifier is shown in Figure 2.96b and displays three well-defined regions. In region I, M_1 is off since $v_I < V_{T1}$. M_2, however, is always in the saturation region and is conducting a small current. The voltage across it is V_{T2} and hence the output voltage, v_o, is $V_{DD} - V_{T2}$. In region II, M_1 is conducting and is operating in saturation and the transfer curve is linear. Finally in region III, M_1 leaves the saturation region and enters the triode region. For the circuit to operate as an amplifier, the dc operating point must be located in the linear region (region II). Assuming that both M_1 and M_2 have the same threshold voltage V_T, but different values of K (K_1 and K_2) and neglecting both channel length modulation and body effects, we write

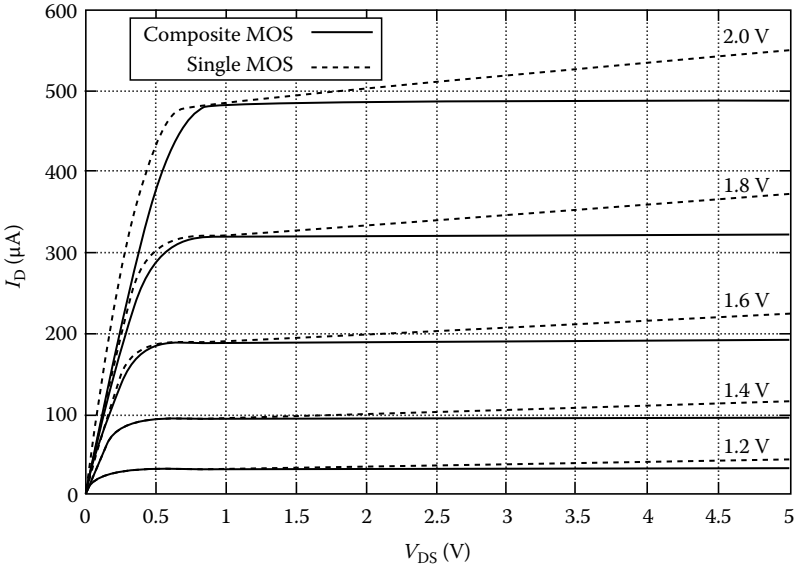


FIGURE 2.95 Simulated I_D curves for the high-output impedance composite transistor shown in Figure 2.94b and for a single transistor.

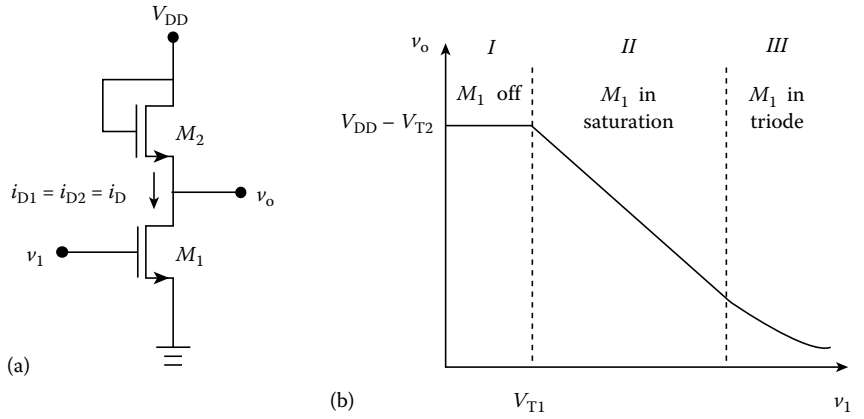


FIGURE 2.96 (a) The NMOS amplifier. (b) Transfer characteristics.

$$i_{D1} = i_{D2} = i_D = \frac{K_1}{2}(v_1 - V_T)^2 \tag{2.172}$$

and

$$\begin{aligned} i_D &= \frac{K_2}{2}(v_{GS2} - V_T)^2 \\ &= \frac{K_2}{2}(V_{DD} - v_o - V_T)^2 \end{aligned} \tag{2.173}$$

Combining Equations 2.172 and 2.173 and with some manipulations, we obtain

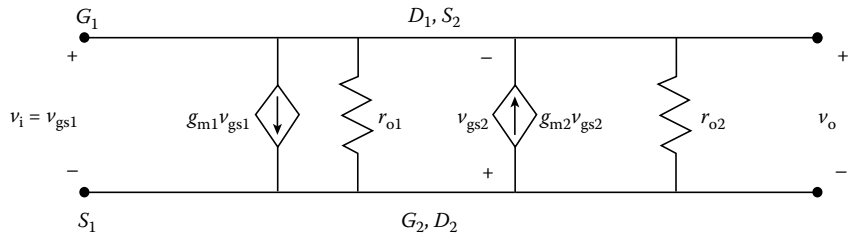


FIGURE 2.97 Small-signal equivalent circuit of the NMOS amplifier.

$$v_o = \left(V_{DD} - V_T + \sqrt{\frac{K_1}{K_2}}V_T\right) - \sqrt{\frac{K_1}{K_2}}v_I \tag{2.174}$$

which is a linear equation between v_o and v_I. This is obviously the equation of the straight-line portion (region II) of the transfer curve.

The first term in Equation 2.174 represents the dc component of the output voltage V_o. The second term represents the small-signal component and thus the ac small-signal gain of the amplifier A_v is

$$A_v = \frac{v_o}{v_i} = -\sqrt{\frac{K_1}{K_2}} = -\sqrt{\frac{(W/L)_1}{(W/L)_2}} \tag{2.175}$$

The small-signal equivalent circuit of the amplifier in Figure 2.96a is shown in Figure 2.97. Since D_2 and G_2 are connected in M_2, the voltage across the controlled current–source $g_{m2}v_{gs2}$ is v_{gs2}. Therefore, the controlled current–source can be represented by a resistance $1/g_{m2}$. Since $v_{gs1} = v_i$, we obtain the voltage gain as follows:

$$A_v = \frac{v_o}{v_i} = -\frac{g_{m1}}{g_{m2} + (1/r_{o1}) + (1/r_{o2})} \tag{2.176}$$

Now, if r_{o1} and r_{o2} are much larger than $(1/g_{m2})$, the gain reduces to $A_v \simeq -g_{m1}/g_{m2}$, which can easily be shown to lead to the expression in Equation 2.175. Note that the gain can also be determined by inspection from the circuit in Figure 2.96a as $-g_{m1}$ multiplied by the equivalent small-signal resistance seen at the drain of M_1, which is $(1/g_{m2}||r_{o1}||r_{o2})$.

Practically, M_1 and M_2 share the same substrate, which is normally connected to the most negative supply voltage in the circuit (ground in this case). It follows that M_2 suffers from body effect, which is modeled in the small-signal equivalent circuit by a controlled current-source $g_{mb2}v_{bs2}$ connected between the two output terminals in Figure 2.97, where $v_{bs2} = v_{gs2}$ and $g_{mb2} = \chi g_{m2}$ and χ is a function of the dc source–body voltage V_{SB} and lies in the range 0.1–0.3 [13]. Taking the body effect of M_2 into account, the amplifier gain becomes

$$A_v = -\frac{g_{m1}}{g_{m2} + g_{mb2}} = -\frac{g_{m1}}{g_{m2}}\frac{1}{1+\chi} \tag{2.177}$$

2.4.5.2 CMOS Amplifier

In CMOS technology, both n-channel and p-channel devices are available, and are usually fabricated in a way that eliminates the body effect. The basic CMOS amplifier is shown in Figure 2.98. Here, M_2 and M_3 in Figure 2.98c are a pair of PMOS devices operating as a current source active load and implement the

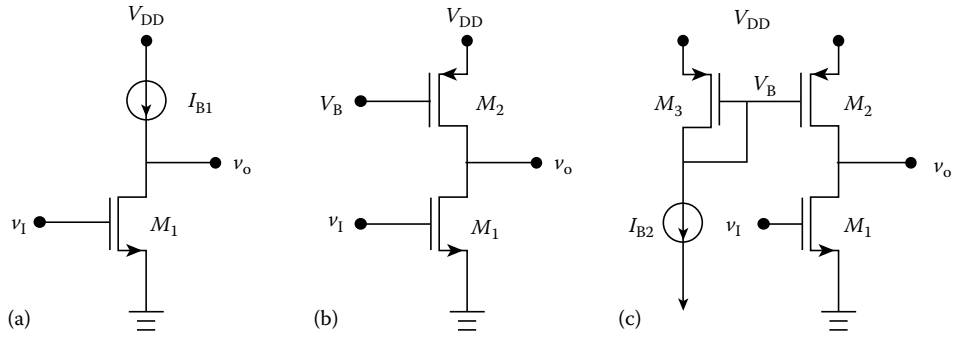


FIGURE 2.98 CMOS amplifier: (a) basic circuit, (b and c) CMOS implementations.

current source I_{B1} in Figure 2.98a. M_2 is biased in the saturation region and when M_1 is operating in the saturation region, the small-signal voltage gain will be equal to $-g_{m1}$, multiplied by the total resistance seen between the output and ground which is $(r_{o1}||r_{o2})$.

Cascode versions of the amplifier as shown previously in Figure 2.93 can be used to boost the gain significantly. For instance, the cascode amplifier in Figure 2.93a has a gain equal to the effective transconductance $-g_{meff}$, multiplied by $r_{o,a}$ given by Equation 2.168, where g_{meff} is given by [14]

$$g_{meff} = g_{m1}\frac{g_{m2}r_{o1} + (r_{o1}/r_{o2})}{g_{m2}r_{o1} + (r_{o1}/r_{o2}) + 1} \tag{2.178}$$

2.4.5.3 BiCMOS Amplifiers

A BiCMOS technology combines bipolar and CMOS transistors on a single substrate. A bipolar transistor has the advantage over an MOS of a much higher transconductance (g_m) for the same dc bias current. Also, bipolar transistors have better high-frequency performance than their MOS counterparts. On the other hand, the practically infinite input resistance at the gate of a MOSFET makes it possible to design amplifiers with extremely high input resistance and an almost zero input bias current. For these reasons, there has been an increasing interest in BiCMOS technologies for implementing high-performance integrated circuits. While most BiCMOS processes offer high-quality NMOS, PMOS, and NPN transistors, advanced BiCMOS processes offer PNP transistors as well.

Figure 2.99 shows three basic folded-cascode single-ended high-performance BiCMOS amplifiers [15]. The main features of these amplifiers are a high gain-bandwidth product and extremely high dc input

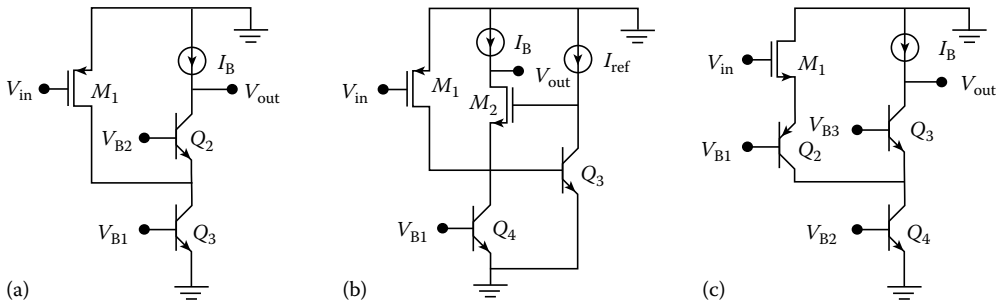


FIGURE 2.99 BiCMOS basic amplifier circuits: (a) common-source, common-base, (b) common-source, common-gate with active-feedback, and (c) common-drain, common-base, common-base.

impedance. The high gain is achieved by cascoding transistors in the signal path (M_1, Q_2 in Figure 2.99a, M_1, M_2 in Figure 2.99b, and M_1, Q_2, Q_3, in Figure 2.99c). The high bandwidth is achieved by exploiting the exponential nature of the current–voltage characteristics of bipolar transistors. For instance, let us consider the amplifier circuit in Figure 2.99a. The internal node in the signal path at the emitter of Q_2 has an extremely low impedance. This is due to the fact that while the emitter current can change significantly with the input signal, the emitter voltage remains almost constant. The same argument can be made about the internal nodes in the signal paths of the other two amplifiers (the base of Q_3 in Figure 2.99b and the emitters of Q_2 and Q_3 in Figure 2.99c). The low impedance of internal nodes in the signal path places nondominant poles at very high frequencies.

2.4.5.4 Differential Amplifier

The amplifier circuits discussed previously are of the single-ended type. A single-ended amplifier has both input and output voltage signals referred to ground. In most IC applications, a differential amplifier is utilized. In this case, the amplifier has a differential input and may also have a differential output, in which case it is called a fully differential amplifier. It is usually easy and straightforward to convert single-ended amplifiers to differential architectures.

The most widely used differential amplifier is based on the common-source or common-gate differential pairs shown respectively in Figure 2.81b and c. The common-source pair is shown here again in Figure 2.100. The only difference here is that the circuit is biased by a constant current source I that is usually implemented using a current-mirror circuit (see Figure 2.81d).

Assuming that M_1 and M_2 are identical and neglecting both channel length modulation and body effect, we write

$$i_{D1} = \frac{K}{2}(v_{GS1} - V_T)^2 \tag{2.179}$$

$$i_{D2} = \frac{K}{2}(v_{GS2} - V_T)^2 \tag{2.180}$$

Taking the square root of both sides in each of the two equations above and defining the differential input as $v_{id} = v_{GS1} - v_{GS2}$, we get

$$\sqrt{i_{D1}} - \sqrt{i_{D2}} = \sqrt{\frac{K}{2}}\, v_{id} \tag{2.181}$$

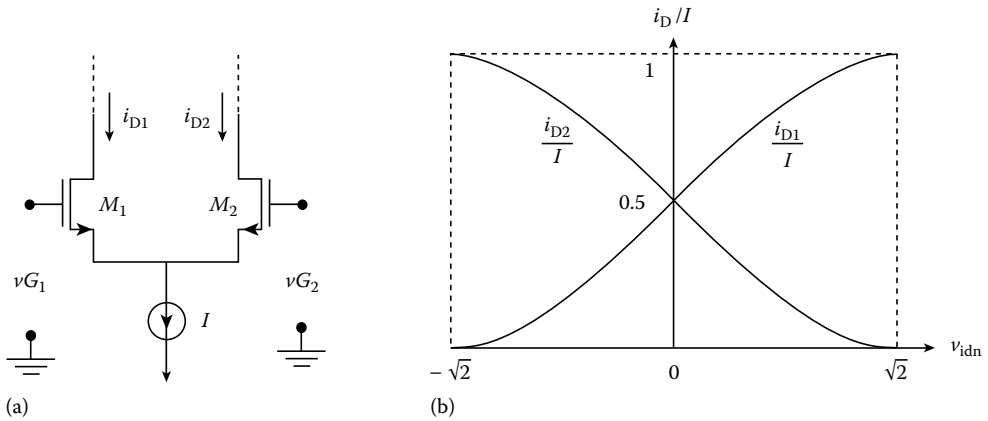


FIGURE 2.100 MOS differential pair: (a) the circuit and (b) the transfer characteristic.

But since the current–source bias imposes the constraint that $i_{D2} + i_{D2} = I$, one can easily show that

$$i_{D1,2} = \frac{I}{2} \pm \sqrt{KI}\left(\frac{v_{id}}{2}\right)\sqrt{1 - \frac{(v_{id}/2)^2}{(I/K)}} \tag{2.182}$$

At the bias point, the small-signal differential input voltage v_{id} is zero,

$$v_{GS1} = v_{GS2} = V_{GS} \quad \text{and} \quad i_{D1} = i_{D2} = I/2$$

This can be used to rewrite the preceding equation as follows:

$$i_{D1,2} = \frac{I}{2} \pm \left(\frac{I}{V_{GS} - V_T}\right)\left(\frac{v_{id}}{2}\right)\sqrt{1 - \left(\frac{v_{id}/2}{V_{GS} - V_T}\right)^2} \tag{2.183}$$

And for $v_{id}/2 \ll V_{GS} - V_T$ (small-signal approximation),

$$i_{D1,2} \simeq \frac{I}{2} \pm \left(\frac{I}{V_{GS} - V_T}\right)\left(\frac{V_{id}}{2}\right) \tag{2.184}$$

The differential pair tranconductance g_m, defined as $g_m = (i_{D1} - i_{D2})/v_{id}$ is then given by $I/(V_{GS} - V_T)$. We recall that a single MOS transistor biased at a drain current I_D has a transconductance $2I_D/(V_{GS} - V_T)$. Thus, we see that each transistor in the pair has a transconductance $2(I/2)/(V_{GS} - V_T)$, which is equal to the differential pair transconductance, g_m. Equations 2.183 and 2.184 indicate that for small-signal inputs, the current in M_1 increases by i_d and that in M_2 decreases by i_d. From Equation 2.183, we can find v_{id} at which current steering between M_1 and M_2 occurs that is $i_{D1} = I$ and $i_{D2} = 0$ or vice versa for negative v_{id}. Equating the second term in Equation 2.183 to $I/2$, we get

$$|v_{id}|\,\max = \sqrt{2}(V_{GS} - V_T) \tag{2.185}$$

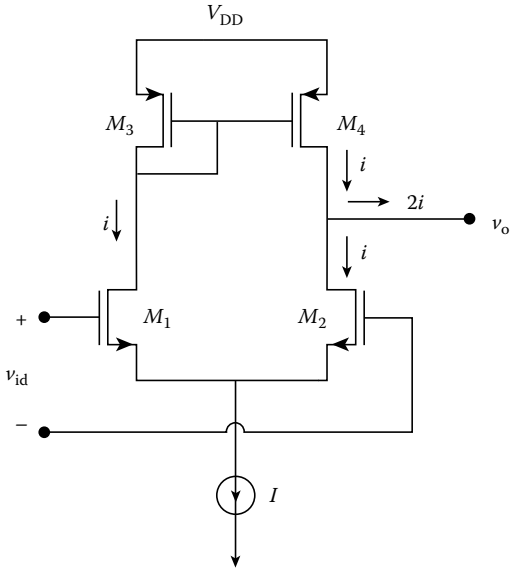


FIGURE 2.101 Simple differential-input, single-ended output CMOS amplifier.

Figure 2.100b shows plots of the normalized currents i_{D1}/I and i_{D2}/I versus the normalized differential input voltage $v_{idn} = v_{id}/(V_{GS} - V_T)$.

A simple CMOS differential amplifier is shown in Figure 2.101, where the PMOS pair is used as an active load. The small-signal current i is given by $g_m(v_{id}/2)$, where $g_m = I/(V_{GS} - V_T)$. The small-signal output voltage is given by $v_o = 2i(r_{o2}||r_{o4})$ and the voltage gain is $A_v = v_o/v_{id} = g_m((r_{o2}||r_{o4})$.

When $v_{id} = 0$, the bias current I does not actually split equally between M_1 and M_2. This is due to mismatches in K, ΔK, and V_T, ΔV_T, which contribute to a dc offset voltage that is usually larger than that in differential amplifiers implemented with bipolar transistors. For instance, modern silicon-gate MOS technologies have ΔV_T as high as 2 mV [13]. Note that ΔV_T has no counterpart in BJTs.

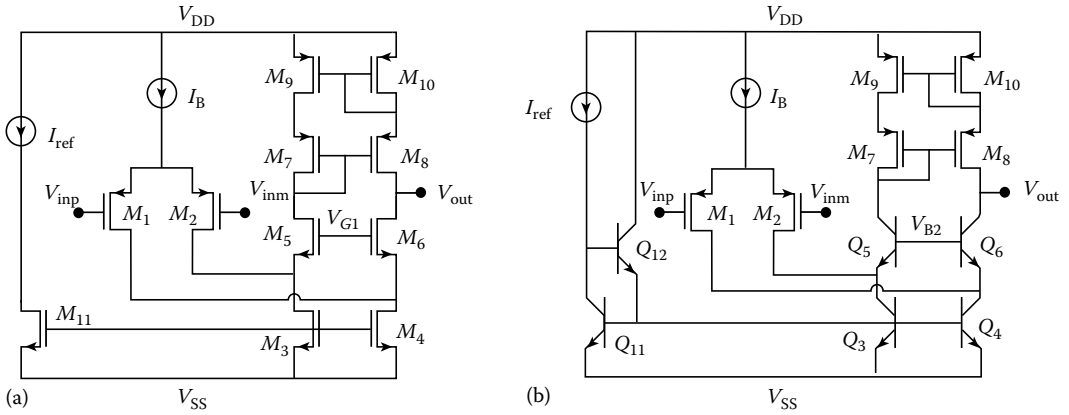


FIGURE 2.102 Folder-cascode amplifier: (a) common-source, common-gate and (b) common-source, common-base.

2.4.5.5 Folded-Cascode Operational Amplifier

The folded-cascode operational amplifier (op-amp) is a basic building block in modern analog-integrated circuits. Figure 2.102 shows two folded-cascode op-amp circuits in CMOS and BiCMOS technologies [15]. Actually, several combinations of bipolar and CMOS devices could be used in the design of this amplifier. Here, we assume that PNP bipolar transistors are not available, which implies a BiCMOS process having less complexity. The input common-source MOS pair is of the PMOS type. The cascode common-gate or common-base pair (M_5 and M_6 in Figure 2.102a and Q_5 and Q_6 in Figure 2.102b) is ''folded'' and, therefore, is implemented with devices of the opposite type to that used in the input pair. This is unlike the basic ''unfolded'' cascode amplifier in Figure 2.93, where both input and cascode devices are of the same type.

The greater values of transconductance associated with the common-base bipolar devices in the BiCMOS op-amp place the nondominant parasitic poles at much higher frequencies. Note that the BiCMOS op-amp circuit is based on the single-ended amplifier shown in Figure 2.99a. The BiCMOS op-amp combines the increased bandwidth the advantages of an MOS input stage; namely a nearly infinite input impedance, a zero input bias current, and a higher slew rate [13].

2.4.6 Conclusion

Analog design is more complicated and less systematic than digital design and involves many trade-offs to meet certain design specifications. It strongly relies on human heuristics. The transfer of these human experiences into a computer-aided design environment is essential to the success of analog design in the context of VLSI of both analog and mixed analog/digital integrated circuits. This transfer, however, requires the development of systematic approaches to the analysis and design of analog integrated circuits. To this end, understanding the basic operations of the analog cells discussed here is critical. The use of these cells in the systematic design of analog VLSI systems, such as filters and data converters, is discussed in Ref. [10].

References

1. M. Ismail and T. Fiez, *Analog VLSI: Signal and Information Processing*, New York: McGraw-Hill, 1994.
2. M. Ismail, S.-C. Huang, and S. Sakurai, Continuous-time signal processing, in *Analog VLSI: Signal and Information Processing*, M. Ismail and T. Fiez, Eds., New York: McGraw-Hill, 1994, Chapter 3.

3. Z. Wang, Analytical determination of output resistance and DC matching errors in MOS current mirrors, *IEE Proc.: Pt. G*, 137, 397–404, Oct. 1990.
4. M. C. H. Cheng and C. Toumazou, Linear composite MOSFETs (COMFETs), *Electron. Lett.*, 27, 1802–1802, Sept. 1991.
5. Y. Tsividis, M. Banu, and J. M. Khoury, Continuous-time MOSFET-C filters in VLSI, *IEEE J. Solid-State Circuits*, SC-21, 15–30, Feb. 1986.
6. M. Banu and Y. Tsividis, Floating voltage-controlled resistors in CMOS technology, *Electron. Lett.*, 18, 678, 679, July 1982.
7. M. Ismail, Four-transistor continuous-time MOS transconductor, *Electron. Lett.*, 23, 1099, 1100, Sept. 1987.
8. B. Nauta, A CMOS transconductance-C filter technique for very high frequencies, *IEEE J. Solid-State Circuits*, 27, 142–153, Feb. 1992.
9. J. Ramirez-Angulo, Applications of composite BiCMOS transistors, *Electron. Lett.*, 27, 2236–2238, Nov. 1991.
10. S.-C. Huang, Systematic design solutions for analog VLSI circuits, PhD dissertation, Department of Electrical Engineering, Ohio State University, Columbus, OH, 1994.
11. K. Bult and G. J. Geelen, The CMOS gain-boosting technique, *J. Analog Integrat. Circuits Signal Process.*, 1, 119–135, 1991.
12. U. Gatti, F. Maloberti, and G. Torelli, A novel CMOS linear transconductance cell for continuous-time filters, in *Proc. IEEE Int. Symp. Circuits Syst.* (ISCAS), 1990, pp. 1173–1176.
13. A. S. Sedra and K. C. Smith, *Microelectronic Circuits* (Series in Electrical Engineering), 3rd ed., Philadelphia: Holt, Rinehart & Winston, 1991, Chapters 5, 6, and 10.
14. K. Bult, Basic CMOS circuit techniques, in *Analog VLSI: Signal and Information Processing*, M. Ismail and T. Fiez, Eds., New York: McGraw-Hill, 1994, Chapter 2.
15. S. R. Zarabadi, M. Ismail, and F. Larsen, Basic BiCMOS circuit techniques, in *Analog VLSI: Signal and Information Processing*, M. Ismail and T. Fiez, Eds., New York: McGraw-Hill, 1994, Chapter 5.

3

High-Performance Analog Circuits

3.1 Broadband Bipolar Networks **3**-1
Introduction • Miller's Theorem • Bipolar Transistor Modeling at High Frequencies • Single-Gain Stages • Neutralization of C_μ • Negative Feedback • RF Bipolar Transistor Layout • Bipolar Current-Mode Broadband Circuits • Broadband Amplifier Stability • Conclusions
Appendix A: Transfer Function and Bandwidth Characteristic of Current-Feedback **3**-41
Appendix B: Transfer Function and Bandwidth Characteristic of Voltage-Feedback **3**-43
Appendix C: Transconductance of the Current-Feedback Op-Amp Input Stage **3**-44
Appendix D: Transfer Function of Widlar Current Mirror **3**-45
Appendix E: Transfer Function of Widlar Current Mirror with Emitter Degeneration Resistors **3**-47
References **3**-47
3.2 Bipolar Noise **3**-48
Thermal Noise • Shot Noise • Generation–Recombination Noise • *1/f* Noise • Noise $1/f^2$ • Burst Noise—RTS Noise • Avalanche Noise • Noise Characterization
References **3**-55

Chris Toumazou
Imperial College of Science, Technology, and Medicine

Alison Payne
Imperial College of Science, Technology, and Medicine

John Lidgey
Oxford Brookes University

Alicja Konczakowska
Gdańsk University of Technology

Bogdan M. Wilamowski
Auburn University

3.1 Broadband Bipolar Networks

Chris Toumazou, Alison Payne, and John Lidgey

3.1.1 Introduction

Numerous textbooks have presented excellent treatments of the design and analysis of broadband bipolar amplifiers. This chapter is concerned with techniques for integrated circuit amplifiers, and is written mainly as a tutorial aimed at the practicing engineer.

For broadband polar design, it is first important to identify the key difference between lumped and distributed design techniques. Basically when the signal wavelengths are close to the dimensions of the integrated circuit, then characteristic impedances become significant, lines become lossy, and we essentially need to consider the circuit in terms of transmission lines. At lower frequencies where the signal

wavelength is much larger than the dimensions of the circuit, the design can be considered in terms of lumped components, allowing some of the more classical low-frequency analog circuit techniques to be applied. At intermediate frequencies, we enter the realms of hybrid lumped/distributed design. Many radio-frequency (RF) designs fall into this category, although every day we see new technologies and circuit techniques developed that increase the frequency range for which lumped approaches are possible. In broadband applications, integrated circuits (ICs) are generally designed without the use of special microwave components, so broadband techniques are very similar to those employed at lower frequencies. However, several factors still have to be considered in RF design: all circuit parasitics must be identified and included to ensure accurate simulation; feedback can generally only be applied locally as phase shifts per stage are significant; the cascading of several local feedback stages is difficult since alternating current (ac) coupling is often impractical; the NPN bipolar transistor is the main device used in silicon, since it has potentially a higher f_t than PNP bipolar or MOSFET devices; active PNP loads are generally avoided due to their poor frequency and noise performance and so resistive loads are used instead.

The frequency performance of an RF or broadband circuit will depend on the frequency capability of the devices used, and no amount of good design can compensate for transistors with an inadequate range. As a rule, designs are kept as simple as possible, since at high frequencies all components have associated parasitics.

3.1.2 Miller's Theorem

It is important to describe at the outset a very useful approximation that will assist in simplifying the high-frequency analysis of some of the amplifiers to be described. The technique is known as Miller's theorem and will be briefly discussed here. A capacitor linking input to output in an inverting amplifier results in an input-referred shunt capacitance that is multiplied by the voltage gain of the stage, as shown in Figure 3.1. This increased input capacitance is known as the Miller capacitance.

It is straightforward to show that the input admittance looking into the inverting input of the amplifier is approximately $Y_{in} = j\omega C_f(1 + A)$. The derivation assumes that the inherent poles within the amplifier are at a sufficiently high frequency so that the frequency response of the circuit is dominated by the input of the amplifier. If this is not the case, then Miller's approximation should be used with caution as it will be discussed later. From the preceding model, it is apparent that the Thévenin input signal sources see an enlarged capacitance to ground. Miller's approximation is often a useful way of simplifying circuit analysis by assuming that the input dominant frequency is given by the simple low-pass *RC* filter in Figure 3.1. However, the effect is probably one of the most detrimental in broadband amplifier design, affecting both frequency performance and/or stability.

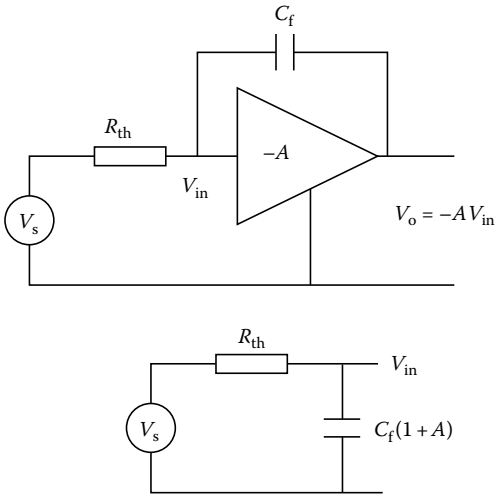


FIGURE 3.1 Example of the Miller effect.

3.1.3 Bipolar Transistor Modeling at High Frequencies

In this section, we consider the high-frequency small-signal performance of the bipolar transistor. The section assumes that the reader has some knowledge of typical device parameters, and has some familiarity with the technology. For small-signal analysis, the simplified hybrid-π model shown in Figure 3.2 is used,

where

r_b is the base series resistance
r_c is the collector series resistance
r_π is the dynamic base–emitter resistance
r_o is the dynamic collector–emitter resistance

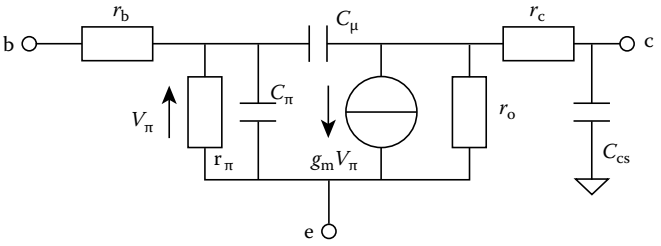


FIGURE 3.2 Hybrid π model of BJT.

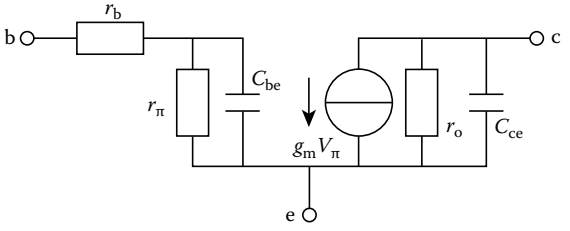


FIGURE 3.3 Simplified Miller-approximated hybrid π model of BJT.

C_π is the base–emitter junction capacitance
C_μ is the collector–base junction capacitance
C_{cs} is the collector–substrate capacitance
g_m is the small-signal transconductance

At low frequencies, the Miller approximation allows the hybrid-π model to be simplified to the circuit shown in Figure 3.3, where the net input capacitance now becomes $C_{be} = C_\pi + C_\mu (1 - A_v)$, the net output capacitance becomes $C_{ce} = C_\mu(1 - 1/A_v)$, where A_v is the voltage gain given by $A_v = (V_{ce}/V_{be}) \approx -g_m R_1$ where R_1 is the collector load resistance. r_c and C_{cs} have been neglected. Thus, $C_{be} \approx C_\pi + g_m R_1 C_\mu$ and $C_{ce} \approx C_\mu$. The output capacitance C_{ce} is often neglected from the small-signal model. The approximation $A_v = -g_m R_1$ assumes that $r_\pi \gg r_b$, and that the load is purely resistive. At high frequencies, however, we cannot neglect the gain roll-off due to C_π and C_μ, and even at frequencies as low as 5% of f_t the Miller approximation can introduce significant errors.

A simplified hybrid-π model that takes the high-frequency gain roll-off into account is shown in Figure 3.4. C_μ is now replaced by an equivalent current source $sC_\mu(V_\pi - V_{ce})$.

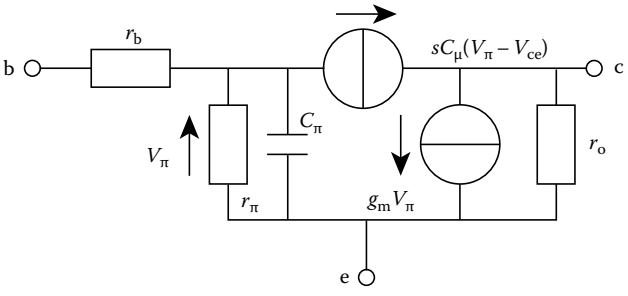


FIGURE 3.4 Simplified high-frequency model.

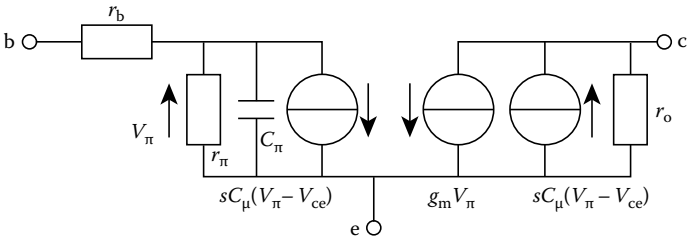


FIGURE 3.5 Split current sources.

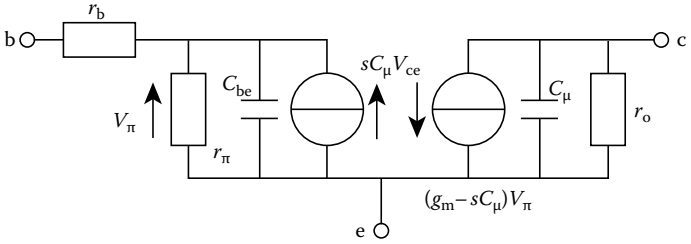


FIGURE 3.6 Modified equivalent circuit.

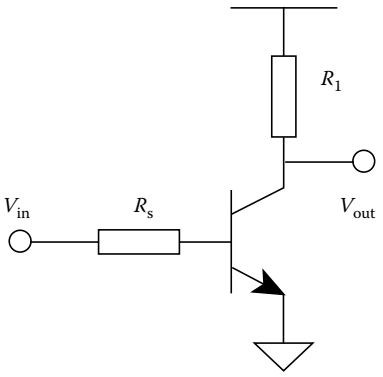


FIGURE 3.7 CE amplifier.

A further modification is to split the current source between the input and output circuits as shown in Figure 3.5.

Finally, the input and output component terms can be rearranged leading to the modified equivalent circuit shown in Figure 3.6, which is now suitable for broadband design. From Figure 3.6, the transconductance $(g_m - sC_\mu)$ shows the direct transmission of the input signal through C_μ. The input circuit current source $(sC_\mu V_{ce})$ shows the feedback from the output to the input via C_μ. Depending on the phase shift between V_{ce} and V_{be}, this feedback can cause high-frequency oscillation. At lower frequencies, $sC_\mu \ll g_m$ and $V_{ce}/V_\pi \approx -g_m R_1$, which is identical to the Miller approximation. The model of Figure 3.6 is the most accurate for broadband amplifier design, particularly at high frequencies.

3.1.4 Single-Gain Stages

Consider now the high-frequency analysis of single-gain stages.

3.1.4.1 Common-Emitter (CE) Stage

Figure 3.7 shows a CE amplifier with load R_1 and source R_s. External biasing components are excluded from the circuit.

First analysis using the Miller approximation yields the small-signal high-frequency model shown in Figure 3.8,

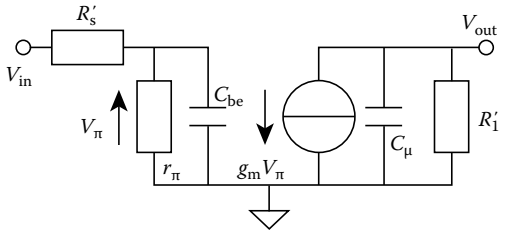


FIGURE 3.8 High-frequency model of the CE.

where

$$\begin{aligned} &R_{1'} = (R_1 \| r_0), \quad R_{s'} = R_s + r_b \quad \text{and} \quad C_{be} = C_\pi + g_m R_{1'} C_\mu \\ &\frac{V_\pi}{V_{in}} = \left(\frac{r_\pi}{r_\pi + R_{s'}}\right)\left(\frac{1}{1 + s(r_\pi \| R_{s'})C_{be}}\right) \\ &\frac{V_{out}}{V_\pi} = \frac{-g_m R_{1'}}{1 + sC_\mu R_{1'}} \end{aligned} \tag{3.1}$$

and thus

$$\frac{V_{out}}{V_{in}} = -\left(\frac{g_m R_{1'} r_\pi}{r_\pi + R_{s'}}\right)\left(\frac{1}{(1 + sC_\mu R_{1'})(1 + s(r_\pi \| R_{s'})C_{be})}\right) \tag{3.2}$$

This approximate analysis shows

- "Ideal" voltage gain $= -g_m R_1$
- Input attenuation caused by $R_{s'}$ in series with r_π
- Input circuit pole p_1 at $s = 1/C_{be}(r_\pi/R_{s'}) \approx 1/C_{be}R_{s'}$
- Output attenuation caused by r_0 in parallel with R_1
- Output circuit pole p_2 at $s = 1/C_\mu R_{1'}$

The input circuit pole is generally dominant, and thus the output pole p_2 can often be neglected. With a large load capacitance C_1, $p_2 \approx 1/C_1 R_{1'}$, and the gain and phase margin will be reduced. However, under these conditions the Miller approximation will no longer be valid, since the gain roll-off due to the load capacitance is neglected.

If we now consider analysis using the broadband hybrid-π model of Figure 3.6, then the equivalent model of the CE now becomes that shown in Figure 3.9, where

$$C_{be} = C_\pi + C_\mu, \quad R_{s'} = R_s + r_b \quad \text{and} \quad R_{1'} = R_1 \| r_0$$

From the model, it can be shown that

$$\frac{V_{out}}{V_\pi} = \frac{-(g_m - sC_\mu)R_{1'}}{1 + sC_\mu R_{1'}} \tag{3.3}$$

$$(V_{in} - V_\pi)/R_{s'} + sC_\mu V_{out} = V_\pi/r_\pi + sC_{be}V_\pi \tag{3.4}$$

and

$$V_{in} r_\pi + V_{out} sC_\mu r_\pi R_{s'} = V_\pi(r_\pi + R_{s'})(1 + sC_{be}(r_\pi \| R_{s'})) \tag{3.5}$$

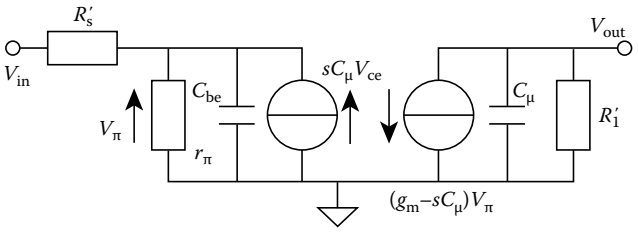


FIGURE 3.9 Equivalent circuit model of the CE.

Rearranging these equations yields

$$\frac{V_{out}}{V_{in}} = -\left(\frac{g_m R_{1'} r_\pi}{r_\pi + R_{s'}}\right) \times \left(\frac{1 - sC_\mu/g_m}{(1 + sC_{be}R_{s'})(1 + sC_\mu R_{1'}) + sC_\mu g_m R_{1'} R_{s'} - s^2 C_\mu^2 R_{s'} R_{1'}}\right) \tag{3.6}$$

This analysis shows that there is a right-hand-plane (RHP) zero at $s = 1/(C_\mu r_e)$, which is not predicted by the Miller approximation. Assuming $R_\pi \gg R_{s'}$ and $C_\pi \gg C_\mu$, the denominator can be written as

$$1 + s\left(R_{s'}\left(C_\pi + C_\mu g_m R_{1'}\right) + C_\mu R_{1'}\right) + s^2 C_\mu C_\pi R_{1'} R_{s'} \tag{3.7}$$

which can be described by the second-order characteristic equation

$$1 + s(1/p_1 + 1/p_2) + s^2/p_1 p_2 \tag{3.8}$$

By comparing coefficients in Equations 3.7 and 3.8, the sum of the poles is the same as that obtained in Equation 3.2 using the Miller approximation, but the pole product $p_1 p_2$ is greater. This means that the poles are farther apart than predicted by the Miller approximation. In general, the Miller approximation should be reserved for analysis at frequencies of operation well below f_t, and for situations where the capacitive loading is not significant. The equivalent circuit of Figure 3.9 therefore gives a more accurate result for high-frequency analysis. For a full understanding of RF behavior, computer simulation of the circuit including all parasitics is essential.

Since the CE stage provides high current and voltage gain, oscillation may well occur. Therefore, care must be taken during layout to minimized parasitic coupling between the input and output. The emitter should be at ground potential for ac signals, and any lead inductance from the emitter to ground will generate phase-shifted negative feedback to the base, which can result in instability.

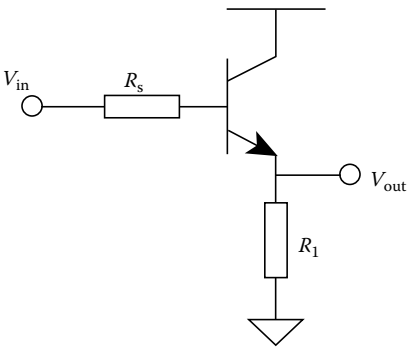


FIGURE 3.10 Common-collector amplifier.

3.1.4.2 Common-Collector (CC) Stage

The CC or emitter follower shown in Figure 3.10 is a useful circuit configuration since it generally serves to isolate a high-gain stage from a load. The high-frequency performance of this stage must be good enough not to degrade the frequency performance or stability of the complete amplifier. An equivalent high-frequency small-signal model of the CC is shown in Figure 3.11.

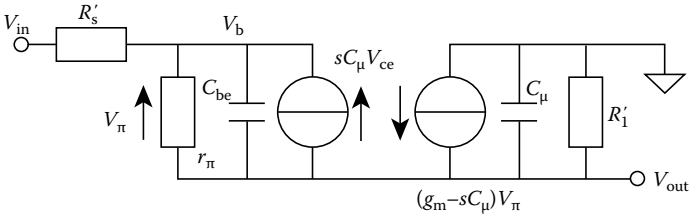


FIGURE 3.11 Equivalent circuit of the CC.

The following set of equations can be derived from Figure 3.11:

$$(V_{in} - V_b)/R_{s'} = V_\pi/r_\pi + sC_{be}V_\pi + sC_\mu V_{out}, \quad V_b = V_{out} + V_\pi \tag{3.9}$$

and

$$V_\pi/r_\pi + sC_{be}V_\pi + sC_\mu V_{out} + (g_m - sC_\mu)V_\pi - sC_\mu V_{out} - V_{out}/R_{1'} = 0 \tag{3.10}$$

Rearranging these equations yields

$$\frac{V_{out}}{V_{in}} = \frac{R_{1'}(1 + g_m r_\pi + sC_\pi r_\pi)}{(R_{s'} + r_\pi)(1 + s(R_{s'} \| r_\pi)C_{be}) + R_{1'}(1 + sC_\mu R_{s'})(1 + g_m r_\pi + sC_\pi r_\pi)} \tag{3.11}$$

The preceding expression can be simplified by assuming $R_\pi \gg R_{s'}$, $g_m r_\pi \gg 1$, $C_\pi \gg C_\mu$ to,

$$\frac{V_{out}}{V_{in}} = \left(\frac{r_\pi}{r_\pi + R_{s'}}\right)\left(\frac{1 + sC_\pi/g_m}{(1 + sC_\mu R_{s'})(1 + sC_\pi/g_m) + (1 + sC_\pi R_{s'})/g_m R_{1'}}\right) \tag{3.12}$$

This final transfer function indicates the presence of a left-half-plane zero at $s = (g_m/C_\pi) = \omega_t$. The denominator can be rewritten as approximately

$$(1 + 1/g_m R_{1'}) + s(C_\mu R_{s'} + C_\pi/g_m + C_{be}R_{s'}/g_m R_{1'}) + s^2 C_\mu C_\pi R_{s'}/g_m \tag{3.13}$$

which simplifies to

$$1 + s(C_\pi r_e + C_\mu R_{s'} + (C_\mu + C_\pi) r_e R_{s'}/R_{1'}) + s^2 C_\mu C_\pi R_{s'} R_{1'} \tag{3.14}$$

Assuming a second-order characteristic form of $1 + s(1/p_1 + 1/p_2) + s^2/p_1 p_2$, if $p_1 \ll p_2$, the above reduces to $1 + s/p_1 + s^2/p_1 p_2$. If $(R_{s'}/R_{1'}) \ll 1$, then $p_1 \approx 1/(C_\pi r_e)$, and this dominant pole will be approximately canceled by the zero. The frequency response will then be limited by the nondominant pole $p_2 \approx 1/C_\mu R_{s'}$.

The frequency response of a circuit containing several stages is thus rarely limited by the CC stage, due to this dominant pole-zero cancellation. For this analysis to be valid, $R_{s'} \ll R_{1'}$. As $R_{s'}$ increases the poles will move closer together, and the pole-zero cancellation will degrade. In practice, the CC stage is often used as a buffer, and is thus driven from a high source resistance into a low value load resistance.

A very important parameter of the common-collector stage is output impedance. It is generally assumed that the output impedance of a CC is low, also that there is good isolation between a load and the amplifying stage, and that any amount of current can be supplied to the load. Furthermore, it is assumed that capacitive loads will not degrade the frequency performance since the load will be driven by an almost short circuit. While this may be the case at low frequencies, it is a different story at high frequencies. Consider the following high-frequency analysis. We first assume that the small-signal model shown in Figure 3.12 is valid.

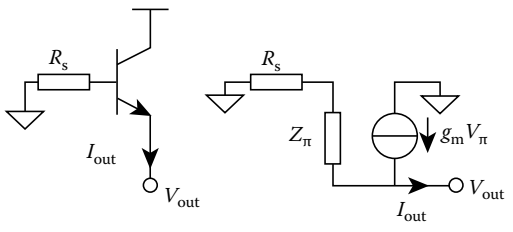


FIGURE 3.12 Equivalent circuit of the CC output stage.

From the Figure 3.12, the output impedance can be approximated as

$$\frac{V_{out}}{I_{out}} = \frac{Z_\pi + R_{s'}}{1 + g_m Z_\pi} \tag{3.15}$$

where $Z_\pi = (r_\pi \| C_{be})$ and $R_{s'} = R_s + r_b$. At very low frequencies ($\omega \to 0$):

$$R_{out} = \frac{r_\pi + R_{s'}}{1 + g_m r_\pi} \approx 1/g_m + R_{s'}/g_m r_\pi \approx r_e + R_{s'}/\beta \tag{3.16}$$

At very high frequencies ($\omega \to \infty$):

$$R_{out} = \frac{1/sC_{be} + R_{s'}}{1 + g_m/sC_{be}} \approx R_{s'} \tag{3.17}$$

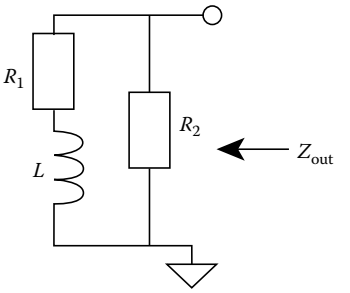


FIGURE 3.13 Equivalent high-frequency model of CC output stage.

If $r_e > R_{s'}$, then the output impedance decreases with frequency, that is, Z_{out} is capacitive. If $R_{s'} > r_e$, then Z_{out} increases with frequency, and so Z_{out} appears inductive. It is usual for an emitter follower to be driven from a high source resistance, thus the output impedance appears to be inductive and can be modeled as shown in Figure 3.13, where

$$R_1 = r_e + R_{s'}/\beta, \quad R_2 = R_{s'}, \quad L = R_{s'}/\omega_t$$

The inductive behavior of the CC stage output impedance must be considered in broadband design since any capacitive loading on this stage could result in peaking or instability. The transform from base resistance to emitter inductance arises because of the 90° phase shift between base and emitter currents at high frequencies, due principally to C_π. This transform property can be used to advantage to simulate an on-chip inductor by driving a CC stage from a high source resistance. Similarly, by loading the emitter with an inductor, we can increase the effective base series resistance $R_{s'}$ without degrading the noise performance of the circuit. A capacitive load will also be transformed by 90° between the base and emitter; for example, a capacitive loading on the base can look like a negative resistance at the emitter.

3.1.4.3 Common-Base (CB) Stage

The CB amplifier shown in Figure 3.14 offers the highest frequency performance of all the single-stage amplifiers. When connected as a unity gain current buffer, the CB stage operates up to the f_t of the transistor.

Using the simplified hybrid π model of Figure 3.3, it follows that

$$\frac{I_{out}}{I_{in}} \approx \frac{\beta}{\beta + 1} \quad \text{where} \quad \beta = \frac{\beta_o}{1 + s/\omega_o} \tag{3.18}$$

$$\frac{I_{out}}{I_{in}} \approx \frac{a_o}{1 + s/\omega_t} \quad \text{where} \quad a_o = \beta_o/(\beta_o + 1) \quad \text{and} \quad \omega_t = \beta_o \omega_o \tag{3.19}$$

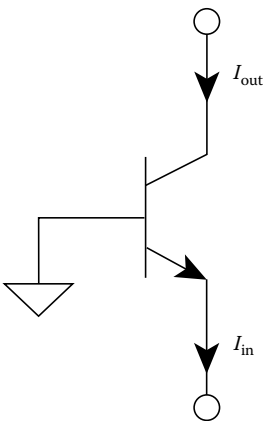


FIGURE 3.14 CB configuration.

The CB stage thus provides wideband unity current gain. Note that the input impedance of the CB stage is the same as the output impedance of the CC stage, and thus can appear inductive if the base series resistance is large.

In many situations, the CB stage is connected as a voltage amplifier, an example of this being the current-feedback amplifier, which will be discussed in a later section. Consider the following high-frequency analysis of the CB stage being employed as a voltage gain amplifier. Figure 3.15 shows the circuit together with a simplified small-signal model. From the equivalent model, the gain of the circuit can be approximated as

$$\frac{V_{out}}{V_{in}} = \frac{kR_1}{R_s}\left(\frac{1 - sC_\mu/g_m}{1 + s(C_\pi/g_m)(kR_{s'}/R_s)}\right) \tag{3.20}$$

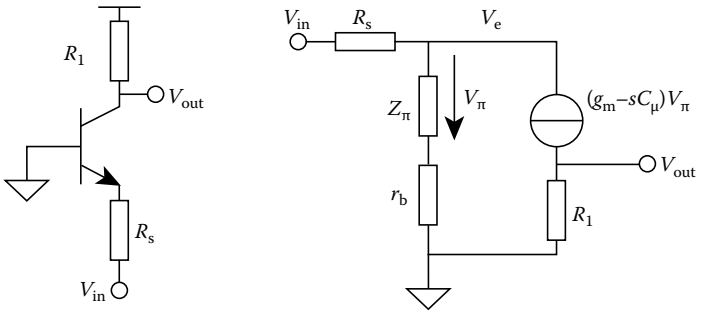


FIGURE 3.15 CB stage as a voltage amplifier.

where

$$R_{s'} = R_s + r_b, \quad \text{and} \quad k \approx \frac{R_s}{R_s + 1/g_m}$$

If $R_s \gg 1/g_m$, then $k \approx 1$ and so

$$\frac{V_{out}}{V_{in}} = \frac{R_1}{R_s}\left(\frac{1 - sC_\mu/g_m}{1 + s(C_\pi/g_m)(1 + r_b/R_s)}\right) \tag{3.21}$$

Thus, it can be seen that the circuit has an RHP zero at $s = 1/(r_e\ C_\mu)$, since $r_e = 1/g_m$ and a pole at $1/C_\pi r_e(1 + r_b/R_s) = \omega_t/(1 + r_b/R_s)$. Note that in the case of a current source drive ($R_s \gg r_b$), the pole is at the ω_t of the transistor. However, this does assume that the output is driven into a short circuit. Note also that there is an excellent isolation between the input and output circuits, since there is no direct path through C_μ and so no Miller effect.

3.1.5 Neutralization of C_μ

Many circuit techniques have been developed to compensate for the Miller effect in amplifiers and hence extend the frequency range of operation. The CE stage provides the highest potential power gain, but the bandwidth of this configuration is limited since the amplified output voltage effectively appears across the collector–base junction capacitance resulting in the Miller capacitance multiplication effect. This bandwidth limiting due to C_μ can be overcome by using a two-transistor amplifying stage such as the CE–CB cascode stage or the CC–CE cascade. Consider now a brief qualitative description of each in turn. The circuit diagram of the CE–CB cascode is shown in Figure 3.16.

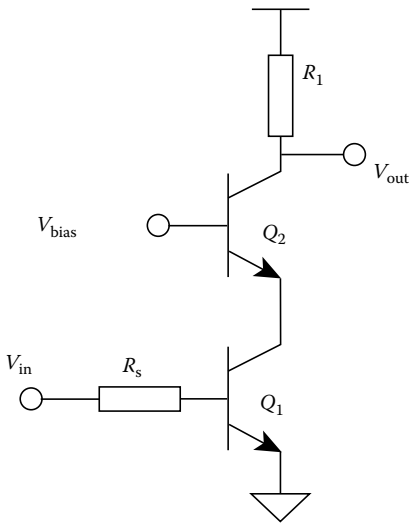


FIGURE 3.16 CE–CB cascode.

The CE transistor Q_1 provides high current gain of approximately β and a voltage gain of $A_{v1} \approx -g_{m1}R_1 = -g_{m1}r_{e2}$, which in magnitude will be close to unity. Therefore, the Miller multiplication of C_μ is minimized, and the bandwidth of Q_1 is maximized. The CB transistor Q_2 provides a voltage gain $A_{v2} \approx R_1/r_{e2}$. The total voltage gain of the circuit can be approximated as $A_v \approx -g_{m1}R_1$, which is equal to that of a single CE stage. The total frequency response is given by the cascaded response of both stages. Since both transistors exhibit wideband operation, then the dominant poles of each stage may be close in frequency. As a result, the total phase shift through the cascode

configuration is likely to be greater than that obtained with a single device, and care should be taken when applying negative feedback around the pair.

Consider now the CC–CE stage of Figure 3.17. In this case, voltage gain is provided by the CE stage transistor Q_2 and is $A_{v2} \approx -g_{m2}R_1$. This transistor is being driven from the low-output impedance of Q_1 and so the input pole frequency of this device ($\approx 1/C_{be2}R_{s2}$) is maximized. The CC stage transistor Q_1 is effectively a buffer that isolates C_μ of Q_2 from the source resistance R_s. The low-frequency voltage gain of this circuit is reduced when compared with a single-stage configuration because the input signal effectively appears across two base–emitter junctions.

The two-transistor configurations help to maintain a wideband frequency response by isolating the input and output circuits. In integrated circuit design, another method of neutralizing the effect of C_μ is possible when differential gain stages are used.

For example, Figure 3.18 shows a section of a differential input amplifier. If the inputs are driven differentially, then the collector voltages V_{c1} and V_{c2} will be 180° out of phase. The neutralization capacitors C_n thus inject a current into the base of each transistor that is equal and opposite to that caused by the intrinsic capacitance C_μ. Consequently, the neutralization capacitors should be equal to C_μ in order to provide good signal cancellation, and so they may be implemented from the junction capacitance of two dummy transistors with identical geometries to Q_1 and Q_2 as shown in Figure 3.19.

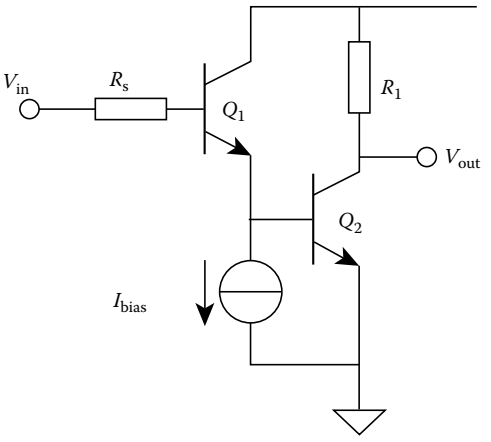


FIGURE 3.17 CC–CE stage.

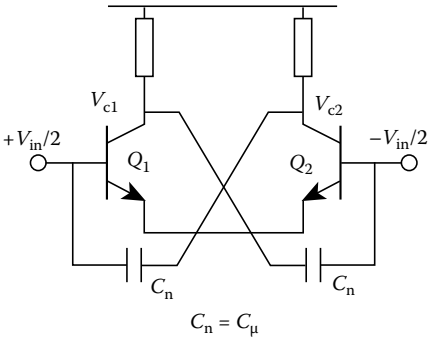


FIGURE 3.18 Differential gain stage.

3.1.6 Negative Feedback

Negative feedback is often employed around high-gain stages to improve the frequency response. In effect, the gain is reduced in exchange for a wider, flatter bandwidth. The transfer function of a closed-loop system can be written

$$H(s) = \frac{A(s)}{1 + A(s)B(s)} \tag{3.22}$$

where $A(s)$ is the open-loop gain and $B(s)$ is the feedback fraction. If the open-loop gain $A(s)$ is large, then $H(s) \approx 1/B(s)$. In RF design, compound or cascaded stages can produce excessive phase shifts that result in instability when negative feedback is applied. To overcome this problem, it is generally accepted to apply local negative feedback around a

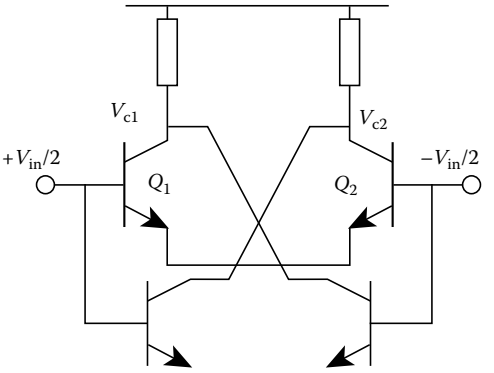


FIGURE 3.19 Implementation of neutralization capacitors.

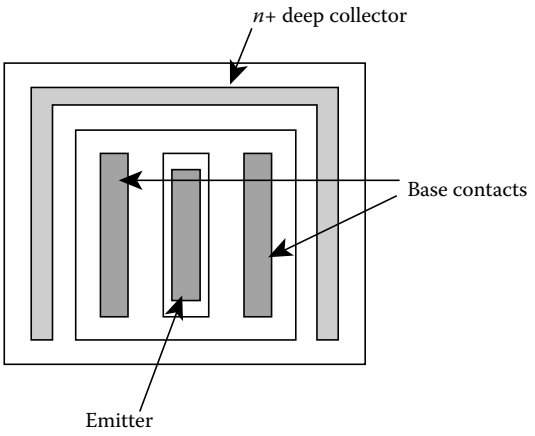


FIGURE 3.20 Stripe geometry.

single stage only. However, the open-loop gain of a single stage is usually too low for the approximation $H(s) = 1/B(s)$ to hold.

3.1.7 RF Bipolar Transistor Layout

When laying out RF transistors, the aim is to

- Minimize C_μ and C_π
- Minimize base width to reduce the forward transit time t_∂ and thus maximize ∂_t
- Minimize series resistance r_b and r_c

To minimize junction capacitance, the junction area must be reduced; however, this will tend to increase the series resistance. Transistors are generally operated at fairly high currents to maximize ∂_t. However, if the emitter gets too crowded, then the effective value of β will be reduced. The requirements given above are generally best met by using a stripe geometry of the type shown in Figure 3.20.

The stripe geometry maximizes the emitter area-to-periphery ratio, which reduces emitter crowding while minimizing the junction capacitance. The length of the emitter is determined by current-handling requirements. The base series resistance is reduced by having two base contacts and junction depths are minimized to reduce capacitance. The buried layer, or deep collector, reduces the collector series resistance. High-power transistors are produced by paralleling a number of transistors with interleaving "fingers," as shown in Figure 3.21. This preserves the frequency response of the stripe geometry while increasing the total current-handling capability.

3.1.8 Bipolar Current-Mode Broadband Circuits

Recently there has been strong interest in applying so-called current-mode techniques to electronic circuit design. Considering the signal operating parameter as a current and driving into low-impedance nodes has allowed the development of a wealth of circuits with broadband properties. Many of the following circuit and system concepts date back several years; it is progress in integrated circuit technology that has given a renewed impetus to "practical" current-mode techniques.

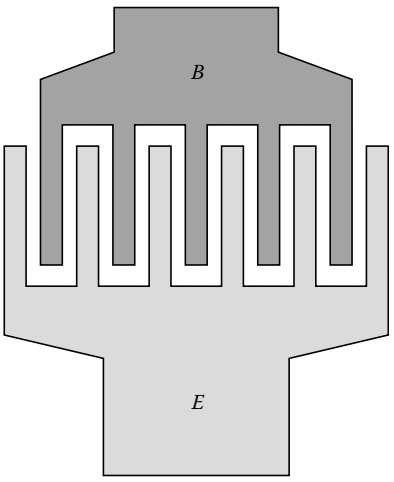


FIGURE 3.21 Transistor layout with interleaving fingers.

The NPN bipolar transistor, for example, is used predominantly in analog IC design because electron mobility is greater than hole mobility in silicon. This means that monolithic structures are typically built on P-type substrates, because vertical NPN transistors are then relatively easy to construct and to isolate from each other by reverse biasing the substrate.

Fabricating a complementary PNP device on a P-type substrate is less readily accomplished. An N type substrate must be created locally and the PNP device placed in this region. Early bipolar processes created PNP devices as lateral transistors and engineers dealt with their inherently poor, low-frequency characteristics by keeping the PNP transistors out of the signal path whenever possible.

However, high-speed analog signal-processing demands symmetrical silicon processes with fully complementary BJTs.

Newer, advanced processes have dielectrically isolated transistors rather than reversed-biased pn junction isolation. These processes are able to create separate transistors, each situated in a local semiconductor region. Then, both PNP and NPN devices are vertical and their performance characteristics are much more closely matched.

Dielectric isolation processes have revolutionized high-speed analog circuit design and have been key in making high-performance current-conveyor and current-feedback op-amp architectures practical. In the following sections, we will briefly review the development of the current-conveyor and current-feedback op-amp.

3.1.8.1 Current Conveyor

The current conveyor is a versatile broadband analog amplifier that is intended to be used with other circuit components to implement many analog signal-processing functions. It is an analog circuit building block in much the same way as a voltage op-amp, but it presents an alternative method of implementing analog systems that traditionally have been based on voltage op-amps. This alternative approach leads to new methods of implementing analog transfer functions, and in many cases the conveyor-based implementation offers improved performance when compared to the voltage op-amp-based implementation in terms of accuracy, bandwidth, and convenience. Circuits based on voltage op-amp are generally easy to design since the behavior of a voltage op-amp can be approximated by a few simple design rules. This is also true for current conveyors, and once the appropriate design rules are understood, the application engineer is able to design conveyor-based circuits just as easily.

The first-generation current conveyor (CCI) was proposed by Smith and Sedra in 1968 [1] and the more versatile second-generation current conveyor (CCII) was introduced by the same two authors in 1970 [2], as an extension of the CCI. The CCII, is without doubt the more valuable and adaptable building block of the two, and we will concentrate mostly on this device. Figure 3.22a shows the voltage–current describing matrix for the CCII, while Figure 3.22b shows the schematic normally used for the CCII with the power supply connections omitted.

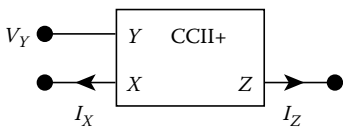
The voltage at the low-impedance input node X follows that at the high-impedance input node Y, while the input current at node X is mirrored or "conveyed" to the high-impedance output node Z. The $\pm$ sign indicates the polarity of the output current with respect to the input current; by convention, a positive sign indicates that both the input and output currents simultaneously flow into or out of the device, thus Figure 3.22b illustrates a CCII+. For the CCI, the input current at node X was reflected to input Y, that is the two inputs had equal currents. In the case of the second-generation conveyor input, Y draws no current, and this second generation, or CCII formulation, has proved to be much more adaptable and versatile than its first-generation predecessor. Because of the combined voltage and current following properties, CCIIs may be used to synthesize a number of analog circuit functions that are not so easily or accurately realizable using voltage op-amps.

Some of these application areas are shown in Figure 3.23. As current-conveyors become more readily available and circuits designers become more familiar with the versatility of this device, it is certain that further ingenious uses will be devised.

The ideal transistor and the current-conveyor. So far a transistor-level realization of the CCII has not been discussed. The current–voltage transfer relationship for the CCII+ is given by

$$\begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \pm 1 & 0 \end{bmatrix} \begin{bmatrix} V_Y \\ I_X \\ V_Z \end{bmatrix}$$

(a)



(b)

FIGURE 3.22 The CCII current conveyor. (a) I–V describing matrix. (b) Schematic.

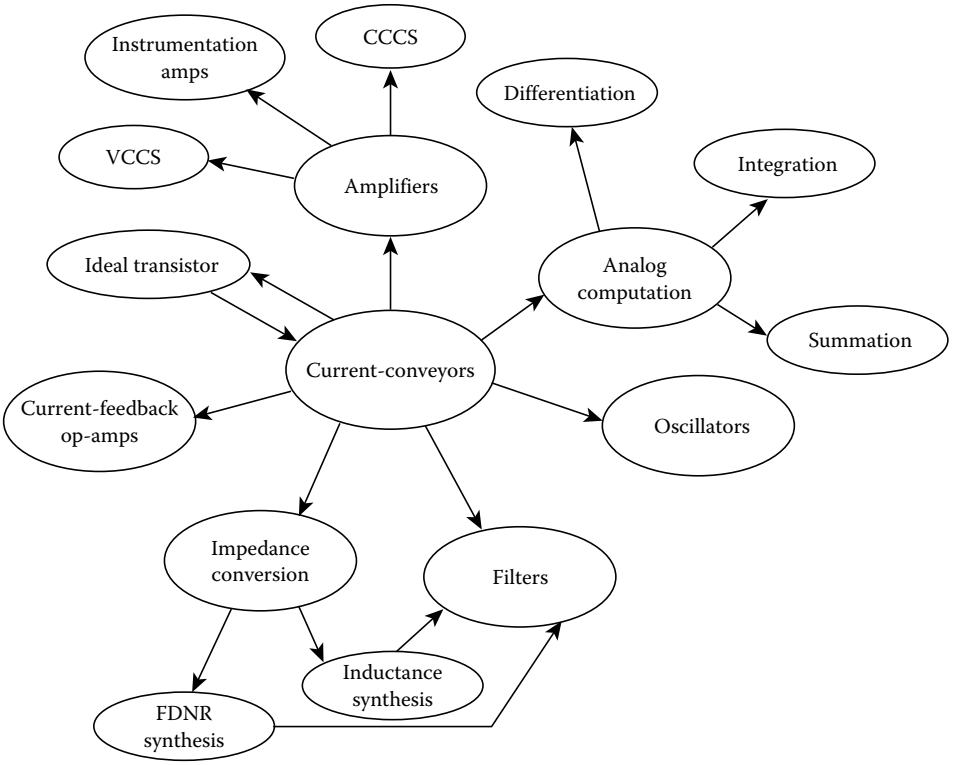


FIGURE 3.23 Current-conveyor applications.

$$V_X = V_Y, \quad I_Y = 0, \quad \text{and} \quad I_Z = I_X \tag{3.23}$$

These equations show that a simple voltage-following action exists between input node Y and output node X, and that there is a simple current-following action between input node X and output node Z. Also, these characteristic equations tell us that the impedance relationship for the ideal current conveyor is

$$Z_{\text{in}Y} = \infty, \quad Z_X = 0, \quad \text{and} \quad Z_{\text{out}Z} = \infty \tag{3.24}$$

Figure 3.24 shows a schematic representation of a CCII—built with a single BJT and on reflection it is clear that the current conveyor is effectively an ideal transistor, with infinite β and infinite g_m.

Driving into the base of a BJT gives almost unity voltage gain from input base to output emitter, with high input impedance and low-output impedance, and driving into the emitter of a BJT gives almost unity current gain from emitter input to collector output, with low input impedance and high output impedance. Drawing the comparison further, the high-input-impedance Y node corresponds to the base (or gate) of a transistor, the low-input-impedance X node corresponds to the emitter (or source) of a transistor, and the high-output-impedance Z node corresponds to the collector (or drain) of a transistor. Clearly, one transistor cannot function alone as a complete current conveyor since an unbiased single transistor at best can only handle unipolar signals and the high-accuracy unity voltage and unity current gain required for a high-performance current conveyor cannot be obtained. However, the generic relationship between the current conveyor and an ideal transistor is valid, and it provides valuable insight into the development and operation of monolithic current conveyors described in the next section.

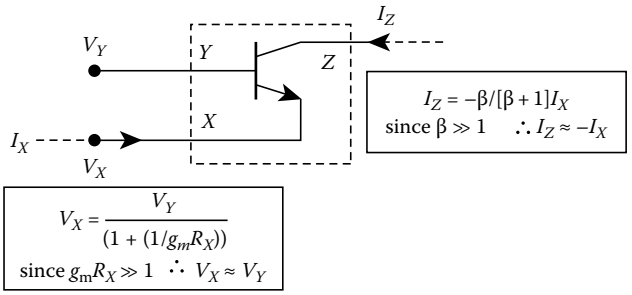


FIGURE 3.24 Single BJT CCII–.

Supply-current sensing. Many of the current-conveyor theories and applications have been tested out in practice using "breadboard" conveyor circuits, due to the lack of availability of a commercial device. Some researchers have built current conveyors from matched transistor arrays, but the most common way of implementing a fairly high-performance current conveyor has been based on the use of supply-current sensing on a voltage op-amp [3,4], as shown in Figure 3.25. The high-resistance op-amp input provides the current-conveyor *Y* node, while the action of negative feedback provides the low-resistance *X* node. Current-mirrors in the op-amp supply leads copy the current at node *X* to node *Z*.

Using this type of architecture, several interesting features soon became apparent. Consider the two examples shown in Figure 3.26. In Figure 3.26b, R_s represents the output resistance of the current source. The open-loop gain of an op-amp can generally be written

$$\frac{V_{out}}{V_{in}} = \frac{A_o}{1 + j(f/f_o)} \tag{3.25}$$

where A_o is the open-loop direct current (dc) gain magnitude and ∂_o is the open-loop -3 dB bandwidth. Since $A_o \gg 1$, the transfer function of the voltage follower of Figure 3.26a can be written as

$$\frac{V_{out}}{V_{in}} \approx \frac{1}{1 + j(f/GB)} \tag{3.26}$$

where $GB = A_o \partial_o$. From Equation 3.26, the -3 dB bandwidth of the closed-loop voltage follower is equal to the open-loop gain-bandwidth product or *GB* of the op-amp. If the op-amp is configured instead to give a closed-loop voltage gain *K*, it is well known that the closed-loop bandwidth correspondingly reduces by the factor *K*.

The transfer function for the current-follower circuit of Figure 3.26b, as shown in Ref. [4], is given by

$$\frac{I_{out}}{I_{in}} \approx \lambda \frac{1 + j(f/GB)}{1 + j(f/kGB)} \tag{3.27}$$

where λ is the current transfer ratio of the current mirrors and $k = (R_s + r_o/A_o)/(R_s + r_o)$, and r_o represents the output resistance of the op-amp. Since $A_o \gg R_s \gg r_o$, then $K \approx 1$, and the pole and zero in Equation 3.27 almost

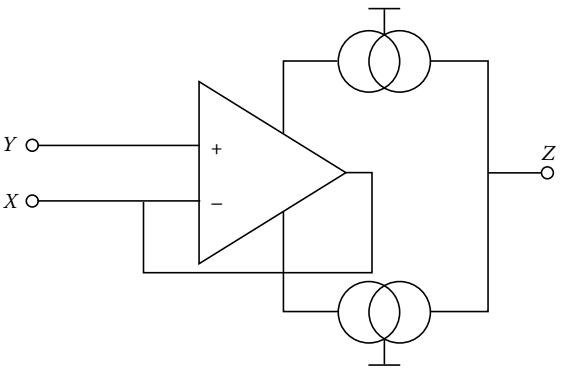


FIGURE 3.25 Supply-current sensing on a voltage op-amp.

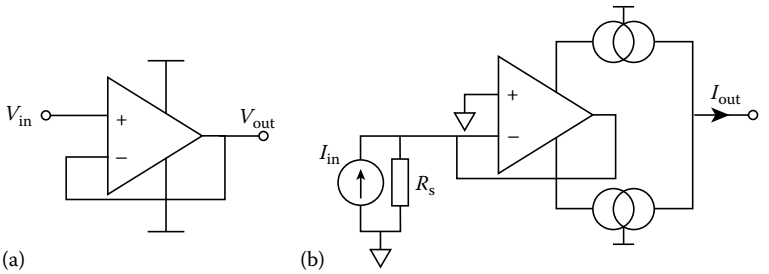


FIGURE 3.26 (a) Voltage follower. (b) Current follower.

cancel. The current-follower circuit thus operates well above the gain-bandwidth product *GB* of the op-amp, and the −3 dB frequency of this circuit will be determined by higher frequency parasitic poles within the current mirrors.

This "extra" bandwidth is achieved because the op-amp is being used with input and output nodes held at virtual ground. The above example is generic in the development of many of the circuits that follow. It demonstrates that reconfiguring a circuit topology to operate with current signals can often result in a superior frequency performance.

First-generation current conveyors. Smith and Sedra's original paper presenting the first-generation CCI current conveyor showed a transistor-level implementation based on discrete devices, shown in Figure 3.27. Assuming that transistors Q_3–Q_5 and resistors R_1–R_3 are matched, then to first order the currents through these matched components will be equal. Transistors Q_1 and Q_2 are thus forced to have equal currents, and equal V_{be}s. Input nodes *X* and *Y* therefore track each other in both voltage and current. In practice, there will be slight differences in the collector currents in the different transistors, due to the finite β of the devices. These differences can be reduced, for example, by using more elaborate current mirrors. The polarity of the output current at node *Z* can be inverted easily by using an additional mirror stage, and the entire circuit can also be inverted by replacing NPN transistors with PNPs, and vice versa. Connecting two complementary current conveyors, as shown in Figure 3.28, results in a class *AB* circuit capable of bipolar operation. Note that in practice this circuit may require additional components to guarantee start-up.

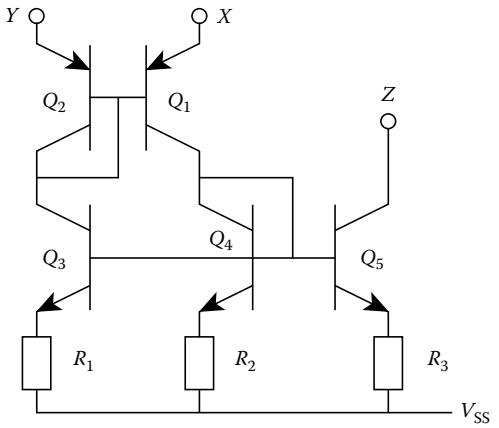


FIGURE 3.27 First-generation current conveyor.

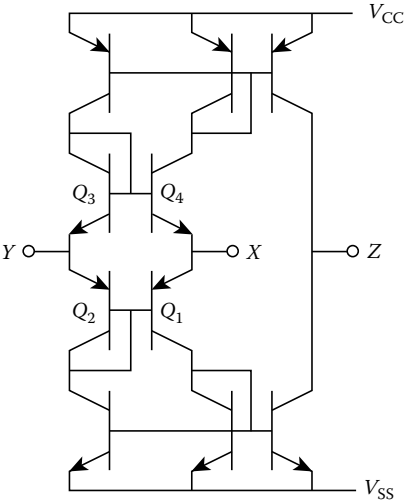


FIGURE 3.28 Class *AB* current conveyor.

An integrated current conveyor based on the architecture shown in Figure 3.27 is commercially available as the PA630 [5], and the basic topology of this device is shown in Figure 3.29. An NPN Wilson mirror (Q_1–Q_3) and a PNP Wilson mirror (Q_4–Q_6) are used to provide the current and voltage following properties between inputs X and Y, similar to the circuit of Figure 3.27. Taking a second output from the PNP current mirror to provide the Z output would destroy the base-current compensation scheme of the Wilson mirror. Therefore, a second NPN Wilson mirror (Q_7–Q_9) is used to perform a current-splitting action and so the combined emitter current of Q_7 and Q_8 is divided in two, with one half being shunted via Q_9 to the supply rail, and the other half driving an output PNP Wilson mirror (Q_{10}–Q_{12}). This results in an output current at node Z that to first order is virtually equal to that at the X and Y inputs. Q_{13} is included to ensure that the device always starts up when turned on. The complete architecture of the PA630 CCI also includes frequency compensation to ensure stability, and modified output current mirrors that use the ''wasted'' collector current of Q_9 to effectively double the output resistance at node Z. A full description of the architecture and operation of this device can be found in Ref. [6].

The current-conveyor architecture shown in Figure 3.29 includes both NPN and PNP transistors in the signal path, and thus the bandwidth and current-handling capability of this device will be poor if only lateral PNPs are available. The development of complementary bipolar processes, with vertical PNP as well as NPN transistors, has made possible the implementation of high-performance integrated circuit current conveyors.

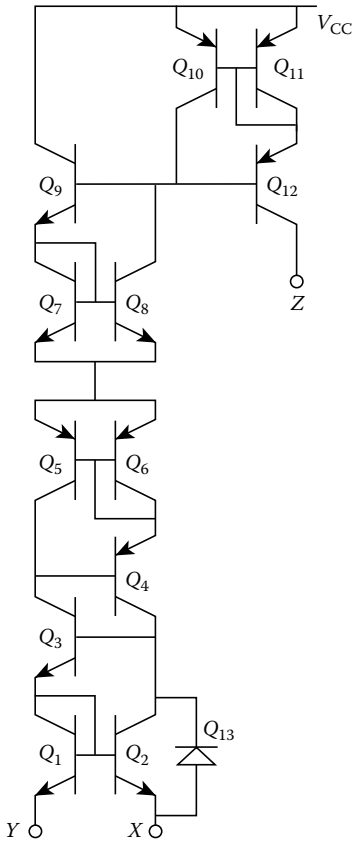


FIGURE 3.29 Simplified PA630 current conveyor.

Second-generation current conveyors. A CCII can also be simply implemented on a complementary bipolar process, by replacing the diode at the CCI Y input with a transistor, and taking the input from the high resistance base terminal, as shown in Figure 3.30a. This can be extended to a class AB version, as shown in Figure 3.30b. Referring to Figure 3.30b, transistors Q_1–Q_4 act as a voltage buffer that transfers the voltage at node Y to node X. The current source and sink ($I_{B1} = I_{B2} = I_B$) provide the quiescent bias current for these input transistors. Any input current (I_x) at node X is split between Q_2 and Q_3, and is copied by current mirrors CM_1 and CM_2 to the output node Z. This CCII architecture forms the basis of the commercially available CCII01 current conveyor [7]. As we shall see later, it is also used as the basic input stage of the current-feedback op-amp, which has emerged as a high-speed alternative to the more conventional voltage op-amp [8].

The simple CCII architecture of Figure 3.30b will clearly exhibit a quiescent voltage offset between nodes X and Y due to the mismatch between the V_{be}s of the NPN and PNP transistors Q_1/Q_2 and Q_3/Q_4, as

$$\begin{aligned} V_Y - V_X &= V_{BE}(p) - V_{BE}(n) \\ &= V_T \ln\left(I_{sp}/I_{sn}\right) \end{aligned} \tag{3.28}$$

where I_{sp} and I_{sn} are the reverse saturation currents of the PNP and NPN transistors, respectively, and V_T is the thermal voltage. This process-dependent voltage offset can be reduced by including additional matching diodes in the input stage, as shown in Figure 3.31. Referring to this diagram,

$$\begin{aligned} V_Y - V_X &= V_{BE}(Q_1) + V_{D_2} - V_{BE}(Q_2) - V_{D_1} \\ V_Y - V_X &= [V_{BE}(Q_1) - V_{D_1}] - [V_{BE}(Q_2) - V_{D_2}] \end{aligned} \tag{3.29}$$

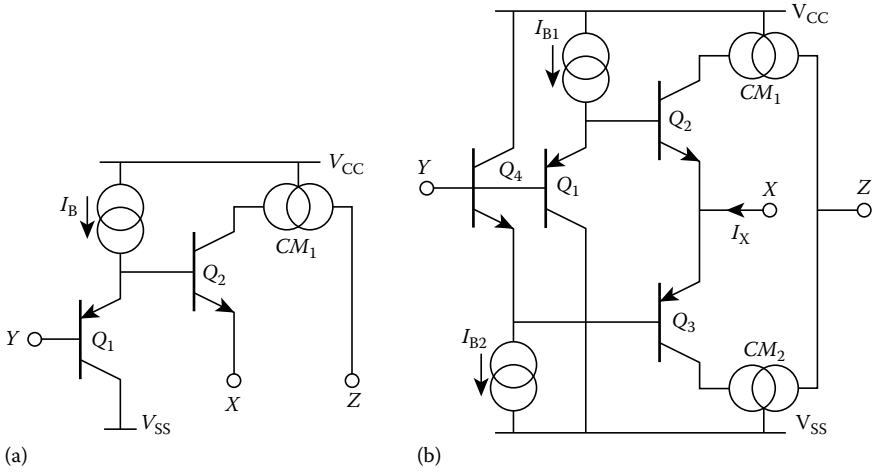


FIGURE 3.30 (a) Class *A* CCII. (b) Class *AB* CCII.

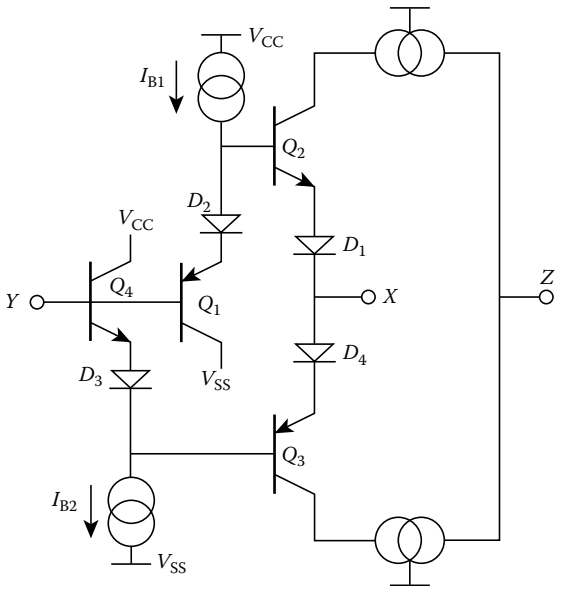


FIGURE 3.31 CCII with input matching diodes.

Inclusion of these diodes clearly reduces the quiescent input voltage offset, provided that D_1 is matched to Q_1, D_2 is matched to Q_2, etc. However, the addition of diodes D_1 and D_2 has several disadvantages. First, the input voltage dynamic range of the circuit will be reduced by the forward voltage across the additional diode. Second, the small-signal input resistance seen looking into node X will be double that for the basic architecture given in Figure 3.30b. This nonzero input resistance at node X (R_x) will compromise the performance of the current conveyor, especially in applications where a nonzero input voltage is applied at node Y. The effect of the small-signal input resistance R_x is to produce a signal-dependent voltage offset V_d between nodes X and Y, where

$$V_{\dot{a}} = R_x I_x \tag{3.30}$$

Since the value of R_x is determined by the small-signal resistance $(r_{e2} + r_{d2})$ in parallel with $(r_{e3} + r_{d3})$, its value could be reduced by increasing the value of the quiescent bias current I_B. However, an increase in bias current will lead to an increase in the total power consumption, as well as a possible increase in offsets, and so is certainly not an ideal solution. Further techniques for CCII implementation are discussed in Ref. [14].

The previous conveyor is typical of commercial conveyor architectures [7], which are generally built on a high-speed dielectric isolation (fully complementary) bipolar process. Such devices feature an equivalent slew rate of some 2000 V/μs and a bandwidth of around 100 MHz.

Until high-performance current conveyors are widely available, these devices will continue to be used in research laboratories rather than in the applications arena. Process technologies and design techniques

have now advanced to the stage where the implementation of an integrated current conveyor is both desirable and viable, and a whole host of applications are waiting for its arrival.

3.1.8.2 Current-Feedback Operations Amplifier

In this section, the design and development of a high-gain wide-bandwidth transimpedance or current-feedback operational amplifier is considered. The design of conventional operational amplifiers has remained relatively unchanged since the introduction of the commercial operational amplifier in 1965. Recently, a new amplifier architecture, called a current-feedback operational amplifier, has been introduced. This amplifier architecture is basically a transimpedance amplifier, or a current-controlled voltage source, while the classical voltage-feedback operational amplifier is a voltage-controlled voltage source.

The current-feedback operational amplifier has two major advantages, compared to its voltage-feedback counterpart. First, the closed-loop bandwidth of the current-feedback amplifier is larger than that of classical voltage-feedback design for comparable open-loop voltage gain. Second, the current-feedback operational amplifier is able to provide a constant closed-loop bandwidth for closed-loop voltage gains up to about 10. A further advantage of the current-feedback architecture is an almost unlimited slew rate due to the class-*AB* input drive, which does not limit the amount of current available to charge up the compensation capacitor as is the case in the conventional voltage-feedback op-amp. This high-speed performance of the current-feedback operational amplifier is extremely useful for analog signal-processing applications within video and telecommunication systems.

The generic relationship between the CCII+ and the current-feedback op-amp is extremely close and several of the features offered by the CCII are also present in the current-feedback op-amp. The basic structure of the current-feedback op-amp is essentially that of a CCII+ with the Z node connected directly to an output voltage follower, as shown in Figure 3.32. Any current flowing into the low-impedance inverting input is conveyed to the gain node (Z_T), and the resulting voltage is buffered to the output. Z_T is thus the open-loop transimpedance gain of the current-feedback op-amp, which in practice is equal to the parallel combination of the CCII+ output impedance, the voltage buffer input impedance and any additional compensation capacitance at the gain node. Generally, in current-feedback op-amps, the gain node is not connected to an external pin, and so the Z node of the CCII+ cannot be accessed.

Current-feedback op-amp architecture. In the following sections, we review the basic theory and design of the current-feedback op-amp and will identify the important features and mechanisms that result in broadband performance. We will begin by reviewing the voltage-feedback op-amp and comparing it with the current-feedback op-amp in order to see the differences clearly.

A schematic of the classical voltage-feedback op-amp comprising a long-tail pair input stage is shown in Figure 3.33a, which contrasts a typical current-feedback architecture, which is shown in Figure 3.33b. In both circuits, current mirrors are represented by two interlocking circles with an arrow denoting the input side of the mirror.

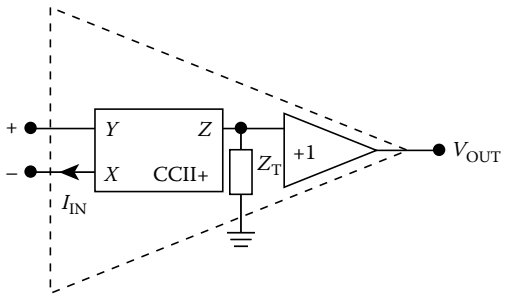


FIGURE 3.32 Current-feedback op-amp structure.

The current-feedback op-amp of Figure 3.33b shows that the noninverting input is a high-impedance input that is buffered to a low-impedance inverting terminal via a class *AB* complementary common-collector stage (Q_1, Q_2, D_1, D_2). Note that this classical input buffer architecture is used here for simplicity. In practice, a higher performance topology such as that described in Figure 3.31 would more likely be employed. The noninverting input is a voltage input; this voltage is then buffered to the inverting low-impedance current input to which feedback is applied. In contrast, both the noninverting and

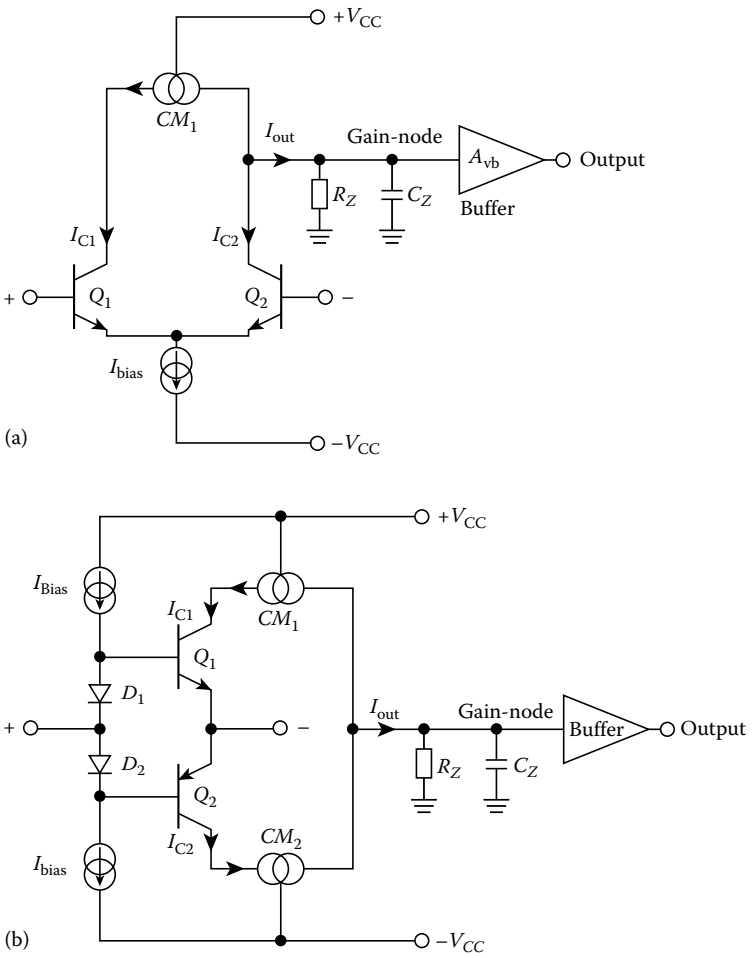


FIGURE 3.33 (a) Simplified classic voltage-feedback op-amp architecture. (b) Typical current-feedback op-amp architecture.

inverting input of the voltage-feedback op-amp are high-impedance voltage inputs at the bases of transistors Q_1 and Q_2.

In both architectures, the collector currents of Q_1 and Q_2 are transferred by the current mirror to a high-impedance node represented by resistance R_Z and capacitance C_Z. This voltage is then transferred to the output by voltage buffers that have a voltage gain A_{vb}, providing the necessary low-output impedance for current driving. In the case of the current-feedback op-amp, the output buffer is usually the same topology as the input buffer stage shown in the Figure 3.33b, but with slightly higher output current bias levels and larger output devices to provide an adequate output drive capability. Ideally, the bias currents I_{CQ1} and I_{CQ2} will be canceled at the gain node giving zero offset current.

Differential-mode operation of the current-feedback op-amp. A schematic diagram of the current-feedback op-amp with a differential input voltage applied at the noninverting and inverting input is shown in Figure 3.34.

The positive input voltage is applied to the base of transistor Q_1 (NPN) via D_1, and the negative input voltage is applied to the emitter of Q_1, causing the V_{BE} of Q_1 to increase and the V_{BE} of Q_2 to

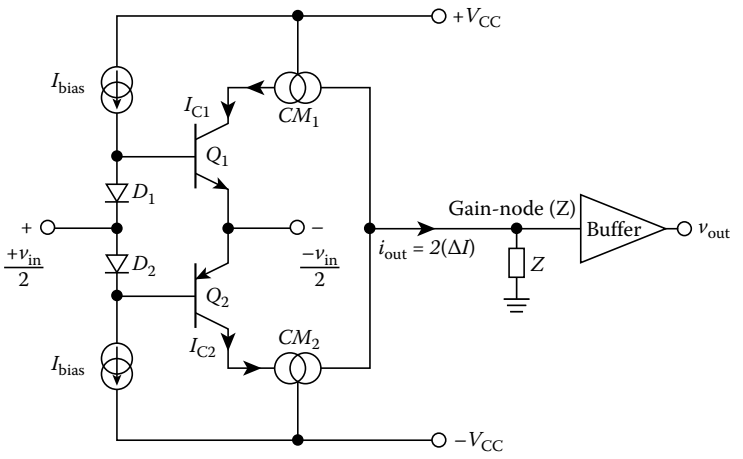


FIGURE 3.34 Current-feedback op-amp with differential input voltage applied.

reduce. I_{C1} will therefore increase by an amount ΔI and so I_{C2} will decrease by the same amount $-\Delta I$. A net current of $2\Delta I$ is therefore sourced out of the high-impedance node (Z) giving rise to a positive voltage ($2\Delta IZ$). This voltage is then buffered to the output.

With negative feedback applied around the current-feedback op-amp, the low-impedance inverting input will sense the current "feedback" from the output via the feedback network. This feedback current flowing into the inverting input is given by

$$i_{in-} = I_{C2} - I_{C1} \tag{3.31}$$

The difference between the collector current I_{C1} and I_{C2}, i_{in-}, will thus be driven into gain node Z, giving rise to the output voltage

$$V_{out} = Zi_{in-} \tag{3.32}$$

It is clear that the output voltage is dependent on the current that flows into the inverting input, hence the amplifier has a high open-loop transimpedance gain Z.

Closed-loop noninverting operation of the current-feedback op-amp. A schematic diagram of the current-feedback op-amp connected with negative feedback as a noninverting amplifier is shown in Figure 3.35. For a positive input voltage v_{in}, the output voltage v_{out} will swing in the positive direction and the inverting input current i_{in-} will flow out:

$$i_{in-} = \frac{v_{in-}}{R_1} - \frac{(v_{out} - v_{in-})}{R_2} \tag{3.33}$$

The input stage is simply a voltage follower and so ideally, $v_{in+} = v_{in-} = v_{in}$. Because $v_{out} = Zi_{in-}$, then substituting for v_{in-} and i_{in-} in Equation 3.33 yields

$$\frac{v_{out}}{Z} = \frac{v_{in}}{R_1} - \frac{(v_{out} - v_{in})}{R_2} \tag{3.34}$$

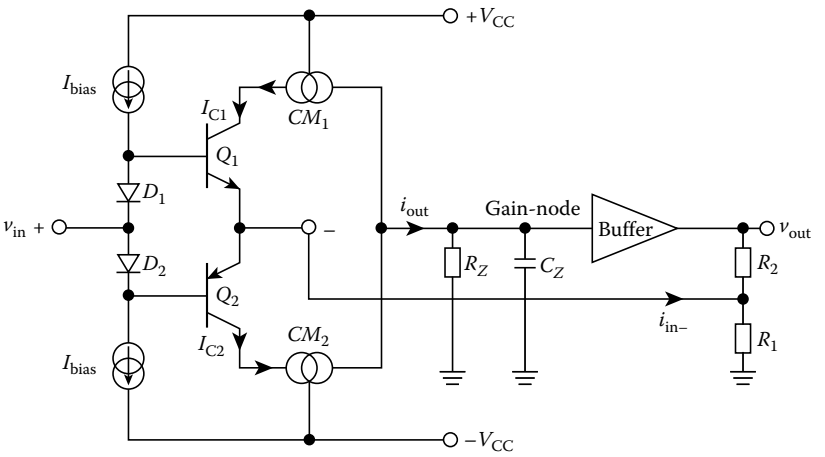


FIGURE 3.35 Noninverting current-feedback op-amp.

rearranging for v_{out}/v_{in}

$$v_{out}\left(\frac{1}{R_2}+\frac{1}{Z}\right)=v_{in}\left(\frac{1}{R_1}+\frac{1}{R_2}\right) \tag{3.35}$$

$$\frac{v_{out}}{v_{in}}=\left(1+\frac{R_2}{R_1}\right)\left(\frac{1}{1+(R_2/Z)}\right) \tag{3.36}$$

This result shows that the closed-loop noninverting gain of the current-feedback op-amp is similar to that of a classical voltage-feedback op-amp. From Equation 3.36, the open-loop transimpedance gain Z must be as large as possible to give good closed-loop gain accuracy. Since v_{out}/Z represents the error current i_{in-}, then maximizing the Z term will minimize the inverting error current. Note that at this stage it is only the R_2 term in the denominator of the second term in Equation 3.36 that sets the bandwidth of the amplifier; the gain-setting resistor R_1 has no effect on the closed-loop bandwidth.

Closed-loop inverting operation of current-feedback op-amp. A current-feedback op-amp connected as an inverting amplifier is shown in Figure 3.36. The low-impedance inverting input samples the input current and drives the output until the voltage at its terminal is at a virtual ground because of negative feedback. Ideally the closed-loop gain is given by

$$A_{CL}=-\frac{R_2}{R_1} \tag{3.37}$$

From Figure 3.36, application of Kirchhoff's current law to the current i_1, i_{in-}, and i_2 gives

$$i_{in-}+i_2=i_1$$
$$i_{in-}-\frac{v_{out}}{R_2}=\frac{v_{in}}{R_1}$$

because $v_{out}/Z=-i_{in-}$, then

$$-\frac{v_{out}}{Z}-\frac{v_{out}}{R_2}=\frac{v_{in}}{R_1}$$

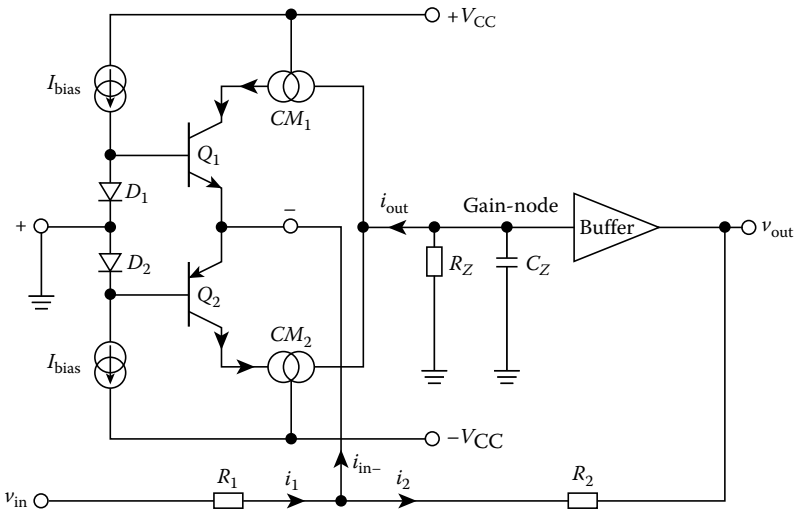


FIGURE 3.36 Inverting current-feedback op-amp amplifier.

which can be rearranged as

$$\frac{v_{\text{out}}}{v_{\text{in}}} = -\frac{R_2}{R_1}\left(\frac{1}{1+\frac{R_2}{Z}}\right) \tag{3.38}$$

Again, the high-Z term is required to provide good closed-loop gain accuracy.

More detailed analysis of the current-feedback op-amp. A simplified macromodel of the current-feedback architecture configured as a noninverting amplifier is shown in Figure 3.37. The input stage is represented by a semi-ideal voltage buffer to the inverting input. The output resistance of the input stage buffer R_{inv} is included since it has a significant effect on the bandwidth of the amplifier, as will be shown later. The current that flows out from the inverting terminal i_3 is transferred to the gain node, which is represented by R_Z and C_Z, via a current mirror that has a current gain K. The voltage at the gain node is transferred to the output in the usual way by a voltage buffer, with voltage gain A_{vb}. The net transfer function is given by

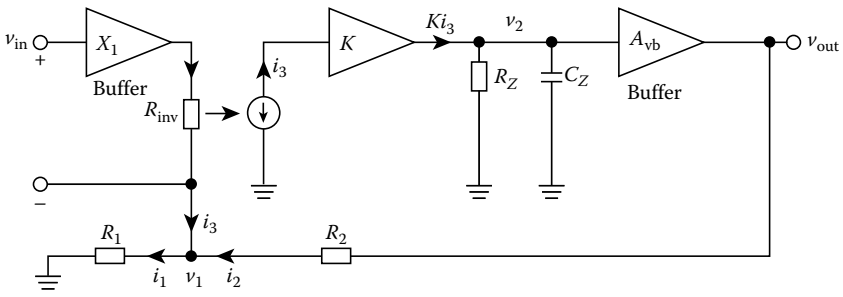


FIGURE 3.37 Inverting amplifier with current-feedback op-amp macromodel.

$$\frac{v_{out}}{v_{in}} = \frac{1 + \dfrac{R_2}{R_1}}{1 + j\omega C_Z \left[\dfrac{R_{inv}\left(1 + \dfrac{R_2}{R_1}\right) + R_2}{A_{vb}K} \right]} \tag{3.39}$$

Hence, the pole frequency is also given by

$$f_{-3dB} = \frac{A_{vb}K}{2\pi C_Z \left[R_{inv}\left(1 + \dfrac{R_2}{R_1}\right) + R_2 \right]} \tag{3.40}$$

(A full derivation of this transfer function is given in Appendix A.)

To compare this result to the classical voltage-mode op-amp architecture, a simplified schematic diagram of the voltage-feedback op-amp configured as a noninverting amplifier is shown in Figure 3.38.

Again from a full analysis, given in Appendix B, the transfer function obtained is

$$\frac{v_{out}}{v_{in}} = \frac{1 + \dfrac{R_2}{R_1}}{1 + j\omega \left[\dfrac{R_z C_Z}{1 + \dfrac{g_m A_{vb} R_Z}{\left(1 + \dfrac{R_2}{R_1}\right)}} \right]} \tag{3.41}$$

The pole frequency is given by

$$f_{-3dB} = \frac{1 + \dfrac{g_m A_{vb} R_Z}{\left(1 + \dfrac{R_2}{R_1}\right)}}{2\pi R_Z C_Z} \tag{3.42}$$

Pole frequency comparison. If one compares the closed-loop pole frequency Equations 3.40 and 3.42 for the current-feedback and voltage-feedback op-amp, respectively, it is clear that the bandwidth of the voltage-feedback op-amp is dependent on the closed-loop gain $(1 + R_2/R_1)$ resulting in the well-known constant gain-bandwidth product $f_{max} = (A_v)_{CL} f_T$. This means that an increase in the closed-loop gain results in a decrease in the bandwidth by the same factor as illustrated in Figure 3.39. In contrast, the pole

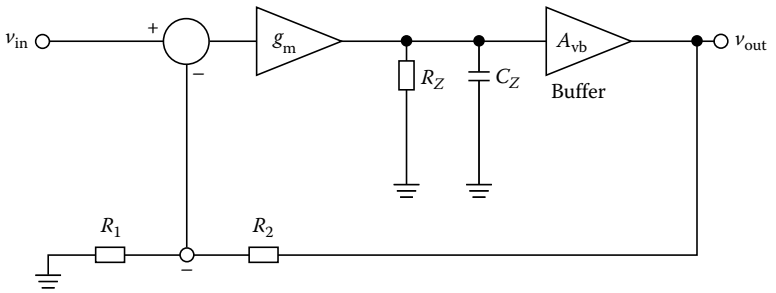


FIGURE 3.38 Noninverting amplifier with voltage-feedback op-amp macromodel.

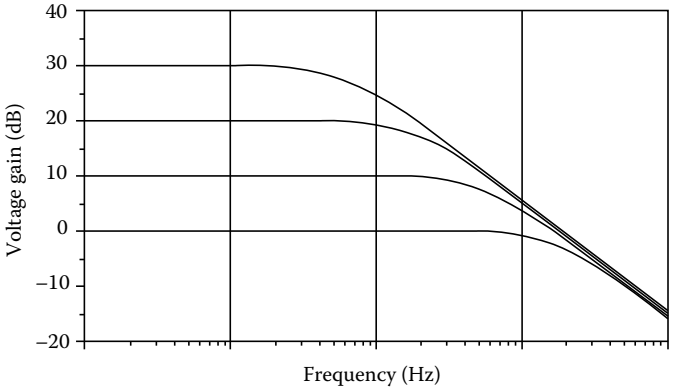


FIGURE 3.39 Frequency response of voltage-feedback op-amp amplifier for various closed-loop gains.

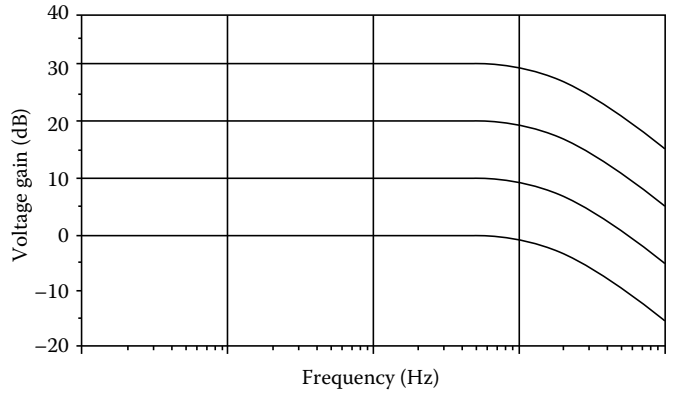


FIGURE 3.40 Frequency response of current-feedback op-amp amplifier for various closed-loop gains.

frequency of the current-feedback op-amp is directly dependent on R_2 and can be set almost independently of the closed-loop gain. Thus, the closed-loop bandwidth is almost independent of closed-loop gain as shown in Figure 3.40, assuming that R_{inv} is close to zero. Intuitively, this is the case since the feedback error current that is set by the feedback resistor R_2 is the current available to charge up the compensation capacitor. However, if one considers Equation 3.40 in some detail it can be seen that for high closed-loop gains and a nonzero R_{inv}, then the R_{inv} term starts to dictate and so the bandwidth will become more dependent on the closed-loop gain.

Slow rate of the current-feedback op-amp. As mentioned earlier, one other advantage of the current-feedback op-amp over the classical voltage-feedback op-amp is the high slew rate performance. For the classical long-tail, or emitter-coupled pair input stage shown in Figure 3.41, the maximum current available to charge up the compensation capacitor C_Z at the gain node is I_{bias}, and this occurs when Q_1 or Q_2 is driven fully on. The resulting transconductance plot shown in Figure 3.42 limits the slew rate of the amplifier.

In contrast, the slew rate of the current-feedback op-amp is virtually infinite, as can be seen from the input stage schematic shown in Figure 3.43. Referring to Figure 3.43, a change in the input voltage ΔV_{in} at $V(+)$ will be copied by the input buffer to $V(-)$. When connected as noninverting amplifier, the current through R_1 will change by $\Delta V_{in}/R_1$, while the current through R_2 will change by $\Delta V_{in}/R_2$, since

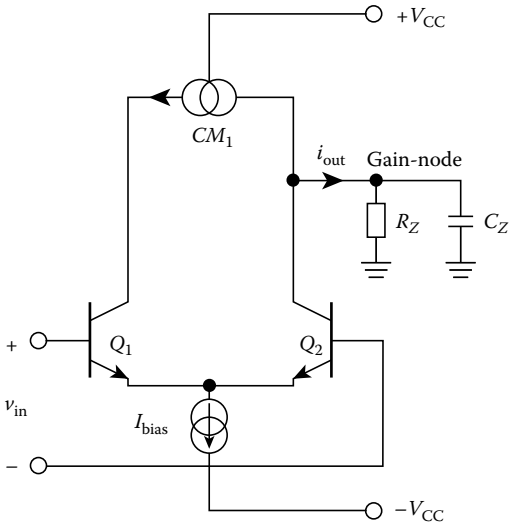


FIGURE 3.41 Long-tail pair input stage.

the output voltage at this point remains stationary. The total change in current through R_1 and R_2 must be supplied by the internal input buffer, and will be $\Delta I(-) = \Delta V_{in}((R_2 + R_1)/(R_2 \times R_1))$. This large input error current causes a rapid change in the output voltage, until V_{out} is again at the value required to balance the circuit once more, and reduce $I(-)$ to zero. The larger the input voltage slew rate, the larger the change in input error current, and thus the faster the output voltage slew rate. Current-feedback op-amps theoretically have no slew-rate limit. A typical current-feedback op-amp will exhibit a slew rate of between 500 and 2000 V/μS.

An analysis of this input stage (see Appendix C) shows that the transconductance follows a sinh(x) type function, as shown in Figure 3.44. In theory, this characteristic provides nearly unlimited slew-rate capability [9]. However, in practice a maximum slew rate will be limited by the maximum current drive into the gain node, which depends on the power dissipation of the circuit, the ability of power supply to deliver sufficient current, and the current-handling capability of the current mirrors.

Wideband and high-gain current-feedback op-amp. Previously, we have shown that the bandwidth of the current-feedback op-amp is almost independent of the closed-loop gain setting. Therefore, the closed-loop gain-bandwidth GB increases linearly with the closed-loop gain. However, the bandwidth of the practical current-feedback op-amp starts decreasing with high gain as a result of the finite inverting-input impedance [10], as shown by Equation 3.40. This is because for high gain, $R_{inv}(1 + R_2/R_1) > R_2$, and so the $R_{inv}(1 + R_2/R_1)$ term dominates the expression for closed-loop bandwidth, resulting in a direct conflict between gain and bandwidth.

At low gains when $R_2 > R_{inv}(1 + R_2/R_1)$, the closed-loop pole frequency is determined only by the compensation capacitor and the feedback resistor R_2. Thus, the absolute value of the feedback resistor R_2 is important, unlike the case of the voltage-feedback op-amp. Usually, the manufacturer species a minimum value of R_2 that will maximize bandwidth but still ensure stability. Note that because of the

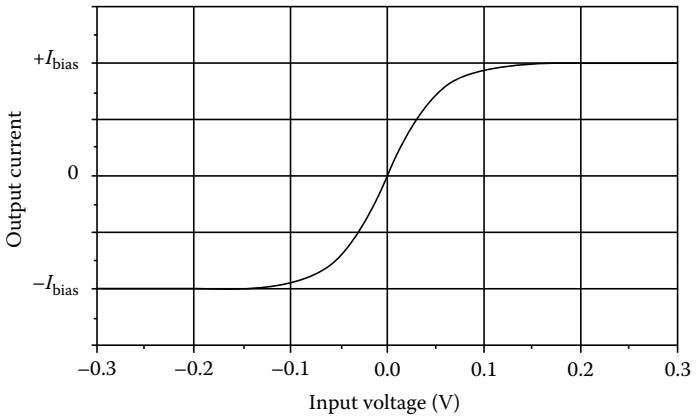


FIGURE 3.42 Long-tail pair input transconductance.

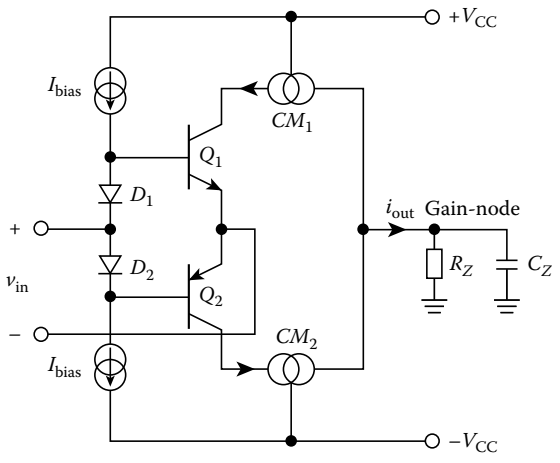


FIGURE 3.43 Current-feedback op-amp input stage.

inherent architecture a very high bandwidth can be achieved with the current-feedback design for a given value of R_2.

In practice, for gains higher than about 10, the $R_{inv}(1 + R_2/R_1)$ term in Equation 3.40 becomes dominant and the amplifier moves toward constant gain-bandwidth product behavior. The *GB* can be increased by reducing R_2 [11] but this will compromise stability and/or bandwidth, or alternatively, C_Z can be reduced. The latter option is limited since the minimum value of C_Z is determined by the device parameters and layout parasitics. Two possible ways of improving the high-gain constant bandwidth capability of the current-feedback op-amp can be seen by inspection of Equation 3.40. Either the K factor, which represents current gain in the current mirrors at the Z-node can be increased from unity to increase the bandwidth as it rolls off with high gain, or the inverting input impedance of the amplifier should be reduced toward zero. In the following section we consider the design of a suitable broadband variable-gain current-mirror circuit with a possible application being to improving the maximum bandwidth capability of current-feedback op-amps.

Basic current mirror. A typical current-feedback op-amp circuit is shown in Figure 3.45. It includes a complementary common-collector input stage (Q_1–Q_4) and a similar output buffer (Q_5–Q_8), with linking cascode current mirrors setting the Z-node impedance (Q_{12}–Q_{14}, Q_9–Q_{11}). The cascoded mirror provides unity current gain. Any attempt to increase the current gain via emitter degeneration usually results in much poorer current-mirror bandwidth. Consider now the development of a suitable broadband, variable gain current mirror.

A schematic diagram of a simple Widlar current mirror and its small-signal equivalent circuit are shown in Figures 3.46 and 3.47, respectively. For simplicity, we will assume that the impedance of the diode-connected transistor Q_1 is resistive and equal to R_D. The dc transfer function of the mirror is derived in Appendix D and is given by

$$\frac{I_{out}}{I_{in}} = \frac{\beta}{\beta + 2} \tag{3.43}$$

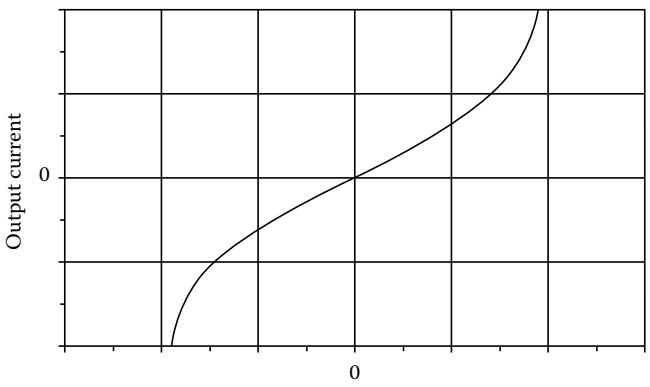


FIGURE 3.44 Input-stage transconductance of the current-feedback op-amp.

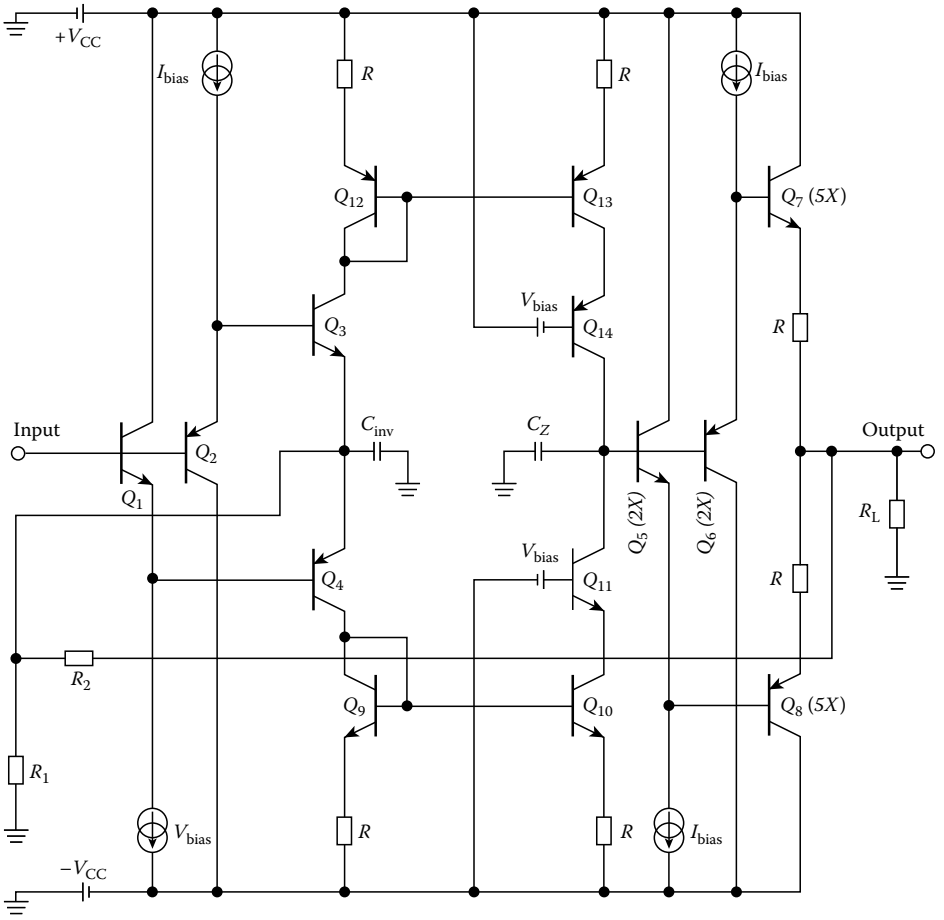


FIGURE 3.45 Transistor-level schematic of a typical current-feedback op-amp. $x =$ Unit transistor area.

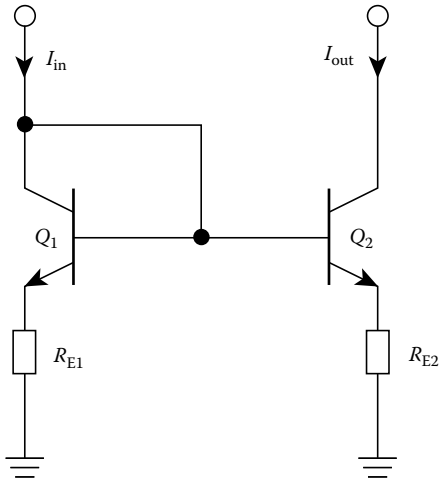


FIGURE 3.46 Simple Widlar current mirror with emitter degeneration.

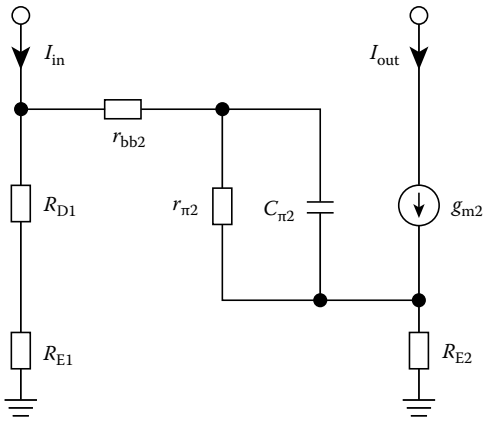


FIGURE 3.47 Small-signal equivalent circuit of Figure 3.46 current mirror.

and the −3 dB bandwidth is given by

$$f_{-3\text{ dB}} = \frac{1}{2\pi C_\pi \left\{ \dfrac{r_{\pi 2}(r_{\text{bb2}} + R_D)}{r_\pi + r_{\text{bb2}} + R_D} \right\}} \tag{3.44}$$

In order to increase the current gain it is usual to insert an emitter-degeneration resistor R_{E1} in the emitter of Q_1. The dc transfer function, derived in the Appendix E, is then

$$I_{\text{in}} R_{\text{E1}} = V_{\text{T}} \ln \frac{I_{\text{out}}}{I_{\text{in}}} \tag{3.45}$$

and the ac small-signal current gain is given by

$$\frac{i_{\text{out}}}{i_{\text{in}}} = (R_{\text{E1}} + R_{\text{D1}}) g_{\text{m2}} \tag{3.46}$$

where

$$R_{\text{D1}} = \frac{I_{\text{in}}}{\frac{KT}{q}} \tag{3.47}$$

The −3-dB bandwidth now becomes

$$f_{-3\text{dB}} = \frac{1}{2\pi C_{\pi 2} \left\{ \dfrac{r_{\pi 2}(r_{\text{bb2}} + R_{\text{D1}} + R_{\text{E1}})}{r_{\pi 2} + r_{\text{bb2}} + R_{\text{D1}} + R_{\text{E1}}} \right\}} \tag{3.48}$$

It can be seen that increasing R_{E1} to increase the gain results in a reduction in the mirror bandwidth. The method of increasing the area of Q_2 to increase the current gain is not advantageous because the capacitance $C_{\pi 2}$ increases simultaneously, and so again, the bandwidth performance is compromised. We can conclude that this approach, though apparently well founded, is flawed in practice.

Improved broadband current mirror. A current mirror with current gain is shown in Figure 3.48 and the small-signal equivalent circuit is shown in Figure 3.49. In this current mirror Q_1 and Q_2 are connected as diodes in series with R_{E1}. Q_3 is connected as a voltage buffer with the bias current source I_{EQ3}. Q_4 is the output transistor with degeneration resistor R_{E4} for current gain setting. The basic idea is to introduce the CC Q_3 to buffer the output from the input and hence isolate gain setting resistor R_{E4} from the bandwidth determining capacitance of the input. The dc transfer function is given by

$$I_{\text{in}} R_{\text{E1}} - I_{\text{out}} R_{\text{E4}} + V_{\text{T}} \ln \frac{I_{\text{in}}^2}{I_{\text{CQ3}} I_{\text{out}}} = 0 \tag{3.49}$$

and the ac small-signal current gain is given by

$$\frac{i_{\text{out}}}{i_{\text{in}}} = \frac{(R_{\text{E1}} + R_{\text{D1}} + R_{\text{D2}}) g_{\text{m4}}}{1 + g_{\text{m4}} R_{\text{E4}}} \tag{3.50}$$

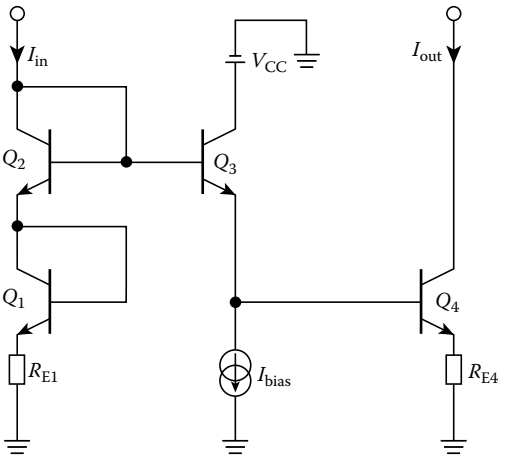


FIGURE 3.48 Improved current mirror with current gain.

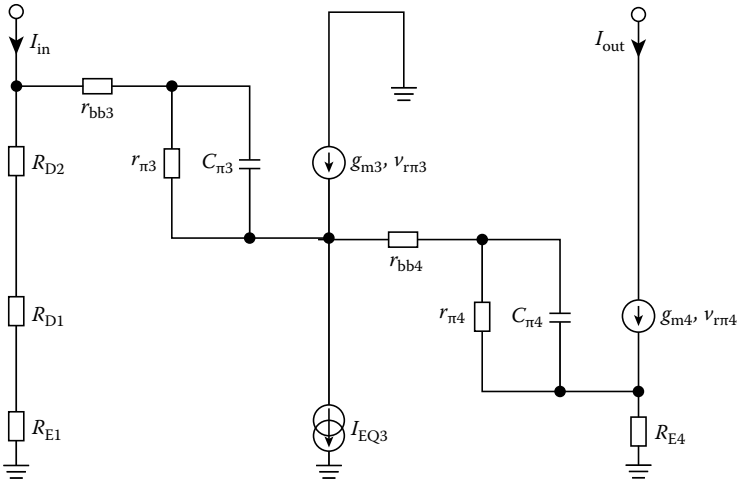


FIGURE 3.49 Equivalent circuit of improved current mirror with current gain.

and the −3 dB bandwidth now becomes

$$f_{-3\text{ dB}} = \frac{1}{2\pi C_{\pi 4}\left(\dfrac{r_{\pi 4}R_x}{r_{\pi 4}+R_x}\right)} \tag{3.51}$$

where

$$R_x = r_{bb4} + \frac{r_{\pi 3} + r_{bb3} + R_{D1} + R_{D2} + R_{E1}}{\beta_3} \tag{3.52}$$

It can be seen clearly that the dominant pole Equation 3.51 of the current mirror with current gain is now only slightly decreased when we increase the current gain by increasing R_{E1}. However, the nondominant pole at the input node is increased, and this will marginally effect the resultant overall stability performance if employed in a current-feedback op-amp. This current mirror with current gain has been employed successfully in current-feedback op-amp design for increased gain-bandwidth capability [12].

Phase linearity. The internal signal path in a current-feedback op-amp is very linear due largely to the symmetrical architecture. Consequently, these devices have a very linear phase response. Furthermore, all the frequency components of a signal are delayed by the same amount when passing through the amplifier, and so the waveform is reproduced accurately at the output. Current-feedback op-amps typically exhibit differential phase error of around ±1° at frequencies of approximately half the bandwidth.

Choosing the value of R_2. From Equation 3.40, we can see that for a fixed value of C_z, a smaller feedback resistor R_2 will give a higher closed-loop bandwidth. It might be expected that the maximum bandwidth would be obtained with the minimum feedback resistance; that is, with $R_2 = 0$. In practice, current-feedback op-amps are generally unstable when their feedback resistance is reduced below a particular value. The reason for this is that the dominant closed-loop pole at frequency of $f \approx 1/2\pi C_z R_2$ must be significantly lower than any nondominant parasitic pole frequency within the op-amp, so that a reasonable gain and phase margin is maintained. If the value of R_2 is reduced, then this dominant pole will move upward in frequency toward the parasitic poles, reducing the gain and phase margin, and eventually leading to instability. Obviously, the "correct" value for R_2 will depend on the internal value of C_z and the location of any parasitic poles within the device. These are the sort of parameters

that are known to the manufacturer, but are generally not listed in a data sheet. Therefore, the manufacturer of a particular device will generally recommend a value of R_2 that guarantees stability, while maintaining a reasonably wide bandwidth. Reducing R_2 below this recommended or optimum value will tend to lead to peaking and instability, while increasing R_2 above the optimum value will reduce the closed-loop bandwidth. If band limiting is required, then a larger value of R_2 than the optimum can be chosen to limit the bandwidth as required.

Since a current-feedback op-amp requires a minimum value of R_2 to guarantee stability, these devices cannot be used with purely capacitive feedback because the reactance of a capacitor reduces at high frequencies. This means that the conventional voltage op-amp integrator cannot be implemented using a current-feedback op-amp.

Practical considerations for broadband designs.

1. *Ground planes.* The purpose of a ground plane is to provide a low-impedance path for currents flowing to ground, since any series impedance in the ground connections will mean that not all ground nodes are at the same potential. In addition, the inductance of a printed circuit track is approximately inversely proportional to the track width, and so the use of thin tracks can result in inductive ground loops, leading to ringing or even oscillations. The use of an unbroken ground plane on one side of the circuit board can minimize the likelihood of inductive loops within the circuit. However, any particularly sensitive ground-connected nodes in the circuit should be grounded as physically close together as is possible.
2. *Bypass capacitors.* Power supply lines often have significant parasitic inductance and resistance. Large transient load currents can therefore result in voltage spikes on the power supply lines, which can couple onto the signal path within the device. Bypass capacitors are therefore used to lower the impedance of the power supply lines at the point of load, and thus short out the effect of the supply line parasitics. The type of bypass capacitor to use is determined by the application and frequency range of interest. High-speed op-amps work best when their power supply pins are decoupled with RF-quality capacitors.

   Manufacturers often recommend using a composite large-small parallel bypass capacitor with something like a 4.7 uF tantalum capacitor on all supply pins, with a parallel 100 nF ceramic to ensure good capacitive integrity at higher frequencies, where the tantalum becomes inductive. However, a note of caution here: This large-small double capacitor technique relies on the large capacitor having sufficiently high ESR so that at resonance the two capacitors do not create a high-Q parallel filter. In surface-mount designs, a single bypass capacitor may well be better than two due to the inherent high-Q of surface-mount capacitors.

   All bypass capacitor connections should be minimized, since track lengths will simply add more series inductance and resistance to the bypass path. The capacitor should be positioned right next to the power supply pin, with the other lead connected directly to the ground plane.
3. *Sensitive nodes.* Certain nodes within a high-frequency circuit are often sensitive to parasitic components. A current-feedback op-amp, for example, is particularly sensitive to parasitic capacitance at the inverting input, since any capacitance at this point combines with the effective resistance at that node to form a second nondominant pole in the feedback loop. The net result of this additional pole is a reduced phase margin, leading to peaking and even instability. Clearly, great care must be taken during layout to reduce track lengths, etc., at this node. In addition, the stray capacitance to ground at $V(-)$ can be reduced by putting a void area in the ground plane at this point. If the op-amp is used as an inverting amplifier, then the potential of the inverting input is held at virtual ground, and any parasitic capacitance will have less effect. Consequently, the current-feedback op-amp is more stable when used in the inverting rather than the noninverting configuration.
4. *Unwanted oscillations.* Following the preceding guidelines should ensure that your circuit is well behaved. If oscillations still occur, a likely source is unintentional positive feedback due to poor

layout. Output signal paths and other tracks should be kept well away from the amplifier inputs to minimize signal coupling back into the amplifier. Input track lengths should also be kept as short as possible for this same reason.

3.1.9 Broadband Amplifier Stability

Operational amplifiers are generally designed with additional on-chip frequency compensation capacitance in place. This is done to present the applications engineer with an op-amp that is simple to use in negative feedback, with minimal chance of unstable operation. In theory, all will be well, but for three main reasons, op-amps become unstable in the real world of analog electronic circuit design. This section outlines the three main causes for unstable operation of broadband amplifiers and shows practical ways of avoiding these pitfalls.

3.1.9.1 Op-Amp Internal Compensation Strategy

Before dealing with specific stability problems in broadband amplifiers and how to solve them, we will look briefly at the internal frequency compensation strategy used in op-amp design. Generally, op-amps can be classified into two groups, those with two high-voltage gain stages and those with only one stage. The two-stage design provides high open-loop gain but relatively low bandwidth, while the higher speed signal-stage amplifier provides lower open-loop gain but much higher usable bandwidth. Insight into the internal op-amp architecture and the type of compensation used will give the designer valuable information on how to tame the unstable op-amp.

3.1.9.2 Review of the Classical Feedback System

Analyzing the classical feedback system in Figure 3.50 gives the well-known expression for the closed-loop gain, A_c:

$$A_c = A/[1 + B \cdot A] \tag{3.53}$$

where A is the open-loop gain of the amplifier and B the feedback fraction. $T = B \times A$ is referred to as the loop-gain, and the behavior of T over frequency is a key parameter in feedback system design. Clearly, if $T \gg 1$ or $A \gg A_c$, then the closed-loop gain is virtually independent of the open-loop gain A, thus

$$A_c \approx B^{-1} \tag{3.54}$$

This is the most important and desirable feature of negative feedback systems. However, the system will not necessarily be stable as, at higher frequencies, phase lag in the open-loop gain A may cause the feedback to become positive.

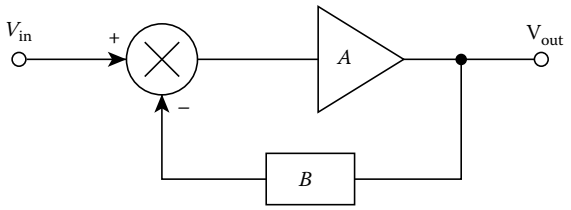


FIGURE 3.50 Classical feedback system.

3.1.9.3 Stability Criteria

Though negative feedback is desirable, it results in potential instability when the feedback becomes positive. The loop-gain T is the best parameter to test whether an amplifier is potentially unstable. The phase margin Φ_M is a common feature of merit used to indicate how far the amplifier is from becoming an oscillator:

$$\Phi_M = 180^\circ + \Phi(|BA| = 1) \tag{3.55}$$

When $\Phi_M = 0°$, the phase of the loop gain, $T = B \times A$ is exactly $-180°$ for $|B \times A| = 1$. The closed-loop gain A_c will become infinite and we have got an oscillator! Clearly, what is required is that $\Phi_M > 0$ and generally the target is to make $\Phi_M \geq 45°$ for reasonably stable performance. However, excessive Φ_M is undesirable if settling time is an important parameter in a particular application.

An op-amp is a general purpose part and so the IC designer strives to produce a maximally versatile amplifier by ensuring that even with 100% feedback, the amplifier circuit will not become unstable. This is done by maintaining a $\Phi_M > 0$ for 100% feedback, that is, when $B = 1$. If the feedback network B is taken to be purely resistive, then any additional phase lag in the loop gain must come from the open-loop amplifier A. Tailoring the phase response of A so that the phase lag is less than 180° up to the point at which $|A| < 1$ or 0 dB ensures that the amplifier is "unconditionally stable"; that is, with any amount of resistive feedback, stable operation is "guaranteed."

Most open-loop op-amps, whether single-stage or two-stage, will exhibit a two-pole response. The separation of these two poles whether at low frequency or high frequency will have a major effect on the stability of the system and it is the op-amp designer's objective to locate these open-loop poles to best advantage to achieve maximum bandwidth, consistent with versatile and stable performance.

3.1.9.4 Two-Stage Op-Amp Architecture

A schematic of the standard two-stage op-amp topology is shown in Figure 3.51. The input differential pair T_1/T_2 provides high gain, as does the second gain stage of T_3/T_4 Darlington pair CE. A high-voltage gain is achieved with this structure, so that output stage is usually a unity voltage gain common-collector output buffer to provide a useful load current drive capability.

The amplifier structure in Figure 3.51 has two internal high-impedance nodes, node X and node Y. These high-impedance nodes are responsible for introducing two dominant poles into the frequency response and their relative location is critical in determining the stability of the amplifier. Each pole contributes a low-pass filter function to the open-loop gain expression of the form

$$[1 + jf/f_P]^{-1} \tag{3.56}$$

Each pole introduces 45° of phase lag at the pole frequency f_P and an additional 45° at $f \approx 10 \times f_P$. With a two-pole amplifier, the open-loop gain A is given by

$$A = A_0/[1 + jf/f_{P1}][1 + jf/f_{P2}] \tag{3.57}$$

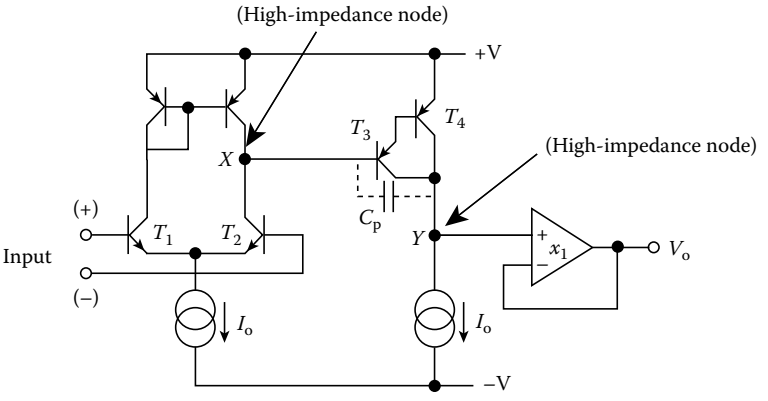


FIGURE 3.51 Architecture of the standard two-stage op-amp.

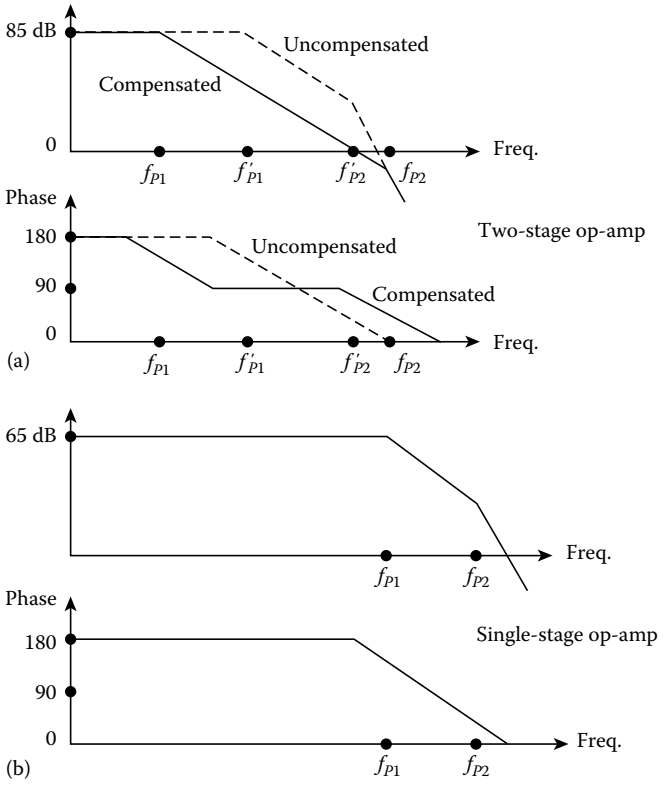


FIGURE 3.52 Pole frequency and phase response for (a) two-stage op-amp and (b) single-stage op-amp.

where A_0 is the dc open-loop gain and f_{P1} and f_{P2} are the two-pole frequencies. A typical plot of A versus f is shown in Figure 3.52a. At low frequencies, where $f \ll f_{P1}$ the gain is flat, and at f_{P1} the gain begins to fall at a rate increasing to −20 dB/decade. The roll-off steepens again at f_{P2} to a final gradient of −40 dB/decade.

It is generally the case that $f_{P1} \ll f_{P2}$ as shown in Figure 3.52a. Turning our attention to the phase plot in Figure 3.52a, at $f = f_{P1}$ the output lags the input by 45°, and as the frequency rises toward f_{P2} the phase lag increases through 135° at f_{P2} to 180° at $f \approx 10 \times f_{P2}$. To ensure unconditionally stable performance, the second pole must be sufficiently far from the first so that the phase margin is large enough.

Figure 3.53 shows curves of the dc value of open-loop gain A_0 versus the ratio N of the pole frequencies ($N = f_{P2}/f_{P1}$) for different values of phase margin. For a given value of $A_0 = 1000$ or +60 dB, the ratio of the pole frequencies must be $N \approx 700$ to obtain a phase margin of 45°.

3.1.9.5 Miller Compensation and Pole Separation

Without any added compensation capacitance, the two open-loop poles of the op-amp are invariably too close to make the amplifier unconditionally stable. The most common compensation method is to add a capacitor between the base and collector of the Darlington pair, shown as C_p in Figure 3.51. This is known as Miller compensation because this strategy makes use of the Miller capacitance multiplication effect discussed earlier. The net result is that the two poles now become significantly far apart, with f_{P1} reducing and f_{P2} increasing, and so the phase margin can be increased to make the op-amp unconditionally stable. However, the penalty of this method is poorer bandwidth and also lower slew rate because of the large capacitance needed, which in practice may be 20 pF or more.

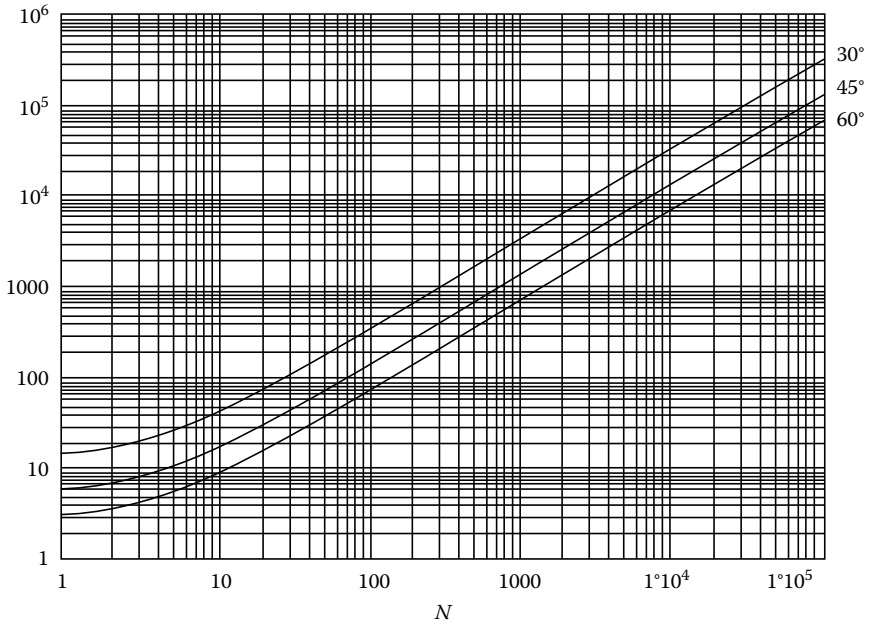


FIGURE 3.53 Low-frequency gain A_0 versus $N\ (=f_{P2}/f_{P1})$ for a two-pole amplifier.

3.1.9.6 Single-Stage Op-Amp Compensation

Figure 3.54 shows a typical simplified circuit schematic of a single-stage op-amp. The input is a differential emitter-coupled pair followed by a folded cascode transistor and an output complementary common-collector buffer. The key difference between this architecture and the two-stage design shown earlier is that X is a low-impedance node, and so the only high-impedance node in the circuit is node Y. Interestingly, the higher frequency nondominant pole of the two-stage amplifier has now become the dominant frequency pole of the single-stage design, as indicated by the second set of curves in Figure 3.52b, which leads to several advantages:

1. The frequency performance off the amplifier is extended. This frequency extension does not lead to a deterioration in phase margin, but simply means that the phase margin problem is shifted up in the frequency domain.

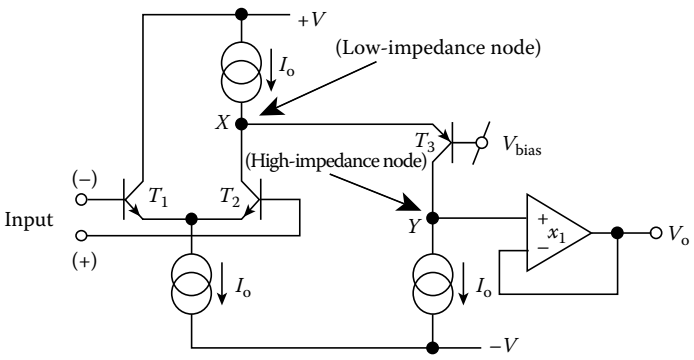


FIGURE 3.54 Architecture of single-stage op-amp.

2. Capacitance at the high-impedance Y node reduces bandwidth, but now improves phase margin.
3. A single value of a few pFs of grounded capacitor at Y will now act as a satisfactory compensation capacitor, unlike the large Miller capacitor required in the two-stage design.
4. The slewing capability of this single-stage structure is very good as a result of the much smaller compensation capacitor.
5. Clearly, it is much more straightforward to develop a stable amplifier for high-frequency applications if it has essentially only one voltage gain stage and so high-frequency op-amp designers generally opt for a single gain stage architecture.

3.1.9.7 Grounded Capacitor Compensation

Typical A_{OL} versus f responses of two single-stage op-amps are shown in Figure 3.55, indicating one high-frequency pole and its proximity to the nondominant pole.

The curves are taken from data for (a) a 2 GHz gain-bandwidth product voltage-feedback op-amp and (b) a 150 MHz current-feedback op-amp. In both cases, the phase characteristics demonstrate the expected 45° lag at the pole frequency, and the slow roll-off in phase at high frequency due to the presence of the very-high-frequency poles.

Both single-stage and two-stage op-amps can be approximated by the two-pole macromodel shown in Figure 3.56. Transconductance G_M and output resistance R_0 represent the gain per stage of $G_M \times R_0$. The difference between the two-stage and single-stage op-amp models is that R_{01} of the single-stage is of the order of $[G_M]^{-1}$ and the dominant compensation capacitor is C_2. C_P in the case of the single stage will

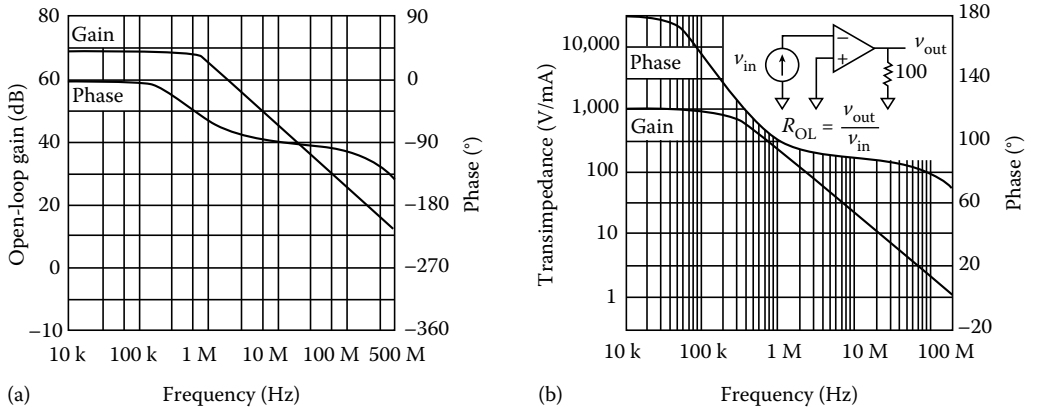


FIGURE 3.55 Single-pole op-amps; open-loop gain and phase frequency characteristics. (a) Voltage feedback. (b) Current feedback.

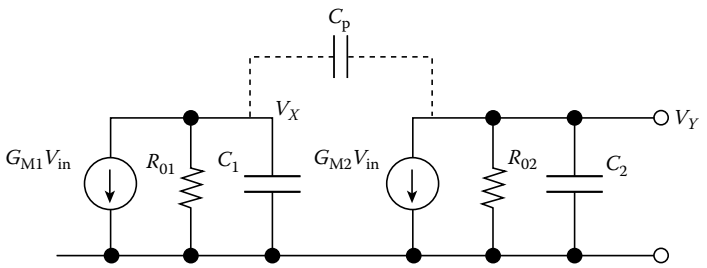


FIGURE 3.56 Partial equivalent circuit of two-pole op-amp.

simply be a feedback parasitic capacitor, while in the case of a two-stage it will be the dominant Miller compensating capacitor. This simple model is an excellent first-cut tool for determining pole locations, and the value of compensation capacitor for a desired bandwidth and stability.

3.1.9.8 High-Frequency Performance

Although the bandwidth in a single-stage design is significantly extended, circuit parasitics become more important. We are confronted with the problem of potential instability, since at higher frequencies the "working environment" of the op-amp becomes very parasitic sensitive; in other words, now op-amp-embedded parasitics cannot be neglected.

An op-amp in closed-loop can be considered at three levels, as shown schematically in Figure 3.57. The inner triangle is the ideal op-amp, internally compensated by the op-amp designer for stable operation using the circuit techniques outlined earlier. High-frequency amplifiers are sensitive to parasitics of the surrounding circuit. The key parasitics within the outer triangle include power supply lead inductance, stray capacitance between power supply pins, and input to ground capacitance. The effect of these parasitics is to destabilize the amplifier, and so the designer is confronted with the task of reestablishing stable operation. The approach needed to achieve this parallels the work of the op-amp designer. The parasitics almost always introduce additional extrinsic nondominant poles, which need to be compensated. The task of compensation cannot be attempte without considering the outer or third level, which includes the closed-loop gain defining components together with the load impedance. Again, stray reactance associated with these components will modify the loop gain, and so to guarantee stable operation of the closed-loop amplifier it is necessary to compensate the complete circuit.

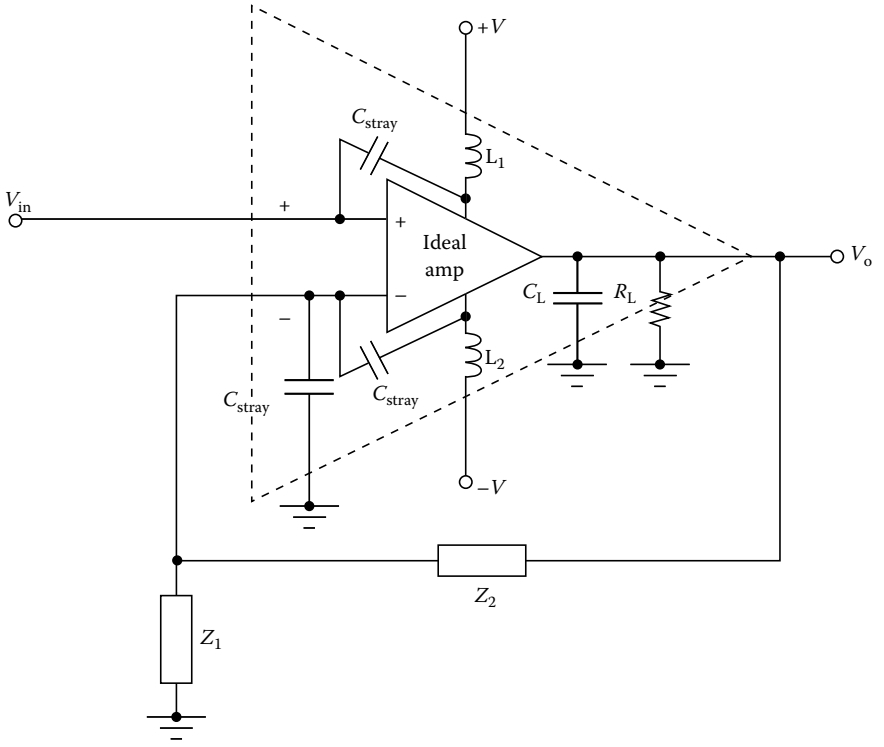


FIGURE 3.57 Real feedback amplifier.

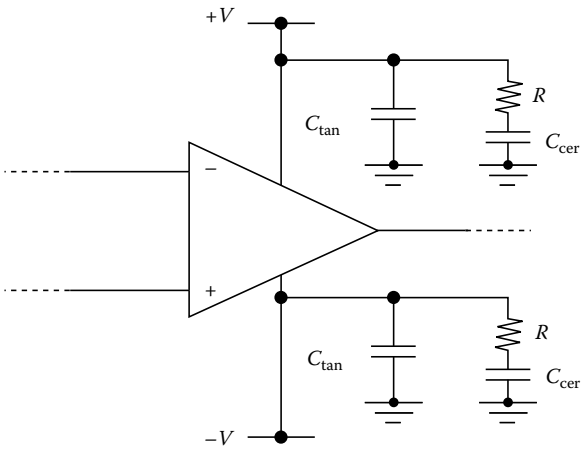


FIGURE 3.58 Supply decoupling circuitry (C_{CER} = ceramic capacitor and C_{TAN} = tantalum).

3.1.9.9 Power Supply Impedance

In this section, we consider the ways in which the impedance of the power supply can affect the frequency response of the amplifier. First, some important rules are

1. There is no such thing as an ideal zero-impedance power supply.
2. Real power supplies have series *R–L* impedance and at high frequencies the inductance matters most.
3. Power supply inductance causes "bounce" on the power supply voltage, generating unwanted feedback via parasitic capacitive links to the inputs. Power supply "bounce" increases with increasing load current.
4. Supply decoupling capacitors act as "short-term local batteries" to maintain power supply integrity, and it is important that they are placed as close as possible to the power supply pins of the op-amp.

Large electrolytic capacitors are fine at low frequencies but are inductive at high frequencies. Figure 3.58 shows commonly used decoupling circuitry. Small-sized tantalum electrolytics are preferred, while a parallel ceramic capacitor with low series inductance takes over the decoupling role at high frequencies. The added series *R* prevents the inductance of the electrolytic resonating with the ceramic capacitor. The waveforms in Figure 3.59 illustrate the benefits of good decoupling.

3.1.9.10 Effects of Resistive and Capacitive Loads

The load presented to an amplifier is likely to have both resistive and capacitive components, as illustrated previously in Figure 3.57. Increasing the load current causes power supply ripple, so good power supply decoupling is vital.

A closed-loop amplifier with voltage-sampled negative feedback results in a very-low output impedance, so it is natural to think that the effects of any load would be shunted out by this low impedance. In reality, the load has an important effect on the amplifier and must not be overlooked. Resistive loads, for example, cause two main effects. First, as a voltage divider with the open-loop output resistance of the op-amp r_0, the open-loop gain is reduced. This effect is small unless the load resistance approaches r_0. Second, the load current is routed to the output pin via the supply pins, and as the load current increases,

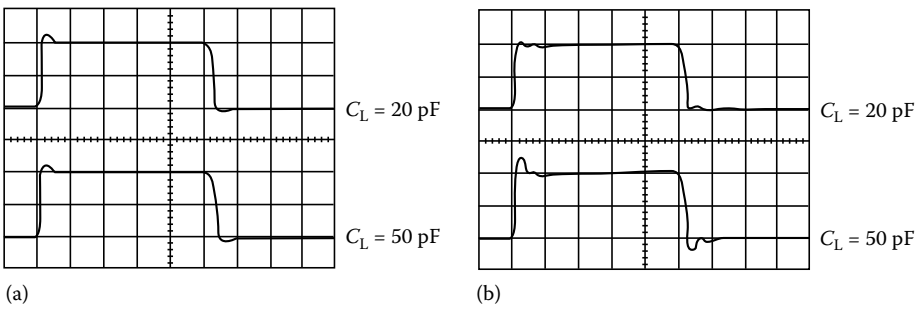


FIGURE 3.59 High-speed voltage buffer: (a) with and (b) without supply decoupling.

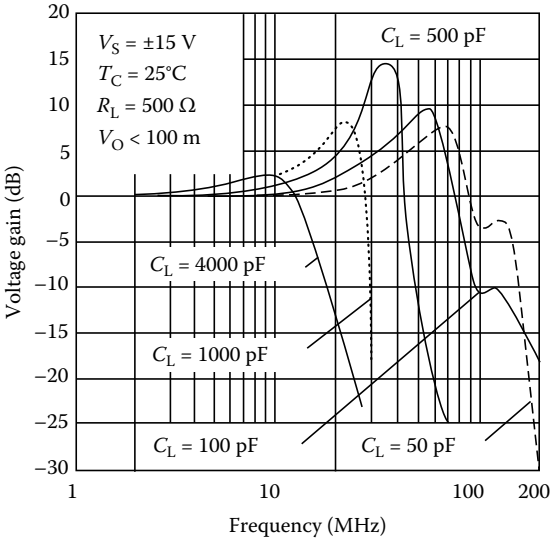


FIGURE 3.60 Load capacitance causes gain peaking.

the supply pin voltage is modulated. This effect is more important, since the integrity of the power supply will be degraded. Again, good supply decoupling is essential to minimize this effect.

Capacitive load current is proportional to the derivative of output voltage, and the maximum capacitive output current demand occurs when dV_{out}/dt is a maximum. Though not directly a stability issue, the designer must remember that a capacitive load demands high-output current at high frequencies and at high amplitude, that is,

$$I_{max} = C_L \cdot 2\pi f_{max} \cdot V_{outpeak} \tag{3.58}$$

Figure 3.60 illustrates the effect of load capacitance on the loop gain.

C_1 together with the equivalent output resistance of the op-amp adds an additional pole into the loop gain of the form

$$V_F/V_{out} = B = 1/[1 + jf/f_L] \quad \text{where } f_L = 1/2\pi r_0 \cdot C_L \tag{3.59}$$

The load resistance has a minor influence on the loop gain compared to the effects of load capacitance by slightly reducing the value of dc open-loop gain by factor K, where $K = R_L/[r_0 + R_L]$, as described above. Since the effective output resistance reduces to $r_{0'} = r_0/R_L$, then f_L changes to $f_{L'} = 1/2\pi r_{0'} C_L$.

3.1.9.11 Neutralizing the Phase Lag

To compensate for high-frequency phase lag, the simplest technique is to add a series resistance R between the output of the op-amp and the load connection point, as shown in Figure 3.61.

The series resistor adds a zero into the V_F/V_{out} equation, which changes to

$$V_F/V_{out} = K \cdot [1 + jf/f_Z]/[1 + jf/f_P] \tag{3.60}$$

where $K = [R + R_L]/[r_0 + R + R_L]$, $f_P = 1/[2\pi(r_0 + R)/R_L \cdot C_L]$ and $f_Z = 1/[2\pi R_{L//R} \cdot C_L] = f_P \cdot [1 + r_0/R]$, so clearly; $f_P < f_Z$.

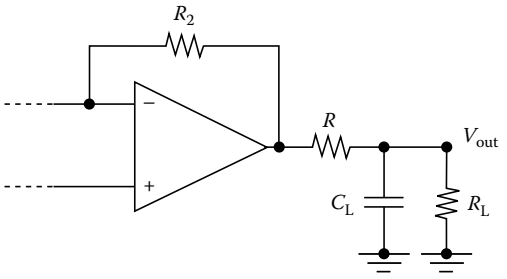


FIGURE 3.61 Load capacitance neutralization.

The phase lag introduced by the pole is compensated by the phase lead of the zero at higher frequencies. The maximum phase lag is limited if the zero is close to the pole, almost eliminating the effects of the load capacitor. Maximum phase lag in V_F/V_{out} occurs at $f = f_M$, where f_M is given by

$$f_M = [f_P \cdot f_Z]^{1/2} = f_P \times (1 + r_0/R)^{1/2} \tag{3.61}$$

and at f_M the phase lag $\Phi = \Phi'$ is given by

$$\begin{aligned} \Phi' &= 90^\circ - 2 \cdot \tan^{-1}[f_M/f_P] = 90^\circ - 2 \cdot \tan^{-1}\left[(1 + r_0/R)^{1/2}\right] \\ \Phi' &\approx -19.5^\circ \quad \text{for} \quad R = r_0 \\ \Phi' &\approx -8.2^\circ \quad \text{for} \quad R = 2 \cdot r_0 \\ \Phi' &\approx -6.4^\circ \quad \text{for} \quad R = 3 \cdot r_0 \end{aligned} \tag{3.62}$$

These values show that the added lag Φ' is not excessive as long as $R > r_0$. The disadvantage with this method is that the series resistor is in direct line with the output current, increasing the output resistance of the amplifier and limiting the output current drive capability. The output impedance also goes inductive at high frequencies.

An alternative way of solving the problem of capacitive load is to view the closed-loop output resistance of the op-amp as being inductive, since the closed-loop output impedance of the op-amp is essentially the open-loop output resistance divided by the loop gain. As the loop gain falls with frequency, the output impedance rises, and thus appears inductive. Adding a load capacitor generates a resonant circuit. The solution is to "spoil" the Q of the resonator, therefore minimizing the added phase lag of C_L.

Adding a so-called series R–C "snubber," as in Figure 3.62, effects a cure. The resistor R is ac coupled by the capacitor at high frequencies and spoils the Q. Effectively, C_L resonates with the inductive output impedance, and at this frequency leaves the R–C snubber as a "new" load. The equivalent circuit is therefore close to the previous compensation method shown in Figure 3.61, but with the added advantage that now the load current is not carried by the series resistance. To select the snubber component values, make $R = 1/2\pi f_0 C$, where f_0 is the resonant frequency, which can simply be determined experimentally from the amplifier without the snubber in place. The value of the series capacitance is a compromise: too big and it will increase the effective load capacitance. Choosing $C = C_L$ works reasonably well in practice.

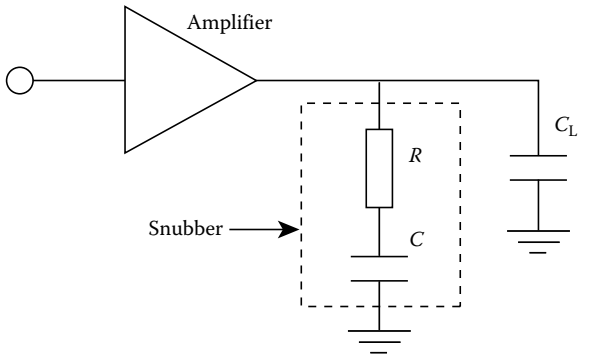


FIGURE 3.62 Snubber cures capacitive load peaking.

3.1.9.12 Inverting Input Capacitance to Ground

With most broadband bipolar op-amps, parasitic capacitance to ground adds an additional pole (and hence phase lag) into the feedback path, which threatens stability. Stray capacitance C_1 at the inverting input pin (shown previously in Figure 3.57) modifies B and adds phase lag in the loop-gain T, compromising stability.

Solving for B with C_1 taken into account will clarify the problem. It is simple to show that

$$B = V_F/V_{out} = Z_1/[Z_1 + Z_2] \tag{3.63}$$

where $Z_1 = R_1/[1 + j\omega R_1 C_1]$ and $Z_2 = R_2$. Substituting, we get

$$B = K/[1 + jf/f_C] \tag{3.64}$$

where $K = R_1\ [R_1 + R_2]$ and $f_C = 1/[2\pi C_1 R_1/R_2]$.

The additional pole at $f = f_C$ will now give the circuit a very undesirable three-pole loop gain, which could cause significant gain peaking, as shown in Figure 3.63. f_C could be made high by choosing relatively low values of $R_1 // R_2$ but the additional pole can be eliminated by adding a feedback capacitor C_2 across resistor R_2 to give pole-zero cancellation.

$$Z_1 = R_1/[1 + j\omega R_1 C_1] \quad \text{and} \quad Z_2 = R_2/[1 + j\omega R_2 C_2] \tag{3.65}$$

If $R_1C_1 = R_2C_2$, then $B = Z_1/[Z_1 + Z_2] = R_1/[R_1 + R_2]$, making B frequency independent. The design equation for C_2 is then

$$C_2 = C_1 \cdot R_1/R_2 \tag{3.66}$$

If the open-loop phase margin Φ_M needs to be increased for the desired value of closed-loop gain, and the inverting capacitance C_1 has its inevitable high-frequency influence, then the optimum solution for C_2 would be to locate the zero on the second pole of the loop-gain response following the procedure given above.

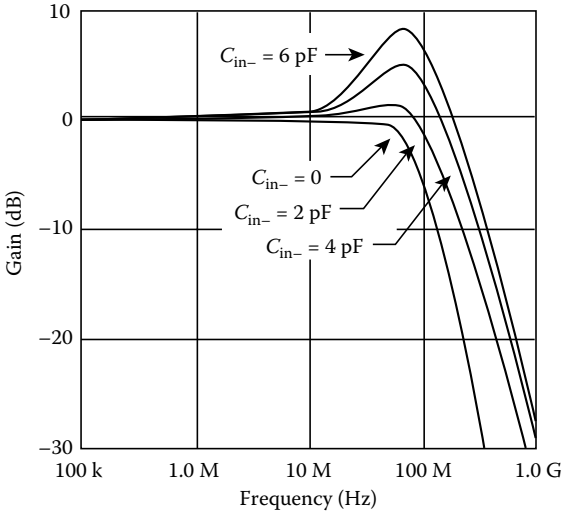
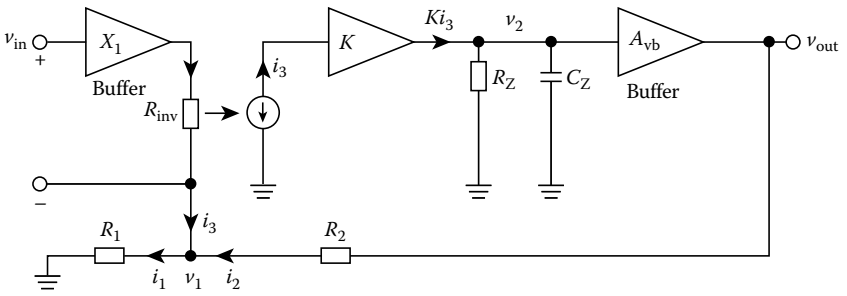


FIGURE 3.63 Stray input capacitance causes gain peaking.

3.1.10 Conclusions

This chapter hopefully serves to illustrate some of the modern techniques the practicing engineer will encounter when designing broadband bipolar amplifiers. It focuses mainly upon key generic building blocks and methodologies for broadband design. Many circuits and design techniques have not been covered, but analysis techniques described should serve as a foundation for the analysis of other broadband designs. Furthermore, comprehensive analytical treatment of many alternative broadband bipolar circuits can be found in the texts [6,13–15].

Appendix A: Transfer Function and Bandwidth Characteristic of Current-Feedback Operational Amplifier



$$-i_1 + i_2 + i_3 = 0 \tag{3.67}$$

$$i_1 = \frac{v_1}{R_1} \tag{3.68}$$

$$i_2 = \frac{v_{out} - v_1}{R_2} \tag{3.69}$$

$$i_3 = \frac{v_{in} - v_1}{R_{inv}} \tag{3.70}$$

$$v_2 = \frac{Ki_3R_Z}{1 + j\omega R_Z C_Z} \tag{3.71}$$

$$v_{out} = A_{vb}v_2 \tag{3.72}$$

Substituting Equations 3.68 through 3.70 into Equation 3.67 yields

$$-\frac{v_1}{R_1} + \frac{v_{out} - v_1}{R_2} + \frac{v_{in} - v_1}{R_{inv}} = 0$$

Rearranging for v_1 gives

$$v_1 = \frac{v_{in}R_2/R_{inv} + v_{out}}{1 + (R_2/R_1) + (R_2/R_{inv})}$$

From Equations 3.71 and 3.72, it is clearly seen that

$$v_{\text{out}} = \frac{A_{\text{vb}} K i_3 R_Z}{1 + j\omega R_Z C_Z} \tag{3.73}$$

Substituting for i_1 and i_2 from Equations 3.68 and 3.69 into Equation 3.67 gives

$$i_3 = v_1\left(\frac{1}{R_1} + \frac{1}{R_2}\right) - \frac{v_{\text{out}}}{R_2}$$

Substitute for v_1:

$$i_3 = \left[\frac{v_{\text{in}} R_2/R_{\text{inv}} + v_{\text{out}}}{1 + (R_2/R_1) + (R_2/R_{\text{inv}})}\right]\left(\frac{1}{R_1} + \frac{1}{R_2}\right) - \frac{v_{\text{out}}}{R_2}$$

Substitute for i_3 from Equation 3.73:

$$\frac{v_{\text{out}}(1 + j\omega R_Z C_Z)}{A_{\text{vb}} K R_Z} = \left\{\left[\frac{v_{\text{in}} R_2/R_{\text{inv}} + v_{\text{out}}}{1 + (R_2/R_1) + (R_2/R_{\text{inv}})}\right]\left(\frac{1}{R_1} + \frac{1}{R_2}\right) - \frac{v_{\text{out}}}{R_2}\right\}$$

rearranging

$$v_{\text{out}}\left[\frac{(1 + j\omega R_Z C_Z)}{A_{\text{vb}} K R_Z} - \frac{(1/R_1 + 1/R_2)}{1 + (R_2/R_1) + (R_2/R_{\text{inv}})} + \frac{1}{R_2}\right] = \frac{(v_{\text{in}} R_2/R_{\text{inv}})(1/R_1) + (1/R_2)}{1 + (R_2/R_1) + (R_2/R_{\text{inv}})}$$

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\frac{R_{\text{inv}}(1 + (R_2/R_1) + (R_2/R_{\text{inv}}))(1 + j\omega R_Z C_Z)}{A_{\text{vb}} K R_Z} - R_{\text{inv}}((1/R_1) + (1/R_2)) + \frac{R_{\text{inv}}(1 + (R_2/R_1) + (R_2/R_{\text{inv}}))}{R_2}}$$

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K R_Z} + \frac{(R_{\text{inv}}(1 + (R_2/R_1)) + R_2) j\omega R_Z C_Z}{A_{\text{vb}} K R_Z} + 1}$$

Factorize the denominator

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\left[1 + \frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K R_Z}\right]\left[1 + \frac{\frac{(R_{\text{inv}}(1 + (R_2/R_1)) + R_2) j\omega R_Z C_Z}{A_{\text{vb}} K R_Z}}{1 + \frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K R_Z}}\right]}$$

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\left[1 + \frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K R_Z}\right]\left[1 + j\omega C_Z\left\{\frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K + \frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{R_Z}}\right\}\right]}$$

If we assume that R_Z is very large, then

$$\frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{R_Z} \approx 0$$

and the transfer function becomes

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{1 + j\omega C_Z\left[\frac{R_{\text{inv}}(1 + (R_2/R_1)) + R_2}{A_{\text{vb}} K}\right]}$$

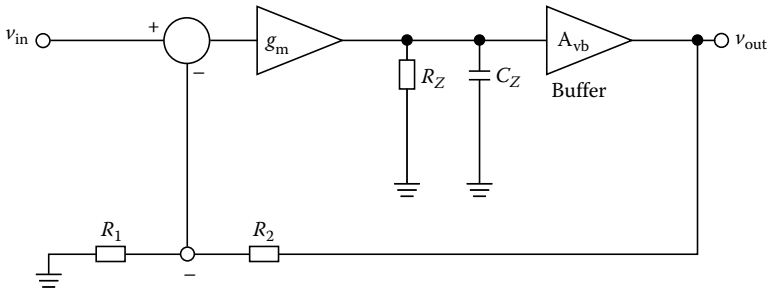
The pole frequency is given by

$$f_{-3\ \mathrm{dB}} = \frac{A_{\mathrm{vb}}K}{2\pi C_Z[R_{\mathrm{inv}}(1+(R_2/R_1))+R_2]}$$

The gain-bandwidth product is given by

$$GBW = \frac{A_{\mathrm{vb}}K[1+(R_2/R_1)]}{2\pi C_Z[R_{\mathrm{inv}}(1+(R_2/R_1))+R_2]}$$

Appendix B: Transfer Function and Bandwidth Characteristic of Voltage-Feedback Operational Amplifier



$$\begin{aligned}
v_{\mathrm{out}} &= \left(v_{\mathrm{in}} - \frac{R_1}{R+R_2 v_{\mathrm{out}}}\right)\frac{g_{\mathrm{m}}R_Z A_{\mathrm{vb}}}{1+j\omega R_Z C_Z} \\
v_{\mathrm{out}}&\left[1+\frac{R_1 g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}{(R_1+R_2)(1+j\omega R_Z C_Z)}\right] = v_{\mathrm{in}}\frac{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}{1+j\omega R_Z C_Z} \\
\frac{v_{\mathrm{out}}}{v_{\mathrm{in}}} &= \frac{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z/(1+j\omega R_Z C_Z)}{1+\frac{R_1 g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}{(R_1+R_2)(1+j\omega R_Z C_Z)}}
\end{aligned} \tag{3.74}$$

Multiply the numerator and denominator by $(1+j\omega R_Z C_Z)/g_{\mathrm{m}}\ A_{\mathrm{vb}}\ R_Z$

$$\begin{aligned}
\frac{v_{\mathrm{out}}}{v_{\mathrm{in}}} &= \frac{1}{\frac{1+j\omega R_Z C_Z}{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}+\frac{R_1}{(R_1+R_2)}} \\
\frac{v_{\mathrm{out}}}{v_{\mathrm{in}}} &= \frac{1+(R_2/R_1)}{\left[\frac{1+j\omega R_Z C_Z}{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}\right][1+(R_2/R_1)]+1} \\
\frac{v_{\mathrm{out}}}{v_{\mathrm{in}}} &= \frac{1+(R_2/R_1)}{\frac{1+(R_2/R_1)}{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}+\frac{j\omega R_Z C_Z(1+(R_2/R_1))}{g_{\mathrm{m}}A_{\mathrm{vb}}R_Z}+1}
\end{aligned}$$

get $1+[1+(R_2/R_1)/g_{\mathrm{m}}\ A_{\mathrm{vb}}\ R_Z]$ out of the denominator

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\left[1 + \frac{1 + (R_2/R_1)}{g_m A_{vb} R_Z}\right]\left[1 + \frac{\frac{j\omega R_Z C_Z (1 + (R_2/R_1))}{g_m A_{vb} R_Z}}{1 + \frac{1 + (R_2/R_1)}{g_m A_{vb} R_Z}}\right]}$$

multiply the denominator bracket by $g_m\, A_{vb}\, R_z/[1 + (R_2/R_1)]$

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\left[1 + \frac{1 + (R_2/R_1)}{g_m A_{vb} R_Z}\right]\left[1 + \frac{j\omega R_Z C_Z (1 + (R_2/R_1))}{g_m A_{vb} R_Z + (1 + (R_2/R_1))}\right]}$$

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{\left[1 + \frac{1 + (R_2/R_1)}{g_m A_{vb} R_Z}\right]\left[1 + \frac{j\omega R_Z C_Z}{1 + \frac{g_m A_{vb} R_Z}{1 + (R_2/R_1)}}\right]}$$

assuming that $g_m\, A_{vb} R_Z$ is much larger than $1 + R_2/R_1$, then

$$\frac{v_{\text{out}}}{v_{\text{in}}} = \frac{1 + (R_2/R_1)}{1 + j\omega\left[\frac{R_Z C_Z}{1 + \frac{g_m A_{vb} R_Z}{1 + (R_2/R_1)}}\right]}$$

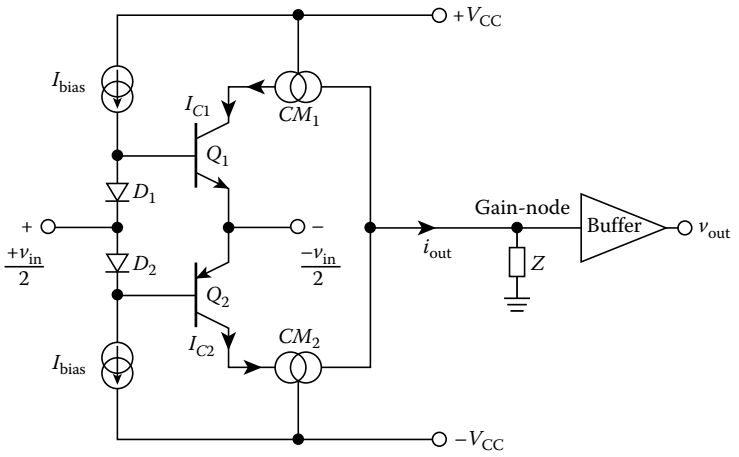
The pole frequency is given by

$$f_{-3\text{ dB}} = \frac{1 + \frac{g_m A_{vb} R_Z}{1 + (R_2/R_1)}}{2\pi R_Z C_Z}$$

The gain-bandwidth product is given by

$$GBW = \frac{(1 + (R_2/R_1))\left[1 + \frac{g_m A_{vb} R_Z}{(1 + (R_2/R_1))}\right]}{2\pi R_Z C_Z}$$

Appendix C: Transconductance of the Current-Feedback Op-Amp Input Stage



$$v_{\text{in}} = v_1 - v_2$$

$$i_{\text{out}} = I_{CI} - I_{C2}$$

$$I_{C1} = I_{S1} e^{\frac{v_{BE1}}{v_T}}$$

$$I_{C2} = I_{S2} e^{\frac{v_{BE2}}{v_T}}$$

$$V_{BE1} = V_{DQ1} + v_{\text{in}}$$

$$V_{BE2} = v_{\text{in}} - V_{DQ2}$$

$$I_{C1} = I_{S1} e^{\left(\frac{V_{DQ1}}{V_T} + \frac{v_{\text{in}}}{v_T}\right)}$$

$$I_{C2} = I_{S2} e^{\left(\frac{V_{DQ1}}{V_T} - \frac{v_{\text{in}}}{v_T}\right)}$$

$$I_{C1} = I_{CQ1} e^{\frac{v_{\text{in}}}{v_T}}$$

$$I_{C2} = I_{CQ2} e^{-\frac{v_{\text{in}}}{v_T}}$$

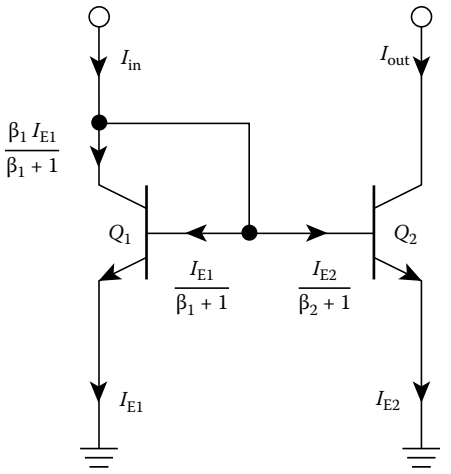
Assuming matched transistors then, $I_{\text{CQ1}} = I_{\text{CQ2}} = I_{\text{CQ}}$

$$i_{\text{out}} = I_{\text{C1}} - I_{\text{C2}} = I_{\text{CQ}} \left[e^{+\left(\frac{v_{\text{in}}}{v_{\text{T}}}\right)} - e^{-\left(\frac{v_{\text{in}}}{v_{\text{T}}}\right)} \right]$$

$$\frac{i_{\text{out}}}{I_{\text{CQ}}} = y = \left[e^{x} - e^{-x}\right] = 2 \sinh(x)$$

where $x = + v_{\text{in}}/V_{\text{T}}$.

Appendix D: Transfer Function of Widlar Current Mirror



$$I_{in} = I_{E1} + \frac{I_{E2}}{\beta_2 + 1}$$

$$I_{in} = \frac{I_{E1}(\beta_2 + 1) + I_{E2}}{\beta_2 + 1}$$

$$I_{out} = \beta_2 I_{B2}$$

$$I_{out} = \frac{\beta_2 I_{E2}}{\beta_2 + 1}$$

$$\frac{I_{out}}{I_{in}} = \frac{(\beta_2 I_{E2})(\beta_2 + 1)}{(\beta_2 + 1)[I_{E1}(\beta_2 + 1) + I_{E2}]}$$

$$\frac{I_{out}}{I_{in}} = \frac{\beta_2 I_{E2}}{I_{E1}(\beta_2 + 1) + I_{E2}}$$

$$\frac{I_{out}}{I_{in}} = \frac{1}{\frac{I_{E1}(\beta_2 + 1)}{I_{E2}\beta_2} + \frac{1}{\beta_2}}$$

For

$$\frac{I_{E1}}{I_{E2}} = \frac{I_{S1}\left(\frac{\beta_1 + 1}{\beta_1}\right) e^{\frac{v_{BE1}}{V_T}}}{I_{S2}\left(\frac{\beta_2 + 1}{\beta_2}\right) e^{\frac{v_{BE2}}{v_T}}}$$

Then, as $V_{BE1} = V_{BE2}$,

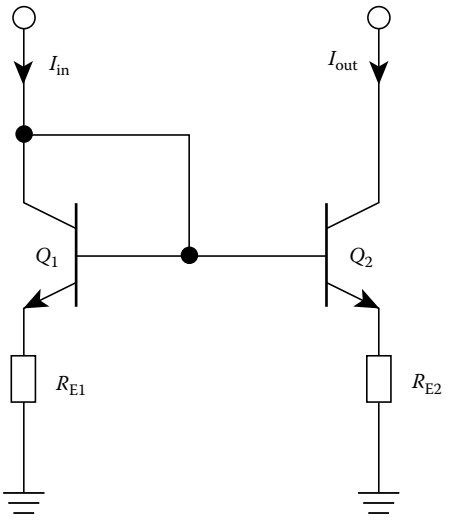
$$\frac{I_{E1}}{I_{E2}} = \frac{I_{S1}(\beta_1 + 1/\beta)}{I_{S2}(\beta_2 + 1/\beta_2)}$$

$$\frac{I_{out}}{I_{in}} = \frac{1}{\frac{I_{S1}(\beta_1 + 1)}{I_{S2}\beta_1} + \frac{1}{\beta_2}}$$

Assume $\beta_1 = \beta_2 = \beta$, $I_{S1} = I_{S2}$. Then

$$\frac{I_{out}}{I_{in}} = \frac{\beta}{\beta + 2}$$

Appendix E: Transfer Function of Widlar Current Mirror with Emitter Degeneration Resistors



Assuming that $\beta \gg 1$, then

$$V_{BE1} + I_{in}R_1 = V_{BE2} + I_{out}R_2$$

$$I_{out} = \frac{I_{in}R_1}{R_2} + \frac{(V_{BE1} - V_{BE2})}{R_2}$$

$$\frac{I_{out}}{I_{in}} = \frac{R_1}{R_2} + \frac{(V_{BE1} - V_{BE2})}{I_{in}R_2}$$

$$\frac{I_{out}}{I_{in}} = \frac{R_1}{R_2} + \frac{V_T \ln\left(\frac{I_{in}}{I_{S1}}\frac{I_{S2}}{I_{out}}\right)}{I_{in}R_2}$$

$$\frac{I_{out}}{I_{in}} = \frac{R_1}{R_2} + \frac{V_T(\ln(I_{in}/I_{out}) + (\Delta V_{BE}/V_T))}{I_{in}R_2}$$

$$\frac{I_{out}}{I_{in}} = \frac{R_1}{R_2} + \frac{V_T(\ln(I_{in}/I_{out}))}{I_{in}R_2} + \frac{\Delta V_{BE}}{I_{in}R_2}$$

Assuming that the term $V_T[\ln(I_{in}/I_{out})]/I_{in}R_2$ is small compared with the other terms, then

$$\frac{I_{out}}{I_{in}} = \frac{R_1}{R_2} + \frac{\Delta V_{BE}}{I_{in}R_2}$$

References

1. K. C. Smith and A. S. Sedra, The current conveyor—A new circuit building block, *Proc. IEEE*, 56, 1368–1369, 1968.
2. A. Sedra and K. C. Smith, A second generation current-conveyor and its applications, *IEEE Trans. Circuit Theory*, CT-17, 132–134, 1970.

3. B. Wilson, High performance current conveyor implementation, *Electron. Lett*., 20(24), 990–991, 1984.
4. C. Toumazou, F. J. Lidgey, and C. Makris, Extending voltage-mode op-amps to current-mode performance, *Proc. IEE: Pt. G*, 137(2), 116–130, 1990.
5. PA630 Data Sheet, Photronics Co., Ottawa, PQ, Canada.
6. C. Toumazou, F. J. Lidgey, and D. Haigh, Eds., *Analogue IC Design—The Current-Mode Approach*, Exeter, England: Peter Peregrinus, 1990.
7. CCII01 Data Sheet, LTP Electronics, Headington, Oxford, England.
8. D. F. Bowers, A precision dual current-feedback operational amplifier, in *Proc. IEEE Bipolar Circuits Technol. Meet*., Minneapolis, MN, Sep. 1988, pp. 68–70.
9. D. F. Bowers, Applying current feedback to voltage amplifier, in *Analogue IC Design: The Current-Mode Approach*, edited by C. Toumazou, F. J. Lidgey, and D. G. Haigh, Eds. Exeter, England: Peter Peregrinus, 1990, Chap. 16, pp. 569–595.
10. I. A. Koullias, A wideband low-offset current-feedback op amp design, in *Proc. IEEE* 1989 *Bipolar Circuits Technol. Meet*., Minneapolis, MN, Sep. 18–19, 1989, pp. 120–123.
11. A. Payne and C. Toumazou, High frequency self-compensation of current feedback devices, in *Proc. IEEE ISCAS*, San Diego, CA, May 10–13, 1992, pp. 1376–1379.
12. T. Vanisri and C. Toumazou, Wideband and high gain current-feedback op-amp, *Electron. Lett*., 28 (18), 1705–1707, 1992.
13. A. Grebene, *Bipolar and MOS Analog Integrated Circuit Design*, New York: Wiley, 1984.
14. C. Toumazou, Ed., *Circuits and Systems Tutorials*, New York: IEEE ISCAS, 1994.
15. *High Performance Analog Integrated Circuits*. Élantec Data Book, Elantec (Intersil Corporation, Milpitas, CA), 1994.

3.2 Bipolar Noise

Alicja Konczakowska and Bogdan M. Wilamowski

Bipolar transistors and other electronic devices generate inherent electrical noise. This limits the device operation at a small-signal range. There are a few different sources of noise, such as thermal noise, shot noise, generation–recombination, $1/f$ (flicker noise), and $1/f^2$ noise, burst noise or random telegraph signal noise (RTS noise), and avalanche noise [1,6].

3.2.1 Thermal Noise

Thermal noise is created by random motion of charge carriers due to the thermal excitation [1]. This noise is sometimes known as the Johnson noise. In 1905 Einstein presented his theory of fluctuating movement of charges in thermal equilibrium. This theory was experimentally verified by Johnson in 1928. The thermal motion of carriers creates a fluctuating voltage on the terminals of each resistive element. The average value of this voltage is zero, but the power on its terminals is not zero. The internal noise voltage source or current source is described by Nyquist equation

$$\overline{v_{\mathrm{n}}^2} = 4kTR\Delta f, \quad \overline{i_{\mathrm{n}}^2} = \frac{4kT\Delta f}{R} \tag{3.75}$$

where
k is the Boltzmann constant
T is absolute temperature
$4kT$ is equal to 1.61×10^{-20} V · C at room temperature

The thermal noise is proportional to the frequency bandwidth Δf. It can be represented by the voltage source in series with resistor R, or by the current source in parallel to the resistor R. The maximum noise

power can be delivered to the load when $R_L = R$. In this case maximum noise power in the load is $kT\Delta f$. The noise power density $dP_n/df = kT$, and it is independent of frequency. Thus, the thermal noise is the white noise. The RMS noise voltage and the RMS noise current are proportional to the square root of the frequency bandwidth Δf. The thermal noise is associated with every physical resistor in the circuit. In a bipolar transistor, the thermal noise is generated mainly by series base, emitter, and collector resistances.

Spectral density of the equivalent voltage and currant thermal noise are given by

$$S_{vR} = 4kT_kR \tag{3.76}$$

or

$$S_{iG} = 4kT_kG \tag{3.77}$$

These spectral noise densities are constant up to 1 THz and it is proportional to temperature and to resistance of elements and as such can be used to indirectly measure:

- The device temperature
- Series distributed resistances of bipolar transistors (primarily base resistance)
- Quality of contacts and connections

3.2.2 Shot Noise

Shot noise is associated with a discrete structure of electricity and the individual carrier injection through the *pn* junction. In each forward biased junction, there is a potential barrier which can be overcome by the carriers with higher thermal energy. This is a random process and the noise current is given by

$$\overline{i_n^2} = 2qI\Delta f \tag{3.78}$$

Spectral density of the shot noise is temperature independent and it is proportional to the junction current:

$$S_{is} = 2qI \tag{3.79}$$

where

q is the electron charge
I is the forward junction current

Shot noise is usually considered as a current source connected in parallel to the small-signal junction resistance. The measurement of shout noise in modern nanoscale devices is relatively difficult since measured values of current are in the range of 100 fA.

Shot noise has to be proportional to the current and any deviation from this relation can be used to evaluate parasitic leaking resistances. It can be used for diagnosis of photodiodes, Zener diodes, avalanche diodes, and Schottky diodes.

3.2.3 Generation–Recombination Noise

The generation–recombination noise is caused by the fluctuation of number of carriers due to existence of the generation–recombination centers. Variation of number of carriers leads to changes of device conductance. This type of noise is function of both temperature and biasing conditions. The spectral density of the generation–recombination noise is described by

$$\frac{S_{g-r}(f)}{N^2} = \frac{(\overline{\Delta N})^2}{N^2} \cdot \frac{4\tau}{1 + (2\pi f \cdot \tau)^2} \tag{3.80}$$

where
$(\overline{\Delta N})^2$ is the variance of the number of carriers N
τ is the carrier lifetime

Spectral density is constant up to the frequency $f_{g-r} = 1/(2\pi\tau)$, and after that is decreasing proportionally to $1/f^2$.

In the case when there are several types of generation–recombination centers with different carrier life time the resultant noise spectrum will be a superposition of several distributions described by Equation 3.80. Therefore the spectral distribution of noise can be used to investigate various generation–recombination centers. This is an alternative method to deep level transient spectroscopy (DLTS) to study generation–recombination processes in semiconductor devices.

3.2.4 *1/f* Noise

The *1/f* noise is the dominant noise in the low-frequency range and its spectral density is proportional to *1/f*. This noise is present in all semiconductor devices under biasing. This noise is usually associated with material failures or with imperfection of a fabrication process. Most of research results conclude that this noise exists even for very low frequencies up to 10^{-6} Hz (frequency period of several weeks). This noise is sometimes used to model fluctuation of device parameters with time. There are two major models of *1/f* noise:

- Surface model developed by McWhorter in 1957 [7]
- Bulk model developed by Hooge in 1969 [8]

The simplest way to obtain $1/f$ characteristics is to superpose many different spectra of generation–recombination noises, where free carriers are randomly trapped and released by centers with different life times. This was the basic concept behind McWhorter model where it was assumed that

- In the silicon oxide near the silicon surface there are uniformly distributed trap centers
- Probability of the carrier penetration to trap centers is decreasing exponentially with the distance from the surface.
- Time constants of trap centers increases with the distance from the surface
- Trapping mechanisms by separate centers are independent

The resulted noise spectral density is given by

$$S_{1/f} \propto (\overline{\Delta N})^2 \int_{\tau_1}^{\tau_2} \frac{1}{\tau} \frac{4\tau}{1 + \omega\tau^2} \cdot d\tau = (\overline{\Delta N})^2 \cdot \frac{1}{f} \quad \text{for } 1/\tau_2 \ll \omega \ll 1/\tau_1 \tag{3.81}$$

The spectral density is constant up to frequency $f_2 = 1/(2\pi\tau_2)$, then is proportional to $1/f$ between f_2 and $f_1 = 1/(2\pi\tau_1)$, from frequency f_1 is proportional to $1/f^2$. The McWhorter model is primarily used for MOS devices.

For bipolar transistor Hooge bulk model is more adequate. In this noise model Hooge uses in the carrier transport two scattering mechanisms of carries: scattering on the silicon lattice and scattering on impurities. He assumed that only scattering on the crystal lattice is the source of the *1/f* noise, while scattering on the impurities has no effect on noise level. All imperfections of the crystal lattice leads to large *1/f* noise.

The noise spectral density for the Hooge model is

$$S_{1/f} = \frac{\alpha_H \cdot I^{\beta}}{f^{\gamma} \cdot N} \tag{3.82}$$

where
$\alpha_H = 2 \cdot 10^{-3}$ is the Hooge constant [8]
β and γ are material constants
N is the number of carriers

Later [9] Hooge proposed to use α_H as variable parameter, which in the case of silicon devices may vary from $5 \cdot 10^{-6}$ to $2 \cdot 10^{-3}$.

The $1/f$ noise is increasing with the reduction of device dimensions and as such is becoming a real problem for devices fabricated in nanoscale. The level of *1/f* noise is often used as the measure of the quality of devices and its reliability. Devices fabricated with well-developed technologies usually have much smaller level of *1/f* noise. The *1/f* noise (flicker noise) sometimes is considered to be responsible for the long term device parameter fluctuation.

3.2.5 Noise $1/f^2$

The noise $1/f^2$ is a derivative of *1/f* noise and it is observed mainly in metal interconnections of integrated circuits. It has become more evident for very narrow connections where there is a possibility of electromigration due to high current densities. In aluminum the electromigration begins at current densities of 200 $\mu A/\mu m^2$ and noise characteristics changes from $1/f^2$ to $1/f^{\gamma}$, where $\gamma > 2$. Also the noise level increases proportionally to the 3rd power of the biasing current:

$$S_{1/f^2}(f) = \frac{C \cdot J^{\beta}}{f^{\gamma} \cdot T} \cdot \exp(-E_a/k \cdot T) \tag{3.83}$$

where
$\beta \geq 3$, $\gamma \geq 2$
C is experimentally found constant
E_a activation energy of the electromigration
$k = 8.62 \cdot 10^{-5}$ eV/K is the Boltzmann constant

The degeneration of metallic layer is described by

$$\nu_d \propto J^n \exp(-E_a/k \cdot T) \tag{3.84}$$

Since Equations 3.83 and 3.84 have a similar character therefore the $1/f^2$ noise can be used as the measure of the quality of metal interconnections. This is a relatively fast and accurate method to estimate reliability of metal interconnections.

3.2.6 Burst Noise—RTS Noise

The burst noise is another type of noise at low frequencies [3,4]. Recently this noise is described as RTS noise. With given biasing condition of a device the magnitude of pulses is constant, but the switching time is random. The burst noise looks, on an oscilloscope, like a square wave with the constant magnitude, but with random pulse widths (see Figure 3.64). In some cases the burst noise may have not two but several different levels.

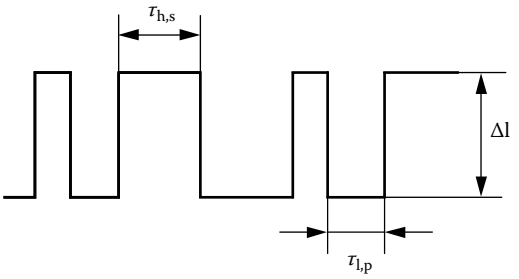


FIGURE 3.64 Example of RTS noise waveform.

Spectral density of the RTS noise has similar form like generation–recombination noise:

$$S_{\mathrm{RTS}} = C\frac{4\cdot(\Delta I)^2}{1+(2\pi f/f_{\mathrm{RTS}})^2} \tag{3.85}$$

where

$$C = \frac{1}{(\overline{\tau_{\mathrm{l}}}+\overline{\tau_{\mathrm{h}}})\cdot f_{\mathrm{RTS}}^2}$$

$f_{\mathrm{RTS}} = \frac{1}{\tau} = \frac{1}{\tau_{\mathrm{l}}} + \frac{1}{\tau_{\mathrm{h}}} = \frac{\overline{\tau_{\mathrm{l}}}+\overline{\tau_{\mathrm{h}}}}{\overline{\tau_{\mathrm{h}}}\cdot\overline{\tau_{\mathrm{l}}}}$ is the corner frequency, below this frequency spectrum of the RTS noise is flat
$\overline{\tau_{\mathrm{l}}}$ is the average time of pulses at low level
$\overline{\tau_{\mathrm{h}}}$ is the average time of pulses at high level

$$\overline{\tau_{\mathrm{l}}} = \frac{1}{P}\sum_{i=1}^{P}\tau_{\mathrm{l,p}}$$

$$\overline{\tau_{\mathrm{h}}} = \frac{1}{S}\sum_{j=1}^{S}\tau_{\mathrm{h,s}}$$

The intensity of the RTS noise depends on the location of the trap center with the reference to the Fermi level. Only centers in the vicinity of Fermi levels are generating the RTS noise. These trapping centers, which are a source for RTS noise, are usually the result of silicon contamination with heavy metals or lattice structure imperfections.

In the SPICE program the burst noise is often approximated by

$$\overline{i_{\mathrm{n}}^2} = K_{\mathrm{B}}\frac{I_{\mathrm{D}}^{A_{\mathrm{B}}}}{1+\left(\frac{f}{f_{\mathrm{RTS}}}\right)^2}\Delta f \tag{3.86}$$

where K_{B}, A_{B}, and f_{RTS} are experimentally chosen parameters, which usually vary from one device to another. Furthermore, a few different sources of the burst noise can exist in a single transistor. In such a case, each noise source should be modeled by separate Equation 3.85 with different parameters (usually different corner frequency f_{RTS})

Kleinpenning [10] showed that RTS noise exists with devices with small number of carriers, where a single electron can be captured by a single trapping center. RTS noise is present in submicrometer MOS transistors and in bipolar transistors with defected crystal lattice. It is present in modern SiGe transistors.

This noise has significant effect at low frequencies. It is function of temperature, collector current, induced mechanical stress, and also radiation. In audio amplifiers the burst noise sounds as random shoots, which are similar to the sound associated with making popcorn. Obviously, bipolar transistors with large burst noise must not be used in audio amplifiers and in other analog circuitry. The burst noise was often observed in epiplanar bipolar transistors with large β coefficients. It is now assumed that devices fabricated with well developed and established technologies do not generate the RTS noise. This is unfortunately not true for modern nanotransistors and devices fabricated with other than silicon materials.

3.2.7 Avalanche Noise

The avalanche noise is another noise component, which can be found in bipolar transistors. For large reverse voltages on the collector junction, the collector current can be multiplied by the avalanche

phenomenon. Carriers in the collector–base junctions gain energies in high electrical field, then lose this energy during collision with the crystal lattice. If the energy gained between collisions is large enough, then during collision another pair of carriers (electron and hole) can be generated. This way the collector current can be multiplied. This is a random process and obviously the noise source is associated with the avalanche carrier generation. The magnitude of the avalanche noise is usually much larger than any other noise component. Fortunately, the avalanche noise exists only in the *pn* junction biased with a voltage close to the breakdown voltage. The avalanche phenomenon is often used to build the noise sources [5].

Spectral density of the avalanche noise is frequency independent

$$S_{\mathrm{I}}(f) = \frac{2qI}{(2\pi f \cdot \tau)^2} \tag{3.87}$$

where I is an average value of the reverse biasing current.

3.2.8 Noise Characterization

Many different methods are used in literature for noise characterization. Sometimes the noise is characterized by an equivalent noise resistance, sometimes by an equivalent noise temperature, sometimes by an equivalent RMS noise voltage or current or sometimes by a noise figure.

3.2.8.1 Equivalent Noise Voltage and Current

The equivalent noise voltage or current is the most commonly used method for modeling the noise in semiconductor devices. The equivalent diagram of the bipolar transistor, including various noise components, is shown in Figure 3.65. The noise components are given by

$$\overline{i_{\mathrm{B}}^2} = \frac{4kT\Delta f}{r_{\mathrm{B}}}, \quad \overline{i_{\mathrm{E}}^2} = \frac{4kT\Delta f}{r_{\mathrm{E}}}, \quad \text{and} \quad \overline{i_{\mathrm{C}}^2} = \frac{4kT\Delta f}{r_{\mathrm{C}}} \tag{3.88}$$

$$\overline{i_{\mathrm{C}}^2} = 2qI_{\mathrm{C}}\Delta f \tag{3.89}$$

$$\overline{i_{\mathrm{B}}^2} = 2qI_{\mathrm{B}}\Delta f + K_{\mathrm{F}}\frac{I_{\mathrm{B}}^{A_{\mathrm{F}}}}{f}\Delta f + K_{\mathrm{B}}\frac{I_{\mathrm{B}}^{A_{\mathrm{B}}}}{1+(f/f_{\mathrm{B}})^2}\Delta f \tag{3.90}$$

Thermal noise is associated with physical resistors only, such as base, emitter, and collector series resistances. The small-signal equivalent resistances, such as r_π and r_o, do not exhibit thermal noise.

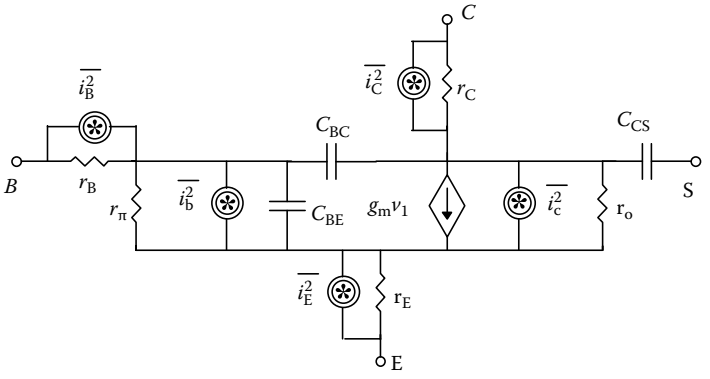


FIGURE 3.65 Equivalent diagram of the bipolar transistor which includes noise sources.

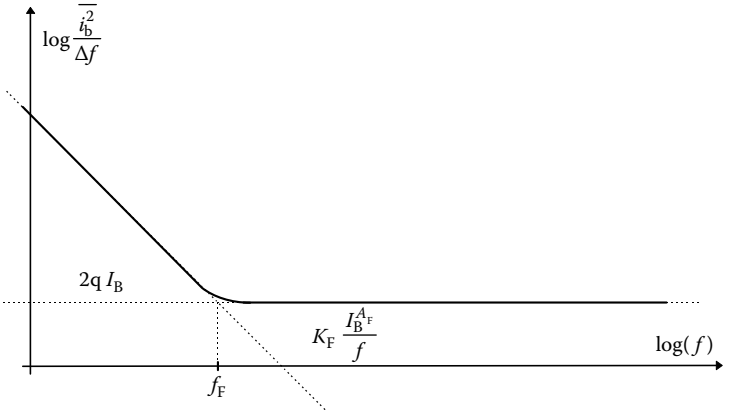


FIGURE 3.66 Bipolar transistor noise as a function of frequency.

The shot noise is associated with both collector and base currents. It was found experimentally that the $1/f$ noise and the burst noise are associated with the base current. The typical noise characteristic of a bipolar transistor is shown in Figure 3.66. The corner frequency of the $1/f$ noise can vary form 10 Hz to 1 MHz.

3.2.8.2 Equivalent Noise Resistance and Noise Temperature

The noise property of a two-port element can be described by a noise current source connected in parallel to the output terminals as Figure 3.67a shows. Knowing that noise current can be expressed as the shot noise of the DC device current the two-port noise can be expressed by means of an equivalent DC noise current

$$I_{eq} = \frac{\overline{i_n^2}}{2q\Delta f} \tag{3.91}$$

Another way to model the two-port noise in the two-port is to use the thermal noise at the input. This can be done using an additional "noisy" resistor connected to the input as Figure 3.67b shows

$$R_n = \frac{\overline{v_{n1}^2}}{4kT\Delta f} = \frac{\overline{v_{n2}^2}}{A_V^2 4kT\Delta f} \tag{3.92}$$

where

A_V is the voltage gain of the two-port

v_{n1}^2 and v_{n2}^2 are equivalent noise voltage sources at the input and the output, respectively

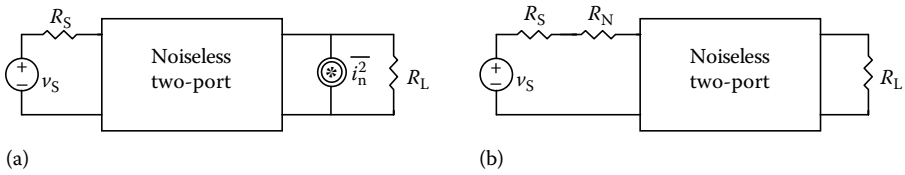


FIGURE 3.67 Noise characterization for two-ports, (a) using the noise source at the output, (b) using noise resistance R_n at the input.

The equivalent noise resistance is not a very convenient way to represent the noise property of the two-port. This additional resistance R_n must not be on the circuit diagram for small-signal analysis. To overcome this difficulty the concept of the equivalent noise temperature was introduced. This is a temperature increment of the source resistance required to obtain the same noise magnitude at the output if this source resistance is the only noise source. The noise temperature can be calculated from the simple formula

$$T_n = \frac{R_n}{R_s} 290^\circ \text{ K} \tag{3.93}$$

where R_n and R_s are shown in Figure 3.67b. It is customary to use 290° K as the reference room temperature for the noise temperature calculations.

3.2.8.3 Noise Figure

The noise figure is the ratio of the output noise of the actual two-port to the output noise of the ideal noiseless two-port when the resistance of the signal source R_s is the only noise source.

$$F = 10 \log \left(\frac{\text{total output noise}}{\text{output noise due to the source resistance}} \right) \tag{3.94}$$

The noise figure F is related to the noise resistance and the noise temperature in the following way

$$F = 10 \log \left(1 + \frac{R_n}{R_s} \right) = 10 \log \left(1 + \frac{T_n}{290^\circ \text{K}} \right) \tag{3.95}$$

The noise figure F is the most common method of noise characterization.

References

1. A. Van der Ziel, *Noise*. Prentice-Hall, New York, 1954.
2. J. L. Plumb and E. R. Chenette, Flicker noise in transistors, *IEEE Transactions on Electronic Devices*, ED-10, 304–308, Sept. 1963.
3. R. C. Jaeger and A. J. Broderson, Low-frequency noise sources in bipolar junction transistors, *IEEE Transactions on Electron Devices*, ED-17, 128–134, Feb. 1970.
4. R. G. Meyer, L. Nagel, and S. K. Lui, Computer simulation of $1/f$ noise performance of electronic circuits, *IEEE Journal of Solid State Circuits*, SC-8, 237–240, June 1973.
5. R. H. Haitz, Controlled noise generation with avalanche diodes, *IEEE Transactions on Electron Devices*, ED-12, 198–207, April 1965.
6. P. R. Gray, R. G. Mayer, *Analysis and Design of Analog Integrated Circuits*, 3rd ed. John Wiley & Sons, New York, 1993.
7. A. L. McWhorter, $1/f$ noise and germanium surface prosperities. *Semiconductor Surface Physics*, Ed. R. H. Kingdton. University of Pennsylvania Press, Philadelphia, PA, 1957, pp. 207–228.
8. F. N. Hooge, $1/f$ noise is no sourface effect. *Physics Letters*, 29A(3), 139–140, 1969.
9. F. N. Hooge, The relation between $1/f$ noise and number of electrons. *Physica B*, 162, 334–352, 1990.
10. T. G. M. Kleinpenning, On $1/f$ noise and random telegraph noise in very small electronic devices. *Physica*, B164, 331–334, 1990.

4

RF Communication Circuits

4.1 Introduction 4-1
4.2 System Level RF Design 4-2
General Overview • RF System Performance Metrics • RF Transceiver Architectures
4.3 Technology 4-9
Active Devices • Passive Devices
4.4 Receiver 4-12
LNA • Down Converter
4.5 Synthesizer 4-16
Topology • Oscillator • Prescaler • Fractional-*N* Synthesis
4.6 Transmitter 4-20
Up versus Down Conversion • CMOS Mixer Topologies • Power Amplifier
References 4-29

Michiel Steyaert
Catholic University of Leuven

Wouter De Cock
Catholic University of Leuven

Patrick Reynaert
Catholic University of Leuven

4.1 Introduction

During the last decade of last century, the world of wireless communications started to grow rapidly. Today, cellular handsets are the largest consumer market in the world. The main trigger was the introduction of digital coding and digital signal processing in wireless communications. The aggressive scaling of CMOS process technology driven by the memory and microprocessor market made CMOS a logical choice for integration of digital signal processing in wireless applications. The development of these high performance, low-cost CMOS technologies allowed integration of enormous amount of digital functionality on one chip. This enabled the use of sophisticated modulation schemes, complex demodulation algorithms, and high-quality error detection and correction to produce high data rate communication channels bringing the Shannon limit in sight [1].

The radio frequency (RF) front-ends are the interface between the antenna and the digital modem of the wireless transceiver. They have to detect very weak signals (μV) that come in at a very high frequency (10 GHz), and at the same time transmit high-power levels (up to several watts) at the same high frequencies. This requires high-performance analog circuits, like filters, amplifiers, and mixers that translate the incoming modulated data between the antenna and the A/D conversion and digital signal processing. Consumer electronic markets are mainly driven by low-cost and low-power consumption. This makes the RF front-ends the bottleneck for future wireless applications. Low-cost and low-power are both linked to high integration level. A high level of integration renders a significant space, cost, weight, and power reduction. A higher degree of integration requires less discrete components reducing the bill of materials cost. Keeping signals on chip greatly reduces power consumption since less I/O drivers are needed. Many

different techniques to obtain a higher degree of integration have been presented over the years [2–5]. This chapter introduces and analyzes some advantages and disadvantages and their fundamental limitations.

Parallel to the trend for further integration, there is the trend to integrate RF circuitry in CMOS technologies. While digital baseband processing has already been implemented in CMOS technology in several product generations, CMOS RF has only recently made its pace forward. For long time, many design houses believed complicated Mixed-Signal RF CMOS chips were impossible to realize. The main objective against CMOS RF was the lack of high-Q passive components and its poor noise performance. It took the persistence of some academic institutions and some pioneering firms to prove them wrong. It is clear that RF CMOS full potential would not have been unfold if only stand-alone radios were developed. CMOS RF systems on chip today implement all radio building blocks including phase-locked-loop (PLL), low-noise amplifier (LNA), power amplifier (PA), up- and down-conversion mixers, filters, and antenna switch. Furthermore, they include all digital baseband processing circuitry and ROM memory [6,7]. This reveals the real strength of CMOS RF over other ''better-suited'' technologies like Si Bipolar, BiCMOS, and Silicon Germanium (SiGe). Putting together RF and baseband in one chip permits compensation of lower radio performance with less expensive digital signal processing circuits, making its performance competitive with SiGe radios. Together with a possible 75% reduction of discrete components, RF CMOS offers the cheapest solution if one pursues the ultimate goal: A single chip including the physical layer (PHY) as well as the media access control (MAC) together with an MAC processor, memory, and I/O such as USBports or peripheral component interconnect (PCI) interfaces.

RF CMOS is not a matter of just replacing bipolar transistors with their CMOS counterpart. It requires a whole range a new architectures, techniques, and a high integration level. When compared with CMOS, SiGe requires less power for a certain gain and achieves a lower noise figure. The biggest drawback of CMOS is its inferior 1/f noise performance. This will only increase with the introduction of high-K dielectric materials in the gate of future CMOS technology nodes. CMOS design engineers therefore went looking for new topologies to reduce the impact of 1/f noise on the radio performance. Another problem that had to be overcome was the lack of high-Q passive components in CMOS technology. Extra processing steps as well as innovative layout and design techniques solved this problem. First, this chapter will analyze some concepts, trends, limitations, and problems posed by technology for high frequency design. Next we will discuss a variety of architectures used in modern RF CMOS transceivers. In the rest of the chapter, we will take a closer look at the different building blocks that appear in a typical RF transceiver. We will split this up between down-conversion, up-conversion, and frequency synthesis.

In the final section, we will take a look at RF CMOS's last barrier: RF power transmission. As CMOS gate lengths shrink, lower voltages are tolerated at the transistor terminals. High-quality impedance converters must therefore be placed between the antenna and the transistor's drain for high power transmissions. These are not available yet in integrated form. One of the major bottlenecks in CMOS PAs is combining high efficiency with high linearity. For high power transmission, designers are obliged to bias the PA high in its saturation region where linearity is low. Therefore, todays integrated PAs are limited to constant envelope modulation schemes like GSM. High efficient PAs still remain out of reach for modulation schemes with large peak-to-average power ratios like orthogonal frequency division modulation (OFDM). This chapter discusses some circuit techniques to circumvent this bottleneck bringing the ultimate goal of a single-chip CMOS solution that is compatible with all standards and is capable of adapting itself a step closer to reality.

4.2 System Level RF Design

4.2.1 General Overview

One of the main challenges facing the RF design engineer originates from the transmission medium used by RF systems. RF systems communicate through AIR by means of electromagnetic waves. Using air as transmission medium has one huge advantage: it gives the transceiver the ability to be mobile. However, there are some disadvantages to this high degree of freedom. There exists only one medium air, which is

consequently used by numerous applications. An overview of these applications and the part of the spectrum they use can be found on the Website of the National Telecommunications and Information Administration (NTIA) [8]. As a result, RF systems operate in a filled spectrum. Receivers will not only detect the wanted signals own to the application, but will also pick up other signals that will consequently be amplified and detected. These unwanted detected signals are called interferes. If the interferer is sufficiently large, it can corrupt the wanted signals preventing them to be properly demodulated and understood. On the transmit side of the application, unwanted signals are generated and transmitted. They are picked up by other applications and can distort their performance. These unwanted transmitted signals are called spurious signals. It is the designer's responsibility to keep these interferers and spurious signals as low as possible. Based on the earlier discussion, it is clear that one needs a regulator to manage this spectrum use. In the United States, this is done using a dual organizational structure; NTIA manages the federal government's use of the spectrum while the Federal Communications Commission (FCC) [9] manages all other uses.

Signals traveling through air also suffer from attenuation. There are several mechanisms causing attenuation such as free-space dispersion, fading, and multipath. These mechanisms depend heavily on the distance between transmitter and receiver, the frequency of transmission, and the environment. Discussion of these mechanisms, however, is beyond the scope of this text. More information concerning these topics can be found in Refs. [10,11]. As a result of these mechanisms, one can expect the received signal power to have a large variation since the distance between transmitter and receiver can change considerably due to the mobility. Performance of RF communication systems is also degraded by thermal noise. Noise is, like in other communication systems, the limiting factor when dealing with weak signals. The noise energy consists of two contributors. First, there is thermal noise which is determined by temperature and bandwidth and is out of control of the designer. On the other hand, there is system noise. This kind of noise can, within limits, be controlled by the designer to allow a certain minimum level of signal power to be detected by the system.

In the next sections, we will take a closer look at the challenges described in the earlier discussion. First, we will take a brief look at the tools and metrics RF designers use to describe and control the performance of their system in the presence of interferers and noise. We will end this section with a discussion of some commonly used transceiver architectures.

4.2.2 RF System Performance Metrics

As described in Section 4.2.1, the lowest signal power level that can be detected correctly by a receiver is limited by noise. The lowest power level that can be detected is usually called the receiver sensitivity. The receiver sensitivity is related to the signal-to-noise ratio (SNR) at the end of the receive chain (baseband). The SNR at baseband is determined by the bit error rate (BER) required by the application. It is usually expressed in terms of E_b/N_o. E_b is the energy per received bit and N_o is the noise power density received together with the bit. The relation between E_b/N_o and BER depends on the modulation scheme used in the application (e.g., Gaussian minimum shift keying (GMSK) in global system for mobile communications (GSM)) and is beyond the scope of this text. More information can be found in Ref. [12]. The SNR can be expressed in function of E_b/N_o as follows:

$$\mathrm{SNR} = \frac{S}{N} = \frac{E_b}{N_o} \times \frac{f_b}{B} \tag{4.1}$$

where

f_b is the bit rate

B is the receiver noise bandwidth

Note that the overall system noise at baseband N is the sum of thermal and circuit noise. This leads to a figure of merit that describes the circuit's performance. It is called noise figure when expressed in decibels and noise factor otherwise. Noise factor or figure is a measure for the excess noise that is contributed by

the circuit to the overall noise and is defined as the ratio between the SNR at the input of the receiver (SNR_i) and the SNR at the output of the receiver (SNR_o)

$$NF = \frac{\mathrm{SNR_i}}{\mathrm{SNR_o}} = \frac{(S/N)_\mathrm{i}}{(S/N)_\mathrm{o}} \tag{4.2}$$

If the receiver consists of different building blocks, one may want to know the noise figures of the different blocks and not only the overall noise figure. One can prove that in case of a series connection [12]

$$\mathrm{NF_{total}} = \mathrm{NF_1} + \frac{\mathrm{NF_2} - 1}{G_1} + \frac{\mathrm{NF_3} - 1}{G_1 G_2} + \cdots \tag{4.3}$$

where

NF_i are the noise factors of successive building blocks
G_i is their respective power gain

One can easily conclude from Equation 4.3 that building blocks earlier in the receive chain have a larger contribution to the overall noise figure than blocks at the end of the chain. This is the reason behind the use of an LNA at the input of an RF receiver. The large power gain combined with a low-noise figure will relax the noise specifications for the following blocks. The principle is explained in Figure 4.1. If an LNA is omitted and the mixer is put directly behind the antenna, the signal is drowned in the mixer noise and the sensitivity will be low. The power gain of the LNA, however, pushes the antenna signal above the noise floor of the mixer. As long as the output noise of the LNA is greater than the input noise of the mixer, the sensitivity is fully determined by the NF of the LNA.

RF systems often operate in an interference limited environment. Interference can also reduce receiver sensitivity. It is therefore more correct to describe the receiver sensitivity by its signal-to-noise plus interference ratio $S/(N+I)$ also known as the signal-to-noise and distortion ratio (SNDR). One of the mechanisms by which interference limits the performance is nonlinearity. It can reduce the signal power as well as increase interference. Large signals can saturate the receiver resulting in a gain compression, which reduces the signal power S. On the other hand, two large interfering signals can, due to non linearity, produce cross-product terms that fall on top of the wanted signal increasing the interference I. This cross-product generation is called intermodulation distortion (IMD). Nonlinearity performance is typically characterized by small signal linearity described by second-and third-order intercept points

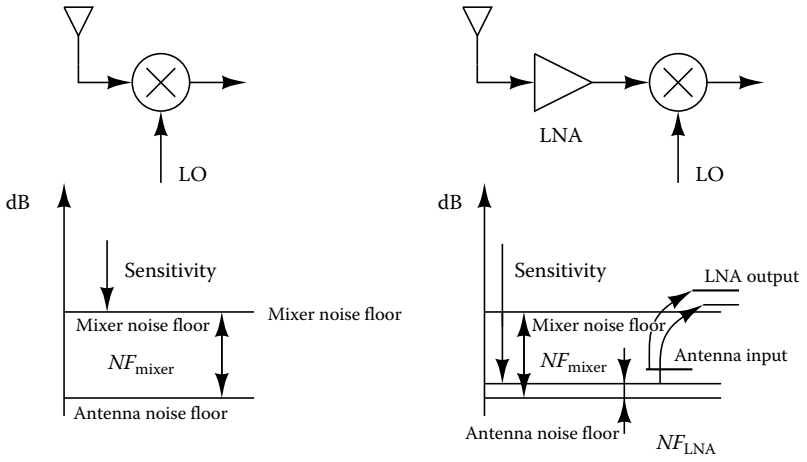


FIGURE 4.1 The benefit of using an LNA.

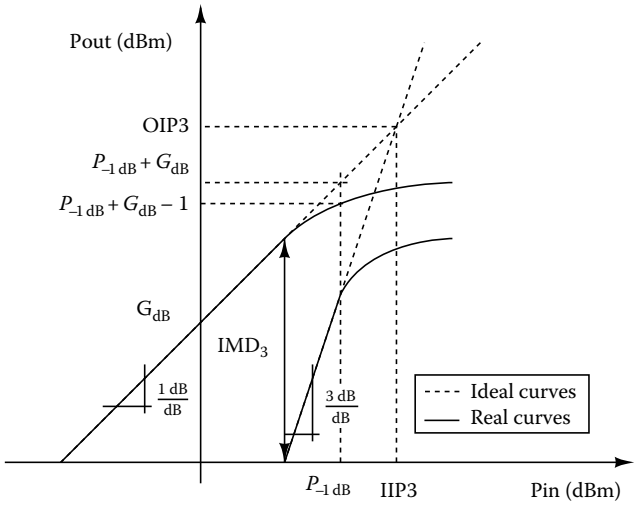


FIGURE 4.2 First- and third-order intermodulation as a function of the input power.

(IP2 and IP3) and large signal linearity described by the 1 dB compression point. Usually balanced topologies are used attenuating the second-order harmonics. Consequently, third-order nonlinearity will become the limiting factor. These concepts will be explained with the help of Figure 4.2. Gain compression is characterized by the 1 dB compression point ($P_{-1\ \mathrm{dB}}$) and is used to evaluate the ability of the system to cope with strong input or interference signals often referred to as blockers. It is defined as the input power for which the gain drops by 1 dB. By identifying the strongest signals at each stage of the design, one can calculate the required 1 dB compression point for each block in a receiver chain. As mentioned earlier, nonlinearity not only causes gain compression, but also generates IMD. This is produced by any pair of blockers that lie near the wanted signal. If two tones at f_1 and f_2 are applied to a nonlinear block, frequencies are produced not only at f_1 and f_2 but also at $2f_1 - f_2$, $2f_2 - f_1$, $3f_1$, $3f_2$, and so on. $f_1, f_2, 3f_1$, and $3f_2$ are not important since they lie far outside the frequency band of interest and can therefore be filtered out. $2f_1 - f_2$ and $2f_2 - f_1$, however, are potential problems as they can fall on top of the wanted signal band and remain unaffected by filtering. The ratio of any of the two cross products is called third-order IMD3. The output power of the intermodulation products grows at a faster rate than that of the wanted signal itself. Therefore, it follows that at a certain input power, the output power of the intermodulation signals will surpass the wanted signal. The input power level where this takes place is called the input-referred third-order intercept point (IIP3). The output power at this point is called the output-referred third-order intercept point (OIP3). Note that this is an imaginary point since gain compression occurs before this point is reached. If the receiver consists of different building blocks, one may want to know the contribution of the different building blocks to the overall linearity performance. One can prove that in case of a cascaded system

$$\frac{1}{\mathrm{IIP3}_{total}} = \frac{1}{\mathrm{IIP3}_1} + \frac{G_1}{\mathrm{IIP3}_2} + \frac{G_1 G_2}{\mathrm{IIP3}_3} + \cdots \tag{4.4}$$

where

$\mathrm{IIP3}_i$ are the input-referred third-order IPs of the successive building blocks
G_i is their respective power gain

One can conclude that, contrary to noise (see Equation 4.3), the last blocks in the receive chain has the largest influence on the overall linearity of the receiver. Equations 4.3 and 4.4 reveal a first trade-off. High gain at the input reduces noise constraints in the rest of the chain but increases the linearity requirements.

A last origin of distortion is due a nonideal local oscillator (LO) signal driving the mixers. In practice, the spectrum of an oscillator is never pure. There is always a certain amount of energy present close to the ideal LO frequency at $\omega_0 + \Delta\omega$. This can translate nearby frequency signals on top of the wanted signal also deteriorating the SNDR of the system. A figure of merit to describe this nonideal LO behavior is called the LO phase noise and is defined as the ratio of the power present in a 1 Hz band at a certain offset frequency $\Delta\omega$ from the carrier frequency ω_0 to the carrier power:

$$\mathcal{L}\{\Delta\omega\} = 10\log\left(\frac{\text{noise power in a 1 Hz band at } \omega_0 + \Delta\omega}{\text{carrier power}}\right) \tag{4.5}$$

4.2.3 RF Transceiver Architectures

In this section, a brief overview of some common transceiver structures will be discussed and contrasted to one another. The discussion will be restricted to the heterodyne transceiver, the zero-IF or direct conversion transceiver, and the low-IF transceiver. There exist numerous other types of transceivers but their properties can be understood by looking at these three structures as they are all variations or combinations of these three structures. First, the different receiver architectures will be discussed followed by there transmitter equivalent.

The heterodyne receiver has been the dominant choice in RF systems for many decades. The reason behind this is its high performance and adaptability to different standards. Figure 4.4 shows the operation of a heterodyne receiver. The broadband antenna signal is first fed to a highly selective RF filter (band select filter) that suppresses all interferers outside the wanted application band. An LNA boosts the wanted signal above the mixer noise floor and an LO generates a signal located at an offset frequency f_{IF} from the wanted signal. The result is that the following signals are down-converted by the mixer to f_{IF}

$$f_{wanted} = f_{LO} - f_{IF} \tag{4.6}$$

$$f_{image} = f_{LO} + f_{IF} \tag{4.7}$$

Not only the wanted signal is mapped onto IF (intermediate frequency), but also another signal called the image or mirror signal. This signal can corrupt the information content in such a way that the information is irreparable. To avoid this, an image reject filter is inserted before the mixer. This way, a highly attenuated version of the image signal is folded on top of the wanted signal, preventing the irreparable corruption of the information content of the signal. Figure 4.3 summarizes this operation. From Equation 4.6 and 4.7, one can see that the center of the image signal is located at a distance $2f_{IF}$ from the wanted signal. The choice of f_{IF} therefore determines the requirements for the image reject filter. If a very low f_{IF} is chosen, a very high-quality filter is needed to suppress the image frequency. To relax the filter specifications, f_{IF} is usually chosen relatively high and a series of down-conversion steps are performed. The heterodyne structure is then referred to as the superheterodyne receiver.

The heterodyne or superheterodyne receiver features a single path topology. Mismatch between different parts is not a issue here. Also LO feedthrough in the mixer is not a problem, since the wanted signal is never close to the LO frequency. In Figure 4.4, it can also be seen that the channel selection is done before the AGC-A/D structure. They will therefore only need to handle a limited dynamic range.

A drawback of the structure, however, is that all critical functions are realized with passive devices. Due to the high demands posed upon these structures, they are mostly implemented off-chip. The integratability of the heterodyne transceiver is therefore rather low. This induces an additional material cost. Moreover, the insertion loss of the passive filters needs to be compensated by a higher gain on-chip to keep the required SNR. Since the filters need to be driven at low impedance (e.g., 50 Ω), one has the choice between using complex impedance transformation structures or using low-output impedance

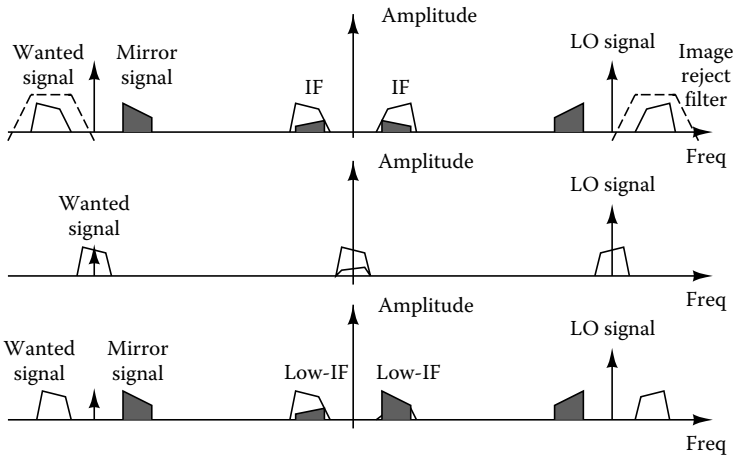


FIGURE 4.3 The down-conversion process in an IF, zero-IF, and a low-IF receiver.

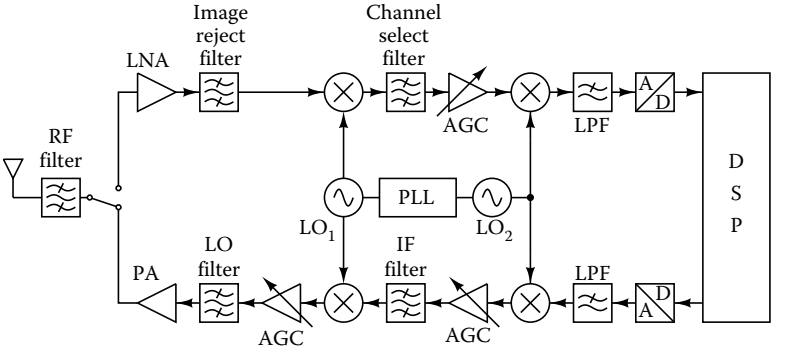


FIGURE 4.4 Heterodyne transceiver architecture.

buffers. Using low output impedance drivers, however, comes at the cost of an extra amount of extra power consumption.

The integratability however can dramatically be improved if one could find a way of getting rid of the external high-quality filters. This means looking for a way of suppressing the image frequency without filters. A first solution to this problem is obvious. Make the image signal the wanted signal or choose $f_{IF} = 0$. This solution is called the zero-IF receiver or direct-conversion receiver [13,14]. Another solution is related to the first one and is called the low-IF topology [3]. This topology takes advantage of the fact that the channels in the direct neighborhood of the wanted channel—the adjacent channels—are usually much weaker than the wanted signal and the signals laying further away. Furthermore, these frequency bands are usually regulated in the application specifications or by the FCC. So, if an IF-frequency is chosen so that the image frequency falls into this lower power bands, less image rejection is needed to keep the required SNR. Figure 4.5 shows the architecture of both a direct or zero-IF receiver and a low-IF receiver. The only difference between both can be found in the choice of IF-frequency. In a zero-IF receiver, the wanted channel is converted to DC and a mirrored version of the channel itself is superimposed onto the clean version of the signal. In a low-IF receiver, the wanted signal is down-converted to a low, nonzero IF, e.g., half the channel bandwidth, such that the mirror signal is the adjacent channel. The antenna signal is first passed through a band select filter. An LNA boosts the signal

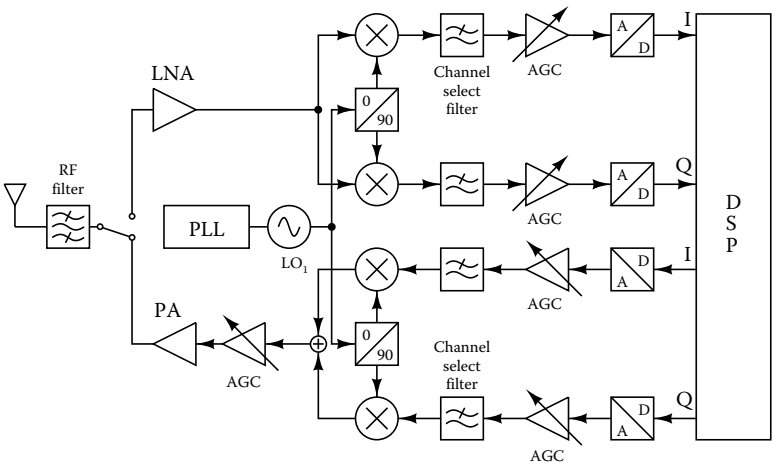


FIGURE 4.5 Direct transceiver architecture.

above the mixer noise floor. So far, there is no difference with the heterodyne receiver. After the LNA, however, the signal is fed to two different signal paths. The two signal paths are then down-converted by two mixers that are steered by two LO signals that are spaced 90° apart. The interstage filter has now become obsolete since the mirror signal will be neutralized by recombining the two signal paths after down conversion. This type of down conversion is called quadrature down conversion. Since the image signal and the wanted signal are separated in the digital signal processor (DSP), the real channel selection and image rejection are done in the digital back-end. This is a positive thing, since the digital domain is the natural biotope of CMOS. Since the image rejection and channel selection no longer rely on high-quality filtering, no external filters are required; therefore, one does not have to cope with their inevitable loss and one does not need low impedance drivers. This allows low power operation. However, the spreading of the signal over two independent signal paths has some drawbacks. The topology relies heavily on the symmetry between the two paths, every mismatch between the two paths will lead to a deterioration of the image suppression and an increased corruption of the wanted information content. Although one could think that that image rejection requirements are more relaxed for a zero-IF receiver since the image signal is a mirrored version of the wanted signal, this is not exactly true. For low-IF receivers, the image signal can be considered as noise for the wanted signal, since there is no correlation at all between the two bands. For zero-IF receivers, there is a strong correlation between image and wanted signal leading to a distortion of the wanted signal. The required image suppression is therefore dependent on the type of modulation that is used in the system. When a quadrature amplitude modulation (QAM) type modulation is used, one can calculate that the required image rejection for zero-IF is 20–25 dB while 32 dB rejection is required for low-IF systems [15]. As the wanted signals in both receivers are located at low frequencies (dc in case of zero-IF), the signal is susceptible to 1/f noise and dc-offset. Complicated feedback structures can get rid of the dc-offset; however, due to the finite time-constants in those loops, part of the signal is also canceled by the feedback. This can corrupt the signal in an unacceptable way. Low-IF topologies are less vulnerable. As long as the dc-offset does not saturate the A/D converters, there is no signal degradation. Due to the absence of filtering in the RF part, the A/D converters, however, have to deal with larger dynamic ranges. Fortunately, as the signals are at low frequencies, oversampled converters can be used which allow higher accuracies.

The same topologies exist for the transmitter side of the transceiver. The heterodyne as well as the direct up-conversion transmitter will be discussed. They are depicted in Figures 4.4 and 4.5. The early up-conversion architectures were in fact multistage architectures. They employed a number of mixing stages

and intermediate frequencies. The main advantage of this type of systems is that only one D/A converter is needed. Quadrature modulated signals are therefore generated in the digital domain. This topology puts high demands on the D/A converter since it must deliver signals at a higher IF frequency. The DSP on the other hand must be able to deliver perfectly matched I/Q signals. This approach requires the use of high-quality passives, multiple LOs. The same conclusions can be drawn as in the receiver. Due to the large number of external components, integratability is limited and power consumption will be high. Another implementation of this multistage architecture includes the use of two D/A converters. Quadrature modulated signals are then generated in the analog domain. Since they are generated at low frequencies, quadrature matching is superior. However, multiple RF filters are still needed, giving rise to a higher cost and power consumption. The topology, however, is not vulnerable to one of the main problems in the direct conversion architecture, oscillator pulling caused by the PA due to the fact that the PA output spectrum is far away from the voltage-controlled oscillator (VCO) frequency. Hereby the main problem in direct up-conversion circuits is addressed. In direct conversion transmitters, the transmitted carrier frequency is equal to the LO frequency. As can be seen in Figure 4.5, modulation and up conversion occur in the same circuit. The I/Q quadrature modulator takes the baseband (or low-IF) input signal and up-converts it directly to the desired RF frequency. This eliminates the need for RF passives and limits the number of amplifiers, mixers, and LOs. The simplicity of the architecture makes it an obvious choice when high integration levels are demanded. However, as mentioned before, the circuit suffers from one major drawback, the disturbance of the LO by the PA. This phenomenon is explained in detail in Refs. [16,17]. As the LO frequency lies in the transmit band, high demands are put on the LO/RF isolation. The system is also susceptible to I/Q mismatch errors, even the least phase mismatch or amplitude difference between I and Q path will result in distortion in the spectrum. However, the elimination of the IF stage in the transmitter leads to large saving in material cost and increases the robustness of the system as the number of discrete components that could fail is reduced. There is not only a cost saving in material cost, the direct up-converter architecture also allows a reduction in equipment size. This makes the circuit first choice for applications with stringent space constraints [18].

4.3 Technology

4.3.1 Active Devices

Since all high level or system level designs in the end need to be implemented in terms of actual active and passive components, it is no surprise that the transistor performance is of major importance for the overall system performance. It is therefore imperative to know the performance limitations of the technology one is working in and to be aware of the shortcomings of the model one is using. It is clear that conformity between measurements and simulation results will strongly depend on the accuracy of the models used with respect to the actual behavior of the devices. Although several compact models exist to describe MOSFET transistors, the BSIM [19] is considered as the de facto standard because it is the model that is generally provided by silicon foundries. Most models are quite accurate for low frequencies; however, most models fail when higher frequencies are to be modeled. ''High frequency'' means operating frequencies around 1/10th of the transistor's cutoff frequency f_t. Figure 4.6 gives an overview of f_t for different technology nodes. For a standard 0.18 μm technology with an f_t of around 50 GHz, this means 5 GHz is considered to be a high frequency. Another parameter is plotted in Figure 4.6, $f_{3\,dB}$ reflects the speed limitation of a transistor in a practical configuration. It is defined as the 3 dB point of a diode connected transistor [20] and takes into account the parasitic speed limitation due to overlap capacitances, drain-bulk junction, and gate-source capacitance while f_t only models the parasitic effect of the gate-source capacitance. In Ref. [21], an extended transistor model is presented that can be used for circuit simulation at RF frequencies. It is shown in Figure 4.7. All the extrinsic components are pulled out of the MOS transistor model, so that the MOS transistor symbol only represents the intrinsic part of the device. This allows to have access to internal nodes and model extrinsic components such as series

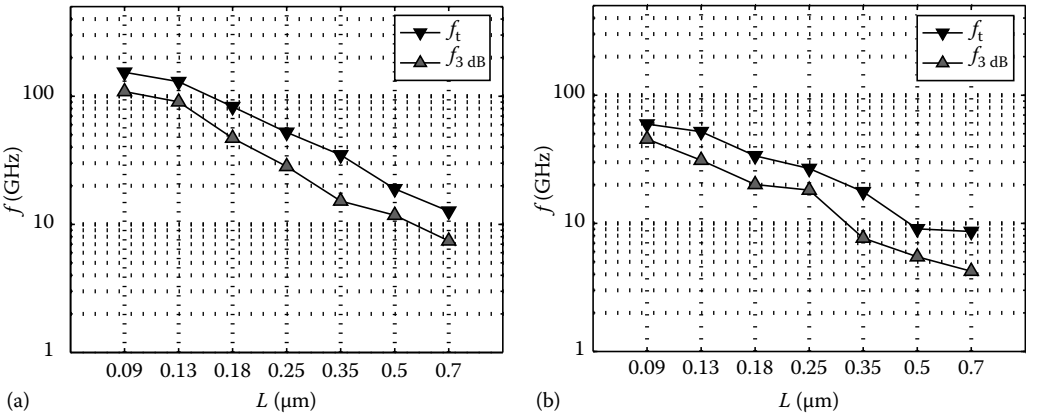


FIGURE 4.6 Maximum operating frequencies for different technology nodes.

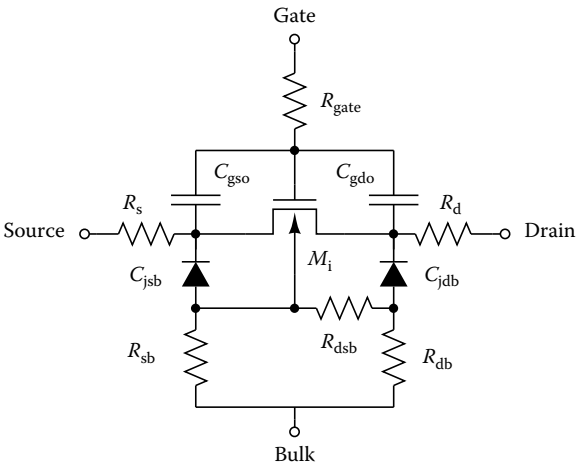


FIGURE 4.7 Extended RF transistor model.

resistances and overlap capacitances in a different way than what is available in the complete model. The source and drain series resistors are added outside the MOS model since the series resistances internal to the compact model are only used in the calculation of the I–V characteristic to account for the dc voltage drop across the source and drain. They do not add any poles and are therefore invisible for ac simulation. The gate resistance is usually not part of a MOSFET model, but plays a fundamental role in RF circuits and is therefore of outmost importance. The substrate resistors R_{dsb}, R_{sb}, and R_{db} have been added to account for the signal coupling through the substrate. Apart from the extra components added in the extended transistor model presented in Ref. [21], another point deserves some attention. The classical transistor model is based on the so-called quasi-static assumption. This means that any positive (negative) change in charge at the gate is immediately compensated by a negative (positive) change of charge in the channel. In reality, however, there will always be a delay in the charge buildup in the channel. Individual electrons (holes) will need a finite time to travel from bulk to the channel. This effect is called the non-quasi-static effect and has been described in Ref. [22–24]. This effect can be modeled by adding a resistance in series with the gate-source capacitance, introducing an extra time constant in the model.

$$\tau_{gs} = \frac{C_{gs}}{5g_m} = \frac{1}{5\omega_t} \tag{4.8}$$

This model is valid in strong innversion and within the long channel approximation. Although one could think that this effect is neglectable at realistic operating frequencies much lower than f_t, in bandpass applications, the gate-source capacitance can be tuned away by an inductor making the input impedance of the transistor purely resistive.

4.3.2 Passive Devices

For a long time, CMOS RF integration was believed to be impossible due to the poor quality of passive devices. Smaller CMOS geometries and innovative design and layout [25–27], however, have enabled high-quality passive components at high frequency to be integrated on chip. Four passive devices (resistors, inductors, capacitors, and varactors) will be discussed. First, one needs of figure of merit to qualify these passive devices. In general, the Q-factor is used for this purpose. Although there exist several definitions for the Q-factor, the most fundamental definition is based on the ratio between the maximum energy storage and the average power dissipation during one cycle in the device.

$$Q = \frac{\omega W_{\max}}{P_{\text{diss}}} \tag{4.9}$$

For an overview of other definitions of the Q-factor, the reader is referred to Ref. [28]. For a purely reactive element (capacitor or inductor), current through the element and voltage over the element are 90° out of phase. Hence, no power is dissipated in it. In real live, however, a certain amount of power will always be dissipated. Power dissipation supposes the presence of a resistance and a resistance always generates thermal noise. The Q-factor consequently is also a way of describing the pureness of a reactive device. Figure 4.8 shows some very common structures used in the modeling of reactive components used in RF circuits together with their Q-factor according to Equation 56.9. Low-Ohmic resistors are commonly available now in all CMOS technologies and their parasitic capacitance is such that they allow for more than high enough bandwidth. A more important passive device is the capacitor. In RF circuits, capacitors can be used for AC coupling. This enables DC-level shifting between different stages resulting in an extra degree of freedom enabling an optimal design of each stage. It also offers the possibility of lowering the power supply voltages. Another field, although not completely RF, where capacitors are commonly used is to implement switched capacitor circuits or arrays. This is favorable to using common resistors since capacitors in general offer better matching properties than resistors. The quality of an integrated capacitor is mainly determined by the ratio between the capacitance value and the value of the parasitic capacitance to the substrate. Too high a parasitic capacitor loads the transistor stages, thus reducing their bandwidth, and it causes an inherent signal loss due to a capacitive division.

The passive device, however, that got the most attention in the past is the inductor. It was long believed that high-quality integrated inductors were simply impossible in standard CMOS processes [29] and

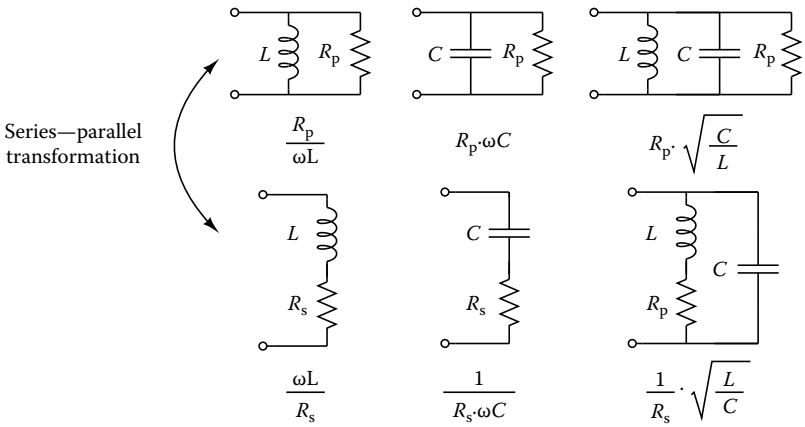


FIGURE 4.8 Quality factors of some common circuits.

could better be avoided if possible. However, due to the use of hollow spiral inductors and slightly altered process technology (thick top metal layer), one is now able to produce high-Q inductors in CMOS. The use of inductors on chip allows a further reduction of the power supply and offers compensation for parasitic capacitors by tuning them away resulting in higher operating frequencies. To be able to use integrated inductors in actual designs, an accurate model is needed. Reference [30] introduces such a model. One of the problems faced when modeling an inductor is how to model the substrate. One of the major drawbacks of inductors is the losses introduced by the substrate underneath the coil by capacitive coupling and eddy currents. This reduces the quality factor of the inductor.

A last passive component that is often encountered in RF CMOS designs is the varactor. It is mostly used for implementing tunable RF filters and VCOs. The different varactor types can be put in two classes: junctions and MOS capacitors. The latter can be used in accumulation and in inversion mode. For all cases, the devices have to be placed in a separate well to be able to use the well potential as the tuning voltage. For a standard NWELL process, the available configurations are therefore limited to p^+/n^- junction diodes and PMOS capacitors. When comparing the different varactor types, one should look at the following specifications: the varactor should offer a high Q-factor, the tuning range over which the capacitance can be varied should be compatible with the supply voltages used in the design, the physical structure should be as compact as possible to limit the area and its capacitance variation should be uniform over the complete tuning range as this makes feedback design easier. For an extended discussion about the different varactor types and their performance, the reader is referred to Ref. [27].

4.4 Receiver

4.4.1 LNA

The importance of the LNA has been explained earlier. The LNA is used to boost the received signal above the mixer noise floor. It is therefore critical that the LNA itself produces little noise. The noise figure of an LNA embedded in a 50 Ω system is defined as

$$\mathrm{NF} = 10 \log_{10} \left(\frac{\text{LNA output noise}}{\text{LNA output noise if the LNA itself was noiseless}} \right) \tag{4.10}$$

that is the real output noise power (dv^2/Hz) of the LNA (consisting of the amplified input noise power and all noise contributions generated in the LNA itself) divided by the amplified input power. Figure 4.9 shows some common input structures. Figure 4.9a shows a nonterminated common source input stage. Figure 4.9b shows the same input stage but now with an impedance matching at the input. Figure 4.9c shows the common gate input structure and finally Figure 4.9d shows a transimpedance amplifier

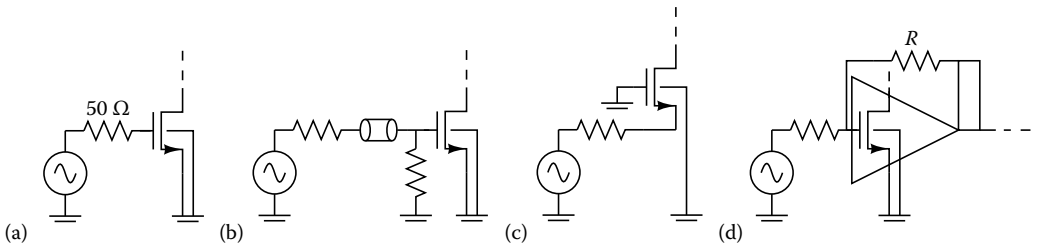


FIGURE 4.9 Some common LNA topologies.

structure that is commonly used for wideband applications. Their respective noise figures can be approximated with the following equations:

$$\text{Common source nonterminated (Figure 4.9a):} \quad \text{NF} = 1 + \frac{1}{50 \cdot g_m} \tag{4.11}$$

$$\text{Common source terminated (Figure 4.9b):} \quad \text{NF} = 2 + \frac{1}{50 \cdot g_m} \tag{4.12}$$

$$\text{Common gate (non)terminated (Figure 4.9c):} \quad \text{NF} = \left[\frac{1 + 50 \cdot g_m}{50 \cdot g_m}\right]^2 + \frac{1}{50 \cdot g_m} \tag{4.13}$$

$$\text{Common source transimpedance (Figure 4.9d):} \quad \text{NF} = 1 + \frac{1}{50 \cdot g_m} \cdot \left[\frac{R + 50}{R}\right]^2 + \frac{50}{R} \tag{4.14}$$

Figure 4.10 compares the noise figures of the different topologies. It is clear that the transimpedance structure and the not terminated common source circuit are far superior compared to the other structures as far as noise is concerned. For those circuits, the NF can be approximated as

$$\text{NF} - 1 \approx \frac{1}{50 \cdot g_m} = \frac{(V_{gs} - V_T)}{2 \cdot 50 \cdot I} \tag{4.15}$$

indicating that a low noise figure requires a large transconductance in the first stage. To generate this transconductance with high power efficiency, we need to bias the transistor in the region with a large transconductance efficiency, i.e., low $V_{gs} - V_T$. This, however, will result in a large gate-source capacitance limiting the bandwidth of the circuit. Together with the 50 Ω source resistance, the achievable bandwidth is limited by

$$f_{3\ \text{dB}} = \frac{1}{2\pi 50 C_{gs}} \tag{4.16}$$

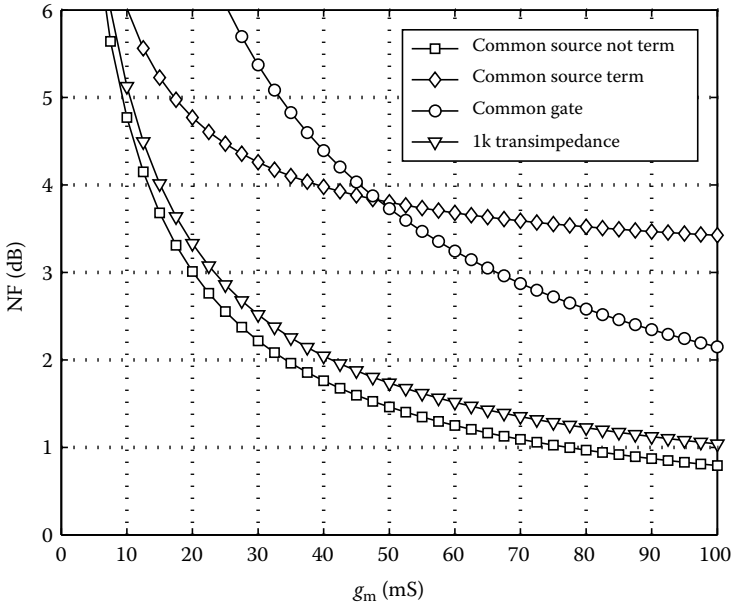


FIGURE 4.10 LNA input structure performance comparison.

When using the well-known approximative expression for the cutoff frequency of a transistor f_T

$$f_T = \frac{g_m}{2\pi C_{gs}}$$

one can conclude that

$$NF - 1 = \frac{f_{3\ dB}}{f_T} \tag{4.17}$$

This means that a low noise figure can only be achieved by making a large ratio between the frequency performance of a transistor, represented by f_T and the theoretical bandwidth $f_{3\ dB}$ of the circuit. Note that the $f_{3\,dB}$ used here is not the same as the one used in Section 4.3. Since f_T is proportional with $V_{gs} - V_T$, a low noise figure requires a large $V_{gs} - V_T$ and associated with it a large power drain. Only by going to deep submicron technologies will f_T become large enough to achieve low noise figures for gigahertz operation with low power consumption. In practice, the noise figure is further optimized by using noise and source impedance matching. These matching techniques often rely on inductors to cancel out parasitics by creating resonant structures. This boosts the maximum operation frequency to higher frequencies. More information concerning the design and optimization of common source LNAs can be found in Refs. [15,31].

At high antenna input powers, the signal quality mainly degrades due to in-band distortion components that are generated by third-order intermodulation in the active elements. Long channel transistors are generally described by a quadratic model. Consequently, a one transistor device ideally only suffers from second-order distortion and produces no third-order intermodulation products. As a result, high IIP3 values should easily be achieved. When transistor lengths shrink, however, third-order intermodulation becomes more important.

To start the analysis of the main mechanisms behind third-order intermodulation, one needs an approximate transistor model. A drain current equation that is strongly related to the SPICE level 2 and level 3 model is

$$I_{ds} = \frac{\mu_0 C_{ox}}{2n} \cdot \frac{W}{L} \cdot \frac{(V_{gs} - V_T)^2}{1 + \Theta \cdot (V_{gs} - V_T)} \tag{4.18}$$

with

$$\Theta = \theta + \frac{\mu_0}{L_{eff} \cdot v_{max} \cdot n} \tag{4.19}$$

where

- θ stands for the mobility degradation due to transversal electrical fields (surface scattering at the oxide–silicon interface)
- $\mu_0/(L_{eff} \cdot v_{max} \cdot n)$ models the degradation due to longitudinal fields (electrons reaching the thermal saturation speed)

As the θ-term is small in todays technologies, it can often be neglected relative to the longitudinal term. It can be seen from Equation 4.18 that for large values of $V_{gs} - V_T$, the current becomes a linear function of $V_{gs} - V_T$. The transistor is then conducting in the velocity saturation region. For smaller values of $V_{gs} - V_T$, the effect of Θ consists apparently in linearizing the quadratic relationship, but in reality, the effect results in an intermodulation behavior that is worse than in the case of quadratic transistors. The second-order modulation will be lower, but it comes at the cost of a higher third-order intermodulation. The following equations can be found by calculating the Taylor expansions of the drain current around a certain $V_{gs} - V_T$ value [32]:

$$\text{IIP2} \cong 10 + 20\log_{10}\left(\left(V_{gs} - V_T\right) \cdot (1 + r) \cdot (2 + r)\right) \tag{4.20}$$

$$\text{IIP3} \cong 11.25 + 10\log_{10}\left(\left(V_{gs} - V_T\right) \cdot V_{sv} \cdot (1 + r)^2 \cdot (2 + r)\right) \tag{4.21}$$

where

$$V_{sv} = \frac{1}{\Theta} \tag{4.22}$$

represents the transit voltage between strong inversion and velocity saturation and

$$r = \frac{V_{gs} - V_T}{V_{sv}} \equiv \Theta \cdot \left(V_{gs} - V_T\right) \tag{4.23}$$

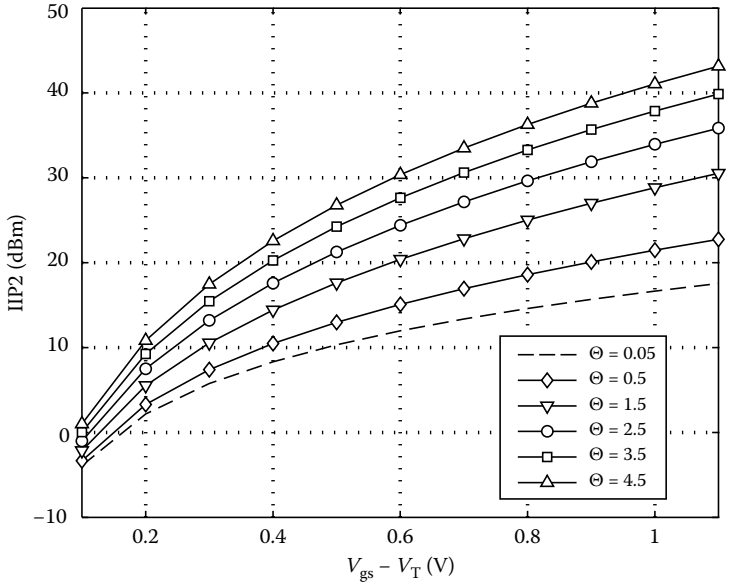
denotes the relative amount of velocity saturation. The transit voltage V_{sv} depends only on technology parameters. For deep submicron processes, this voltage becomes even smaller than 300 mV, which is very close to the $V_{gs} - V_T$ at the boundary of strong inversion. The expressions for IIP2 and IIP3 are normalized to 0 V dBm, the voltage that corresponds to a power of 0 dB in a 50 Ω resistor. For a given L_{eff}, the IIP3-value of a transistor is only a function of the gate overdrive voltage. Figure 4.11 plot the IIP2 and IIP3 in function of the gate overdrive voltage for different values of Θ. It can be seen that for a certain value of $V_{gs} - V_T$, the IIP2 increases for increasing Θ (decreasing gate lengths) which proves former statements. The picture becomes a bit more complicated when looking at the IIP3 plot. For practical values of Θ, one can distinguish two regions in the $V_{gs} - V_T$ domain. For high gate overdrive voltages, deep submicron transistors clearly exhibit better linearity because the saturation voltage becomes lower and the transistor will reach velocity saturation earlier. Short channel transistors therefore offer a maximum amount of linearity at a given power supply and require minimum $V_{gs} - V_T$ for a given IIP3. On the other hand, for low overdrive voltages, short channel transistors perform worse. Thus, to ensure a certain amount of linearity, one has to bias the transistors at a high enough overdrive voltage or apply some linearizing feedback technique (e.g., source degeneration). It can be shown that for the same equivalent g_m and the same distortion level, the required dc current is lower when local feedback is provided at the source. It comes, however, at the cost of a larger transistor and this can compromise the amplifier bandwidth.

4.4.2 Down Converter

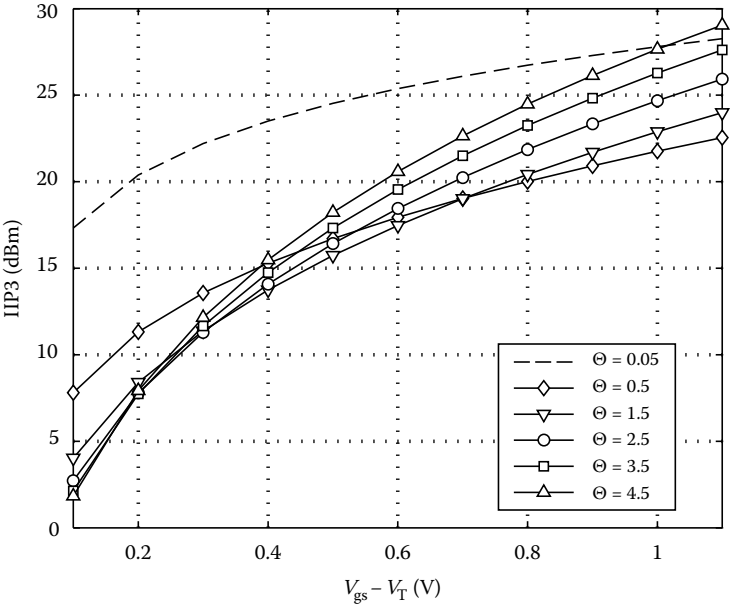
The most often used topology for a multiplier is the multiplier with cross-coupled variable transconductance differential stages. The use of this topology or related topologies (e.g., based on the square law) in CMOS is limited for high-frequency applications. Two techniques are used in CMOS: the use of the MOS transistor as a switch and the use of the MOS transistor in the linear region.

The technique often used in CMOS downconversion for its ease of implementation is subsampling on a switched-capacitor amplifier [33,34]. Here, the MOS transistor is used as a switch with a high input bandwidth. The wanted signal is commutated via these switches. Subsampling is used in order to be able to implement these structures with a low frequency op-amp. The switches and the switched capacitor circuit run at a much lower frequency (comparable to an IF frequency or even lower). The clock jitter must, however, be low so that the high frequency signals can be sampled with a high enough accuracy. The disadvantage of subsampling is that all signals and noise on multiples of the sampling frequency are folded upon the wanted signal. The use of a high-quality HF filter in combination with the switched capacitor subsampling topology is therefore absolutely necessary.

In Ref. [3], a fully integrated quadrature down-converter is presented. The circuit requires no external components, nor does it require tuning or trimming. It uses a double-quadrature structure, which renders a very high performance in quadrature accuracy. The down-converter topology is based on the use of MOS transistors in the linear region. By creating a virtual ground, a low frequency op-amp can be used for down conversion. The MOS transistor in the linear region results in a very high linearity for both the RF and the LO signal.



(a) Second-order intermodulation point



(b) Third-order intermodulation point

FIGURE 4.11 Linearity as a function of the gate overdrive voltage.

4.5 Synthesizer

One fundamental building block in every RF transceiver is the frequency synthesizer. The frequency synthesizer is responsible for generating the LO signal. The signal generated by the frequency synthesizer needs to be clean since low oscillator noise is crucial for the quality and reliability of the information

transfer. The signal should also be programmable and fast switching to be able to address all frequency channel within the specified time frame.

4.5.1 Topology

Synthesizers can usually be divided into three categories: table look-up synthesizer, the direct synthesizer, and the indirect or PLL synthesizer. In a table look-up synthesizer, the required sinusoidal frequency is created piece by piece using digital representations stored in memory of the amplitude at different time points of the sinusoidal waveform. The required building blocks are an accumulator that keeps track of the time, a memory containing a sine, a digital-to-analog converter (DAC), and a low-pass-filter to perform interpolation of the waveform to remove high frequency spurs. This type of synthesis is limited in frequency due to the access time of the memory and due to the maximum operation frequency of the high accuracy DAC. Moreover, high frequency spurs, generated due to the sampling behavior of the system, tend to corrupt the spectral purity of the signal. The direct frequency synthesizer employs multiplication, division, and mixing to generate the wanted frequency from a single reference. By repeatedly mixing and dividing, any accuracy is possible. The output spectrum is as clean as the reference frequency spectrum. Very fast frequency hopping is possible. The main disadvantages of this type of system is the difficult layout of the system, the high power consumption due to the numerous components, and the spectral purity can be corrupted by cross-coupling between stages. For generating high frequencies, the indirect or PLL type of frequency synthesizer often is the best choice. In a PLL, the synthesized frequency is generated by locking a locally generated frequency to an external frequency. The external frequency originates from a low frequency high quality crystal oscillator. To generate to local signal in the PLL, a VCO is used. A simple PLL topology is shown in Figure 4.12. A PLL includes following the building blocks: a VCO, a phase/frequency detector (PD/PFD), a loop filter, and a frequency divider or prescaler. The last building block is needed to derive a low frequency signal from the LO. This allows the signal to be locked to the external frequency through means of the PD. The PD is a circuit that compares the external frequency phase with the locally generated frequency phase and outputs an error voltage proportional to the phase difference. After filtering, this error signal is fed back to the VCO. This constitutes a control system. Under lock conditions, the external frequency and the locally generated frequency have a constant phase relationship.

$$F_{\text{out}} = N \cdot F_{\text{ref}} \tag{4.24}$$

The two signals are locked to each other, hence the name PLL. Even when a low-quality LO signal is generated, a high-quality signal can be synthesized. Due to the phase relationship between the input and the output frequency, the output signal will have the same spectral purity as the input high-quality signal. This is due to the fact that the loop remains locked to the input phase and therefore follows the phase deviations of that signal thus taking over its phase noise. This, however, is only true as long as the loop dynamics can follow the input signal. The loop dynamics are mainly determined by the bandwidth of the loop. For offset frequencies below the loop bandwidth, the phase noise is determined by the phase noise of the reference signal, for frequency offsets above the loop bandwidth, the output phase noise will be determined by the phase noise of the locally generated signal.

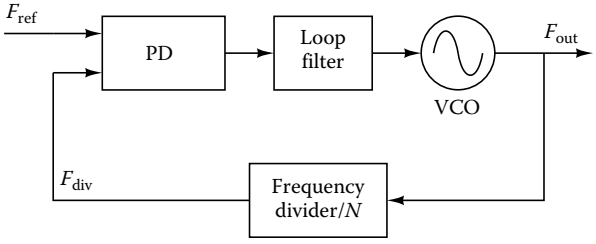


FIGURE 4.12 PLL-based frequency synthesizer.

When a programmable frequency divider is used in the loop, one can see that a set of frequencies can be synthesized. Suppose that the frequency by which the output signal is divided can be varied between N_1 and N_2, the output becomes

$$F_{\text{out}} = N_1 \cdot F_{\text{ref}},\ (N_1 + 1) \cdot F_{\text{ref}}, \ldots, N_2 \cdot F_{\text{ref}} \tag{4.25}$$

The PLL synthesizer is inherently slower than the other two types of synthesizers. The switching speed between two frequencies in Equation 4.25 is mainly determined by the loop bandwidth. Fast switching is only possible if a high loop bandwidth is implemented. Note that the loop bandwidth will also determine the phase noise performance. One, however, cannot indefinitely enlarge the loop bandwidth for stability reasons. A rule of thumb is that the loop bandwidth may not exceed 10% of the reference frequency to maintain stability. The loop bandwidth will also be limited by phase noise constraints. Spurious suppression and in-band phase noise levels will ultimately determine the loop bandwidth. When a low bandwidth has to be implemented, large capacitors will be needed. The total capacitance value is mainly determined by the need for implementing a stabilizing low frequency zero in the loop filter. This makes integration difficult as it will blow up silicon area and therefore increases the cost. One must therefore find ways to implement small bandwidth without having to use large capacitors. One obvious way of doing this is creating a low frequency pole through the use of a large resistance. This, however, will increase the phase noise. Other techniques, however, exist. In Ref. [35], a dual path loop filter is used. The filter consists of one active path and one passive path. Combining both will create a low frequency zero without the need for an extra resistor and capacitor. In Ref. [36], another technique is used to create the low frequency zero. It is created in the digital domain. The signal in the loop filter is combined with a sampled delayed version of itself. If the required switching speed is not achieved with a PLL configuration, one can make a combination of the direct synthesizer with the indirect synthesizer. In this topology, a number of PLLs are implemented and the outputs of all are combined with mixers. In this way, it is possible to synthesize a wide frequency range with a fast switching speed. This technique has recently been adopted for use in ultrawide band systems [37]. The major drawback of this technique, however, is that single sideband mixers have to be used. This requires accurate quadrature phases in all PLLs, low harmonic distortion, and well-matched mixers.

4.5.2 Oscillator

As it was mentioned above, the VCO is the main source of the phase noise outside the loop bandwidth. Therefore, its design is one of the critical parts of a PLL design. For the design of sub-gigahertz VCO, two oscillator types are often used: ringoscillators and oscillators based on a resonant tank composed of an inductor and a capacitor. The last type is referred to as an LC-tank VCO. The inductor in an LC-tank VCO can be implemented in two ways: an active implementation and a passive implementation. It can be shown [38,39] that the phase noise is inversely proportional to the power consumption. In LC-tank VCOs, the power consumption is proportional to the quality factor of the tank. Equations 4.26 through 4.28 show this relationship.

$$\text{Ring Osc. [39]}: \quad \mathcal{L}\{\Delta\omega\} \sim kTR \cdot \left(\frac{\omega}{\Delta\omega}\right)^2 \quad \text{with} \quad g_m = \frac{1}{R} \tag{4.26}$$

$$\text{Active LC [38]}: \quad \mathcal{L}\{\Delta\omega\} \sim \frac{kT}{2\omega C} \cdot \left(\frac{\omega}{\Delta\omega}\right)^2 \quad \text{with} \quad g_m = 2\omega C \tag{4.27}$$

$$\text{Passive LC [38]}: \quad \mathcal{L}\{\Delta\omega\} \sim kTR \cdot \left(\frac{\omega}{\Delta\omega}\right)^2 \quad \text{with} \quad g_m = R(\omega C)^2 \tag{4.28}$$

It is clear that for high frequency, a low power solution is only viable with an LC-tank VCO with a passive inductor. The use of a passive inductor, however, comes at a severe area penalty. Moreover, as it was discussed in Section 4.3, high-quality integrated inductors are difficult to make. For extremely low phase noise VCOs, bond wire inductors have been investigated [38]. The main drawback of using bondwires as inductors lies in reliability and yield. It is very difficult to make two bondwires exactly the same and reproduce this several times.

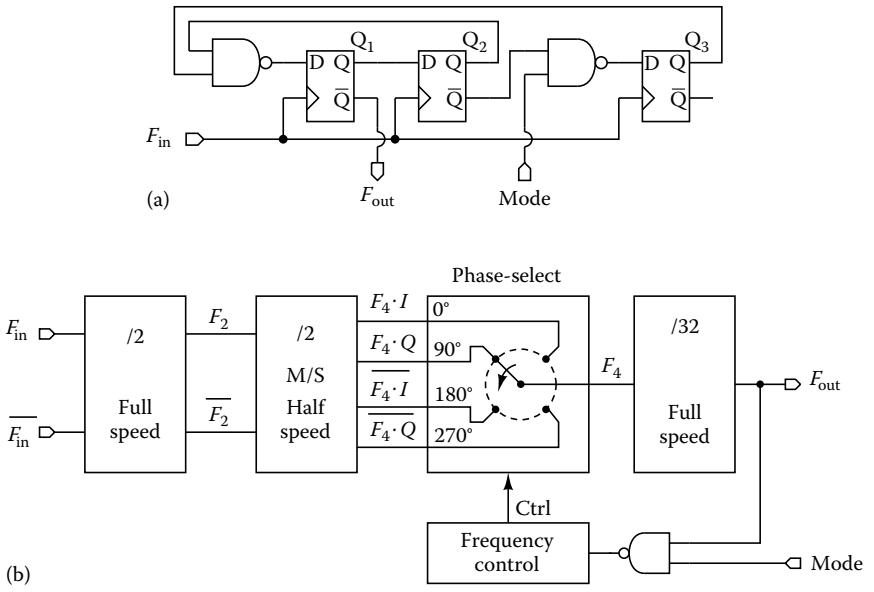


FIGURE 4.13 Dual modulus prescaler architecture: (a) D-flipflop based and (b) phase select topology. (From Craninckx, J. and Steyaert, M. *IEEE J. Solid-State Circuits*, 30(12), 1474, 1995.)

4.5.3 Prescaler

Several structures can be used as programmable divider. Programmable counters are the easiest solutions and are available in standard cell libraries. They are, however, limited in operation frequency. When high frequencies need to be synthesized, one can use a so-called prescaler. A prescaler divides by a fixed ratio and can therefore operate at high frequencies because they do not have to allow for delays involved with counting and presetting. A few high speed prescaller stages lower the speed used in the following counter stages. The disadvantage is that for a certain frequency resolution, the reference frequency has to be lowered. This slows the loop down as a lower bandwidth has to be implemented to maintain stability in the loop. A solution to this resolution problem is the use of dual- or multi-modulus prescalers. This circuit extends the prescaler with some extra logic to allow the prescaler to divide by N and $N+1$ in case of a dual-modulus prescaler and by N to $N+x$ in case of a multi-modulus prescaler. The speed decrease of this extra circuitry can usually be kept limited. Figure 4.13 shows two possible implementations of a dual-modulus prescaler. Implementation given by Figure 4.13a is a straightforward implementation based on d-flipflops. The critical path consists of a NAND gate and a d-flipflop. Implementation given by Figure 4.13b is a more complex implementation. It is based on the 90° phase relationship between the outputs of a master/slave toggle flipflop. It contains no additional logic in the high frequency path. The dual-modulus prescaler is as fast as an asynchronous fixed divider.

4.5.4 Fractional-*N* Synthesis

As it can be concluded from Equation 4.25, the minimal frequency resolution that can be achieved when using the topologies described previously is equal to F_{ref}. In GSM, e.g., the channels are 200 kHz spaced apart, this means that we need a frequency resolution of 200 kHz and therefore a low reference frequency. This results in high division ratios. The in-band phase noise of a PLL, however, is proportional to the division ratio, large ratios mean high in-band noise. As it is already explained, a low reference frequency will also result in a low PLL bandwidth and therefore a slow loop. Therefore, we need a technique that enables us to use a high reference frequency and still achieve the required frequency resolution. Fractional-N synthesizers solve this problem. Figure 4.14 makes things clearer. A basic fractional-N

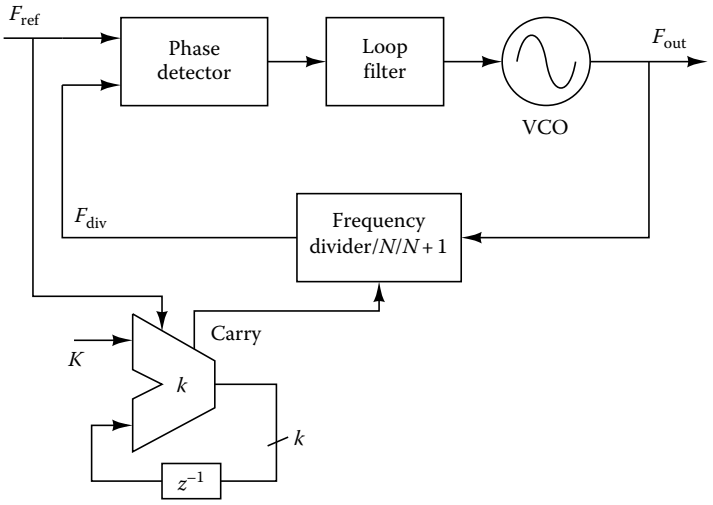


FIGURE 4.14 Fractional-N principle.

synthesizer consists, besides the standard PLL building blocks, of an accumulator and a dual modulus prescaler. By switching fast between the two division ratios, fractional divisions can be synthesized. The accumulator increases its value every reference clock cycle with a certain amount $K = n \cdot 2^k$. The dual-modulus prescaler is controlled by the accumulator overflow bit. If the accumulator overflows, the division ratio is $N+1$, otherwise it is N. On average, the dual-modulus prescaler divides K times by $N+1$ and $2^k - K$ times by N, resulting in a synthesized frequency of

$$N_{\text{frac}} = \frac{(2^k - K) \cdot N + K \cdot (N+1)}{2^k} = N + \frac{K}{2^k} = N + n \tag{4.29}$$

This means that also non-integer ratios can be synthesized and the above-mentioned limitations on the reference frequency is not applicable. There are, of course, drawbacks to the technique. The major one is the generation of spurs in the output spectrum due to pattern noise in the overflow signal. A detailed study of fractional-N synthesis, however, is beyond the scope of this chapter and the reader is referred to the open literature for further information. A thorough study of fractional-N synthesizers and their simulation can be found in Ref. [41].

4.6 Transmitter

Most RF communication systems are based on bidirectional data traffic. This means that apart from the receiver section, also a transmitter section must be implemented to complete a full transceiver. As explained in Section 4.2, a transmitter commonly includes a number of mixers, LO, and a PA. The LO is covered in a previous section; this section will therefore only describe the mixer and the PA used in up-conversion or transmitter systems.

4.6.1 Up versus Down Conversion

Although a huge amount of literature exists concerning the down-conversion process, up conversion has long time been neglected. This is rather surprising. When looking in Figure 4.5, one immediately notices

the parallelism between the receiver and the transmitter. The same functionality occurs. Both paths contain an amplifier (LNA, PA), both contain an interface to the digital domain (A/D, D/A), both contain a mixer and both are steered by the same LO system. The nature of the signals in both paths (input and output) has a huge influence on the circuit implementation. This seems logical for the LNA/PA analogy or the A/D and D/A-converter. Both have completely different topologies. Although the mixers in the up-conversion path and the down-conversion path face the same signals, there is typically not a great difference between the up- and down-conversion mixer topology. Most implementations are variations on the four-quadrant mixer topology, better known as the Gilbert-cell [42]. There are, however, fundamental differences between up conversion and down conversion. The first fundamental difference is located in the input signals of the mixer. In case of a down-conversion mixer, the input usually is a high-frequency, low-power signal surrounded by large blocking signals. In case of an up conversion, the input signal is a locally generated large baseband signal with a clean spectrum. At the output side, the situation is the opposite. A down-converted signal is a low frequency signal. It is, therefore, relatively easy to filter or apply feedback to cope with unwanted signals. At the transmitter side, however, a large and linear signal has to be processed within the technology dependend limited frequency range. Every extra building block placed between the mixer and the PA has to deal with high-frequency signals. Filtering is, therefore, impossible behind the up-conversion mixer as it will require large amount of power. Therefore LO leakage and other unwanted signals like intermodulation products have to be limited. A last, but not least difference lies within one of the design specifications of a mixer, the conversion gain G_c. It is defined as the ratio between the input power of the mixer and the output power. At the receiver side, the mixer input power is a design constraint as it is determined by the application. At the transmitter side, both input and output power are design variables. They can both be chosen freely. As it is easier and more power friendly to amplify a low-frequency signal, a large baseband signal is preferred.

4.6.2 CMOS Mixer Topologies

4.6.2.1 Switching Modulators

Many mixer implementations are based on the traditional variable transconductance multiplier with cross-coupled differential modulator stages [42]. It is depicted in Figure 4.15. The circuit was originally implemented in a bipolar technology and therefore based on its inherent translinear behavior. The MOS counterpart, however, can only be effectively used in switching mode. This induces the use of large LO driving signals and result in large LO feedthrough and power consumption. Moreover, when using a square-wave type modulation signal, a lot of energy is located at the third harmonic.

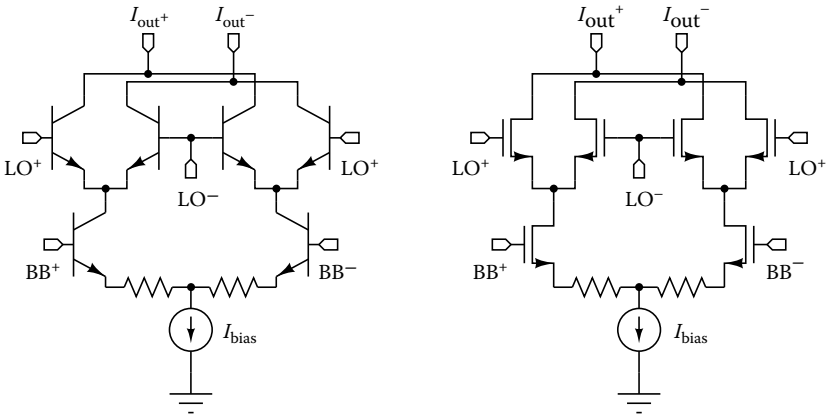


FIGURE 4.15 Bipolar and CMOS version of the Gilbert cell mixer.

This unwanted signal can only be filtered out by an extra blocking filter at the output. In CMOS the variable transconductance is typically implemented using a differential pair biased in the saturation region. To avoid distortion problems, large $V_{gs} - V_T$ values or a large source degeneration resistor is needed. This results in a large power consumption and noise problems. For upconversion, one also has to be aware that the high frequency current has to run through the modulating transistors. The source degeneration is therefore limited by bandwidth constraints. These problems can be circumvented by replacing the bottom differential pair with a pseudo-differential topology biased in the linear region [43].

4.6.2.2 Linear MOS Mixers

Figure 4.16 presents a linear CMOS mixer topology together with an output driver [44,45]. The circuit implements a real single-ended output topology avoiding the use of external combining. Some basic design ideas and some guidelines to optimize the circuit will be presented. The circuit is based on an intrinsically linear mixer topology. The circuit feature four mixer transistors biased in the linear region. Each mixer converts a quadrature LO voltage and a baseband signal to a linearly modulated current. The expression for the source-drain current for an MOS transistor in the linear region is given by

$$I_{DS} = \beta\left[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}\right] \tag{4.30}$$

This equation can be rewritten in terms of a DC and an AC term:

$$\begin{aligned} I_{DS} &= \beta(V_{DS} + v_{ds}) \cdot \left(V_{GS} - V_T - \frac{V_D - V_S}{2} + v_g - \frac{v_d + v_s}{2}\right) \\ &= \underbrace{\beta V_{DS}\left(V_{GS} - V_T - \frac{V_D - V_S}{2}\right)}_{\text{DC component}} \\ &\quad + \underbrace{\beta v_{ds}\left(V_{GS} - V_T - \frac{V_D - V_S}{2}\right) + \beta V_{DS}\left(v_g - \frac{v_d + v_s}{2}\right) + \beta v_{ds}\left(v_g - \frac{v_d + v_s}{2}\right)}_{\text{AC component}} \end{aligned} \tag{4.31}$$

Two signal have to be applied to a mixer transistor, the low frequency baseband signal, and the high frequency LO signal. Applying these signals may only result in the wanted high-frequency currents. Based on Equation 4.31, some conclusions can be drawn.

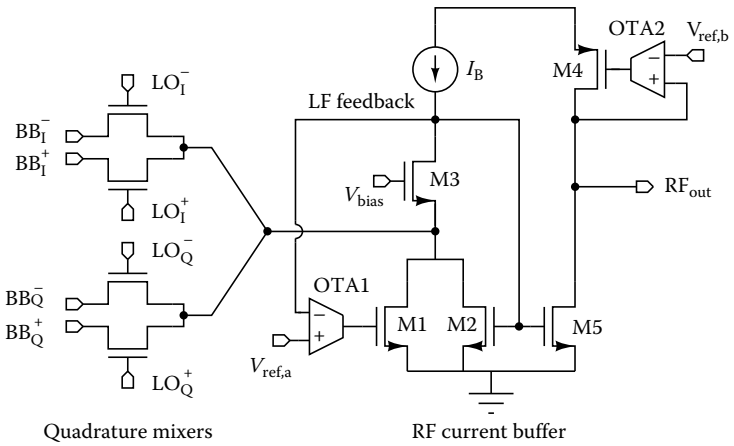


FIGURE 4.16 Schematic of a linear up-conversion mixer with output driver.

If the LO signal is applied to the drain/source of the mixer transistor, a product term

$$\beta v_{ds} \cdot \left(V_{GS} - V_T - \frac{V_D - V_S}{2} \right)$$

is formed. As this contains the product of a DC voltage with the oscillator signal, this component is located at LO frequency. It is preferable to avoid this frequency component to be formed. Therefore, the LO signal should not be applied to this node. Applying the LO signal to the gate of the mixer transistors results in the wanted behavior. According to Equation 4.31, only the high-frequency components are formed by

$$\beta v_g \cdot (V_{DS} + v_{ds})$$

By applying a zero-DC voltage between source and drain, only the high frequency mixer product is generated. The voltage to current conversion is perfectly balanced. The current of the four mixer paths is immediately added at the output of the mixers at a common node. This requires a virtual ground at that point that is achieved due to the low impedance input of the buffer stage (Figure 4.16). The total current flowing into the output buffer is given by

$$I_{MIX} = \beta \left(v_{bb_I}^2 + v_{bb_Q}^2 + 2 \cdot v_{LO_I} v_{bb_I} + 2 \cdot v_{LO_Q} v_{bb_Q} \right) \tag{4.32}$$

Equation 4.32 shows two frequency components in the modulated waveform. $\beta v_{LO} v_{bb}$ is the wanted signal. To prevent intermodulation products of the low frequency baseband signal $\beta v_{bb_I}^2$ with the wanted RF signal, the LF signal has to be suppressed at the current summing node. This is achieved by a low frequency feedback loop in the output buffer.

The low frequency feedback loop consists of OTA1 and transistors M1 and M3. It suppresses the low frequency signals resulting in a higher dynamic range of the output stage and decreases unwanted intermodulation products. It also lowers the input impedance of the output stage at low frequencies. The structure in fact separates the high- and low-frequency components of the input current and prevents the low frequency component to be mirrored to the output stage. The RF current buffer also ensures a low impedance at high frequencies at the mixer current summing node and therefore provides the necessary virtual ground.

4.6.2.3 Nonlinearity and LO Feedthrough Analysis

The difficulty to integrate IF filters is one of the reasons to implement direct conversion transmitters. This implies the the LO is at the same frequency as the RF signal and cannot be filtered out. To minimize the spurious signal components at the LO frequency, one has to isolate the origins of the unwanted frequency components. They can be categorized in three topics: capacitive feedthrough due to gate-source and gate-drain, parasitic overlap capacitances, and intrinsic nonlinearity of the mixers, mixer products due to a nonideal virtual ground.

When an ideal virtual ground is provided at the output of the mixer, capacitive LO feedthrough is canceled. This cancellation is never perfect, however, due to technology mismatch. The capacitive LO feedthrough for a single mixer transistor, biased in the linear region, is therefore given by

$$I_{LO} = 2\pi f \cdot v_{LO} \cdot WL \cdot \left(\frac{C_{ox}}{2} + \frac{C_{ov}}{L} \right) \tag{4.33}$$

where

c_{ox} is the oxide capacitance
C_{ov} is the gate-drain/source overlap capacitance
v_{LO} is the amplitude of the LO signal
f is its frequency

Based on Equations 4.32 and 4.33, the ratio between the LO feedthrough current and the modulated current is given by

$$\frac{i_{\text{signal}}}{\Delta(i_{\text{LO}})} = \frac{2\mu C_{\text{ox}} \frac{W}{L} v_{\text{bb}} v_{\text{LO}}}{\delta(i_{\text{LO}}) \cdot 2\pi f \cdot v_{\text{LO}} \cdot WL \cdot \left(\frac{C_{\text{ox}}}{2} + \frac{C_{\text{ov}}}{L}\right)} = \frac{\mu C_{\text{ox}} v_{\text{bb}}}{\delta(i_{\text{LO}}) \cdot \pi f \cdot L^2 \cdot \left(\frac{C_{\text{ox}}}{2} + \frac{C_{\text{ov}}}{L}\right)} \tag{4.34}$$

where $\delta(i_{\text{LO}})$ accounts for the relative difference in LO feedthrough for the different mixer transistors due to mismatch. Equation 4.34 shows that the ratio between the modulated current and the LO feedthrough current is independent from the LO amplitude and from the transistor width. Feedthrough will be less if shorter transistor lengths are used. The relative matching between the different mixer transistor will become worse, however, when shorter lengths are used [46]. One must therefore use Equation 4.34 with care. The δ will increase for smaller transistor length. With proper design and optimization, one should, however, be able to achieve a 30 dBC signal to LO feedthrough ratio even if two LO feedthrough currents are added instead of being canceled by the virtual ground ($\delta(i_{\text{LO}} = 1)$). When more realistic numbers of mismatch are considered (e.g., 10%), 50 dBC is easily achieved. The presented equations can, therefore, be used by the experienced designer to estimate the matching requirements and check if these requirements are realistic.

Another problem one faces is a possible DC-offset between the source and drain terminal of the mixer transistor. Equation 4.31 explains the problem. Ideally, no DC is present. The mixer then shows the required behavior. When a DC is present, however, one can see that components are generated at DC, the LO frequency due to multiplication with v_{g} and a component at baseband. While the low frequency components can be filtered out by the low frequency feedback in the output buffer, the component at the LO frequency remains. This component will, therefore, set the requirements for the tolerated DC-offset. A possible solution for this problem is measuring the DC-offset between source and drain. The offset is then controllable. The offset requirement is translated into an offset specification on the op-amps used in the feedback loops in the output buffer.

If the common mixer node is not a ideal virtual ground, the modulated current will be converted to a voltage dependent on the impedance seen on that node. The spectrum of the modulated signal will therefore be a combination of the modulated current spectrum and the frequency dependence of the impedance. When an impedance Z_{c} is considered at the common mixer node, the modulated current is given as the result of a second-order equation

$$\left(\beta Z_{\text{c}}^2\right) \cdot I^2 - (1 + 2\beta Z_{\text{c}} \cdot (V_{\text{GS}} - V_{\text{T}})) \cdot I + 2\beta v_1 v_{\text{g}} + \beta v_1^2 = 0 \tag{4.35}$$

It can be noticed that when $Z_{\text{c}} = 0$, Equation 4.35 is reduced to Equation 4.32. As Equation 4.35 is a second-order equation, it is a possible origin of distortion and therefore has to be taken into account. One side note should be made to the previous. Only currents that are not canceled out by the differential character of the mixer are converted in a voltage. This advantage of a balanced structure, however, is not valid for a nonideal voltage source at the input of the mixer transistors. If a nonideal voltage source is used at this node, each frequency component of the modulated current will be converted in a voltage according to the specific frequency-dependent impedance. These voltages are then similar to the baseband signal up-converted to the LO frequency. It is therefore essential to keep this node as low-impedance as possible.

Equation 4.32 is only valid if a very low impedance is seen at the source and drain terminals of the mixer transistors. If this condition is fulfilled, no unwanted high-frequency mixing components are present in the modulated signal. However, both in measurements and in simulations, a significant unwanted signal is noticed at $f_{\text{LO}} \pm 3f_{\text{bb}}$. One expects this component to originate from a $v_{\text{LO}} v_{\text{bb}}^3$ product term. However, Equation 4.35 only shows a second-order relationship. The observed product term must therefore find its origin in another effect. It is proved to be a result of short channel effects in an

MOS transistor. Both the effective mobility and the threshold voltage are affected by the gate-source and drain-source voltage. The calculated impact of the threshold voltage modulation cannot explain the observed effect; it is therefore assumed that it is a result of the mobility modulation. After some calculations, one can prove that the effective mobility is

$$\mu_{\text{eff}} = \frac{\mu_0}{1 + \theta \cdot (V_{\text{GS}} - V_{\text{T}})_{\text{dc}} + \theta \cdot \left(v_{\text{LO}} - \frac{v_{\text{bb}}}{2}\right) + \frac{\mu_0}{V_{\max} \cdot L} + \frac{\theta}{2} \cdot |v_{\text{bb}}|} \tag{4.36}$$

Substituting v_{bb} with $A\sin(\omega_{\text{bb}}t)$ and making a Fourier series expansion of $|v_{\text{bb}}|$ results in

$$\mu_{\text{eff}} = \frac{\mu_0}{B\left(1 + \frac{\theta}{B} v_{\text{LO}} - \frac{\theta A}{2B} \sin(\omega_{\text{bb}}t) + C\cos(2\omega_{\text{bb}}t) + D\cos(4\omega_{\text{bb}}t) + \cdots\right)} \tag{4.37}$$

with

$$\begin{aligned} A\sin(\omega_{\text{bb}}t) &= \text{the baseband signal} \\ B &\approx 1 + (V_{\text{GS}} - V_{\text{T}}) \\ C &= \frac{A}{B} \cdot \frac{4}{3\pi} \cdot \left(\frac{\mu_0}{V_{\max} \cdot L} + \frac{\theta}{2}\right) \\ D &= \frac{A}{B} \cdot \frac{4}{5 \cdot 3\pi} \cdot \left(\frac{\mu_0}{V_{\max} \cdot L} + \frac{\theta}{2}\right) \end{aligned}$$

Equation 4.37 shows that a second-order baseband frequency component $\cos(2\omega_{\text{bb}}t)$ appears. In the DC reduction factor B, the third term is an order of magnitude smaller than 1. Hence it appears that the magnitude C of the second-order component has a first order relationship to the baseband signal amplitude A. In the voltage to current relationship, μ_{eff} is multiplied with $v_{\text{LO}}v_{\text{bb}}$. As a result a mixing component at $f_{\text{LO}} \pm 3f_{\text{bb}}$ occurs. In the amplitude C of this distortion component, $\mu_0/(V_{\max} \cdot L)$ is dominant to $\theta/2$ for most submicron technologies. It is also important to notice that the distortion is inversely proportional to the gate length. This indicates that this effect will become even more apparent as gate lengths continue to scale down.

4.6.3 Power Amplifier

4.6.3.1 CMOS Power Amplification

The integration of PAs in a CMOS technology is impeded by the low supply voltage of the current deep-submicron and nanometer technologies. Apart from this, the relative high parasitic capacitances of the MOS transistor, at least compared to GaAs or SiGe transistors, and the relative low quality factor of on-chip inductors, further hinders the integration. On the other hand, the digital MOS transistor is optimized for switching and as such a lot of switching amplifiers have been integrated in CMOS with great success recently [47–53]. Furthermore, CMOS RF amplifiers are capable to break the 1 W barrier of output power performance [54]. In this section, the topic of switching RF amplifier is discussed first. In the second part, some linearization techniques will be discussed.

4.6.3.2 Switching Class E Amplifier

The Class E amplifier was invented in 1975 [55], but the first implementation of this amplifier in CMOS was reported in 1997 [47]. In contrast to the Class A, B, C, and F amplifiers, the Class E is designed in the time-domain. In theory, the Class E amplifier is capable to achieve an efficiency of 100%. In order to achieve this, the transistor and output network are designed in such a way that the drain through the transistor is separated in time from the voltage across the transistor. This avoids power dissipation in the transistor, a necessary requirement to achieve a high efficiency. If all other elements are assumed to be

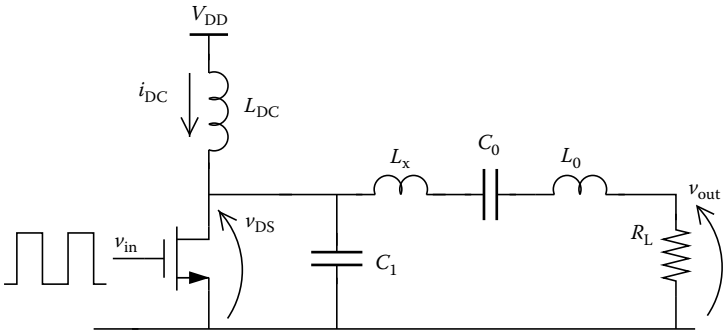


FIGURE 4.17 Basic Class E PA.

lossless, the amplifier is then indeed capable to achieve an efficiency of 100%. Figure 4.17 depicts the basic circuit of a CMOS Class E amplifier. The nMOS transistor should act as a switch and therefore it is driven by a square wave between zero and the maximum permissible gate voltage, which is normally equal to V_{DD}, the supply voltage of the technology. As such the nMOS transistor can be modeled by an ideal switch with a series resistance R_{on}. Inductor L_1 can be seen as the DC feed inductance, and in the original Class E theory, this inductor is assumed to be very large, and can be replaced by an ideal current source. Finally, inductor L_x and capacitor C_1 are the two crucial elements that create the Class E waveform at the drain of the nMOS transistor.

In a fully integrated CMOS implementation, the DC feed inductor L_1 cannot be made very large. First, this would require a huge silicon area, but more important, the relative high power loss of CMOS integrated inductors does not allow for such a large value. As such, the value of L_1 has to be reduced, and the current through the latter will not be a constant. The amplifier can still be designed to meet the Class E conditions, even with a small value of L_1. In fact, reducing the value of L_1 will result in a larger value for C_1 and a smaller value for L_x.

The capacitor C_1 and inductor L_x are constrained by the two Class E requirements, given below.

$$\text{Class E} \Leftrightarrow \begin{cases} v_{DS}(t = t_1) & = & 0 \\ \left.\dfrac{dv_{DS}(t)}{dt}\right|_{t=t_1} & = & 0 \end{cases}$$

In Figure 4.18, the drain voltage and current for Class E operation are shown. Solving the two Class E equations will give a value for C_1 and L_x. Finally, the value of the load resistance is constrained by the required output power. To achieve sufficient output power in a low voltage CMOS technology, an impedance matching network is required between the 50 Ω load or antenna impedance and the Class E amplifier. The on resistance of the nMOS transistor can be written as

$$r_{on} = \frac{L}{\mu_n C_{ox} W (V_{GS} - V_T)} \tag{4.38}$$

The lower the on resistance, the higher the efficiency of the amplifier, and thus it is beneficial to increase the width of the nMOS transistor. However, that large transistor cannot be directly connected to the up-conversion mixer, and several amplifying stages are needed between them. If the nMOS transistor has a large gate width, more power will be consumed by the driver stages and thus the overall efficiency of the amplifier, defined as

$$\eta_{oa} = \frac{P_{out}}{P_{DC,PA} + P_{DC,DRV}} \tag{4.39}$$

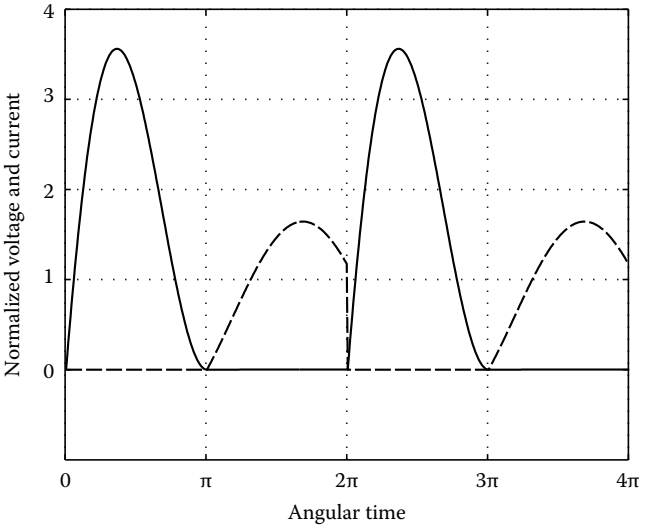


FIGURE 4.18 Voltage (solid line) and current (dashed line) of the Class E PA.

will have a maximum value for a specific transistor width. The overall efficiency is not always a good figure to compare PAs, since that figure can never reach 100%, even if each of the stages has a conversion efficiency of 100%. After all, the power consumed by the driver stages will never flow to the output load, but is only needed to switch on and of the next stage in line.

The power added efficiency (PAE) defined as

$$\text{PAE} = \frac{P_{\text{out}} - P_{\text{in}}}{P_{\text{DC}}} \tag{4.40}$$

is a useful definition for stand-alone PAs that have an input matched to 50 Ω. However, one should be aware whether the DC power consumption of the driver stages is included in P_{DC}.

Another important aspect of switching amplifiers is the reliability. A drawback of the Class E amplifier, at least compared to the Classes B and F, is that the drain voltage goes up to several times the supply voltage of the amplifier. This might cause reliability problems. On the other hand, the switching nature of the amplifier alleviates this. After all, due to the switching, voltage and current are separated in time. In other words, the high voltage peaks are not accompanied by a drain current, and when the drain current is high, the voltage across the switch is low. This is a big advantage compared to other types of amplifiers.

Figure 4.17 depicts another benefit of the Class E amplifier. For Class E operation, a shunt capacitance C_1 is required at the drain. However, in CMOS, there is already a large parasitic drain capacitance, and this capacitance can now become part of the amplifier. In Classes B and F amplifiers, that parasitic capacitance will create a low impedance for the harmonics that are crucial for the high efficiency of Classes B and F. Therefore, CMOS seems to be the natural habitat of the Class E amplifier.

4.6.3.3 Linearization of CMOS RF PAs

Switching amplifiers only have phase linearity, and therefore are only useful for constant envelope systems like Bluetooth and GSM. However, modern RF communication systems like UMTS, CDMA-2000, and WLAN allow amplitude modulation to increase the datarate of a wireless link. The only way to recover or restore the amplitude linearity of a switching amplifier is by modulating the supply voltage or by combining two nonlinear amplifiers. Systems that modulate the supply voltage of a switching

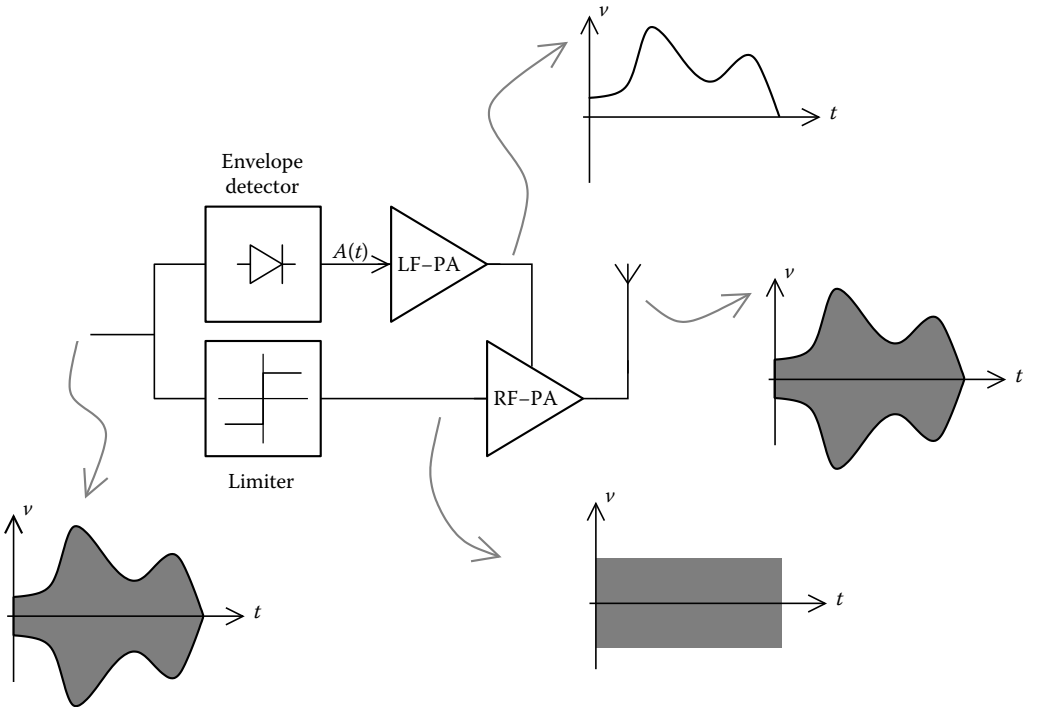


FIGURE 4.19 Kahn technique linearized PA.

amplifier are denoted as ''envelope elimination and restoration,'' ''polar modulation,'' or ''supply modulation.'' They originate from the Kahn technique (see Figure 4.19) that was already employed in vacuum tube amplifiers. In CMOS, one can make use of the availability of digital signal processing to directly create the amplitude and phase signal, and as such, the limiter and envelope detector of Figure 4.19 can be avoided. Furthermore, AM-AM and AM-PM predistortion is relative easy to implement. The general picture of polar modulation is shown in Figure 4.20. An another advantage of polar-modulated amplifiers is that the entire phase path carries a constant envelope signal and thus one can use nonlinear or saturated blocks in the upconversion path. Furthermore, amplitude and phase feedback are relatively easy to implement. Another group of techniques combine the output of two constant envelope amplifiers that have a different in phase. The two amplifiers are combined through a transformer, a power combiner, or through transmission lines, and the output is, in general, the sum of the two amplifiers. These systems are called ''outphasing'' or ''LINC,'' depending on the used combiner. Depending on their phase difference, the resulting output envelope can be higher or lower, and thus has amplitude modulation, as shown in Figure 4.21. The major drawback of these techniques is the difficulty to implement the power combiner in CMOS. Also, feedback is not as easy to implement in these systems. On the other hand, these systems allow to efficiently amplify signals that have a very high modulation bandwidth.

Apart from the two groups discussed in this section, several other techniques exist to amplify an amplitude-modulated signal. There is no ''ideal'' solution for CMOS integration. An alternative solution or approach is to use a linear amplifier with an efficiency improvement technique, such as the Doherty amplifier or the bias adoption technique. However, the linearization of nonlinear amplifiers has the advantage that switching or nonlinear amplifiers can be used, which are easier to implement in CMOS. Furthermore, the RF driver stages and all the blocks preceding the RF amplifier can be nonlinear as well. Needless to say, this is a huge advantage in low voltage technologies.

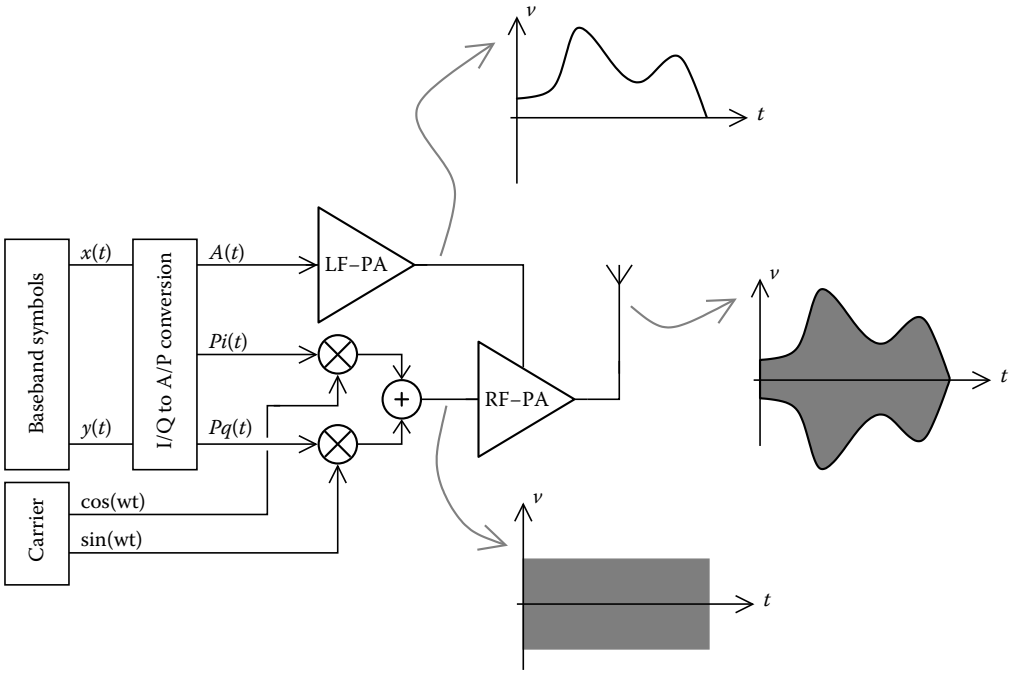


FIGURE 4.20 DSP based polar modulation architecture.

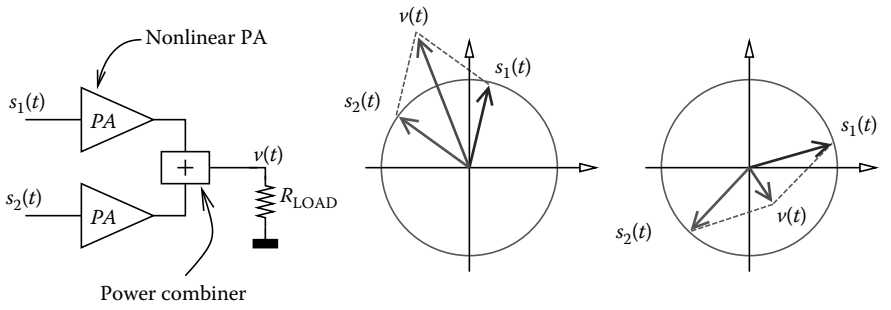


FIGURE 4.21 LINC or outphasing architecture.

References

1. C. E. Shannon, Communication in the presence of noise, in *Proceedings of the IRE*, vol. 37, Jan. 1949, pp. 10–21.
2. J. Sevenhans, A. Vanwelsenaers, J. Wenin, and J. Baro, An integrated Si bipolar transceiver for a zero-IF 900MHZ GSM digital mobile radio front-end of a hand portable phone, in *Proceedings of the Custom Integrated Circuits Conference, CICC*, May 1991, pp. 7.7.1–7.7.4.
3. J. Crols and M. Steyaert, A single Chip 900MHz CMOS receiver front-end with a high performance low-IF topology, *IEEE J. Solid-State Circuits*, 30(12), 1483–1492, 1995.
4. P. R. Gray and R. G. Meyer, Future directions in silicon ICs for RF personal communications, in *Proceedings of the Custom Integrated Circuits Conference, CICC*, May 1995, pp. 91–95.

5. A. A. Abidi, Low-power radio-frequency ICs for portable communications, *Proc. IEEE*, 83(4), 544–569, 1995.
6. S. Mehta, D. Weber, M. Terrovitis, K. Onodera, M. Mack, B. Kaczynski, H. Samacati, S. Jen, W. Si, M. Lee, K. Singh, S. Mendis, P. Husted, N. Zhang, B. McFarland, D. Su, T. Meng, and B. Wooley, An 802.11g WLAN SoC, in *ISSCC Digest of Technical Papers*, vol. 48, Feb. 2005, pp. 94–95.
7. H. Darabi, S. Khorram, Z. Zhou, T. Li, and B. Marholev, A fully integrated SoC for 802.11b in 0.18μm CMOS, in *ISSCC Digest of Technical Papers*, vol. 48, Feb. 2005, pp. 96–97.
8. National Telecommunications and Information Administration, Office of Spectrum Management (NTIA-OSM), http://www.ntia.doc.gov/osmhome/osmhome.html.
9. Federal Communications Commission (FCC), http://www.fcc.gov/.
10. R. L. Freeman, *Radio System Design for Telecommunications*, 2nd ed., Wiley, New York, 1997.
11. T. S. Rappaport, *Wireless Communications: Principles and Practice (Communication, Engineering, and Emerging Technologies)*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2001.
12. L. W. Couch, *Digital and Analog Communication System*, Prentice Hall, Upper Saddle River, NJ, 1997.
13. A. Abidi, Direct conversion radio transceivers for digital communications, *IEEE J. Solid-State Circuits*, 30(12), 1399–1410, 1995.
14. B. Razavi, Design considerations for direct conversion receivers, *IEEE Trans. Circuits Syst. II*, 44(6), 428–453, 1997.
15. J. Janssens and M. Steyaert, *CMOS Cellular Receiver Front-Ends: From Specification to Realization (The International Series in Engineering and Computer Science)* Springer, The Netherlands, 2002.
16. K. Kurokawa, Injection locking of microwave solid-state oscillators, *Proc. IEEE*, 61, 1386–1410, Oct. 1973.
17. B. Razavi, A study of injection locking and pulling in oscillators, *IEEE J. Solid-State Circuits*, 39(9), 1415–1424, 2004.
18. M. Feulner, Direct up-conversion lowers base-station costs, *Wireless Europe Magazine*, April–May 2005, pp. 22–25, wireless.iop.org.
19. W. Liu, X. Jin, X. Xi, J. Chen, M.-C. Jeng, Z. Liu, Y. Cheng, K. Chen, M. Chan, K. Hui, J. Huang, R. Tu, P. K. Ko, and C. Hu, *BSIM3v3.3 MOSFET Model, User's Manual*, University of California, Berkeley, 1999.
20. M. Steyaert and W. Sansen, Opamp design towards maximum gain-bandwidth, in *Proceedings of the AACD Workshop*, Delft, Mar. 1993, pp. 63–85.
21. C. Enz and Y. Cheng, MOS transistor modeling for RF IC design, *IEEE J. Solid-State Circuits*, 35(2), 186–201, 2000.
22. J. Janssens, J. Crols, and M. Steyaert, Design of broadband low-noise amplifiers in deep submicron CMOS technology, in *Analog Circuit Design. 1 Volt Electronics, Mixed-Mode Systems, Low-Noise and RF Power Amplifiers for Telecommunication*, J. Huijsing, R. Van de Plassche, and W. Sansen, Eds., Kluwer Academic Publishers, Amsterdam, the Netherlands, 1999, pp. 317–335.
23. Y. P. Tsividis, *Operation and Modelling of the MOS Transistor*, McGraw-Hill, New York, 1987.
24. D. Leenaerts, J. van der Tang, and C. Vaucher, *Circuit Design for RF Transceivers*, Kluwer Academic Publishers, Boston, MA, 2001.
25. J. Craninckx and M. Steyaert, A 1.8GHz low phase noise CMOS VCO using optimized hollow spiral inductors, *IEEE J. Solid-State Circuits*, 32(5), 736–744, 1997.
26. N. Itoh, B. D. Muer, and M. Steyaert, Low supply voltage integrated CMOS VCO with three terminals spiral inductor, in *Proceedings of the European Solid State Circuits Conference, ESSCIRC*, Sept. 1999, pp. 194–197.
27. A.-S. Porret, T. Melly, C. C. Enz, and E. A. Vittoz, Design of high-Q varactors for low-power wireless applications using a standard CMOS process, *IEEE J. Solid-State Circuits*, 35(3), 337–345, 2000.
28. O. Kenneth, Estimation methods for quality factors of inductors fabricated in silicon integrated circuit process technologies, *IEEE J. Solid-State Circuits*, 33(8), 1249–1252, 1998.

29. C. S. Meyer, D. K. Lynn, and D. J. Hamilton, *Analysis and Design of Integrated Circuits*. McGraw-Hill, New York, 1968.
30. J. Crols, P. Kinget, J. Craninckx, and M. Steyaert, An analytical model of planar inductors on lowly doped silicon substrates for high frequency analog design to 3 GHz, in *Digest of Technical Papers, Symposium on VLSI Circuits*, 1996.
31. P. Leroux and M. Steyaert, *LNA-ESD Co-Design for Fully Integrated CMOS Wireless Receivers*, The International Series in Engineering and Computer Science, vol. 843, Springer, The Netherlands, 2005.
32. D. Rabaey and J. Sevenhans, The challenges for analog circuit design in mobile radio VLSI chips, in *Proceedings of the AACD Workshop* (*Leuven*), Mar. 1993, pp. 225–236.
33. D. Shen, H. Chien-Meen, B. Lusignan, and B. Wooley, A 900 MHz integrated discrete-time filtering RF front-end, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 1996, pp. 54–55, 417.
34. S. Sheng, L. Lynn, J. Peroulas, K. Stone, I. O'Donnell, and R. Brodersen, A low-power CMOS chipset for spread spectrum communications, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 1996, pp. 346–347, 471.
35. J. Craninckx and M. Steyaert, A fully integrated CMOS DCS-1800 frequency synthesizer, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 1998, pp. 372–373.
36. B. Zhang, P. Allen, and J. Huard, A fast switching PLL frequency synthesizer with an on-chip passive discrete-time loop filter in 0.25 μm CMOS, *IEEE J. Solid-State Circuits*, 38(6), 855–865, 2003.
37. D. Leenaerts1, R. van de Beek1, G. van der Weide, H. Waite, J. Bergervoet, K. Harish, Y. Zhang, C. Razzell, and R. Roovers, SiGe BiCMOS 1ns fast hopping frequency synthesizer for UWB radio, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 2005.
38. J. C. and M. Steyaert, Low-noise voltage controlled oscillators using enhanced LC-tanks, *IEEE Trans. Circuits Syst. II*, 42(12), 794–804, 1995.
39. B. Razavi, Analysis, modeling and simulation of phase noise in monolythic voltage controlled oscillators, in *Proceedings of the Custom Integrated Circuits Conference, CICC*, May 1995, pp. 323–326.
40. J. Craninckx and M. Steyaert, A 1.8GHz low phase noise voltage controlled oscillator with prescaler, *IEEE J. Solid-State Circuits*, 30(12), 1474–1482, 1995.
41. B. de Muer and M. Steyaert, *CMOS Fractional-N Synthesizers: Design for High Spectral Purity and Monolithic Integration*, Springer, Norwell, MA, 2003.
42. B. Gilbert, A precise four-quadrant multiplier with sub-nanosecond response, *IEEE J. Solid-State Circuits*, 3(4), 365–373, 1968.
43. A. Rofougaran, J. Y.-C. Chang, M. Rofougaran, S. Khorram, and A. A. Abidi, A 1 GHz CMOS RF front-end IC with wide dynamic range, in *Proceedings of the European Solid State Circuits Conference, ESSCIRC*, Sep. 1995, pp. 250–253.
44. M. Borremans and M. Steyaert, A 2V low distortion 1 GHz CMOS up-converter mixer, *IEEE J. Solid-State Circuits*, 33(3), 359–366, 1998.
45. M. Borremans, M. Steyaert, and T. Yoshitomi, A 1.5V wide band 3GHz CMOS quadrature direct up-converter for multi-mode wireless communications, in *Proceedings of the Custom Integrated Circuits Conference, CICC*, May 1998, pp. 79–82.
46. M. Pelgrom, A. Duinmaijer, and A. Welbers, Matching properties of MOS transistor, *IEEE J. Solid-State Circuits*, 24(5), 1433–1439, 1989.
47. D. Su and W. McFarland, A 2.5V, 1-W monolithic CMOS RF power amplifier, in *Proceedings of the Custom Integrated Circuits Conference, CICC*, May 1997, pp. 189–192.
48. K. C. Tsai and P. R. Gray, 1.9-GHz 1-W CMOS RF power amplifier for wireless communication, *IEEE J. Solid-State Circuits*, 34(7), 962–970, 1999.
49. C. Yoo and Q. Huang, A common-gate switched 0.9-W class-E power amplifier with 41% PAE in 0.25-μm CMOS, *IEEE J. Solid-State Circuits*, 36(5), 823–830, 2001.
50. T. Sowlati and D. Leenaerts, A 2.4GHz 0.18 μm CMOS self-biased cascode power amplifier with 23dBm output power, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 2002, pp. 294–295.

51. C. Fallesen and P. Asbeck, A 1W 0.35 μm CMOS power amplifier for GSM-1800 with 45% PAE, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 2001, pp. 158–159.
52. A. Shirvani, D. K. Su, and B. A. Wooley, A CMOS RF power amplifier with parallel amplification for efficient power control, in *ISSCC Digest of Technical Papers*, San Francisco, Feb. 2001.
53. V. R. Vathulya, T. Sowlati, and D. Leenaerts, Class 1 bluetooth power amplifier with 24 dBm output power and 48% PAE at 2.4GHz in 0.25 μm CMOS, in *Proceedings of the European Solid State Circuits Conference, ESSCIRC*, Villach, Austria, 2001.
54. I. Aoki, S. D. Kee, D. B. Rutledge, and A. Hajimiri, Fully integrated CMOS power amplifier design using the distributed active transformer architecture, *IEEE J. Solid-State Circuits*, 37(3), 1–13, 2002.
55. N. O. Sokal and A. D. Sokal, Class E—A new class of high-efficiency tuned single-ended switching power amplifiers, *IEEE Journal of Solid-State Circuits*, 10(3), June 1975, pp. 168–176.

5

PLL Circuits

5.1 Introduction **5**-1
What Is and Why Phase-Locked? • Basic Operation Concepts of PLLs • Classification of PLL Types
5.2 PLL Techniques **5**-2
Basic Topology • Loop Orders of the PLL • Tracking Process • Lock-In Process • Acquisition Process • Aided Acquisition • PLL Noise Performance
5.3 Building Blocks of PLL Circuit **5**-10
Voltage-Controlled Oscillators • Phase and Frequency Detectors • Loop Filters • Charge-Pump PLL • PLL Design Considerations
5.4 PLL Applications **5**-27
Clock and Data Recovery • Delay-Locked Loop • Frequency Synthesizer
Bibliography **5**-34

Muh-Tian Shiue
National Central University

Chorng-Kuang Wang
National Taiwan University

5.1 Introduction

5.1.1 What Is and Why Phase-Locked?

Phase-locked loop (PLL) is a circuit architecture that causes a particular system to track with another one. More precisely, PLL synchronizes a signal (usually a local oscillator output) with a reference or an input signal in frequency as well as in phase.

Phase locking is a useful technique that can provide effective synchronization solutions in many data transmission systems such as optical communications, telecommunications, disk drive systems, and local networks, in which data are transmitted in baseband or passband. In general, only data signals are transmitted in most of these applications, namely, clock signals are not transmitted in order to save hardware cost. Therefore, the receiver should have some mechanisms to extract the clock information from the received data stream in order to recover the transmitted data. The scheme is called a timing recovery or clock recovery.

The cost of electronic interfaces in communication systems raises as the data rate gets higher. Hence, high-speed circuits are the critical issue of the high data rate systems implementation, and the advanced very large scale integration (VLSI) technology plays an important role in cost reduction for the high-speed communication systems.

5.1.2 Basic Operation Concepts of PLLs

Typically, as shown in Figure 5.1, a PLL consists of three basic functional blocks: a phase detector (PD), a loop filter (LF), and a voltage-controlled oscillator (VCO). PD detects the phase difference between the VCO output and the input signal, and generates a signal proportional to the phase error. The PD output

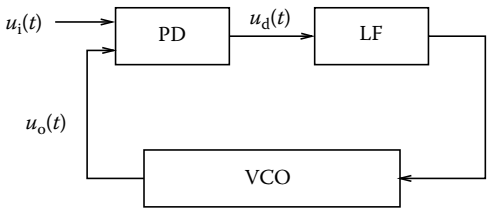


FIGURE 5.1 Basic block diagram of the PLL.

contains a direct current (DC) component and an alternative current (AC) component, the former is accumulated and the latter is filtered out by the LF. The LF output that is near a DC signal is applied to the VCO. This almost DC control voltage changes the VCO frequency toward a direction to reduce the phase error between the input signal and the VCO. Depending on the type of LF used, the steady-state phase error will be reduced to zero or to a finite value.

PLL has an important feature, which is the ability to suppress both the noises superimposed on the input signal and generated by the VCO. In general, the more narrow bandwidth the PLL has, the more effectively the filtering of the superimposed noises can be achieved. Although a narrow bandwidth is better for rejecting large amounts of the input noise, it also prolongs the settling time in the acquisition process. Then, the error of the VCO frequency cannot be reduced rapidly. So there is a trade-off between jitter filtering and fast acquisition.

5.1.3 Classification of PLL Types

Different PLL types have been built from different classes of building blocks. The first PLL integrated circuit (IC) appeared around 1965 and consisted of purely analog devices. In the so-called linear PLLs (LPLLs), an analog multiplier (four-quadrant) is used as the PD, the LF is built of a passive or an active RC filter, and the VCO is used to generate the output signal of the PLL. In most cases, the input signal to this LPLL is a sine wave, whereas the VCO output signal is a symmetrical square wave.

The classical digital PLL (DPLL) uses a digital PD such as an exclusive OR (XOR) gate, a JK-flipflop, or a phase-frequency detector (PFD). The remaining blocks are still the same as LPLL. In many aspects, the DPLL performance is similar to the LPLL.

The function blocks of the all digital PLL (ADPLL) are implemented by purely digital circuits, and the signals within the loop are digital, too. Digital versions of the PD are the same as DPLL. The digital LF is built of an ordinary up/down counter, N-before-M counter or K-counter [1]. The digital counterpart of the VCO is the digital-controlled oscillator [2,3].

In analogy to filter designs, PLLs can be implemented by software such as a microcontroller, microcomputer, or digital signal processing (DSP); this type of PLL is called software PLL (SPLL).

5.2 PLL Techniques

5.2.1 Basic Topology

A PLL is a feedback system that operates and minimizes the phase difference between two signals. Referring to the basic function block diagram of a PLL as shown in Figure 5.1, it typically consists of a PD, an LF, and a VCO. The PD works as a phase error detector and an amplifier. It compares the phase of the VCO output signal $u_o(t)$ with the phase of the reference signal $u_i(t)$ and develops an output signal $u_d(t)$ that is proportional to the phase error θ_e. Within a limited range, the output signal can be expressed as

$$u_d(t) = k_d \theta_e \tag{5.1}$$

where k_d with the unit of V/rad represents the gain of the PD.

The output signal $u_d(t)$ of the PD consists of a DC component and a superimposed AC component. The latter is undesired and removed by the LF. In general, the LF is a low-pass filter (LPF) to generate an almost DC control voltage for the VCO to oscillate at the frequency equal to the input frequency.

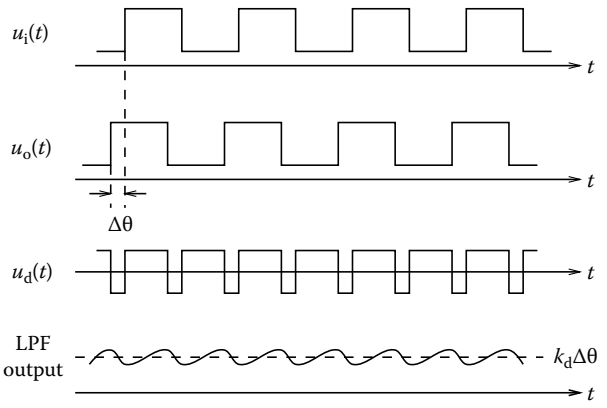


FIGURE 5.2 Waveforms in a PLL.

How the building blocks of a basic PLL work together will be explained below. At first, assume both the waveforms of input signal and VCO output are rectangular. Furthermore, it is assumed that the angular frequency ω_i of the input signal $u_i(t)$ is equivalent to the central frequency ω_o of the VCO signal $u_o(t)$. Now a small positive frequency step is applied to $u_i(t)$ at $t = t_0$ (shown in Figure 5.2). $u_i(t)$ accumulates the phase increments faster than $u_o(t)$ of VCO does. If the PD can response wider pulses increasingly, a higher DC voltage is accordingly generated at the LF output to increase the VCO frequency. Depending on the type of the LF that will be discussed later, the final phase error will be reduced to zero or a finite value.

It is important to note from the descriptions above that the loop locks only after the two conditions are satisfied: (1) ω_i and ω_o are equal and (2) the phase difference between the input $u_i(t)$ and the VCO output $u_o(t)$ settles to a steady-state value. If the phase error varies with time so fast that the loop is unlocked, the loop must keep on the transient process, which involves both "frequency acquisition" and "phase acquisition."

To design a practical PLL system, it is required to know the status of the responses of the loop if (1) the input frequency is varied slowly (tracking process), (2) the input frequency is varied abruptly (lock-in process), and (3) the input and the output frequencies are not equal initially (acquisition process). Using LPLL as an example, these responses will be shown in Sections 5.2.3 through 5.2.5.

5.2.2 Loop Orders of the PLL

Figure 5.3 shows the linear model of a PLL. According to the control theory, the closed-loop transfer function of PLL can be derived as

$$H(s) \triangleq \frac{\theta_o(s)}{\theta_i(s)} = \frac{k_d k_o F(s)}{s + k_d k_o F(s)} \tag{5.2}$$

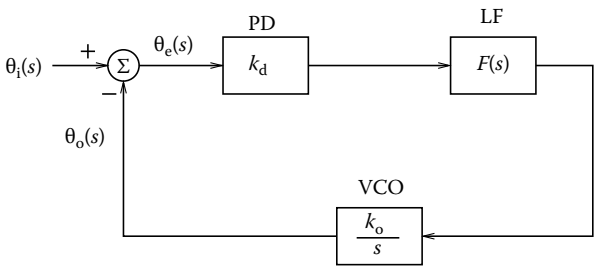


FIGURE 5.3 Linear model of PLL.

where

k_d with units V/rad is called the PD gain

k_o is the VCO gain factor and has units rad/(s-V)

In addition to the phase transfer function, a phase-error transfer function $H_e(s)$ is derived as follows:

$$H_e(s) \triangleq \frac{\theta_e(s)}{\theta_i(s)} = \frac{s}{s + k_d k_o F(s)} \tag{5.3}$$

The loop order of the PLL depends on the characteristics of the LF. Therefore the LF is a key component that affects the PLL dynamic behavior. A PLL with an LF consisted of simple amplifier or attenuator is called a first-order PLL. As shown in Figure 5.3, set $F(s) = 1$ and the closed-loop transfer function can be derived as

$$H(s) = \frac{k}{s + k} \tag{5.4}$$

where the DC loop gain $k = k_d k_o$. If fast tracking is required, a high DC loop gain k is necessary for the bandwidth of the PLL being wide enough because the DC loop gain k is the only parameter available. Such a design is not suitable for noise suppression. Therefore, fast tracking and narrow bandwidth are incompatible in a first-order loop.

A commonly used LF is the passive lag filter. The transfer function is

$$F(s) = \frac{1}{1 + s\tau} \tag{5.5}$$

The closed-loop transfer function can be derived as

$$H(s) = \frac{k_d k_o/\tau}{s^2 + (1/\tau)s + k_d k_o/\tau} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.6}$$

where

$\omega_n = \sqrt{\frac{k_d k_o}{\tau}}$ is the "natural frequency"

$\zeta = \frac{1}{2}\sqrt{\frac{1}{\tau k_d k_o}}\,\omega_n$ is the "damping factor"

These two parameters are important to characterize a PLL. Now, a second-order PLL is obtained and there are two parameters (τ and $k = k_o k_d$) available to achieve fast tracking as well as the noise suppression. Then three loop parameters (ω_n, ζ, k) must be determined. In addition, the phase-error transfer function $H_e(s)$ can be further derived as follows:

$$H_e(s) = \frac{s(s + 1/\tau)}{s^2 + (1/\tau)s + k_d k_o/\tau} \tag{5.7}$$

5.2.3 Tracking Process

The linear model of a PLL shown in Figure 5.3 is suitable for analyzing the tracking performance of a PLL that is almost in lock, that is, only with a small phase error. If the phase error changes too abruptly, the PLL fails to lock, and a large phase error is induced even though the change happens only momentarily. The unlock condition is a nonlinear process that cannot be analyzed via the linear model. The acquisition process will be described in Section 5.2.5.

At first, consider that a step phase error expressed as $\theta_i(t) = \Delta\theta u(t)$ is applied to the input. The Laplace transform of the input is $\theta_i(s) = \frac{\Delta\theta}{s}$ that is substituted into Equation 5.7 to get

$$\theta_e(s) = \frac{\Delta\theta}{s} \frac{s(s + 1/\tau)}{s^2 + (1/\tau)s + k_d k_o/\tau} \tag{5.8}$$

According to the final value theorem of the Laplace transform,

$$\lim_{t\to\infty} \theta_e(t) = \lim_{s\to 0} s\theta_e(s) = \frac{\Delta\theta}{k_d k_o}$$

In another word, the loop will eventually track on the step phase change with a steady-state phase error proportional to the DC loop gain. If it is necessary to have a high DC loop gain in order to reduce the steady-state phase error and a very narrow bandwidth for improving the noise suppression, the loop will be severely underdamped and the transient response will be poor.

When a step change of frequency $\Delta\omega$ is applied to the input, the input phase change is a ramp, that is, $\theta_i(t) = \Delta\omega t$, therefore $\theta_i(s) = \frac{\Delta\omega}{s^2}$. Substituting $\theta_i(s)$ in Equation 5.7 and applying the final value theorem, then

$$\begin{aligned} \theta_v &= \lim_{t\to\infty} \theta_e(t) = \lim_{s\to 0} s\theta_e(s) \\ &= \lim_{s\to 0} \frac{\Delta\omega}{s^2} \frac{s(s + 1/\tau)}{s^2 + (1/\tau)s + (k_d k_o/\tau)} \end{aligned} \tag{5.9}$$

where θ_v is called the "velocity error" or "static phase error" [4]. In practice, the input frequency almost never agrees exactly with the VCO free-running frequency, that is, usually there is a frequency difference $\Delta\omega$ between the two. From Equation 5.9, the velocity error will be infinite while there is a frequency difference $\Delta\omega$. Another commonly used LF is the active lead-lag LF with the transfer function $F(s)$ described as follows:

$$F(s) = k_a \frac{1 + s\tau_2}{1 + s\tau_1} \tag{5.10}$$

where τ_1, τ_2, and k_a are the two time constants and DC gain of an active lead-lag LF, respectively. Substituting $\theta_i(s)$ in Equation 5.3 and applying the final value theorem, then

$$\begin{aligned} \theta_v &= \lim_{s\to 0} s\theta_e(s) = \lim_{s\to 0} \frac{\Delta\omega}{s + k_d k_o F(s)} \\ &= \frac{\Delta\omega}{k_d k_o F(0)} = \frac{\Delta\omega}{k_v} \end{aligned} \tag{5.11}$$

From Equation 5.11, if the PLL has a high DC loop gain, that is, $k_d k_o F(0) \gg \Delta\omega$, the steady-state phase error corresponding to a step frequency error input approaches to zero. This is the reason that a high gain loop has a good tracking performance. Now the advantage of a second-order loop using an active LF with high DC gain is evident. The active lead-lag LF with a high DC gain will make the steady-state phase error approach to zero and the noise bandwidth be narrow simultaneously, which is impossible in a first-order loop.

If the input frequency is changed linearly with time at a rate of $\Delta\omega$, that is $\theta_i(t) = \frac{1}{2}\Delta\omega\ t^2$, $\theta_i(s) = \frac{\Delta\omega}{s^3}$. Using an active LF with high DC gain and applying the final value theorem of Laplace transform, it is derived that

$$\theta_a = \lim_{t\to\infty} \theta_e(t) = \lim_{s\to 0} s\theta_e(s) = \frac{\Delta\omega}{\omega_n^2} \tag{5.12}$$

where θ_a is called an "acceleration error" (sometimes calls "dynamic tracking error" or "dynamic lag") [4].

In some applications, PLL needs to track an accelerating phase error without static tracking error. When frequency ramp is applied, the static phase error will be

$$\theta_e(s) = \lim_{s \to 0} \frac{\Delta\omega}{s(s + k_d k_o F(s))} \tag{5.13}$$

In order to have θ_e zero, it is necessary to make $F(s)$ be a form of $\frac{G(s)}{s^2}$, where $G(0) \neq 0$. $\frac{G(s)}{s^2}$ implies that the LF has two cascade integrators. This results in a third-order loop. In order to eliminate the static acceleration error, a third-order loop is very useful for some special applications such as satellite and missile systems.

Based on Equation 5.12, a large natural frequency ω_n is used to reduce the static tracking phase error in a second-order loop; however, a wide natural frequency has an undesired noise filtering performance. In the contrast, the zero tracking phase error for a frequency ramp error is concordant with a small loop bandwidth in a third-order loop. In practice, there are three basic types of LF: passive lead-lag filter, active lead-lag filter, and active proportional and integral (PI) filter. The characteristics of the three types of LF and their effects on the PLL will be described in Section 5.3.3. Besides, a high-order filter is used for critical applications because it provides better noise filtering, initial acquisition, and fast tracking. However it is difficult to design a high-order loop due to some problems such as loop stability.

All the preceding analysis on the tracking process is under the assumption that the phase error is relatively small and the loop is linear. If the phase error is large enough to make the loop drop out of lock, the linear assumption is invalid. For a sinusoidal-characteristic PD, the exact phase expression of Equation 5.11 should be

$$\sin\theta_v = \frac{\Delta\omega}{k_v} \tag{5.14}$$

The sine function has solutions only when $\Delta\omega \leq k_v$. However, there is no solution if $\Delta\omega > k_v$. This is the case the loop loses lock and the output of the PD will be beat notes signal rather than a DC control voltage. Therefore, k_v can be used to define the ''hold range'' of the PLL, that is

$$\Delta\omega_H = \pm k_v = k_o k_d F(0) \tag{5.15}$$

The hold range is the frequency range in which a PLL is able to maintain lock ''statically.'' Namely, if input frequency offset exceeds the hold range statically, the steady-state phase error would drop out of the linear range of the PD and the loop loses lock. k_v is the function of k_o, k_d, and $F(0)$. The DC gain $F(0)$ of the LF depends on the filter type. Therefore, it is important to make an LF have a high DC gain for extending the hold range. Referring to the characteristics of the three basic types of LF described in Section 5.3.3, the hold range $\Delta\omega_H$ can be $k_o k_d$, $k_o k_d k_a$, and ∞ for passive lead-lag filter, active lead-lag filter, and active PI filter, respectively. The hold range expressed in Equation 5.15 is not correct when some other components in PLL are saturated earlier than the PD. When the PI filter is used, the real hold range is actually determined by the control range of the VCO.

Considering the dynamic phase error θ_a in a second-order loop, the exact expression for a sinusoidal characteristic PD is

$$\sin\theta_a = \frac{\Delta\omega}{\omega_n^2} \tag{5.16}$$

which implies that the maximum change rate of the input frequency is ω_n^2. If the rate exceeds ω_n^2, the loop will fall out of lock.

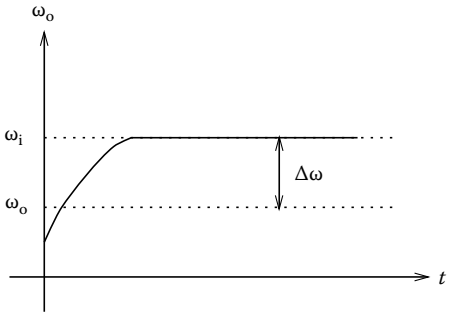


FIGURE 5.4 Lock-in process of the PLL.

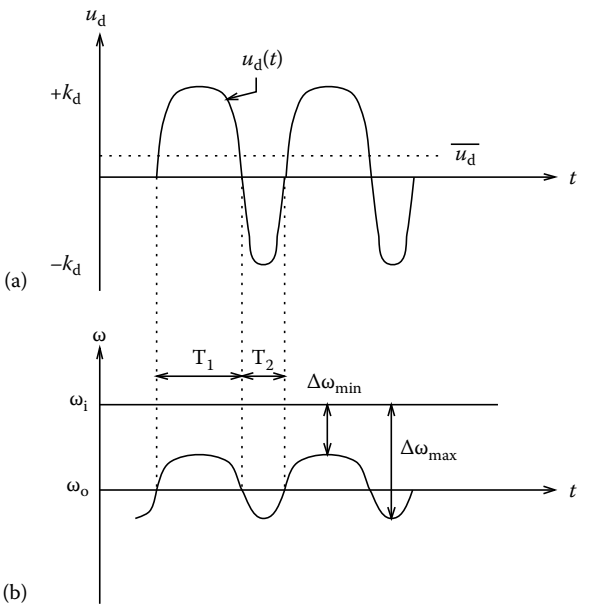


FIGURE 5.5 Pull-in process of the PLL.

5.2.4 Lock-In Process

The "lock-in" process is defined as PLL locks within one single beat note between the input and the output (VCO output) frequency. The maximum frequency difference between the input and the output that PLL can lock within one single beat note is called the "lock-in range" of the PLL.

Figure 5.4 shows a case of PLL lock-in process that a frequency offset $\Delta\omega$ is less than the lock-in range, and the lock-in process happens. Then PLL will lock within one single beat note between ω_i and ω_o. In Figure 5.5b, the frequency offset $\Delta\omega$ between input (ω_i) and output (ω_o) is larger than the lock-in range, hence the lock-in process will not take place, at least not instantaneously.

Suppose the PLL is unlocked initially. The input frequency ω_i is $\omega_o + \Delta\omega$. If the input signal $v_i(t)$ is a sine wave and given by

$$v_i(t) = A_i \sin(\omega_o t + \Delta\omega t) \tag{5.17}$$

And the VCO output signal $v_o(t)$ is usually a square wave written as a Walsh function [5]

$$v_o(t) = A_o W(\omega_o t) \tag{5.18}$$

$v_o(t)$ can be replaced by the Fourier series,

$$v_o(t) = A_o \left[\frac{4}{\pi}\cos(\omega_o t) + \frac{4}{3\pi}\cos(3\omega_o t) + \cdots\right] \tag{5.19}$$

So the PD output v_d is

$$\begin{aligned} v_d(t) = v_i(t)v_o(t) &= A_i A_o \left[\frac{2}{\pi}\sin(\Delta\omega t) + \cdots\right] \\ &= k_d \sin(\Delta\omega t) + \text{high} - \text{frequency terms} \end{aligned} \tag{5.20}$$

The high frequency components can be filtered out by the LF. The output of the LF is given by

$$v_f(t) \approx k_d |F(\Delta\omega)| \sin(\Delta\omega t) \tag{5.21}$$

The peak frequency deviation based on Equation 5.21 is equal to $k_d k_o |F(\Delta\omega)|$. If the peak deviation is larger than the frequency error between ω_i and ω_o, the lock-in process will take place. Hence the lock-in range is consequently given by

$$\Delta\omega_L = k_d k_o |F(\Delta\omega_L)| \tag{5.22}$$

The lock-in range is always larger than the corner frequency $\frac{1}{\tau_1}$ and $\frac{1}{\tau_2}$ of the LF in practical cases. An approximation of the LF gain $F(\Delta\omega_L)$ is shown as follows:

For the passive lead-lag filter

$$F(\Delta\omega_L) \approx \frac{\tau_2}{\tau_1 + \tau_2}$$

For the active lead-lag filter

$$F(\Delta\omega_L) \approx k_a \frac{\tau_2}{\tau_1}$$

For the active PI filter

$$F(\Delta\omega_L) \approx \frac{\tau_2}{\tau_1}$$

τ_2 is usually much smaller than τ_1, the $F(\Delta\omega_L)$ can be further approximated as follows:

For the passive lead-lag filter

$$F(\Delta\omega_L) \approx \frac{\tau_2}{\tau_1 + \tau_2}$$

For the active lead-lag filter

$$F(\Delta\omega_L) \approx k_a \frac{\tau_2}{\tau_1}$$

For the active PI filter

$$F(\Delta\omega_L) \approx \frac{\tau_2}{\tau_1}$$

Substituting above equations in Equation 5.22 and assuming a high gain loop,

$$\Delta\omega_L \approx 2\zeta\omega_n \tag{5.23}$$

can be gotten for all three types of LF shown in Figure 5.12.

5.2.5 Acquisition Process

Suppose that the PLL does not lock initially, the input frequency is $\omega_i = \omega_o + \Delta\omega$, where ω_o is the initial frequency of VCO. If the frequency error $\Delta\omega$ is larger than the lock-in range, the lock-in process will not happen. Consequently the output signal $u_d(t)$ of the PD shown in Figure 5.5a is a sine wave that has the frequency of $\Delta\omega$. The AC PD output signal $u_d(t)$ passes through the LF. Then the output $u_f(t)$ of the LF modulates the VCO frequency. As shown in Figure 5.5b, when ω_o increases, the frequency difference between ω_i and ω_o becomes smaller and vice versa. Therefore, the PD output $u_d(t)$ becomes asymmetric when the duration of positive half-periods of the PD output is larger than the negative ones. The average value $\overline{u_d(t)}$ of the PD output therefore goes positive slightly. Then the frequency of VCO will be pulled up until it reaches the input frequency. This phenomenon is called a ''pull-in process.''

Because the pull-in process is a nonlinear behavior, the mathematical analysis is quite complicated. According to the results of [1], the pull-in range and the pull-in time depend on the type of LF. For an active lead-lag filter with a high gain loop, the pull-in range is

$$\Delta\omega_P \approx \frac{4\sqrt{2}}{\pi}\sqrt{\zeta\omega_n k_o k_d} \tag{5.24}$$

and the pull-in time is

$$T_P \approx \frac{\pi^2}{16}\frac{\Delta\omega_0^2 k_a}{\zeta\omega_n^3} \tag{5.25}$$

where $\Delta\omega_0$ is the initial frequency error. Equations 5.24 and 5.25 should be modified for different types of PDs [1].

5.2.6 Aided Acquisition

The PLL bandwidth is always too narrow to lock a signal with large frequency error. Furthermore, the frequency acquisition is slow and impractical. Therefore, there are aided frequency-acquisition techniques to solve this problem such as the frequency locked-loop (FLL) and the bandwidth-widening methods.

The FLL, which is very much similar to a PLL, is composed of a frequency discriminator, an LF, and a VCO. PLL is a coherent mechanism to recover a signal buried in noise. An FLL, in contrast, is a noncoherent scheme that cannot distinguish the phase error between input signal and VCO signal. Therefore an FLL can only be useful to provide the signal frequency which exactly synchronizes with the reference frequency source.

The major difference between PLL and FLL is the PD and the frequency discriminator. The frequency discriminator is the frequency detector in the FLL. It generates a voltage proportional to the frequency difference between the input and the VCO. The frequency difference will be driven to zero in a negative feedback fashion. If a linear frequency detector is employed, it can be shown that the frequency-acquisition time is proportional to the logarithm of the frequency error [6]. In the literature, some frequency detectors-like quadricorrelator [7], balance quadricorrelator [8], rotational frequency detector [9], and frequency delimiter [10] are disclosed.

5.2.7 PLL Noise Performance

In high-speed data recovery applications, a better performance of the VCO and the overall PLL itself is desired. In a consequence, the random variations of the sampling clock, so-called jitter, is the critical performance parameter.

Jitter sources of PLL in the case of using a ring VCO mainly come from the input and the VCO itself. The ring oscillator jitter is associated with the power supply noise, the substrate noise, $1/f$ noise, and the thermal noise. The former two noise sources can be reduced by fully differential circuit structure. $1/f$ noise, on the other hand, can be rejected by the tracking capability of the PLL. Therefore, the thermal noise is the worst noise source. From the analysis of [18], the one stage RMS timing jitter error of the ring oscillator normalized to the time delay per stage can be shown as

$$\frac{\Delta\tau_{rms}}{t_d} \approx \sqrt{\frac{2KT}{C_L}}\left(\sqrt{1+\frac{2}{3}a_v}\right)\frac{1}{V_{pp}} \tag{5.26}$$

where

C_L is the load capacitance

$\sqrt{1+\frac{2}{3}a_v}$ is called the noise contribution factor ζ

a_v is the small-signal gain of the delay cell

V_{pp} is the VCO output swing

From Equation 5.26, for a fixed output bandwidth, higher gain contributes larger noise.

Because the ring oscillator is a feedback architecture, the noise contribution of a single delay cell may be amplified and filtered by the following stage. To consider two successive stages, Equation 5.26 can be rearranged as [18]

$$\frac{\Delta\tau_{rms}}{t_d} \approx \sqrt{\frac{2KT}{C_L}}\frac{1}{(V_{gs}-V_t)}\zeta \tag{5.27}$$

Therefore, the cycle-to-cycle jitter of the ring oscillator in a PLL can be predicted by [18]

$$\overline{(\Delta\tau_N)^2} = \frac{KT}{I_{ss}}\frac{a_v\zeta^2}{(V_{gs}-V_t)}T_o \tag{5.28}$$

where

I_{ss} is the rail current of the delay cell

T_o is the output period of the VCO

Based on Equation 5.28, designing a low jitter VCO ($V_{gs}-V_t$) should be as large as possible. For fixed delay and fixed current, a lower gain of each stage is better for jitter performance, but the loop gain must satisfy the Barkhausen criterion. From the viewpoint of VCO jitter, a wide bandwidth of PLL can correct the timing error of the VCO rapidly [14]. If the bandwidth is too wide, the input noise jitter may be so large that dominates the jitter performance of the PLL. Actually this is a trade-off.

For a PLL design, the natural frequency and the damping factor are the key parameters to be determined by designers. If the input signal-to-noise ratio $(SNR)_i$ is defined, then the output signal-to-noise ratio $(SNR)_o$ can be obtained [4]

$$(SNR)_o = (SNR)_i\frac{B_i}{2B_L} \tag{5.29}$$

where

B_i is the bandwidth of the prefilter

B_L is the noise bandwidth

Hence the B_L can be derived using Equation 5.29. And the relationship of B_L with ω_n and ζ is

$$B_L = \frac{\omega_n}{2}\left(\zeta+\frac{1}{4\zeta}\right) \tag{5.30}$$

Therefore the ω_n and ζ can be designed to satisfy the $(SNR)_o$ requirement.

Besides the system and the circuit designs, jitter can be reduced in the board level design. Board jitter can be alleviated by better layout and noise decoupling schemes such as appending proper decouple and bypass capacitances.

5.3 Building Blocks of PLL Circuit

5.3.1 Voltage-Controlled Oscillators

The function of a VCO is to generate a stable and periodic waveform whose frequency can be varied by an applied control voltage. The relationship between the control voltage and the oscillation frequency

depends upon the circuit architecture. A linear characteristic is generally preferred because of its wider applications. As a general classification, VCO can be categorized roughly into two types by the output waveforms: (1) harmonic oscillators that generate nearly sinusoidal outputs and (2) relaxation oscillators that provide square or triangle outputs.

In general, a harmonic oscillator is composed of an amplifier that provides an adequate gain and a frequency-selective network that feeds a certain output frequency range back to the input. LC tank oscillators and crystal oscillators belong to this type. Generally, the harmonic oscillators have the following advantages: (1) superior frequency stability while the conditions of temperature, power supply, and noise are included; and (2) good frequency accuracy control due to that the oscillation frequency is determined by a tank circuit or a crystal.

Essentially, harmonic oscillators are not compatible with monolithic IC technology and their frequency tuning range is limited. On the contrary, relaxation oscillators are easy to be implemented in monolithic ICs. Since frequency is normally proportional to a controlled-current or -voltage and inversely proportional to timing capacitors, the frequency of oscillation can be varied linearly over a very wide range. On the other hand, the ease of frequency tuning brings in drawbacks, such as poor frequency stability and frequency inaccuracy.

Relaxation oscillators are the most commonly used oscillator configuration in monolithic IC design because they can operate in a wide frequency range with a minimum number of external components. According to the mechanism of the oscillator topology employed, relaxation oscillators can be further categorized into three types: (1) grounded capacitor VCO [20], (2) emitter-coupled VCO, and (3) delay-based ring VCO [21]. The operation of the first two oscillators are similar in the sense that the time duration spent in each state is determined by the timing components and the charge/discharge currents. The ring oscillator is one of the relaxation oscillators and has received considerable attentions recently in high frequency PLL applications for clock synchronization and timing recovery. Because the ring oscillator can provide high frequency oscillation with simple digital-like circuits that are compatible with digital technology, it is suitable for VLSI implementations.

In order to achieve high rejection of power supply and substrate noises, both the signal path and the control path of a VCO must be fully differential. A common ring oscillator topology in monolithic PLLs is shown in Figure 5.6. The delay-based ring VCO operates quite differently since the timing relies on the delay in each gain stages that are connected in a ring configuration. The loop oscillates with a period equal to $2NT_d$, where T_d is the delay of each stage. The oscillation can be obtained when the total phase shift is zero and the loop gain is greater or equal to unity at a certain frequency. To vary the frequency of oscillation, the effective number of stages or the delay of each stage must be changed. The first approach is called ''delay interpolating'' VCO [21], where a shorter delay path and a longer delay path are used in parallel. The total delay is tuned by increasing the gain of one path and decreasing the other, and the total delay is a weighted sum of the two delay paths. The second approach is to vary the delay time of each stage to adjust the oscillation frequency. The delay of each stage is tuned by varying the capacitance or the resistance seen at the output node of each stage.

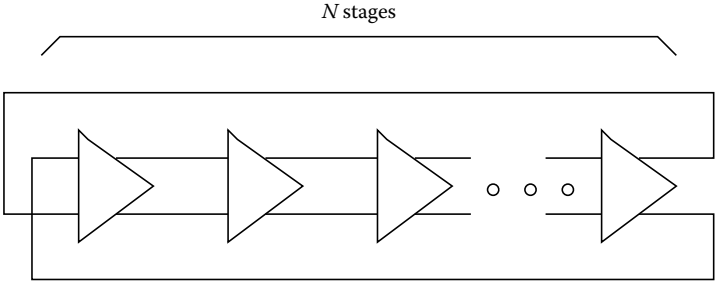


FIGURE 5.6 Ring oscillator.

Because the tuning range of the capacitor is small and the maximum oscillation frequency is limited by the minimum value of the load capacitor, the ''resistive tuning'' is a better alternative technique. Resistive tuning method provides a large, uniform frequency tuning range and leads itself easily to a differential control. In Figure 5.7a, the on-resistance of the triode PMOS loads is adjusted by V_{cont}. The more V_{cont} decreases, the more the delay of the stage drops because the time constant at the output node is decreased. However, the small-signal gain decreases as well when V_{cont} decreases. The circuit eventually fails to oscillate when the loop gain at the oscillation frequency is less than unity. In Figure 5.7b, the delay of gain stage is tuned by adjusting the tail current, but the small-signal gain remains constant. So the circuit is better than Figure 5.7a. As shown in Figure 5.7c [22], the PMOS current source with a pair of cross-coupled diode loads provides a differential load impedance that is independent of common-mode voltage. This makes the cell delay insensitive to common-mode noise. Figure 5.7d is a poor delay cell for a ring oscillator because the tuning range is very small.

The minimum number of stages that can be used while maintaining a reliable operation is an important issue in a ring oscillator design. When the number of stages decreases, the required phase

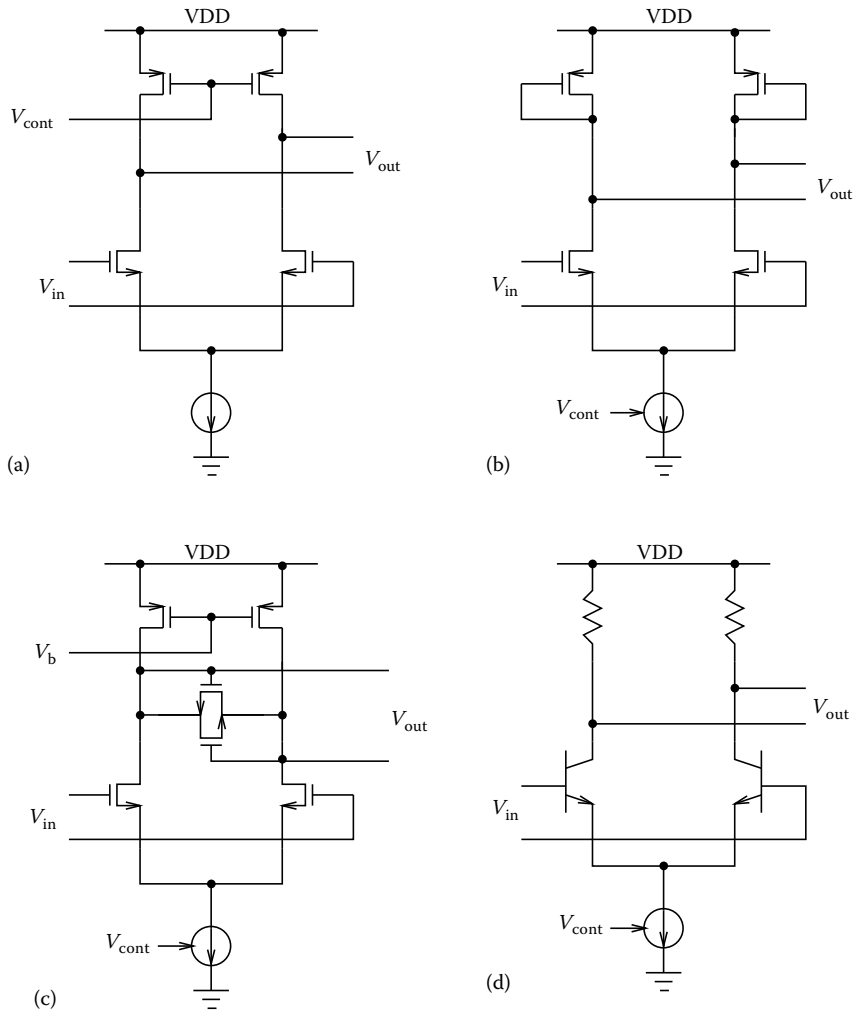


FIGURE 5.7 The gain stages using of resistive tuning.

shift and DC gain per stage increases. Two-stage bipolar ring oscillator can be designed reliably [23], but CMOS implementations are not. Thus, CMOS ring oscillators utilize three or more stages typically.

5.3.2 Phase and Frequency Detectors

The PD type has the influence on the dynamic range of PLLs. Hold range, lock-in range, and pull-in range are analyzed in Section 5.2 based on the multiplier PD. Most of the other types of PD have a greater linear output span and a larger maximum output swing than a sinusoidal characteristic PD. A larger tracking range and a larger lock limit are available if the linear output range of PD increases. The three widely used PDs are XOR PD, edge-triggered JK-flipflop, and PFD. The characteristics of these PDs are plotted in Figure 5.8.

The XOR PD can maintain phase tracking when the phase error θ_e is confined in the range of

$$\frac{-\pi}{2} < \theta_e < \frac{\pi}{2}$$

as shown in Figure 5.8a. The zero phase error takes place when the input signal and the VCO output are quadrature in phase as shown in Figure 5.9a. As the phase difference deviates from $\frac{\pi}{2}$, the output duty cycle is no longer 50%, which provides a DC value proportional to the phase difference as shown in Figure 5.9b. But the XOR PD has a steady-state phase error if the input signal or the VCO output are asymmetric.

The JK-flipflop PD shown in Figure 5.10, also called a two-state PD, is barely influenced by the asymmetric waveform because it is edge-triggered. The zero phase error happens when the input signal and the VCO output are out-off phase as illustrated in Figure 5.10a. As shown in Figure 5.8b, the JK-flipflop PD can maintain phase tracking when the phase error is within the range of

$$-\pi < \theta_e < \pi$$

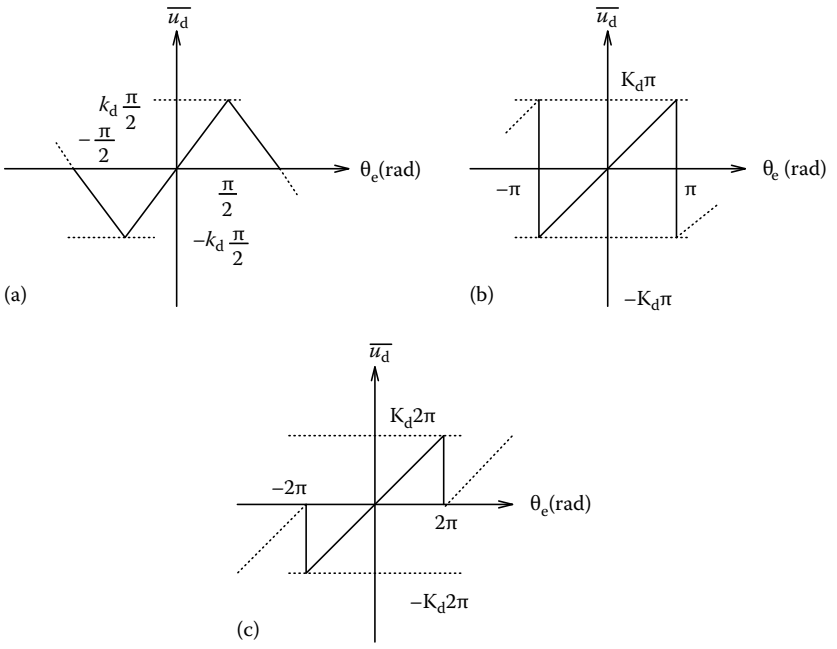


FIGURE 5.8 PD characteristics of (a) XOR, (b) JK-flipflop, and (c) PFD.

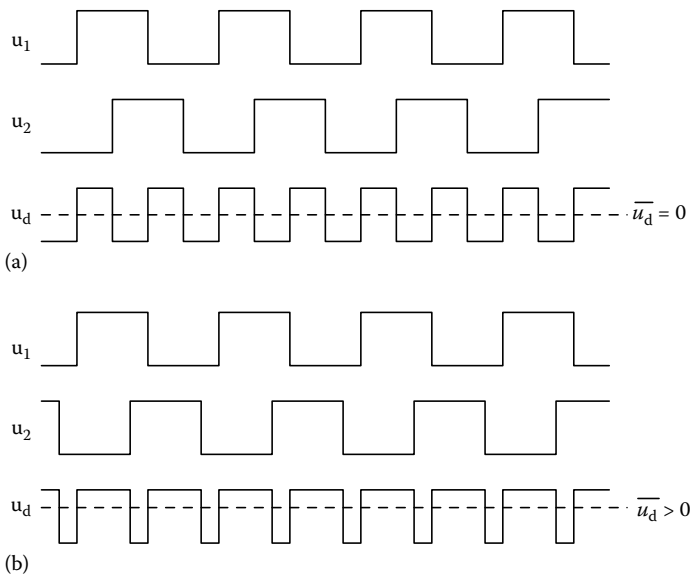


FIGURE 5.9 Waveforms of the signals for the XOR PD: (a) waveforms at zero phase error and (b) waveforms at positive phase error.

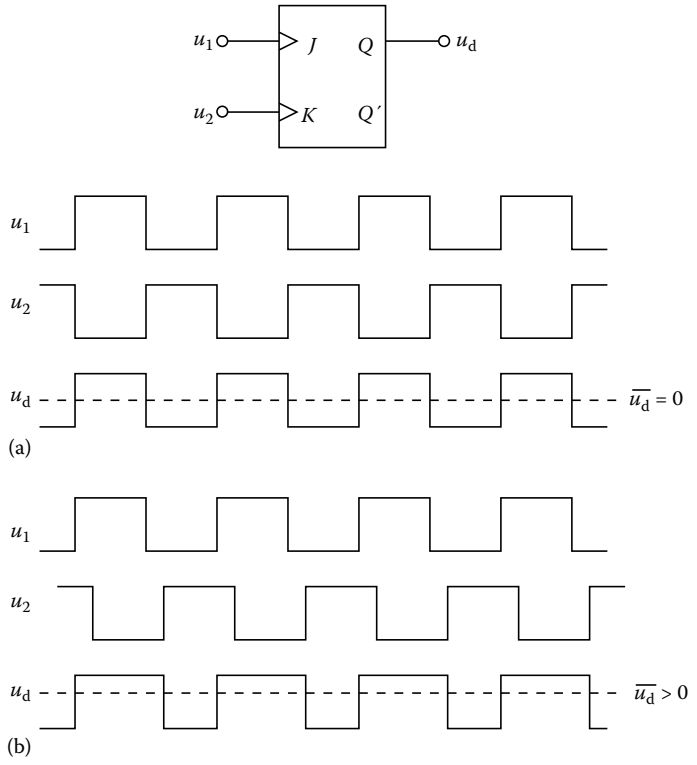


FIGURE 5.10 Waveforms of the signals for the JK-flipflop PD: (a) waveforms at zero phase error and (b) waveforms at positive phase error.

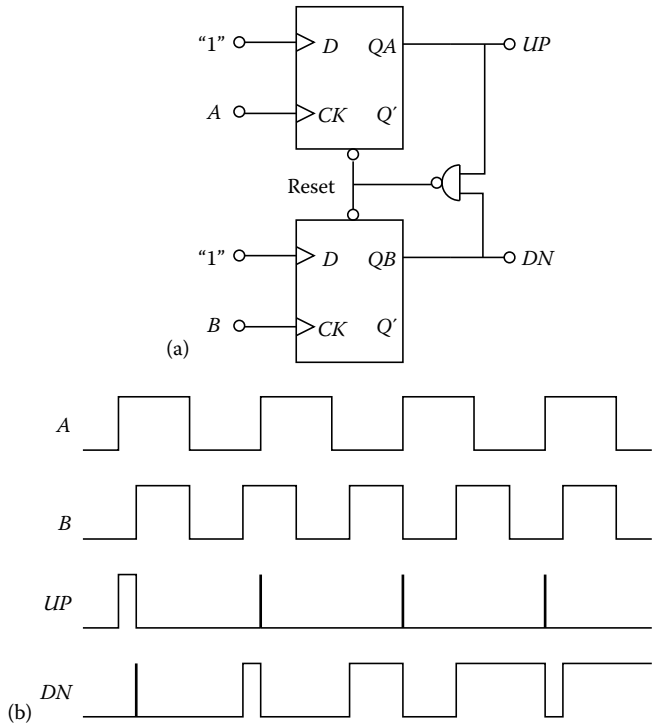


FIGURE 5.11 (a) PFD diagram and (b) inputs and outputs waveforms of PFD.

Here, a positive edge appearing at the J input triggers the flipflop into "high" state ($Q = 1$), and the rising edge of u_2 drives Q to zero. Figure 5.10b shows the output waveforms of the JK-flipflop PD for $\theta_e > 0$.

The PFD output depends not only on the phase error, but also on the frequency error. The characteristic is shown in Figure 5.8c. When the phase error is greater than 2π, the PFD works as a frequency detector. The operation of a typical PFD is as follows and the waveforms is shown in Figure 5.11. If the frequency of input A, ω_A, is less than the frequency of input B, ω_B, then the total width of positive pulses appearing at Q_A is shorter than at Q_B. Conversely, if $\omega_A > \omega_B$, the total width of positive pulses appears at Q_A is longer than at Q_B. If $\omega_A = \omega_B$, then the PFD generates pulses at either Q_A or Q_B with a width equal to the phase difference between the two inputs. The outputs Q_A and Q_B are usually called the "up" and "down" signals, respectively. If the input signal fails, which usually happens at the non-return-to-zero (NRZ) data recovery applications during missing or extra transmissions, the output of the PFD would stick on the high state (or low state). This condition may cause VCO to oscillate fast or slow abruptly, which results in noise jitter or even losing lock. This problem can be remedied by additional control logic circuits to make the PFD output to toggle back and forth between the two logic level with 50% duty cycle [19], the loop is interpreted as zero phase error. The "rotational FD" described by Messerschmitt can also solve this issue [9]. The output of a PFD can be converted to a DC control voltage by driving a three-state charge-pump which will be described in Section 5.3.4.

5.3.3 Loop Filters

For a PLL with given VCO and PD, LF is the critical component to determine the PLL characteristics, such as the damping factor that determines the relative stability, open-loop/closed-loop bandwidths that relate to the convergence speed in the initial state and the tracking capability in the steady state, and so on. Various types of LF will be introduced in this section.

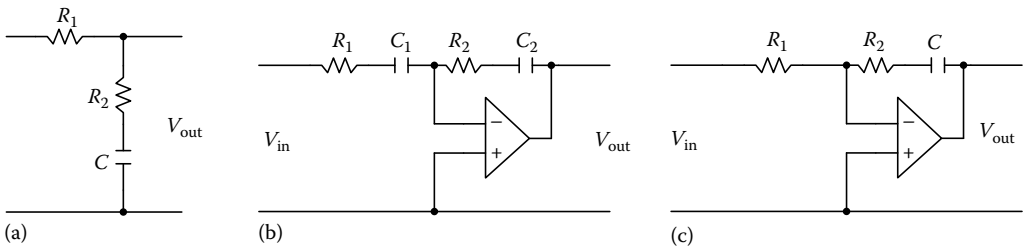


FIGURE 5.12 (a) Passive lead-lag filter, (b) active lead-lag filter, and (c) active PI filter.

5.3.3.1 Continuous-Time LFs

Figure 5.12 shows three types of LF that are widely used. Figure 5.12a is a passive lead-lag filter with transfer function $F(s)$ given by

$$F(s) = \frac{1 + s\tau_2}{1 + s(\tau_1 + \tau_2)} \tag{5.31}$$

where

$\tau_1 = R_1 C$
$\tau_2 = R_2 C$

Figure 5.12b shows an active lead-lag filter, whose transfer function is repeated here for convenience

$$F(s) = k_a \frac{1 + s\tau_2}{1 + s\tau_1} \tag{5.32}$$

where

$\tau_1 = R_1 C_1$
$\tau_2 = R_2 C_2$
$k_a = -\frac{C_1}{C_2}$

A "PI" filter is shown in Figure 5.12c. The transfer function is given by

$$F(s) = \frac{1 + s\tau_2}{s\tau_1} \tag{5.33}$$

where

$\tau_1 = R_1 C$
$\tau_2 = R_2 C$

Their Bode plots are shown in Figure 5.13a through c, respectively. High order filters could be used in some applications, but additional filter poles introduce a phase shift. In general, it is not trivial to maintain the stability of high order systems.

The transfer functions of the LFs shown in Figure 5.12 are substituted for $F(s)$ in Equation 5.2 in order to analyze the phase transfer function. We obtain the phase transfer functions as follows: for the passive lead-lag filter

$$H(s) = \frac{k_d k_o (1 + s\tau_2/\tau_1 + \tau_2)}{s^2 + s(1 + k_d k_o \tau_2/\tau_1 + \tau_2) + (k_d k_o/\tau_1 + \tau_2)} = \frac{\omega_n(2\zeta - (\omega_n/k_d k_o))s + \omega_n^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \tag{5.34}$$

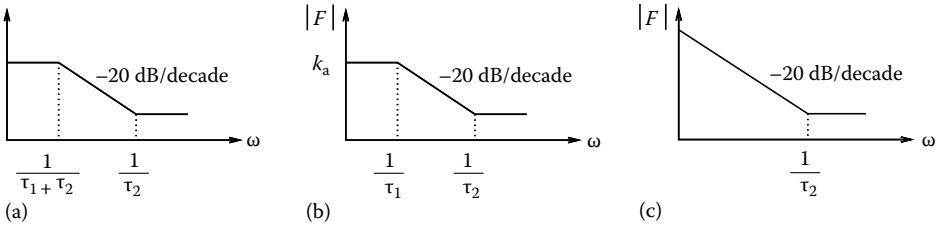


FIGURE 5.13 Bode plots of (a) passive lead-lag filter, (b) active lead-lag filter, and (c) active PI filter.

for the active lead-lag filter

$$H(s) = \frac{k_d k_a k_o (1 + s\tau_2/\tau_1)}{s^2 + s(1 + k_d k_a k_o \tau_2/\tau_1) + (k_d k_a k_o/\tau_1)} = \frac{\omega_n(2\zeta - (\omega_n/k_d k_a k_o))s + \omega_n^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \tag{5.35}$$

and for the active PI filter

$$H(s) = \frac{k_d k_o (1 + s\tau_2/\tau_1)}{s^2 + s(k_d k_o \tau_2/\tau_1) + (k_d k_o/\tau_1 + \tau_2)} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \tag{5.36}$$

If the condition $k_d k_o \gg \omega_n$ or $k_d k_o k_a \gg \omega_n$ is true, this PLL system is called a "high gain loop." If the reverse is true, the system is a "low gain loop." Most practical PLLs are a high gain loop for good tracking performance. For a high gain loop, Equations 5.34 through 5.36 become approximately

$$H(s) \approx \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \tag{5.37}$$

Similarly, assuming a high gain loop, the approximate expression of the phase-error transfer function $H_e(s)$ for all three LF types becomes

$$H_e(s) \approx \frac{s^2}{s^2 + 2s\zeta\omega_n + \omega_n^2} \tag{5.38}$$

The magnitude frequency responses of $H(s)$ for a high gain loop with several values of damping factor are plotted in Figure 5.14. It exhibits that the loop performs an LPF on the input phase signal. That is, the second-order PLL is able to track both phase and frequency modulations of the input signal as long as the modulation frequency remains within the frequency band roughly between zero and ω_n.

The transfer function $H(s)$ has a −3 dB frequency, $\omega_{-3\text{ dB}}$, that stands for the close loop bandwidth of the PLL. The relationship between $\omega_{-3\text{ dB}}$ and ω_n is presented here to provide a comparison with a familiar concept of bandwidth.

In a high gain loop case, by setting $|H(j\omega)| = \frac{1}{\sqrt{2}}$ and solving for ω, we can find

$$\omega_{-3\text{ dB}} = \omega_n\left[2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2}\right]^{\frac{1}{2}} \tag{5.39}$$

The relationship between $\omega_{-3\text{ dB}}$ and ω_n for different damping factors is plotted in Figure 5.15 [4].

The magnitude frequency responses of $H_e(s)$ are plotted in Figure 5.16. A high pass characteristic is observed. It indicates that the second-order PLL tracks the low frequency phase error but cannot track high frequency phase error.

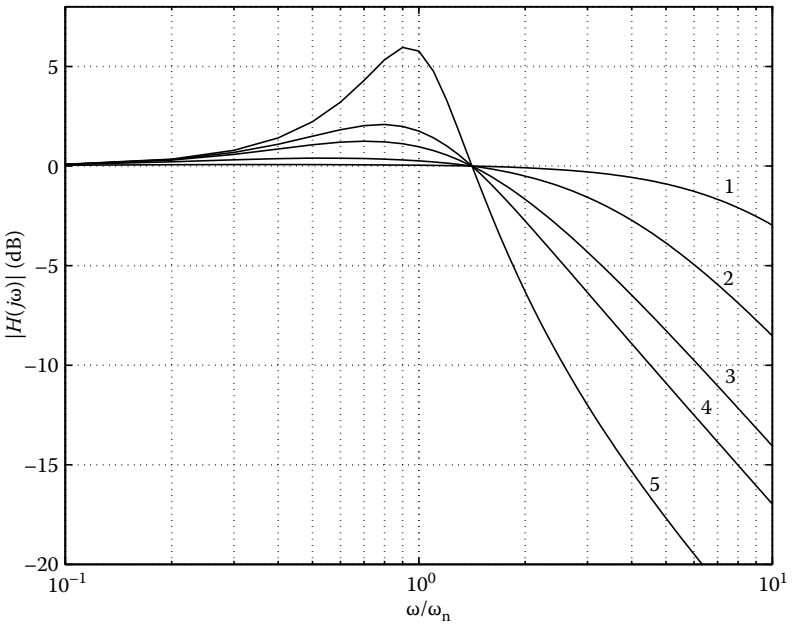


FIGURE 5.14 Frequency responses of the phase transfer function $H(j\omega)$ for different damping factors. Trace1: $\zeta = 5$, Trace2: $\zeta = 2$, Trace3: $\zeta = 1$, Trace4: $\zeta = 0.707$, Trace5: $\zeta = 0.3$.

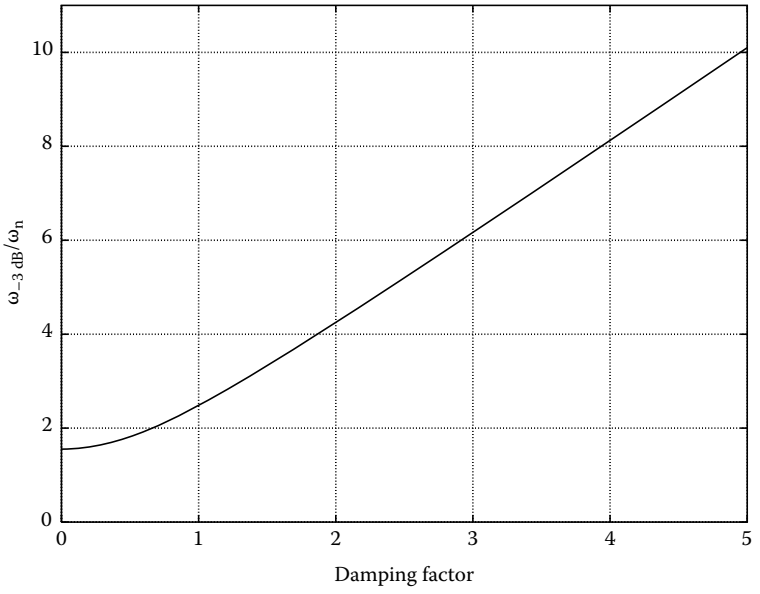


FIGURE 5.15 $\omega_{-3\ dB}$ bandwidth of a second-order loop versus different damping factors.

5.3.3.2 Transformations from *s*-Domain to *z*-Domain

As mentioned in the Section 5.1.3, the function blocks of the ADPLL is implemented by purely digital circuits, and the signals within the loop are digital too. In addition, the SPLL implemented by a microcontroller, microcomputer, or DSP is another type of PLL in discrete-domain (*z*-domain).

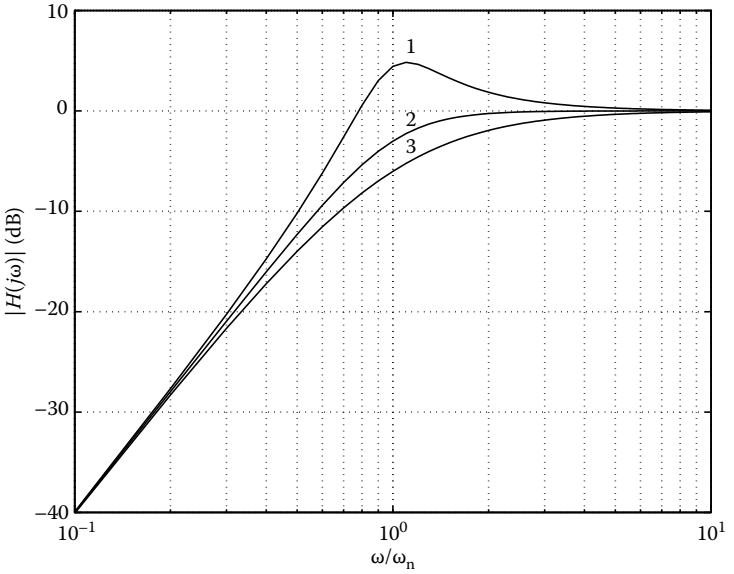


FIGURE 5.16 Frequency responses of the phase-error transfer function $H_e(j\omega)$ for different damping factors. Trace1: $\zeta = 0.3$, Trace2: $\zeta = 0.707$, Trace3: $\zeta = 1$.

Therefore, the analysis and design of a PLL had better be in discrete-domain. The basic types of LF and their features have been described in Section 5.3.3.1. Here, the corresponding discrete-time version of the three basic types of LF will be described after the introduction of transformations from continuous-domain (*s*-domain) to discrete-domain. There are two popular methods to transform a filter from continuous-domain to discrete-domain: backward difference method and bilinear transformation method. Figure 5.17 shows the principle of the backward difference method that we approximate the area under each segment of continuous curve by a rectangular area. Referring to Figure 5.17, the backward difference method means to approximate the integration areas of $\int_{(k-1)}^{kT} y(t)\mathrm{d}t$ by $y(kT)T$. Based on the backward difference method, the *z*-domain equivalent transfer function $H(z)$ of an *s*-domain transfer function $H(s)$ is simple and obtained by the substitution

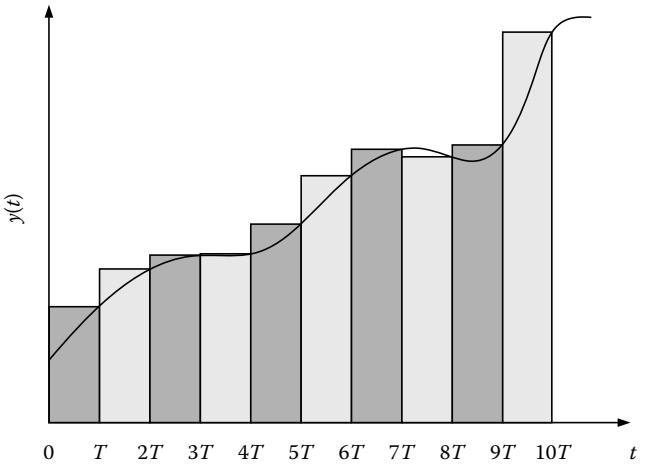


FIGURE 5.17 Backward difference method using a rectangular area approximation.

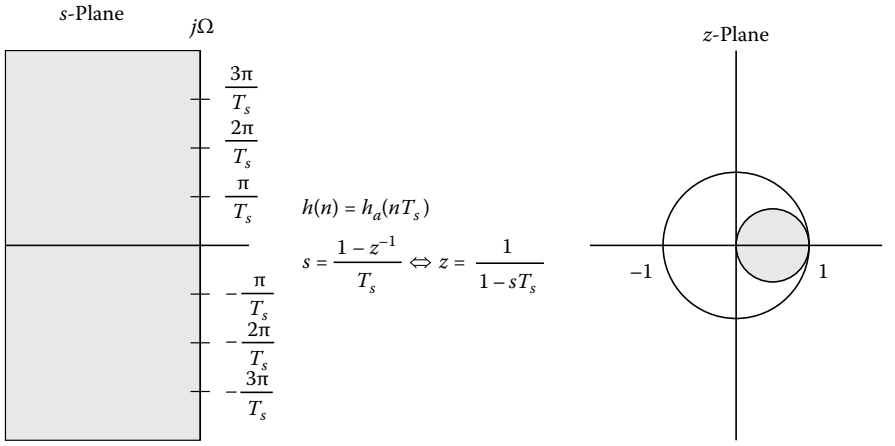


FIGURE 5.18 Mapping of the left half of the s-plane into the z-plane by the backward difference method.

$$H(z) = H(s)\big|_{s=\frac{1-z^{-1}}{T_s}} \tag{5.40}$$

One of the advantages of the backward difference method is that it will produce a stable discrete-time filter for a stable continuous-time filter. Figure 5.18 shows the mapping of the left half of the s-plane into the z-plane by the backward difference method. However, there is considerable distortion in the transient and frequency response characteristics of the discrete-time filter obtained in this method since the stable region is mapped into only a circle within the unit circle of z-plane.

From calculus and Figure 5.17, a good approximation is obtained only if the continuous-time signal changes very slowly over the sampling interval T_s. In other words, the signal bandwidth has to be much smaller than the sampling rate since the mapping from s-domain to z-domain should become distorted while the sampling period is too long. To reduce the distortion, it is desired to use a faster sampling frequency, that is, a smaller sampling period.

Figure 5.19 shows the principle of the bilinear transformation method that we approximate the area under each segment of continuous curve by a trapezoidal area. Therefore, the bilinear transformation method is also called the trapezoidal integration method to approximate the integration areas $\int_{(k-1)}^{kT} y(t)\mathrm{d}t$

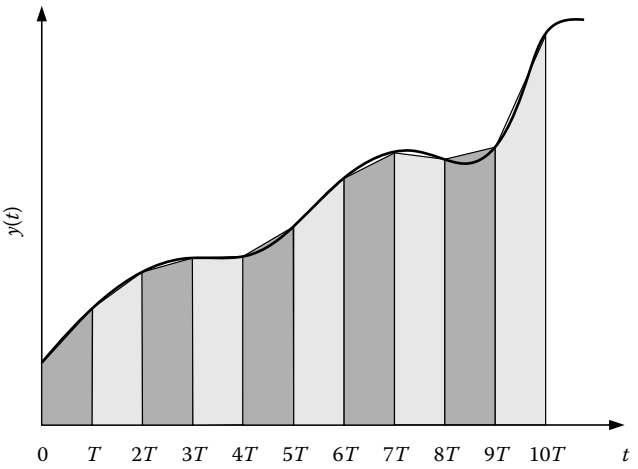


FIGURE 5.19 Bilinear transformation method using trapezoidal area approximation.

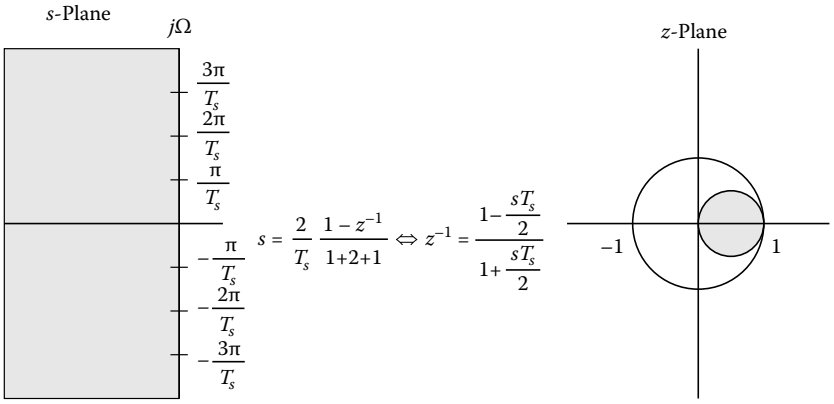


FIGURE 5.20 Mapping of the left half of the *s*-plane into the *z*-plane by the bilinear transformation method.

by $\frac{1}{2}[y(kT) + y((k-1)T)]T$. Thus, the *z*-domain equivalent transfer function $H(z)$ of a continuous-time filter $H(s)$ is obtained by

$$H(z) = H(s)\big|_{s=\frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}} \tag{5.41}$$

By means of bilinear transformation method, the entire left half of the *s*-plane is mapped into the unit circle with center at the origin of the *z*-plane as shown in Figure 5.20. Hence, the bilinear transformation method produces a stable discrete-time filter for a stable continuous-time filter. Furthermore, there is no frequency folding by means of the bilinear transformation method since it maps the entire *jw* axis of the *s*-plane into one complete revolution of the unit circle in the *z*-plane.

5.3.3.3 Discrete-Time LFs

As mentioned in the Section 5.3.3.1, there are three typical LFs for a phase-locked filter. Using backward difference transformation, the discrete-time transfer functions of passive lead-lag filter, active lead-lag filter, and active PI filter can be obtained as follows:

Passive lead-lag filter:

$$F_{\text{back,PLL}}(z) = \frac{T_s + \tau_2 - \tau_2 z^{-1}}{T_s + \tau_1 + \tau_2 - (\tau_1 + \tau_2)z^{-1}} \tag{5.42}$$

Active lead-lag filter:

$$F_{\text{back,ALL}}(z) = k_a \frac{T_s + \tau_2 - \tau_2 z^{-1}}{T_s + \tau_1 - \tau_1 z^{-1}} \tag{5.43}$$

Active PI filter:

$$F_{\text{back,PI}}(z) = \frac{T_s + \tau_2 - \tau_2 z^{-1}}{\tau_1(1 - z^{-1})} \tag{5.44}$$

On the other hand, the discrete-time transfer functions of passive lead-lag filter, active lead-lag filter, and active PI filter can be written, using the bilinear transformation, as follows:

Passive lead-lag filter:

$$F_{\text{bilinear,PLL}}(z) = \frac{T_s + 2\tau_2 + (T_s - 2\tau_2)z^{-1}}{T_s + 2\tau_1 + 2\tau_2 + (T_s - 2\tau_1 - 2\tau_2)z^{-1}} \tag{5.45}$$

Active lead-lag filter:

$$F_{\text{bilinear,ALL}}(z) = k_a \frac{T_s + 2\tau_2 + (T_s - 2\tau_2)z^{-1}}{T_s + 2\tau_1 + (T_s - 2\tau_1)z^{-1}} \tag{5.46}$$

Active PI filter:

$$F_{\text{bilinear}}(z) = \frac{T_s + 2\tau_2 + (T_s - 2\tau_2)z^{-1}}{2\tau_1(1 - z^{-1})} \tag{5.47}$$

From the viewpoint of implementation, all the discrete-time LFs have the transfer function format of a first-order infinite impulse response filter:

$$F_{\text{LF}}(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \tag{5.48}$$

The differences in hardware requirements are small, but the system characteristics and performance are dramatic. Using an approach similar to the Weiner filter theory, Jaffe and Rechtin [30] investigated the optimal LFs for PLLs with different inputs. For a frequency step input, the form of active PI filter is shown to be optimal.

5.3.4 Charge-Pump PLL

A charge-pump PLL usually consists of four major blocks as shown in Figure 5.21. The PD is a purely PFD. The charge-pump circuit converts the digital signals UP, DN, and null (neither up nor down) generated by the PD into a corresponding charge-pump current I_p, $-I_p$, and zero. The LF is usually a passive RC circuit converting the charge-pump current into an analog voltage to control VCO. The purpose of the "charge-pump" is to convert the logic state of the phase-frequency detector output into an analog signal suitable for controlling the VCO. The schematic of the charge-pump circuit and the LF is shown in Figure 5.22. The linear model shown in Figure 5.3 can be employed to describe a charge-pump PLL. k_d is the equivalent gain of a charge-pump circuit. If the loop bandwidth is much smaller than the input frequency, the detailed behavior within a single cycle can be ignored. Then the state of a PLL can be assumed to be only changed by a small amount during each input cycle. Actually the "average" behavior over many cycles is what we are interested in. The average current charging the capacitor is given by

$$\begin{aligned} I_{\text{avg}} &= \frac{Q}{T} = \frac{I\Delta t}{T} \\ &= \frac{I\left(\frac{\phi_e}{2\pi}\right)T}{T} \\ &= \frac{I\phi_e}{2\pi} \end{aligned} \tag{5.49}$$

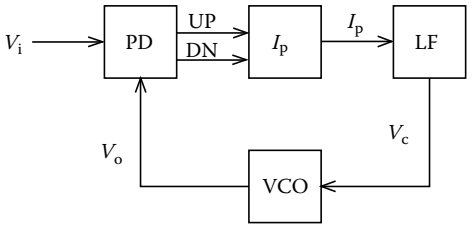


FIGURE 5.21 Charge-pump PLL diagram.

And the average k_d is

$$k_d \triangleq \frac{I_{\text{avg}}}{\phi_e} = \frac{I_{\text{avg}}}{2\pi} \tag{5.50}$$

The charge-pump current is transferred to the control voltage of the following VCO by the LF consisted of a resistor and a capacitor as shown in Figure 5.22. The impedance (transfer function) of the RC LF is given by

$$F(s) = R + \frac{1}{C_p s} = \frac{1 + RC_p s}{C_p s} = K_p + \frac{K_I}{s} \tag{5.51}$$

which has the format of an active PI filter. Therefore, the closed-loop transfer function can be obtained as

$$H(s) \triangleq \frac{\phi_{out}}{\phi_{in}} = \frac{k_d \cdot F(s) \cdot \frac{k_o}{s}}{1 + \frac{k_d k_o}{s}} = \frac{\frac{I_{avg}}{2\pi C_p}(RC_p s + 1)k_o}{s^2 + \frac{I_{avg}}{2\pi} k_o R s + \frac{I_{avg}}{2\pi C_p} k_o} \tag{5.52}$$

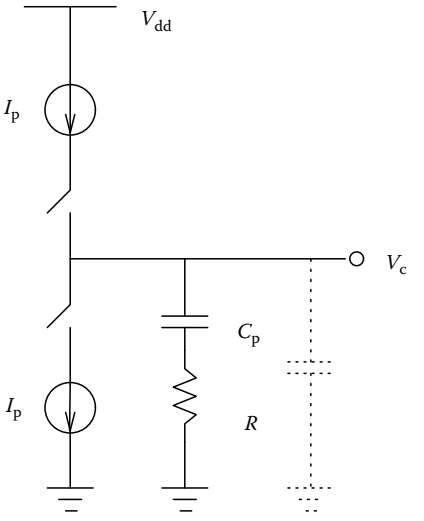


FIGURE 5.22 The schematic of LF.

Generally, a second-order system is characterized by the natural frequency $f_n = \frac{\omega n}{2\pi}$ and the damping factor ζ, and they can be expressed as follows:

$$\omega_n = \sqrt{\frac{I_{avg}}{2\pi C_p} k_o} \text{ rad/s}$$
$$\zeta = \frac{RC_p}{2}\omega_n \tag{5.53}$$

For the stability consideration, there is a limitation of a normalized natural frequency F_N [15],

$$F_N \triangleq \frac{f_n}{f_i} < \frac{\sqrt{1 + \zeta^2} - \zeta}{\pi} \tag{5.54}$$

In the single-ended charge pump, the resistor added in series with the capacitor shown in Figure 5.22 may introduce ''ripple'' in the control voltage V_c even when the loop is locked [16]. The ripple control voltage modulates the VCO frequency and results in phase noise. This effect is especially undesired in frequency synthesizers. In order to suppress the ripple, a second-order LF, as shown in Figure 5.22 with a shunt capacitor in dotted line, is used. This configuration introduces a third pole in the PLL. Stability issues must be taken care of furthermore. Gardner provides criteria for the stability of the third-order PLL [16].

An important property of any PLLs is the static phase error that arises from a frequency offset $\Delta\omega$ between the input signal and the free-running frequency of the VCO. According to the analysis in Ref. [16], the static phase error is

$$\theta_v = \frac{2\pi\Delta\omega}{k_o I_p F(0)} \text{ rad} \tag{5.55}$$

To eliminate the static phase error in conventional PLLs, an active LF with a high DC gain ($F(0)$ is large) is preferred. Nevertheless, the charge-pump PLL allows zero static phase error without the need of

a large DC gain of the LF. This effect arises from the input open circuit during the "null" state (charge-pump current is zero). Real circuits will impose some resistive loading R_s in parallel to the LF. Therefore, the static phase error, from Equation 5.55 will be

$$\theta_v = \frac{2\pi\Delta\omega}{k_o I_p R_s} \text{ rad} \tag{5.56}$$

The shunt resistive loading most likely comes from the input of a VCO control terminal. Compared with the static phase error of a conventional PLL as expressed in Equation 5.11, the same performance can be obtained from a charge-pump PLL without a high DC-gain LF [30].

5.3.5 PLL Design Considerations

5.3.5.1 Typical Procedures of PLL Design

A PLL design usually starts with specifying the key parameters such as natural frequency ω_n, lock-in range $\Delta\omega_L$, damping factor ζ, and the frequency control range which majorly depend on applications. Typical design procedures are described as follows:

Step 1. Specify the damping factor ζ. The damping factor determines the relative stability of a PLL. ζ should be considered as a critical parameter to achieve fast response, small overshoot, and minimum noise bandwidth B_L. If ζ is very small, large overshoot occurs and the overshoot causes phase jitter [19]. If ζ is too large, the response becomes sluggish.

Step 2. Specify the lock-in range $\Delta\omega_L$ or the noise bandwidth B_L. As shown in Equations 5.53 and 5.50, the natural frequency ω_n depends on $\Delta\omega_L$ and ζ (or B_L and ζ). If the noise is not the key issue of the PLL, we may ignore the noise bandwidth and specify the lock-in range. If the noise is concerned, we should specify B_L first, and keep the lock-in range of PLL.

Step 3. Calculate the ω_n according to step 2. If the lock-in range has been specified, Equation 5.53 indicates

$$\omega_n = \frac{\Delta\omega_L}{2\zeta} \tag{5.57}$$

If the noise bandwidth has been specified, Equation 5.50 indicates the natural frequency as

$$\omega_n = \frac{2B_L}{\zeta + \frac{1}{4\zeta}} \tag{5.58}$$

Step 4. Determine the VCO gain factor k_o and the PD gain k_d. k_o and k_d are both characterized by circuit architectures. They must achieve the requirement of the lock-in range specified in step 2. For example, if k_o or k_d is too small, the PLL will fail to achieve the desired lock-in range.

Step 5. Choose the LF. Different types of the LF are available as shown in Figure 5.12. According to Equation 5.9 through 5.11, ω_n and ζ specified above are used to derive the time constants of the LF.

5.3.5.2 PLL Bandwidth Control

As illustrated in previous sections, the PLL noise performance is contrary to the dynamic performances of a PLL such as lock-in range and acquisition speed, which are generally proportional to the PLL bandwidth or the nature frequency ω_n. For example, Equation 5.52 shows that it is desired to have large k_d or $F(\Delta\omega_L)$ to achieve wide lock-in range and fast acquisition. In contrast, Equations 5.59 and 5.60 show that it is desired to have a small nature frequency ω_n in order to reduce the output jitter. Besides, variations in process, voltage, and temperature can lead to uncertainties in loop parameters of a

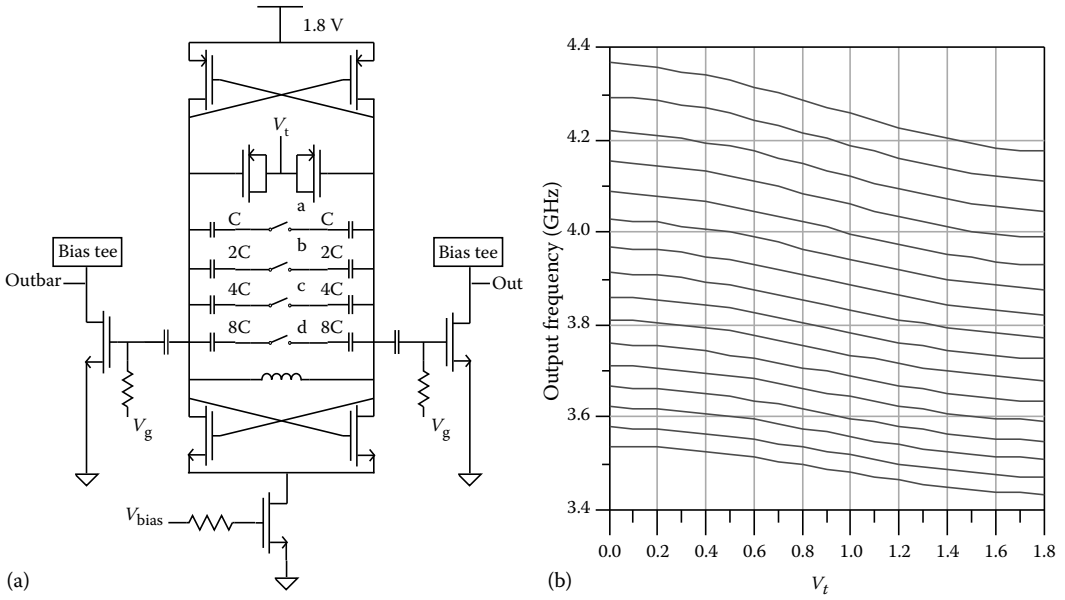


FIGURE 5.23 (a) Wide-range LC-tank VCO with capacitor array and (b) conversion gain k_o.

fixed-bandwidth PLL, such as the conversion gain k_o of VCO, the time-constant ratio τ_2/τ_1 of the LF, and the charge-pump current I_p. For example, variations in process, voltage, and temperature will cause the frequency of a ring oscillator to vary by a factor of 2–3 between its slowest and fastest conditions experientially. Figure 5.23 shows a wide-range LC-tank VCO with capacitor array to achieve low conversion gain k_o for superior phase noise performance. It can be seen that the VCO conversion gain k_o slightly varies from segment to segment of frequencies. Moreover, the curve of conversion gain k_o is nonlinear in any frequency segment. The design result shows a conversion gain range of 27–50 MHz/V. This means that the PLL bandwidth will vary by a factor of 2 due to the variation of VCO conversion gain k_o. Following the design procedures mentioned above, a conservative operating point is usually chosen to guarantee the stability for all conditions. Unfortunately, such a conservative design only achieves a suboptimal performance in most cases.

For achieving fast acquisition as well as a superior noise performance, many schemes of PLL bandwidth control are proposed in the literature [31–37]. Figure 5.24 shows the concept of PLL bandwidth control based on some monitoring techniques of the phase error. In Ref. [31], based on some modifications of the Kalman filtering formulation, an approach is developed to derive the optimal loop gain sequence of dual-loop DPLL, which is independent of measured noise statistics. In order to achieve low noise PLL as

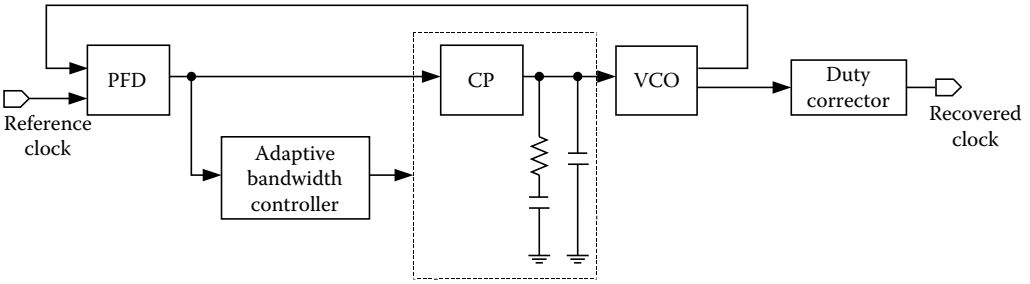


FIGURE 5.24 The concept of an adaptive-bandwidth PLL.

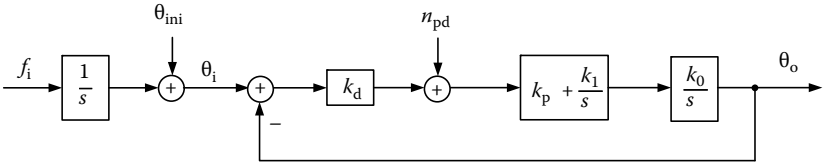


FIGURE 5.25 Linear model of a charge-pump PLL.

well as fast acquistion, Joonsuk Lee and Beomsup Kim [32] realize an analog bandwidth controller to adaptively control the current of charge-pump, that is the PLL bandwidth. Figure 5.25 shows the linear model of charge-pump PLL using a proportional–integral (PI) filter as shown in Figure 5.21. According to Equations 5.51 through 5.53, the damping factor ζ, that is, relative stability of such an adaptive-bandwidth PLL will vary during the lock-in process since the nature frequency ω_n is adaptively adjusted.

In order to develop a PLL and delay-locked loop (DLL) with adaptive-bandwidth to enable optimal performance over a wide frequency range and across process, voltage, and temperature variations, a discrete-time, open-loop dynamic model of the PLL/DLL is proposed to characterize the change in output variables in response to the sampled error, and to express the adaptive-bandwidth criteria in terms of the open-loop gains [37]. Furthermore, the scaling equations for the charge-pump current and the filter resistor are derived to achieve adaptive-bandwidth charge-pump PLL/DLLs with a constant damping factor ζ. It is regrettable that such scaling scheme is a challenge in the analog realization. In contrast, it is easier to maintain a constant damping factor ζ using a digital PI LF to achieve the adaptive bandwidth control. In Ref. [11], a modified structure of digital PI LF is proposed and shown in Figure 5.26. Based on such a modified structure, it is easy to derive the equivalent phase-transfer function in continuous-time domain that is represented in terms of the parameters of loop components as

$$\begin{aligned} H_\theta(s) &= \frac{\Theta_o(s)}{\Theta_i(s)} \\ &= \frac{k_d k_o K_P s + k_d k_o K_P K_I}{s^2 + k_d k_o K_P s + k_d k_o K_P K_I} \\ &= \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned} \tag{5.59}$$

where

$$\begin{aligned} \omega_n &= \sqrt{k_d k_o K_P K_I}, \\ \zeta &= 0.5\sqrt{k_d k_o K_P / K_I} \end{aligned} \tag{5.60}$$

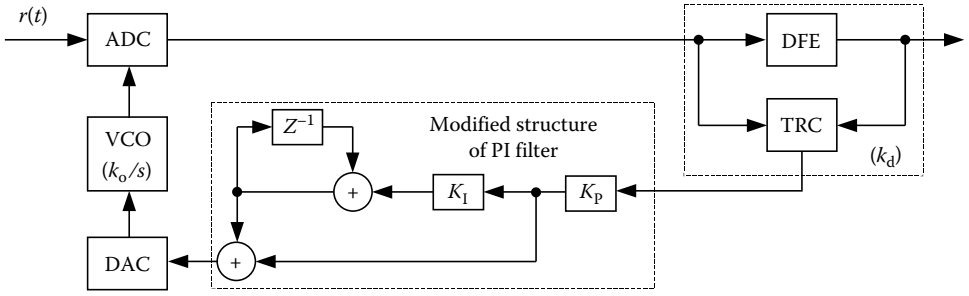


FIGURE 5.26 Timing recovery using a modified structure of digital PI LF.

denote the nature frequency and the damping factor, respectively. Referring to Equation 5.60, if a scaling factor of K_P and K_I is induced into the modified PI LF structure for adaptively adjusting the PLL bandwidth or the nature frequency ω_n, we can mantain a constant damping factor ζ while the PLL bandwidth is proportional to the scaling factor induced.

5.4 PLL Applications

5.4.1 Clock and Data Recovery

In data transmission systems such as optical communications, telecommunications, disk drive systems, and local networks, data are transmitted on baseband or passband. In most of these applications, only data signals are transmitted by transmitter, but clock signals are not transmitted in order to save hardware cost. Therefore, the receiver should have some schemes to extract the clock information from the received data stream and to regenerate transmitted data using the recovered clock. This scheme is called timing recovery or clock recovery.

To recover the data correctly, the receiver must generate a synchronous clock from the input data stream, the recovered clock must synchronize with the bit rate (the baud of data). The PLL can be used to recover the clock from the data stream, but there are some special design considerations. For example, because of the random nature of data, the choice of PFDs is restricted. In particular, three-state PD is not proper, because when there are no transitions in the data stream, the PD interprets that the VCO frequency is higher than the data frequency and remains its output in the ''down'' state, which makes the PLL lose lock as shown in Figure 5.27. Thus, the choice of PFD for random binary data requires a careful examination over whether data transitions are absent. One useful method is the rotational frequency detector described in Ref. [9]. The random data also causes the PLL to introduce undesired phase variation in the recovered clock, it is called timing jitter and this is an important issue of the clock recovery.

5.4.1.1 Data Format

Binary data are usually transmitted in an NRZ format as shown in Figure 5.28a because of the consideration of bandwidth efficiency. In NRZ format, each bit has a duration of T_B (bit period). The signal does not go to zero between adjacent pulses representing 1's. It can be shown in Ref. [24] that the corresponding spectrum has no line component at $f_B = \frac{1}{T_B}$, most of the spectrum of this signal lines below $\frac{f_B}{2}$. The term ''NRZ'' distinguishes itself from another data type called ''return-to-zero'' (RZ) as shown in Figure 5.28b, where the signal goes to zero between consecutive bits. Therefore, the spectrum of RZ data have a frequency component at f_B. For a given bit rate, RZ data need wider transmitting bandwidth; therefore, NRZ data are preferable when channel or circuit bandwidth is a concern.

Due to the lack of a spectral component at the bit rate of NRZ format, a clock recovery circuit may lock to spurious signals or fail to lock at all. Thus, a nonlinear process for the NRZ data is essential to create a frequency component at the baud rate.

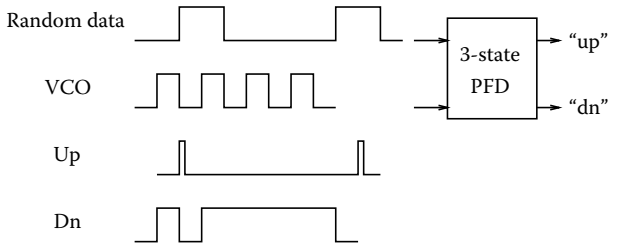


FIGURE 5.27 Response of a three-state PD to random data.

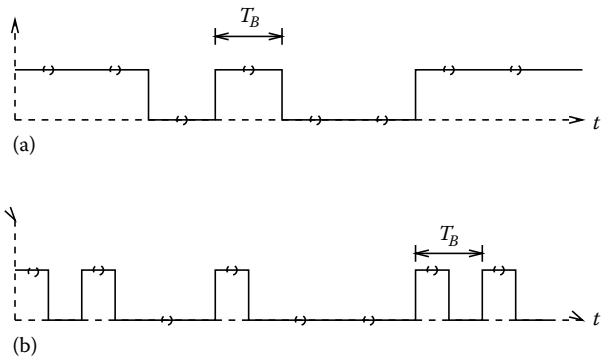


FIGURE 5.28 (a) NRZ data and (b) RZ data.

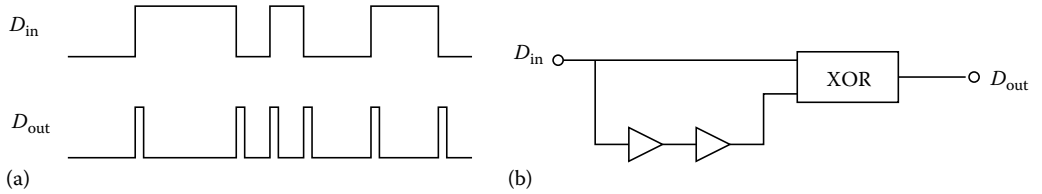


FIGURE 5.29 Edge detection of NRZ data.

5.4.1.2 Data Conversion

One way to recover the clock signal from the NRZ data is to convert it to an RZ-like data that have a frequency component at the bit rate, and then recover clock from data using a PLL. Transition detection is one of the methods to convert NRZ data to RZ-like data. As illustrated in Figure 5.29a, the edge detection requires a mechanism to sense both positive and negative data transitions. In Figure 5.29b, NRZ data are delayed and compared with itself by an exclusive-OR gate; therefore, the transition edges are detected. In Figure 5.30, the NRZ data V_i is first differentiated to generate pulses corresponding to each transition. These pulses are made to be all positive by squaring the differentiated signal v_i. The result is that the signal V_i' looks just like RZ data, where pulses are spaced at an interval of T_B.

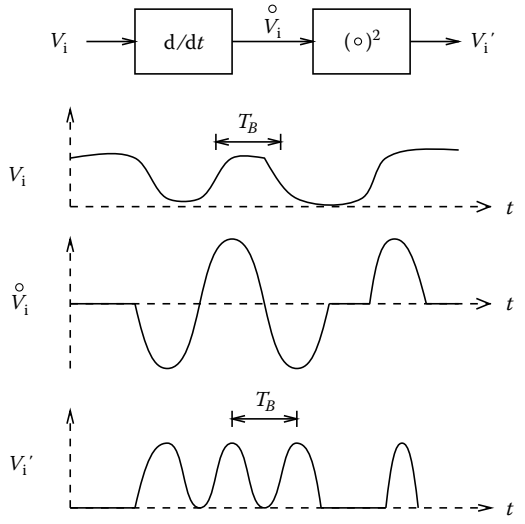


FIGURE 5.30 Converting NRZ to RZ-like signal.

5.4.1.3 Clock Recovery Architecture

Based on different PLL topologies, there are several clock recovery approaches. Here, the early–late and the edge-detector-based methods will be described.

Figure 5.31 shows the block diagram of the early–late method. The waveforms for the case in which the input lags the VCO output are shown in Figure 5.32, where the early integrator integrates the input signal for the early-half period of the clock signal, and holds it for the remainder of the clock signal. On the other

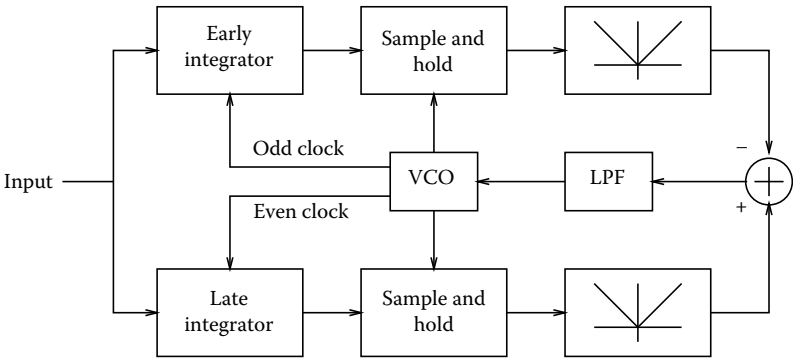


FIGURE 5.31 Early–late block diagram.

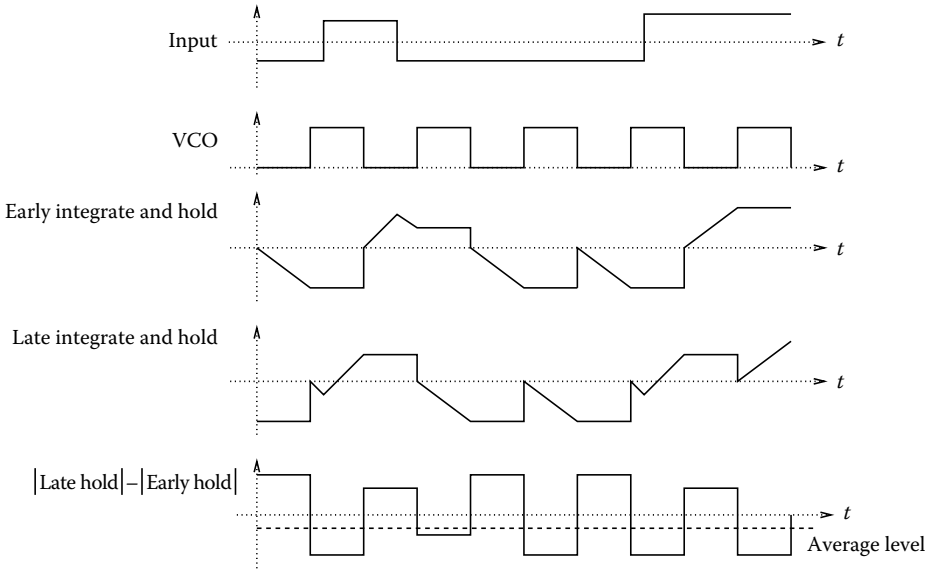


FIGURE 5.32 Clock waveforms for early–late architecture.

hand, the late integrator integrates the input signal for the late-half period of the clock signal and holds it for the next early-half period. The average difference between the absolute values of the late hold and the early hold voltage generated from an LPF gives the control signal to adjust the frequency of the VCO. As mentioned above, this method is popular for rectangular pulses. However, there are some drawbacks in this method. As this method relies on the shape of pulses, a static phase error can be introduced if the pulse shape is not symmetric. In high-speed applications, this approach requires a fast settling integrator that limits the operating speed of the clock recovery circuit and the acquisition time cannot be easily controlled.

The most widely used technique for clock recovery in high performance, wide-band data transmission applications is the edge-detection-based method. The edge-detection method is used to convert data format such that the PLL can lock the correct baud frequency. More details have been described in Section 5.4.1.2. There are many variations of this method depending on the exact implementation of each PLL loop component. The ''quadricorrelator'' introduced by Richman [7] and modified

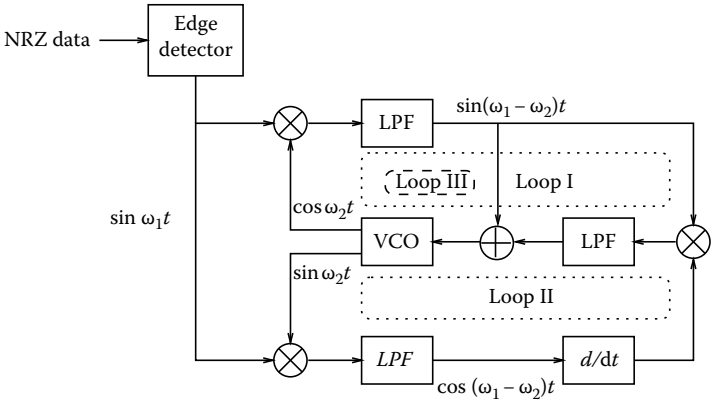


FIGURE 5.33 Quadricorrelator.

by Bellisio [25] is a frequency-difference discriminator and has been implemented in a clock recovery architecture. Figure 5.33 [26] is a phase-frequency locked loop using edge-detection method and quadricorrelator to recover timing information from NRZ data. As shown in Figure 5.33, the quadricorrelator follows the edge-detector with a combination of three loops sharing the same VCO. Loops I and II form a frequency-locked loop that contains the quadricorrelator for frequency detection. Loop III is a typical PLL for phase alignment. Since the phase- and frequency-locked loops share the same VCO, the interaction between two loops is a very important issue. As described in Ref. [26], when $\omega_1 \approx \omega_2$, the DC feedback signal produced by loops I and II approaches zero and loop III dominates the loop performance. A composite frequency- and PLL is a good method to achieve fast acquisition and a narrow PLL loop bandwidth to minimize the VCO drift. Nevertheless, because the wide band frequency-locked loop can response to noise and spurious components, it is essential to disable frequency-locked loop when the frequency error gets into the lock-in range of the PLL to minimize the interaction. More clock recovery architectures are described in Refs. [19,21,23,27–29].

5.4.2 Delay-Locked Loop

Two major elements for adjusting the timing are VCO and voltage-controlled delay line (VCDL). Figure 5.34 shows a typical DLL [12,13] that replaces the VCO of a PLL with a VCDL. The input signal is delayed by an integer multiple of the signal period because the phase error is zero when the phase difference between V_{in} and V_o approaches multiple of the signal periods. The VCDL usually consists a number of cascaded gain stages with variable delay. Delay lines, unlike ring oscillators, cannot generate a signal; therefore, it is difficult to make frequency multiplication in a DLL.

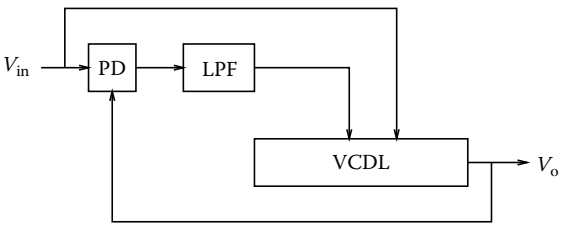


FIGURE 5.34 DLL block diagram.

In a VCO, the output "frequency" is proportional to the input control voltage. The phase transfer function contains a pole, which is $H(s) = \frac{k_o}{s}$ (k_o is the VCO gain). In a VCDL, the output "phase" is proportional to the control voltage, and the phase transfer function is $H(s) = k_{VCDL}$. So the DLL can be easily stabilized with a simple first-order LF. Consequently, DLLs have much more relaxed trade-offs among gain, bandwidth, and stability. This is one of the two important

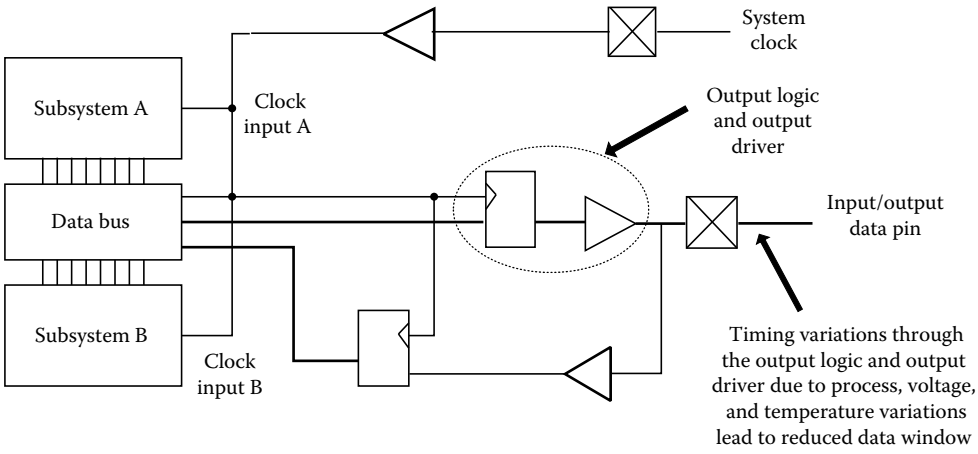


FIGURE 5.35 Modern digital systems use synchronous communication to achieve high-speed signaling to and from the bus between the subsystems.

advantages over PLLs. Another advantage is that delay lines typically introduce much less jitter than VCO [14]. Because a delay chain is not configured as a ring-oscillator, there is no jitter accumulation since the noise does not contribute to the starting point of the next clock cycle.

A typical application of DLL is to synchronize the clock edges of subsystems within a digital system to access the bus between subsystems. Figure 5.35 shows modern digital systems that use synchronous communication to achieve high-speed signaling to and from the bus between the subsystems. Subsystems that communicate synchronously use a clock signal as a timing reference so that data can be transmitted and received with a known relationship to this reference. A difficulty in maintaining this relationship is that process, voltage, and temperature variations can alter the timing relationship between the clock and data signals of subsystems, resulting in reduced timing margins. Figure 5.36 shows that on the left side the data valid window (the time over which data can be sampled reliably by the receiver) can be large at low signaling speeds [38]. Even in the presence of a substantial shift in the data valid window across operational extremes, the resulting data valid window can still be large enough to transmit and receive the data reliably. Unfortunately, the variations in process, voltage, and temperature can result in the loss of

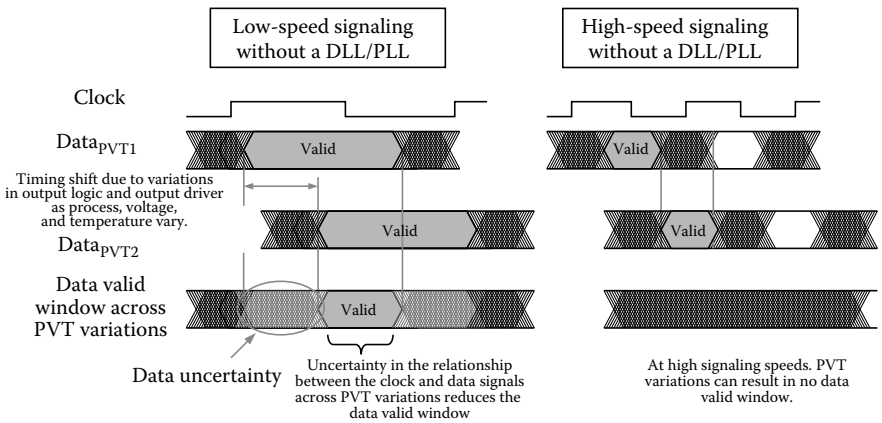


FIGURE 5.36 The timing relationships between the clock and data signals of subsystems in a conventional digital system.

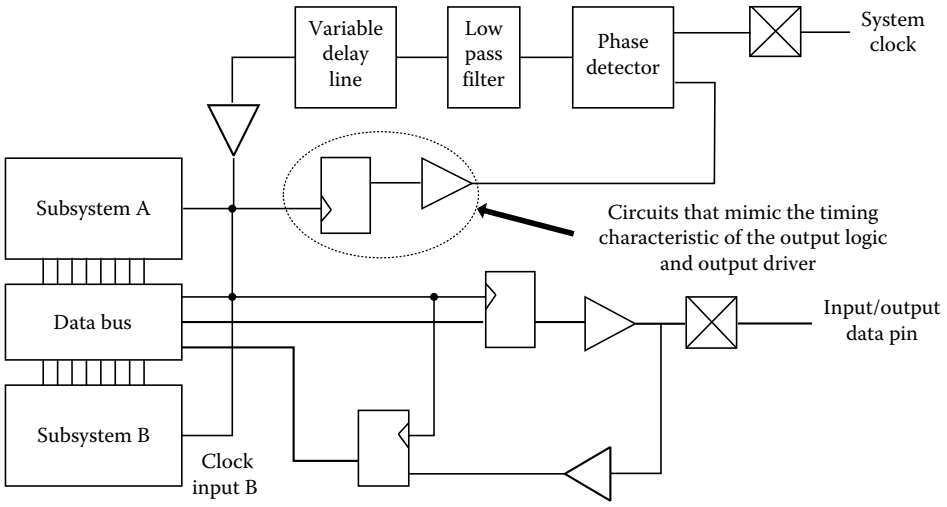


FIGURE 5.37 DLL-on-chip to maintain the timing relationship between a clock signal and an output data signal.

the data valid window when the signal speed is increased as also shown on the right-hand side of Figure 5.36. This problem gets worse as signaling speeds increase, limiting the ability of subsystems to communicate data at higher speeds.

The function of DLLs and PLLs to synchronize a signal with a reference or an input signal in frequency as well as in phase can be used to maintain such a fixed timing relationship between signals of subsystems. Figure 5.37 shows how a DLL is used to maintain the timing relationship between a clock signal and an output data signal. The PD detects phase differences between the clock and output data and sends control information through an LPF to a variable delay line that adjusts the timing of the internal clock to maintain the desired timing relationship. The PD must account for the timing characteristics of the output logic and output driver. This is important since it estimates the phase differences between the clock and the data driven by the output driver, where the timing relationships of subsystems are changed over time due to the process, voltage, and temperature variations. Maintaining the timing relationships between the clock and output data with DLLs and PLLs results in improved timing margins as shown in Figure 5.38. Then, the important limitation to increasing signaling speeds is addressed.

5.4.3 Frequency Synthesizer

A frequency synthesizer generates any of a number of frequencies by locking a VCO to an accurate frequency source such as a crystal oscillator. For example, RF systems usually require a high-frequency local oscillator whose frequency can be changed in small and precise steps. The ability of multiplying a reference frequency makes PLLs attractive for synthesizing frequencies.

The basic configuration used for frequency synthesis is shown in Figure 5.39a. The system is capable of generating the frequency at an integer multiple of the reference frequency. A quartz crystal is usually used as the reference clock source because of its low jitter characteristic. Due to the limited speed of CMOS device, it is difficult to generate frequency directly in the range of GHz or more. To generate higher frequencies, prescalers are used, which are implemented with other IC technologies such as ECL. Figure 5.39b shows a synthesizer structure using a prescaler V, where the output frequency becomes

$$f_{\text{out}} = \frac{NVf_{\text{i}}}{M} \tag{5.61}$$

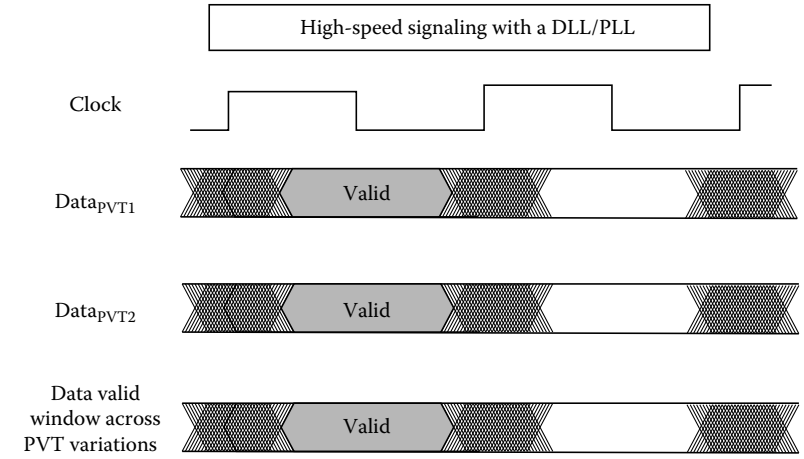


FIGURE 5.38 Timing relationships between subsystems when a DLL is employed for synchronizing the bus access.

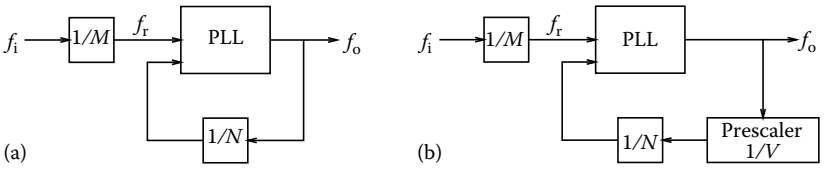


FIGURE 5.39 Frequency-synthesizer block diagrams: (a) basic frequency-synthesizer system and (b) system extends the upper frequency range by using an additional high-speed prescaler.

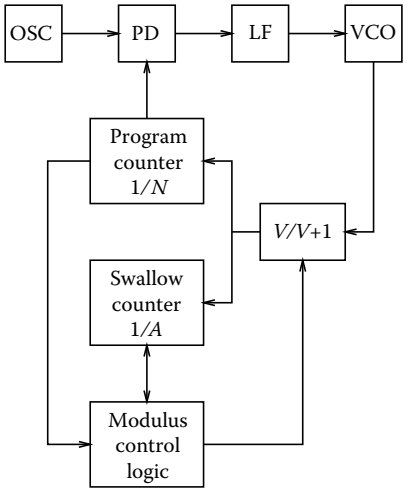


FIGURE 5.40 The block diagram of dual-modulus frequency synthesizer.

Because the scaling factor V is obviously much greater than one, it is no longer possible to generate any desired integer multiple of the reference frequency. This drawback can be circumvented by using a so-called dual-modulus prescaler as shown in Figure 5.40. A dual-modulus prescaler is a divider whose division can be switched from one value to the other by a control signal. The following shows that the dual-modulus prescaler makes it possible to generate a number of output frequencies that are spaced only by one reference frequency. The VCO output is divided by $V/V+1$ dual-modulus prescaler. The output of the prescaler is fed into a "program counter" $1/N$ and a "swallow counter" $1/A$. The dual-modulus prescaler is set to divide by $V+1$ initially. After "A" pulses out of the prescaler, the swallow counter is full and changes the prescaler modulus to V. After additional "N–A" pulses out of the prescaler, the program counter changes the prescaler modulus back to $V+1$ and restarts the swallow counter. Then the cycle

is repeated. In this way, the VCO frequency is equal to $(V+1)A + V(N-A) = VN + A$ times of the reference frequency. Note that N must be larger than A. If this is not the case, the program counter would be full earlier than the $1/A$ and both counters would be reset. Therefore, the dual-modulus prescaler would never be switched from $V+1$ to V. For example, if $V=64$, then A must be in the range of 0–63 such that $N_{\min}=64$. The smallest realizable division ratio is

$$(N_{\text{tot}})_{\min} = N_{\min} V = 4096 \tag{5.62}$$

The synthesizer of Figure 5.40 is able to generate all integer multiple of the reference frequency starting from $N_{\text{tot}}=4096$. For extending the upper frequency range of frequency synthesizers but still allowing the synthesis of lower frequency, the four-modulus prescaler is a solution [1].

Based on the above discussions, the synthesized frequency is an integer multiple of a reference frequency. In RF applications, the reference frequency is usually larger than the channel spacing for loop dynamic performance considerations, in which the wider loop bandwidth for a given channel spacing allows faster settling time and reduces the phase jitter requirements to be imposed on the VCO. Therefore a ''fractional'' scaling factor is needed. Fractional division ratios of any complexity can be realized. For example, a ratio 3.7 is obtained if a counter is forced to divide by 4 in seven cycles of each group of 10 cycles and by 3 in the remaining three cycles. On the average, this counter divides the input frequency by 3.7 effectively.

Bibliography

1. R. E. Best, *Phase-Locked Loops Theory, Design, Applications*, McGraw-Hill, New York, 1984.
2. T. G. Donald and J. D. Gallia, *Digital Phase-Locked Loop Design Using SN54/74LS297*, Application Note AN 3216, Texas Instruments Inc., Dallas, TX.
3. W. Rosink, All-digital phase-locked loops using the 74HC = HCT297, *Electron. Components and Appl.*, 9, 66–89, 1989.
4. F. M. Gardner, *Phaselock Techniques*, 2nd ed., John Wiley & Sons, New York, 1979.
5. S. G. Tzafestas, *Walsh Functions in Signal and Systems Analysis and Design*, Van Nostrand Company, 1985.
6. F. M. Gardner, Acquisition of phaselock, *Conference Record of the International Conference on Communications*, ICC'76 vol. I, pp. 10.1–10.5, June 1976.
7. D. Richman, Color carrier reference phase synchronization accuracy in NTSC color television, *Proc. IRE*, 42, 106–133, 1954.
8. F. M. Gardner, Properties of frequency difference detector, *IEEE Trans. Commun.*, COM-33, 131–138, 1985.
9. D. G. Messerschmitt, Frequency detectors for PLL acquisition in timing and carrier recovery, *IEEE Trans. Commun.*, COM-27, 1288–1295, 1979.
10. R. B. Lee, Timing recovery architecture for high speed data communication system, Master thesis, Department of Electrical Engineering, National Taiwan University, Taiwan, ROC, 1993.
11. M.-T. Shiue, Transceiver VLSI design for high speed local access modems, PhD thesis, Department of Electrical Engineering, National Central University, Chung-Li, Taiwan, ROC, 1998.
12. M. Bazes, A novel precision MOS synchronous delay lines, *IEEE J. Solid-State Circuits*, 20, 1265–1271, 1985.
13. M. G. Johnson and E. L. Hudson, A variable delay line PLL for CPU-coprocessor synchronization, *IEEE J. Solid-State Circuits*, 23, 1218–1223, 1988.
14. B. Kim, T. C. Weigandt, and P. R. Gray, PLL/DLL systems noise analysis for low jitter clock synthesizer design, *IEEE International Symposium on Circuits and Systems*, ISCAS'94, 4, pp. 31–34, 1994.
15. M. V. Paemel, Analysis of a charge-pump PLL: A new model, *IEEE Trans. Commun.*, COM-42, 131–138, 1994.

16. F. M. Gardner, Charge-pump phase-locked loops, *IEEE Trans. Commun*., COM-28, 1849–1858, 1980.
17. F. M. Gardner, Phase accuracy of charge pump PLL's, *IEEE Trans. Commun*., COM-30, 2362–2363, 1982.
18. T. C. Weigandt, B. Kim, and P. R. Gray, Analysis of timing recovery jitter in CMOS ring oscillator, *IEEE International Symposium on Circuits and Systems*, ISCAS'94, 4, pp. 27–30, 1994.
19. T. H. Lee and J. F. Bulzacchelli, A 155-MHz clock recovery delay- and phase-locked loop, *IEEE J. Solid-State Circuits*, 27, 1736–1746, 1992.
20. M. P. Flyun and S. U. Lidholm, A 1.2 μm CMOS current-controlled oscillator, *IEEE J. Solid-State Circuits*, 27, 982–987, 1992.
21. S. K. Enam and A. A. Abidi, NMOS IC's for clock and data regeneration in gigabit-per-second optical-fiber receivers, *IEEE J. Solid-State Circuits*, 27, 1763–1774, 1992.
22. M. Horowitz et al., PLL design for a 500 MB/s Interface, *IEEE International Solid-State Circuit Conference/Digest of Technical Papers*, pp. 160–161, 1993.
23. A. Pottbacker and U. Langmann, An 8 GHz silicon bipolar clock-recovery and data-regenerator IC, *IEEE J. Solid-State Circuits*, 29, 1572–1751, 1994.
24. B. P. Lathi, *Modern Digital and Analog Communication System*, HRW, Philadelphia, 1989.
25. J. A. Bellisio, A new phase-locked loop timing recovery method for digital regenerators, *IEEE Int. Comm. Conf. Rec*., 1, 10.17–10.20, 1976.
26. B. Razavi, A 2.5-Gb/s 15-mW clock recovery circuit, *IEEE J. Solid-State Circuits*, 31, 472–480, 1996.
27. R. J. Baumert, P. C. Metz, M. E. Pedersen, R. L. Pritchett, and J.A. Young, A monolithic 50–200 MHz CMOS clock recovery and retiming circuit, *Proceedings of the IEEE Custom Integrated Circuits Conference*, 14.5.5–14.5.4, San Diego, CA, 1989.
28. B. Lai and R. C. Walker, A monolithic 622 Mb/s clock extraction data retiming circuit, *IEEE International Solid-State Circuit Conference/Digest of Technical Papers*, pp. 144–145, 1991.
29. B. Kim, D. M. Helman, and P. R. Gray, A 30 MHz hybrid analog/digital clock recovery circuit in 2-μm CMOS, *IEEE J. Solid-State Circuits*, 25, 1385–1394, 1990.
30. R. Jaffe and E. Rechtin, Design and performance of phase-lock circuits capable of near-optimal performance over a wide range of input signal and noise levels, *IRE Trans. Inf. Theory*, IT-1, 66–76, 1955.
31. B. Chun, Y. H. Lee, and B. Kim, Design of variable loop gains of dual-loop DPLL, *IEEE Trans. Commun*., COM-45, 1520–1522, 1997.
32. J. Lee and B. Kim, A low-noise fast-lock phase-locked loop with adaptive bandwidth control, *IEEE J. Solid-State Circuits*, 35, 1137–1145, 2000.
33. J. Dunning, et al., An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors, *IEEE J. Solid-State Circuits*, 30, 412–422, 1995.
34. J. G. Maneatis, Low-jitter process-independent DLL and PLL based on self-biased techniques, *IEEE J. Solid-State Circuits*, 31, 1723–1732, 1996.
35. S. Sidiropoulos, et al., Adaptive bandwidth DLLs and PLLs using regulated supply CMOS buffers, in *IEEE Symp. VLSI Circuits Dig. Tech. Papers*, June 2000, pp. 124–127.
36. J. Kim and M. A. Horowitz, Adaptive-supply serial links with sub-1 V operation and per-pin clock recovery, *IEEE J. Solid-State Circuits*, 37, 1403–1413, 2002.
37. J. Kim, et al., Design of CMOS adaptive-bandwidth PLL/DLLs: A general approach, *IEEE Trans. Circuits Systems-II*, 50, 860–869, 2003.
38. http://www.rambus.com/products/innovationslicensing/innovations/dllpll.aspx

6

Synthesis of Reactance Pulse-Forming Networks

6.1 Introduction **6**-1
6.2 Networks Forming Quasi-Rectangular Output Pulses **6**-2
Quasi-Rectangular Output Pulse and Its Laplace Transform • Realization Requirements • Second Approximation Step: Approximation for Realization • Pulse-Forming Networks with Non-Delayed Output Pulses • Pulse-Forming Networks with Delayed Output Pulse
6.3 Transfer Functions of Wideband Amplifiers **6**-15
Parameters of the Step Response and Its Laplace Transform • Transfer Function Approximation • Example of Transfer Function Design • Tabulated Results
6.4 Forming a Sinusoidal Pulse **6**-19
Required Transfer Function • Approximation for Realization • Example
6.5 Summary **6**-24
References **6**-25

Igor M. Filanovsky
University of Alberta

6.1 Introduction

Linear lumped reactance parameter networks loaded by resistors can be used to form pulses of different shapes [1,2]. The most known applications [2] require the networks that form pulses of the so-called quasi-rectangular shape. The network excitation is usually a step voltage, and it is required that the network output response (in this case it is the step response) is a pulse of finite duration. In a realizable network, the fronts of this pulse should have ''rounded'' corners, and the slopes of the fronts should also be finite. Below this is done using a semi-period of $\sin^2 t$ function.

These pulse-forming networks will be considered first, and it will be shown that the required result may be obtained using different input excitations and using, of course, different networks. The pulse of the same shape can be obtained when the excitation is a step, an impulse or even a sinusoidal function. The last result looks surprising for linear networks, yet the reader should understand that the output pulse is shaped during the transient period, when the sinusoidal voltage is turning on. When the transient is over, the steady-state response of the network is zero.

This reconsideration of the old problem establishes a certain difference between synthesis in time domain and frequency domain. The output signal time-domain approximation, in case of pulse-forming

networks, should also result in a realizable transfer function. But this synthesis results in many solutions not only because of many possible realizations of the same transfer function. In addition, one may change input excitation (even though in practice the choice of input excitations is limited), then find modified transfer function, and then synthesize new networks. This modification of input signal is not used in the frequency-domain synthesis where it is assumed that the input excitation is the sinusoidal signal.

The method of approximation proposed here gives new and useful results even for the case of the step voltage input excitation and quasi-rectangular shape of the output response. It is shown that if the output pulse is delayed with respect to the input step, then it is possible to shape the output pulse so that its amplitude will be higher than the amplitude of the applied input step voltage. The increase of amplitude is usually connected with the idea of using a transformer, yet the pulse-shaping circuit may provide transformerless change of amplitude.

The approximation developed here for pulse forming may also be applied to the synthesis of wideband amplifier transfer functions. Unfortunately, an initial attempt of using $\sin^2 t$ for this purpose [3] did not bring any general results, and the required transfer functions were found numerically [4]. But even this attempt was forgotten, and the transfer functions of the wideband amplifiers were found indirectly. They were obtained starting from the frequency domain and investigating the time-domain response of the filter transfer functions. It was found that the step responses of the Bessel filters do not have overshoot. It happened that exactly this transient response is required in realization of the wideband amplifiers. Yet, as shown here, using $\sin^2 t$ in the approximation of the impulse response allows one to find the realizable transfer functions directly, on the basis of requirements formulated in the time domain to the step response.

Finally, the considered approximation method allows one to find realizable transfer functions for the networks with the step or impulse response representing the sinusoidal pulse of finite duration, and with the pulse envelope that can be of an arbitrary shape. As an example, we consider the case when this envelope is also of sinusoidal shape. The pulses of these shapes find applications in ultra-wideband transmission networks and wavelet signal analysis. These pulse-shaping networks are usually realized by active networks. Yet, the fact that at least one physically realizable network is available is important for using of algorithms, which are able to improve an approximation and find a transfer function that is better suitable for the chosen realization method.

Many of the results given here can be found in [5]. Considering that this source of information is not easily accessible, a sincere effort is done to summarize all available material.

6.2 Networks Forming Quasi-Rectangular Output Pulses

In this section, we consider all procedures required in the synthesis of pulse-forming networks [5,6]. They start by approximation of the output pulse in the time domain and finding the Laplace transform of this pulse. Then, we review the requirements imposed on the realizable transfer function. After that we consider the second round of approximation, when the Laplace transform is approximated by an algebraic ratio obtained from the spectral representation of the Laplace transform. Finally, we give examples of realization. Some of these procedures will be omitted in parts dedicated to the wideband amplifier transfer functions and forming of the sinusoidal pulses.

6.2.1 Quasi-Rectangular Output Pulse and Its Laplace Transform

Assume that it is required to find reactance networks forming a quasi-rectangular pulse for three cases of input excitation, $v_i(t)$, namely, the unit impulse, the unit step, and the sinusoid with unit amplitude. The ratio of two polynomials approximating the Laplace transform of the output response will be the same for all three cases. In case of impulse excitation, $v_i(t) = \delta(t)$, this ratio is directly equal to the transfer function of the pulse-forming network. In case of $v_i(t) = u(t)$, a unit step voltage, the required transfer function is obtained by multiplication of the approximating ratio by s, and in case of $v_i(t) = \sin \Omega t$ this ratio is multiplied by $(s^2 + \Omega^2)$.

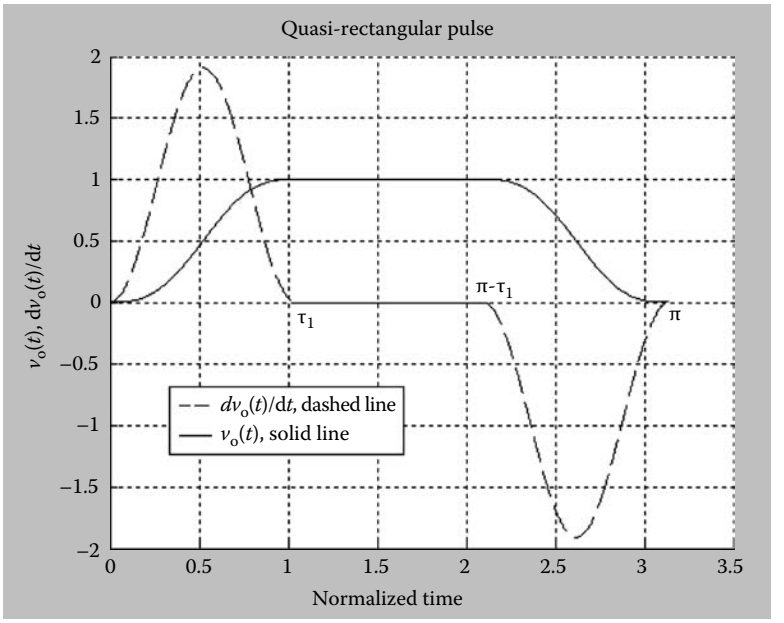


FIGURE 6.1 Quasi-rectangular pulse.

The required output response, $v_o(t)$, is a symmetric pulse (Figure 6.1) of the normalized duration $\tau = \pi$ for all considered cases of excitation. Assume that the derivative, dv_o/dt, of this response is approximated by one semi-period of $\sin^2(\pi t/\tau_1)$ function for the initial front of the pulse, and one delayed negative semi-period of $\sin^2(\pi t/\tau_1)$ for the rear front. The duration of each front is equal to τ_1. Then dv_o/dt is described by the following system of equations:

$$\frac{dv_o}{dt} = \begin{cases} A\sin^2(\pi t/\tau_1) & 0 < t < \tau_1 \\ 0 & \tau_1 < t < (\tau - \tau_1) \\ -A\sin^2[\pi(t - \tau + \tau_1)/\tau_1] & (\tau - \tau_1) < t < \tau \\ 0 & t > \tau = \pi \end{cases}. \tag{6.1}$$

The amplitude, A, will be obtained from the normalization condition

$$A\int_0^{\tau_1} \sin^2(\pi t/\tau_1)dt = 1. \tag{6.2}$$

This gives $A = 2/\tau_1$. This value of A is also equal to the maximal slew rate of the fronts, so that the approximate rise and fall times can be evaluated as

$$\tau_r = \tau_f = 1/A = \tau_1/2. \tag{6.3}$$

These rise and fall times are close to ones defined by 0.1 and 0.9 levels of the output pulse.

The Laplace transform of $(2/\tau_1)\sin^2(\pi t/\tau_1)$ function is equal to [7]

$$F_0(s) = \frac{1}{\tau_1}\frac{4(\pi/\tau_1)^2}{s\left[s^2 + 4(\pi/\tau_1)^2\right]}. \tag{6.4}$$

Using the shift theorem [8], one finds that the Laplace transform of the sine-squared pulse will be

$$F_1(s) = \frac{1}{\tau_1} \frac{4(\pi/\tau_1)^2}{s\left[s^2 + 4(\pi/\tau_1)^2\right]} (1 - e^{-s\tau_1}). \tag{6.5}$$

The derivative dv_0/dt includes two (one positive, another negative) sine-squared pulses shifted with respect to each other by $\tau - \tau_1$. Using the shift theorem again, and considering that

$$\mathbf{L}[dv_0/dt] = sV_o(s), \tag{6.6}$$

where $V_o(s)$ is the Laplace transform of $v_o(t)$, one obtains that

$$V_o(s) = \frac{1}{\tau_1} \frac{4(\pi/\tau_1)^2}{s^2\left[s^2 + 4(\pi/\tau_1)^2\right]} (1 - e^{-s\tau_1}) \left[1 - e^{-s(\pi-\tau_1)}\right]. \tag{6.7}$$

This $V_o(s)$ may be considered as an impulse response of a nonrealizable (using lumped parameters) transfer function. But if, following [5], its real, $\mathrm{Re}V_o(j\omega)$, and imaginary, $\mathrm{Im}V_o(j\omega)$, parts are expanded in infinite products, then the finite number of terms in these products may be used for approximation of $\mathrm{Re}V_o(j\omega)$ and $\mathrm{Im}V_o(j\omega)$. This further approximation allows one to obtain a realizable transfer function $H_0(s)$ of the pulse-forming network excited by the input unit impulse. As one will see below, the obtained transfer function includes zeros on the $j\omega$-axis only, and the denominator order may be controlled.

Then, if $v_i(t)$ is a step function, the required transfer function is equal to $H_1(s) = sH_0(s)$. If $v_i(t)$ is sin Ωt, the required transfer function is $H_2(s) = (s^2 + \Omega^2)H_0(s)$.

6.2.2 Realization Requirements

We remind here the requirements that should be imposed on the algebraic ratio if it represents the transfer function of a reactance network loaded by resistor [9,10].

Let $F(s)$ be the Laplace transform of network impulse response, and $F(j\omega) = P(\omega) + jQ(\omega)$. One can find a realizable network if $F(s)$ allows the approximation

$$H_a(s) = \frac{N(s)}{D(s)} = \frac{A_1 s^\mu N_1(s)}{D_2(s) + A_2 s D_1(s)}. \tag{6.8}$$

Here $H_a(s)$ is the ratio of two polynomials, and deg $N(s) \le$ deg $D(s)$, $N_1(s)$, $D_1(s)$, and $D_2(s)$ are even polynomials with zeros on the $j\omega$-axis only, that is, $N_1(s) = \prod_{l=1}^{\nu} \left(s^2 + c_l^2\right)$, $D_2(s) = \prod_{k=1}^{n} \left(s^2 + a_k^2\right)$, $sD_1(s) = s\prod_{k=1}^{n-1} \left(s^2 + b_k^2\right)$, and a_k and b_k are alternating (for odd denominator polynomials both product superscripts in $D_1(s)$ and $D_2(s)$ should be equal). A_1 and A_2 are the positive real constants. Indeed, the function (Equation 6.8) allows then realization as a transfer function of a reactance network loaded by resistor [9,10].

The conditions that should be imposed on real and imaginary parts of realizable $H_a(s)$ are obtained the following way. To avoid separate discussions for even and odd μ let us consider the function

$$H_{a1}(s) = s^{-\mu} H_a(s) = \frac{A_1 N_1(s)}{D_2(s) + A_2 s D_2(s)}. \tag{6.9}$$

Then one can write that

$$\mathrm{Re}\, H_{a1}(j\omega) = \frac{A_1 \prod_{l=1}^{\nu} \left(c_l^2 - \omega^2\right) \prod_{k=1}^{n} \left(a_k^2 - \omega^2\right)}{D_2^2(-\omega^2) + A_2^2 \omega^2 D_1(-\omega^2)} \tag{6.10}$$

and

$$\operatorname{Im} H_{a1}(j\omega) = -\frac{A_1 A_2 \omega \prod_{l=1}^{\nu}\left(c_l^2 - \omega^2\right)\prod_{k=1}^{n-1}\left(b_k^2 - \omega^2\right)}{D_2^2(-\omega^2) + A_2^2\omega^2 D_1(-\omega^2)}. \tag{6.11}$$

Hence, if $F(s) = s^{\mu_1}F_1(s)$ and $F_1(j\omega) = P_1(\omega) + jQ_1(\omega)$, then (1) both $\mathrm{Re}F_1(j\omega)$ and $\mathrm{Im}F_1(j\omega)$ should be approximated by the ratios so that each of them includes 2ν simple zeros located at $\omega = \pm\gamma_k$, (2) the approximating ratio for $\mathrm{Re}F_1(j\omega)$ includes, in addition, $2n$ simple zeros located at $\omega = \pm\alpha_k$, (3) the approximating ratio for $\mathrm{Im}F_1(j\omega)$ includes, in addition, $2(n-1)$ simple zeros located at $\omega = \pm\beta_k$, and (4) α_k and β_k should alternate. Then the approximating ratio

$$H_a(s) = \frac{A_1 s^{\mu_1}\prod_{l=1}^{\nu}\left(s^2 + \gamma_l^2\right)}{\prod_{k=1}^{n}\left(s^2 + \alpha_k^2\right) + A_2 s\prod_{k=1}^{n-1}\left(s^2 + \beta_k^2\right)}, \tag{6.12}$$

where $\mu_1 = \mu$, and $a_k = \alpha_k$, $b_k = \beta_k$, and $c_l = \gamma_l$, may be used as a transfer function for the network realization. The constants A_1 and A_2 can be calculated equating $F(s)$ and $H_a(s)$ at two points of the s-plane (the most convenient points are usually $s = 0$ and $s = j$, the final choice may be decided by computer calculations).

One can see that if $\deg N(s) \le \deg D(s) - 2$, the multiplication of the numerator in Equation 6.12 by s or by $(s^2 + \Omega^2)$ does not violate the realization condition.

Representing thus obtained $H_a(s)$, for example, as

$$H_a(s) = \frac{[A_1 s^{\mu} N_1(s)]/D_2(s)}{\{[A_2 s D_1(s)]/D_2(s)\} + 1} = -\frac{y_{21}(s)}{y_{22}(s) + 1}, \tag{6.13}$$

one finds $y_{21}(s) = -A_1 s^{\mu} N_1(s)/D_2(s)$ and $y_{22}(s) = A_2 s N_1(s)/D_2(s)$. These two parameters are sufficient to realize the network [9,10] within a constant multiplier for $y_{21}(s)$.

6.2.3 Second Approximation Step: Approximation for Realization

Now we find the approximation to the output response Laplace transform (Equation 6.7). This step is leading to realizable transfer functions. We rewrite $V_o(s)$ as

$$V_o(s) = \frac{16(\pi/\tau_1)^2}{\tau_1 s^2\left[s^2 + 4(\pi/\tau_1)^2\right]} \cdot e^{-\frac{s\pi}{2}} \cdot \sinh\left(\frac{s\tau_1}{2}\right) \cdot \sinh\left[\frac{s(\pi - \tau_1)}{2}\right]. \tag{6.14}$$

Substituting $s = j\omega$ one can find that

$$\mathrm{Re}V_o(j\omega) = \frac{16(\pi/\tau_1)^2}{\tau_1\omega^2\left[4(\pi/\tau_1)^2 - \omega^2\right]} \cdot \cos\left(\frac{\omega\pi}{2}\right) \cdot \sin\left(\frac{\omega\tau_1}{2}\right) \cdot \sin\left[\frac{\omega(\pi - \tau_1)}{2}\right] \tag{6.15}$$

and

$$\mathrm{Im}V_a(j\omega) = -\frac{16(\pi/\tau_1)^2}{\tau_1\omega^2\left[4(\pi/\tau_1)^2 - \omega^2\right]} \cdot \sin\left(\frac{\omega\pi}{2}\right) \cdot \sin\left(\frac{\omega\tau_1}{2}\right) \cdot \sin\left[\frac{\omega(\pi - \tau_1)}{2}\right]. \tag{6.16}$$

Taking into consideration that [11]

$$\sin x = x\prod_{k=1}^{\infty}\left(1-\frac{x^2}{k^2\pi^2}\right) \tag{6.17}$$

and

$$\cos x = \prod_{k=0}^{\infty}\left(1-\frac{x^2}{(2k+1)^2\pi^2}\right), \tag{6.18}$$

one can see that $\mathrm{Re}V_o(j\omega)$ has real zeros at $\alpha_k = \pm(2k+1)$, where $k = 0, 1, 2, \ldots$, and $\mathrm{Im}V_a(j\omega)$ has zeros at $\beta_k = \pm 2k$, where $k = 0, 1, 2, \ldots$ as well. These zeros are alternating on ω-axis. In addition, both $\mathrm{Re}V_o(j\omega)$ and $\mathrm{Im}V_o(j\omega)$ have common zeros located at $\gamma_k = \pm 2\pi\ k$, where $k = 2, 3, 4, \ldots$ (the zeros for $k = 0, 1$ are canceling the poles), and $\gamma_k = \pm 2\pi k/(\pi - \tau_1)$, where $k = 1, 2, 3 \ldots$ (the zero for $k = 0$ is canceling the pole). Then, in accordance with the previous part, to approximate $V_a(s)$ one may use the algebraic ratio

$$H_a(s) = \frac{N(s)}{D(s)} = \frac{A_1\prod_{k=2}^{\nu_1}\left[s^2+(2\pi k/\tau_1)^2\right]\prod_{k=1}^{\nu_2}\left[s^2+(2\pi k/(\pi-\tau_1))^2\right]}{\prod_{k=0}^{n_1}\left[s^2+((2k+1)^2\right]+A_2 s\prod_{k=1}^{n}(s^2+4k^2)} \tag{6.19}$$

To have the approximation with sufficient number of terms in the products, it is rational to find the spectral function

$$|V_o(j\omega)| = \left|\frac{16(\pi/\tau_1)^2\sin(\omega\tau_1/2)\sin[\omega(\pi-\tau_1)/2]}{\tau_1\omega^2\left[4(\pi/\tau_1)^2-\omega^2\right]}\right| \tag{6.20}$$

and evaluate the spectrum bandwidth ω_m. The zeros included in the approximating products should be located in the spectrum bandwidth, and they will determine the subscripts in the products of Equation 6.21. The following two possibilities should be considered for n_1 and n. When $n_1 = n - 1$, then the denominator degree is odd, when $n_1 = n$ then this degree is even. If ω_m is chosen, then, in case of the odd degree $2n \approx \omega_m$ (i.e., $n \approx \omega_m/2$), and in case of the even degree $2n_1 + 1 \approx \omega_m$ (i.e., $n_1 \approx \omega_m - 0.5$). Hence, the odd degree allows one to obtain the approximating transfer function with more "dense" location of real and imaginary part zeros in the spectrum bandwidth, which usually results in a better approximation in the time domain as well.

From the other side, to obtain the simplest transfer function one has to choose $\nu_1 = 1$. Then, the first product in the numerator is equal to unit and will not include any multipliers. Considering that $(2\pi/\tau_1)$ represents now the upper approximation frequency, one can find the other superscripts in the approximating products from the approximate equalities

$$\begin{cases}\dfrac{2\pi}{\tau_1}\approx\dfrac{2\pi\nu_2}{\pi-\tau_1}\\[2ex]\dfrac{2\pi}{\tau_1}\approx 2n \text{ or } (2n_1+1)\end{cases} \tag{6.21}$$

that give $\nu_2 \approx (\pi - \tau_1)/\tau_1$ and $n \approx \dfrac{\pi}{\tau_1}$; $n_1 \approx \dfrac{\pi}{\tau_1} - \dfrac{1}{2}$. This results in

$$H_a(s) = \frac{A_1\prod_{k=1}^{\nu_2}\left[s^2+\left(\frac{2\pi k}{\pi-\tau_1}\right)^2\right]}{\prod_{k=0}^{n_1}\left[s^2+(2k+1)^2\right]+A_2 s\prod_{k=1}^{n}(s^2+4k^2)}. \tag{6.22}$$

The constants A_1 and A_2 can be found equating $V_o(s)$ given by Equation 6.7 and this $H_a(s)$ at two points of the s-plane. Using Equation 6.13 one can then find the realizable $y_{22}(s)$ and $y_{21}(s)$ required for the network synthesis.

6.2.4 Pulse-Forming Networks with Non-Delayed Output Pulses

As examples, we synthesize the pulse-forming networks that should have the output response, which is the quasi-rectangular pulse with duration of fronts equal to $\tau_1 = \pi/3$, so that the rise and fall times are $\pi/6$ each. To obtain the simplest network, one has to put $\nu_2 = 2$, $n = 3$, and $n_1 = 2$. Then $V_o(s)$ for the case of $v_i = \delta(t)$ will be approximated by the ratio

$$H_a(s) = \frac{A_1(s^2+9)(s^2+36)}{[(s^2+1)(s^2+9)(s^2+25) + A_2 s(s^2+4)(s^2+16)(s^2+36)]}. \tag{6.23}$$

Equating Equations 6.7 and 6.23 at $s = 0$ and $s = j$ one can find that $A_1 = 1.453$ and $A_2 = 0.152$. Hence, the network transfer function for this case is equal to

$$H_0(s) = \frac{1.453(s^2+9)(s^2+36)}{[(s^2+1)(s^2+9)(s^2+25) + 0.152s(s^2+4)(s^2+16)(s^2+36)]}. \tag{6.24}$$

The impulse response, $h_o(t)$, of this transfer function is shown in Figure 6.2 (the initial output pulse, $v_o(t)$, is also shown for comparison).

If $v_i = u(t)$ then the network transfer function for this case of excitation is equal to

$$H_1(s) = \frac{1.453s(s^2+9)(s^2+36)}{[(s^2+1)(s^2+9)(s^2+25) + 0.152s(s^2+4)(s^2+16)(s^2+36)]}. \tag{6.25}$$

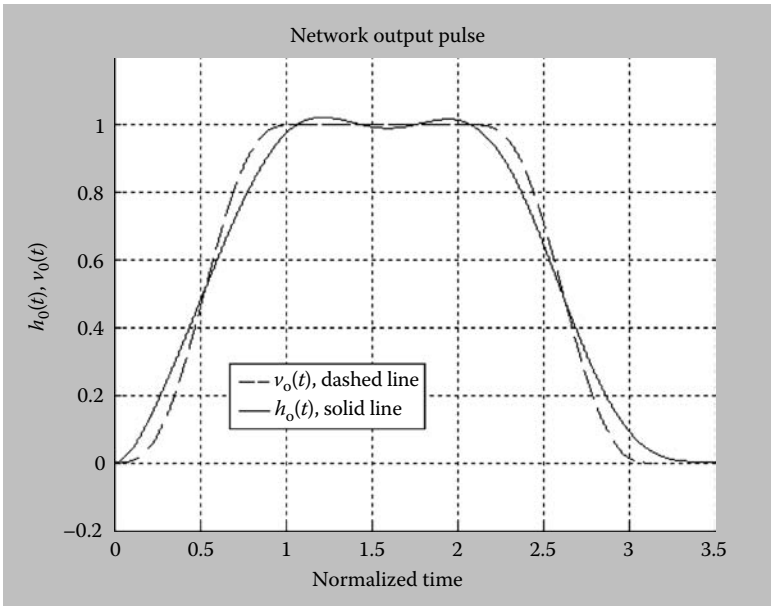


FIGURE 6.2 Pulse-forming network output response.

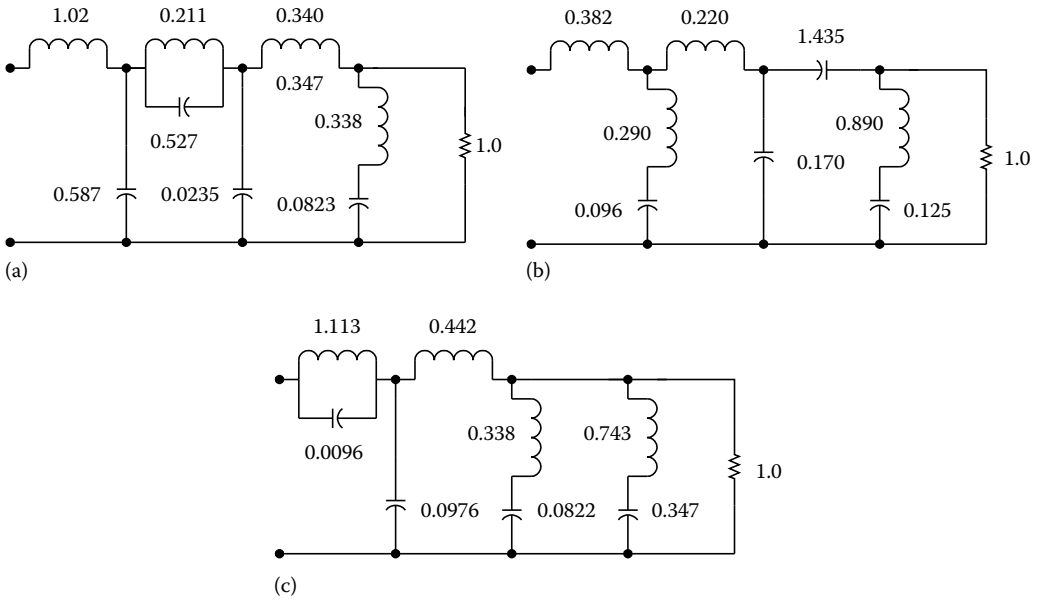


FIGURE 6.3 Pulse-forming networks: (a) $v_i = \delta(t)$, (b) $v_i = u(t)$, and (c) $v_i = \sin \Omega t$.

Finally, assume that $v_i = \sin 4t$. Then the zeros that are due to the input signal will coincide with one pair of zeros in the denominator odd part. This will simplify the realization. The transfer function for this case of excitation is equal to

$$H_2(s) = \frac{1.453(s^2+9)(s^2+16)(s^2+36)}{[(s^2+1)(s^2+9)(s^2+25) + 0.152s(s^2+4)(s^2+16)(s^2+36)]}. \tag{6.26}$$

The realizations of the functions (Equations 6.24 through 6.26) were obtained dividing their numerators and denominators by $(s^2+1)(s^2+9)(s^2+25)$. This gives

$$y_{22}(s) = \frac{0.152s(s^2+4)(s^2+16)(s^2+36)}{(s^2+1)(s^2+9)(s^2+25)}, \tag{6.27}$$

common for all three networks. Then this parameter was realized by standard procedures [9,10] taking into consideration the location of $y_{12}(s)$ zeros for each network. The results of these realizations are shown in Figure 6.3.

One can also verify that for the network of Figure 6.3b, the voltage source efficiency coefficient representing the ratio of the output pulse amplitude to input step amplitude is equal to 0.743, that is, nearly 1.5 times higher than with the usual approach [1,2].

If the required duration of pulse is τ_0 s, and the load resistor is R ohms, then each value of inductance should be multiplied by $\tau_0 R/\pi$ and the value of each capacitor by $\tau_0/(R\pi)$.

6.2.5 Pulse-Forming Networks with Delayed Output Pulse

The above-mentioned voltage source efficiency coefficient increases when the output pulse is delayed [12]. Yet, this delay should be used judiciously. The increase of source efficiency results in a more complicated circuit. In addition, the rise and fall times increase, that is, slew rate of the fronts

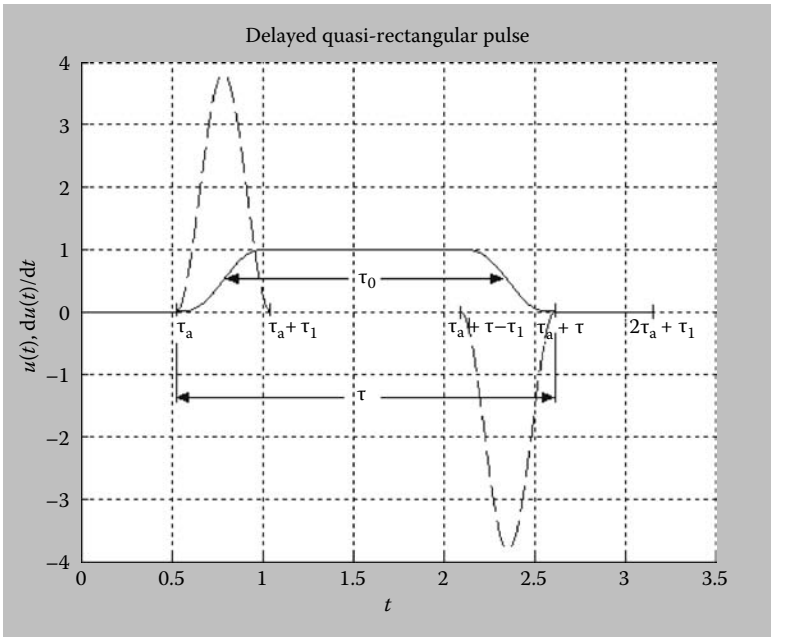


FIGURE 6.4 Delayed quasi-rectangular pulse.

deteriorates. It is possible to prepare a table that helps to visualize a possible trade-off between the delay, the circuit complexity, and the voltage source efficiency coefficient. The calculations given below follow the above-developed pattern.

Let the required output response be a delayed symmetric pulse $u(t)$ (Figure 6.4). Assume that the derivative, $\mathrm{d}u/\mathrm{d}t$, of this response is described by delayed positive and delayed negative semi-periods of sine-squared function. These semi-periods determine the duration of fronts that is equal to τ_1. Then $\mathrm{d}u/\mathrm{d}t$ can be described by the following equations:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \begin{cases} 0 & 0 \le t < \tau_a \\ A\sin^2\left[\frac{\pi}{\tau_1}(t-\tau_a)\right] & \tau_a \le t < \tau_a + \tau_1 \\ 0 & \tau_a + \tau_1 \le t < \tau_a + \tau - \tau_1 \\ -A\sin^2\left[\frac{\pi}{\tau_1}(t+\tau_1-\tau-\tau_a)\right] & \tau_a + \tau - \tau_1 \le t < \tau_a + \tau \\ 0 & t > \tau_a + \tau \end{cases} \tag{6.28}$$

The normalized time is defined by the condition

$$2\tau_a + \tau = \pi. \tag{6.29}$$

To obtain the normalized magnitude of unity for the output pulse amplitude, one has to choose the value of A from the equation

$$A\int_{\tau_a}^{\tau_a+\tau_1} \sin^2\left[\frac{\pi}{\tau_1}(t-\tau_a)\right]\mathrm{d}t = 1. \tag{6.30}$$

This gives

$$A = 2/\tau_1. \tag{6.31}$$

This value is again equal to the maximal slew rate of the fronts, and the approximate rise and fall times can be evaluated as

$$\tau_r = \tau_f = 1/A = \tau_1/2. \tag{6.32}$$

The value of $\tau_0 = \tau - \tau_1$ will be considered as the pulse duration at the level of 0.5.

We repeat here, for convenience, the Laplace transform of $(2/\tau_1)\sin^2(\pi t/\tau_1)$ [7] that is equal to

$$F_0(s) = \frac{1}{\tau_1}\frac{4(\pi/\tau_1)^2}{s\left[s^2 + 4(\pi/\tau_1)^2\right]}. \tag{6.33}$$

The derivative du/dt includes two (one positive, another negative) delayed sine-squared pulses shifted with respect to each other by $\tau_0 = \tau - \tau_1$. Using the shift theorem [8], and, considering that $\mathbf{L}[du/dt] = sU(s)$, where $U(s)$ is the Laplace transform of $u(t)$, one obtains that

$$U(s) = \frac{4(\pi/\tau_1)^2}{\tau_1}\frac{e^{-s\tau_a}(1 - e^{-s\tau_1})\left[1 - e^{-s(\tau-\tau_1)}\right]}{s^2\left[s^2 + 4(\pi/\tau_1)^2\right]}. \tag{6.34}$$

This result can be rewritten as

$$U(s) = \frac{16(\pi/\tau_1)^2}{\tau_1}\frac{e^{-s\left(\tau_a+\frac{\tau}{2}\right)}\sinh\left(\frac{s\tau_1}{2}\right)\sinh\left[\frac{s(\tau-\tau_1)}{2}\right]}{s^2\left[s^2 + 4(\pi/\tau_1)^2\right]}. \tag{6.35}$$

For the normalized time defined by Equation 6.29, one can rewrite this result as

$$U(s) = \frac{16(\pi/\tau_1)^2}{\tau_1}\frac{e^{-s\frac{\pi}{2}}\sinh\left(\frac{s\tau_1}{2}\right)\sinh\left[\frac{\xi\pi s}{2}\right]}{s^2\left[s^2 + 4(\pi/\tau_1)^2\right]}, \tag{6.36}$$

where $\xi = \frac{1}{1+(2\tau_a+\tau)/\tau_0} < 1$. Using the expansion [11] of

$$\sinh(x) = x\prod_{k=1}^{\infty}\left(1 + \frac{x^2}{4k^2\pi^2}\right) \tag{6.37}$$

and approximating

$$\sinh\left(s\frac{\tau_1}{2}\right) \approx \frac{s\tau_1}{2}\left(1 + \frac{s^2\tau_1^2}{4\pi^2}\right) \tag{6.38}$$

and

$$\sinh\left(\frac{\xi\pi s}{2}\right) \approx \frac{\xi\pi s}{2}\prod_{l=1}^{\nu}\left[1 + \frac{s^2}{(2l/\xi)^2}\right] \tag{6.39}$$

(the choice of ν is discussed below), one can write

$$U(s) \approx Ce^{-s\frac{\pi}{2}} \prod_{l=1}^{\nu} \left[s^2 + \left(\frac{2l}{\xi} \right)^2 \right], \tag{6.40}$$

where $C = (\xi\pi)^{2\nu+1} / \left(4^{\nu} \prod_{l=1}^{\nu} l^2 \right)$. The approximation (Equation 6.38) defines the maximum frequency, ω_m, where Equation 6.40 is valid on the $j\omega$-axis, namely,

$$(2\pi)/\tau_1 \leq \omega_m < (4\pi)/\tau_1 \tag{6.41}$$

(the calculations show that using $(3\pi)/\tau_1$ for the right-hand side is usually sufficient).

In this problem, the output pulse $u(t)$ is usually the result of application of a step voltage to the network input. Then, if the Laplace transform of this input voltage is written as

$$U_1(s) = 1/(Ks), \tag{6.42}$$

then the realized value of K (it will be defined below) gives us the voltage source efficiency coefficient.

The ratio $U(s)/U_1(s)$ is a nonrealizable transfer function. Yet, using the previously developed approach, it is possible to find further approximation to Equation 6.40 so that the resulting ratio will be realizable.

We calculate $\mathrm{Re}U(j\omega)$ and $\mathrm{Im}U(j\omega)$ for the approximation (Equation 6.40). Substituting $s = j\omega$ one finds that

$$\mathrm{Re}U(j\omega) \approx C \cos\left(\omega \frac{\pi}{2} \right) \prod_{l=1}^{\nu} \left[\left(\frac{2l}{\xi} \right)^2 - \omega^2 \right] \tag{6.43}$$

and

$$\mathrm{Im}U(j\omega) \approx -C \sin\left(\omega \frac{\pi}{2} \right) \prod_{l=1}^{\nu} \left[\left(\frac{2l}{\xi} \right)^2 - \omega^2 \right]. \tag{6.44}$$

Taking into consideration Equations 6.18 and 6.19, one can see that $\mathrm{Re}U(j\omega)$ has real zeros at $\alpha_k = \pm(2k+1)$ with $k = 0, 1, 2, \ldots$ and $\mathrm{Im}U(j\omega)$ has zeros at $\beta_k = \pm 2k$ with $k = 0, 1, 2, \ldots$ as well. These zeros are alternating (or interlacing). In addition, both $\mathrm{Re}U(j\omega)$ and $\mathrm{Im}U(j\omega)$ have common zeros located at $\gamma_l = \pm 2l/\xi$ with $l = 1, 2, \ldots, \nu$. Then, in accordance with the previous part, to approximate $U(s)$ in the bandwidth defined by Equation 6.41, one can use the algebraic ratio

$$U_a(s) = \frac{A_1 \prod_{l=1}^{\nu} [s^2 + (2l/\xi)^2}{\prod_{k=0}^{n_1} \left[s^2 + (2k+1)^2 \right] + A_2 s \prod_{k=1}^{n} (s^2 + 4k^2)}. \tag{6.45}$$

The coefficients A_1 and A_2 may be found equating $U_a(s)$ given by Equation 6.45 to $U(s)$ given by Equation 6.36 at two points of the s-plane.

The Equation 6.41 defines the maximum approximation bandwidth ω_m for Equation 6.45 as well. To obtain simpler networks, it is better to be closer to the left-hand side of this inequality. Then, one can find ν from the approximate condition

$$2\nu/\xi \approx 2\pi/\tau_1. \tag{6.46}$$

Then, n_1 may have two values, namely $n_1 = n$ for even-order denominators and $n_1 = n - 1$ for odd-order ones. This gives two approximate equations $2\pi/\tau_1 \approx 2n$ and $2\pi/\tau_1 \approx 2n_1 + 1$ for the even- and odd-part superscripts of the denominator.

Finally, to obtain the transfer function for the input step voltage, one has to multiply the approximating ratio $U_a(s)$ by Ks. This finally gives

$$H(s) = \frac{KA_1 s \prod_{l=1}^{\nu} \left[s^2 + (2l/\xi)^2\right]}{\prod_{k=0}^{n_1} \left[s^2 + (2k+1)^2\right] + A_2 s \prod_{k=1}^{n} (s^2 + 4k^2)}. \tag{6.47}$$

Representing this $H(s) = N(s)/[D_2(s) + A_2 s D_1(s)]$ as in Equation 6.13, one obtains $y_{21}(s) = -N(s)/D_2(s)$ and $y_{22}(s) = A_2 s D_1(s)/D_2(s)$, two parameters that are sufficient to realize the network within a constant multiplier for y_{21}.

The maximum value of K is defined by the Fialkov condition [9], which specifies that for the unbalanced two ports, the transfer function numerator coefficients should be less or equal to the denominator coefficients for the corresponding degrees of s. The finalized voltage source efficiency coefficient depends on the realized value of A_1.

As an example, we consider the network with the step response that approximates the delayed quasi-rectangular pulse with the following parameters: the delay time $\tau_a = \pi/6$, the duration of each front is $\tau_1 = \pi/6$, the pulse duration calculated at the level of 0.5 of the output amplitude is $\tau_0 = \pi/2$, and the total pulse duration is $\tau = (2\pi)/3$. The reader may notice that this pulse was used in Figure 6.4 to illustrate the problem. The dashed line shows the pulse shape corresponding to the initial assumptions described by Equation 6.28.

Using the pulse parameters one can find that $\xi = (\tau - \tau_1)/\pi = 0.5$. Then, using Equations 6.41 and 6.46, one can find that $\nu = 3$, $n = 6$, and $n_1 = 6$. This allows one to find the ratio that approximates the Laplace transform (Equation 6.36) as

$$U_a(s) = \frac{A_1(s^2 + 16)(s^2 + 64)(s^2 + 144)}{\prod_{k=0}^{6} \left[s^2 + (2k+1)^2\right] + A_2 s \prod_{k=1}^{6} (s^2 + 4k^2)}. \tag{6.48}$$

Equating Equations 6.36 and 6.48 at $s = 0$ and $s = j$, one can find $A_1 = 1.945 \times 10^5$ and $A_2 = 1.329$. The original, $u_a(t)$, corresponding to this transform is shown in Figure 6.5. The network transfer function, thus, is given by the ratio

$$H(s) = \frac{K \times 1.945 \times 10^5 s(s^2 + 16)(s^2 + 64)(s^2 + 144)}{\prod_{k=0}^{6} \left[s^2 + (2k+1)^2\right] + 1.329 s \prod_{k=1}^{6} (s^2 + 4k^2)}. \tag{6.49}$$

Finally, one divides the numerator and denominator of $H(s)$ by $\prod_{k=0}^{6} \left[s^2 + (2k+1)^2\right]$ and obtains

$$y_{22} = \frac{1.329 s \prod_{k=1}^{6} (s^2 + 4k^2)}{\prod_{k=0}^{6} \left[s^2 + (2k+1)^2\right]} \tag{6.50}$$

and

$$y_{21} = \frac{K \times 1.945 \times 10^5 s(s^2 + 16)(s^2 + 64)(s^2 + 144)}{\prod_{k=0}^{6} \left[s^2 + (2k+1)^2\right]}. \tag{6.51}$$

Using the Fialkov condition, one can find that the maximal achievable voltage source efficiency coefficient is $K = 0.984$.

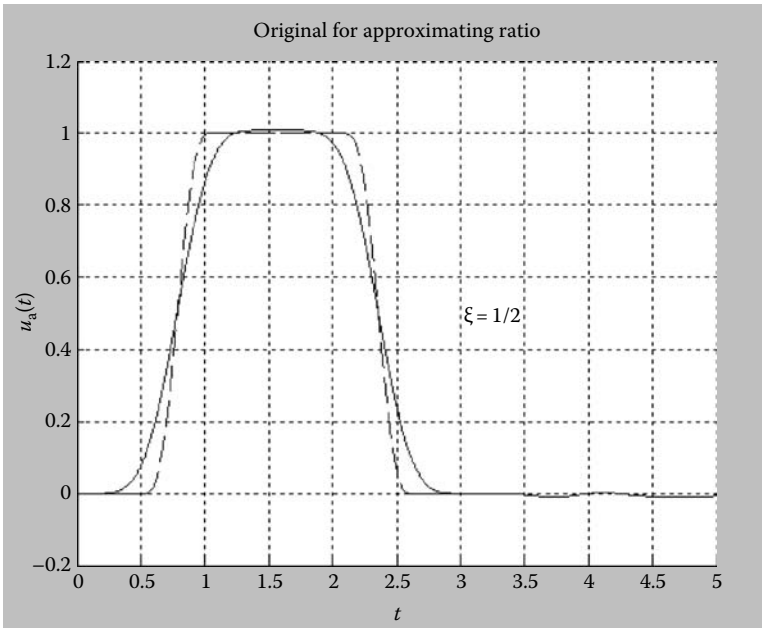


FIGURE 6.5 Step response of the pulse-shaping network.

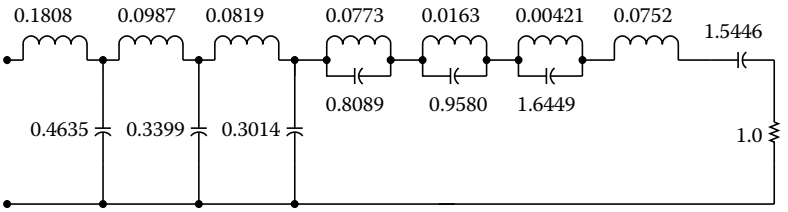


FIGURE 6.6 Realization of the pulse-shaping network.

A realization of this transfer function is shown in Figure 6.6. It starts by realization of common zeros of y_{22} and y_{21} (i.e., by realization of poles of y_{22}^{-1} at $s=0$, $s\to\infty$, $s=\pm j4$, $s=\pm j8$, and $s=\pm j12$) and finishes by the Cauer form to realize other zeros of y_{21} located at $s\to\infty$.

One can verify that the realized value of A_1 is equal to 1.911×10^5. Hence, the realized value of K will be less than the maximum achievable one and equal to 0.967.

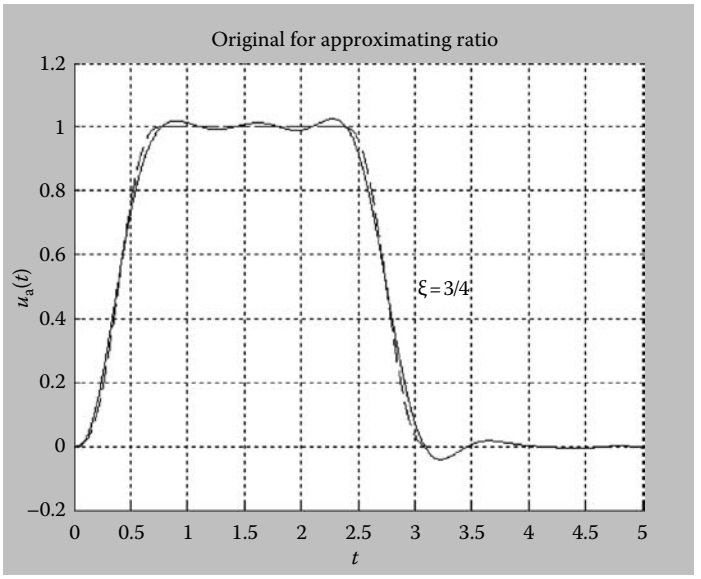
The normalized duration of pulse was $\pi/2$, the normalized value of the load was unit. If the required duration of pulse is τ_r s and the load resistor is R ohms, then each value of inductance in the network of Figure 6.6 should be multiplied by $2\tau_r R/\pi$ and the value of each capacitor by $2\tau_r/(R\pi)$.

Table 6.1 shows that with increase of delay, the maximum voltage source efficiency coefficient increases as well so that the output pulse amplitude can be larger than the amplitude of the input step voltage. The table was calculated for the pulses with the same ratio $\tau_1/\tau_0=1/3$ when the shape of pulse in absolute time is preserved, and in all cases $\nu=3$. We also indicated the required transfer function parameters as well.

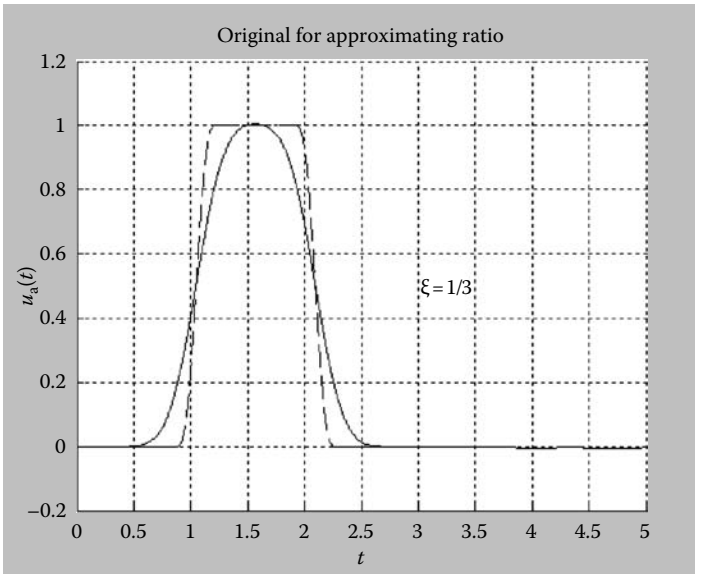
To better visualize the deflection of the realized shapes of the output pulses from the initially assumed shapes, Figure 6.7a and b shows the output pulses for the first and the third lines, respectively, of the

TABLE 6.1 Delay Influence on Voltage Source Efficiency

$\xi = \tau_0/\pi$	τ_1/π	τ_a/π	K	$n = n_1$	A_1	A_2
3/4	1/4	0	0.663	4	1.625×10^{2}	9.462
1/2	1/6	1/6	0.984	6	1.945×10^{5}	13.292
1/3	1/9	5/18	1.476	9	2.673×10^{11}	19.886



(a)



(b)

FIGURE 6.7 Deterioration of output pulse with increasing delay: (a) $\xi = 3/4$ (no delay) and (b) $\xi = 1/3$, $\tau_a/\pi = 5/18$.

Table 6.1 as well. One can see that with increasing delay, the slew rate of the fronts deteriorates and differs from that initially assumed. The duration of the pulse at the level of 0.5 is preserved.

If the zeros of y_{12} coincide with the zeros of y_{22}, the realization is simplified. This was the case in the considered example (it is easy to verify that it is also valid for the transfer function obtained from the third line of Table 6.1). Usually, it can be easily done for one zero only.

6.3 Transfer Functions of Wideband Amplifiers

Wideband amplifiers with a monotonic step response are frequently used for amplification of pulse signals. Their investigation was started by Elmore [13], and the basic results of this work are still used in the design [14–16]. Elmore's approach allowed one to obtain the relationships between the delay time, rise time, and the number of stages for such amplifiers. Yet, this approach introduces a strong limitation: it requires that the poles of transfer functions should be located on the negative real axis of the *s*-plane [13]. The synthesis of the filters with maximally flat delay (Bessel filters) [17] and the Gaussian-response filters [18] removed this restriction. But the transfer functions of these filters were found indirectly, from the requirements in the frequency domain. There is no immediate relationship between the delay-to-rise-time ratio (which is the main parameter in the design of the amplifiers with monotonic step response) and the filter order. One has to simulate their step responses using the available tables of transfer function poles [19], evaluate the delay-to-rise-time ratio, and then decide upon the required order of the transfer function.

It was shown in [4] how to find the transfer function numerically with a monotonic step response starting from the requirements to the delay-to-rise-time ratio. Later on, this problem was solved in [20], yet the results of this work are not easy to apply for design of wideband amplifier transfer functions. Finally, follow the approach proposed in [5], it was shown in [21,22] how to find the transfer functions with monotonic step response and optimized delay-to-rise-time ratio.

Here, we consider a particular case of the solution given in [21]. The amplifier transfer functions tabulated below are obtained when the amplifier impulse response is approximated by the period of $\sin^2(\pi\, t/\tau)$ function (here $\tau \le \pi$). This period is symmetrically located within the interval $0 \le t \le \pi$ of the normalized time t. Then, one repeats the above outlined approximation steps. One finds the Laplace transform of the output response. Approximating further this Laplace transform by a suitable transfer function, one can find an all-pole realizable transfer function of ''maximal for chosen τ order.''

The results of this synthesis procedure are represented by the table of step-response parameters for the transfer functions from the fourth to tenth order. The parameters are also compared with Bessel and Gaussian filter transfer functions of the corresponding order. The table of corresponding transfer functions poles is also given. We do not give the realization of the tabulated transfer functions by LC two-port networks loaded by resistor. The transfer functions of wideband amplifiers are usually realized by active networks, and their realization can be found elsewhere [23]. The proposed method is easily extended on the synthesis of delay networks [24].

6.3.1 Parameters of the Step Response and Its Laplace Transform

Here, the amplifier impulse response, $h(t)$, is approximated by the period of the $\sin^2(\pi\, t/\tau)$ function (here $\tau \le \pi$) symmetrically located within the interval $0 \le t \le \pi$ of the normalized time t, i.e.,

$$h(t) = \begin{cases} 0 & 0 \le t \le (\pi - \tau)/2 \\ A \sin^2 \frac{\pi}{\tau}\left(t - \frac{\pi}{2} + \frac{\tau}{2}\right) & \frac{\pi-\tau}{2} \le t \le \frac{\pi+\tau}{2} \\ 0 & (\pi + \tau)/2 \le t \le \pi \end{cases}. \tag{6.52}$$

The amplitude, A, of the impulse response will be chosen from the normalization condition of the step response, $u(t)$, so that

$$\int_{(\pi-\tau)/2}^{(\pi+\tau)/2} h(t)\mathrm{d}t = 1. \tag{6.53}$$

One finds that $A = 2/\tau$. Then, the delay time $\tau_d = \pi/2$, the rise time $\tau_r = 1/A$, and the delay-to-rise-time ratio, ρ, is equal to

$$\rho = \tau_d/\tau_r = \pi/\tau. \tag{6.54}$$

Using the tables [7] and the delay theorem [8], one finds the Laplace transform of Equation 6.52 as

$$F(s) = \frac{4(\pi/\tau)^2(1-\mathrm{e}^{-\tau s})}{\tau\, s\left[s^2+(2\pi/\tau)^2\right]}\mathrm{e}^{-\left(\frac{\pi-\tau}{2}\right)s}. \tag{6.55}$$

In the following, we will also need the numerical values of this transform at $s = 0$ and $s = j$. One can find that $F(0) = 1$ and $F(j) = -j\frac{8(\pi/\tau)^2\sin(\tau/2)}{\tau\left[(2\pi/\tau)^2-1\right]}$.

6.3.2 Transfer Function Approximation

The Laplace transform (Equation 6.55) can be rewritten as

$$F(s) = \frac{8(\pi/\tau)^2\sinh(\tau s/2)}{\tau s\left[s^2+(2\pi/\tau)^2\right]}\mathrm{e}^{-s\frac{\pi}{2}}. \tag{6.56}$$

Using the expansion (Equation 6.37) for the function of sinh x, and approximating the infinite product with two terms, as

$$\sinh(s\tau/2) \approx \frac{s\tau}{2}\left(1+\frac{s^2\tau^2}{4\pi^2}\right), \tag{6.57}$$

one finds that $F(s)$ can be approximated as

$$F(s) \approx \mathrm{e}^{-s\frac{\pi}{2}}. \tag{6.58}$$

The validity region of this approximation is determined by the first term in the discarded part of the approximation (Equation 6.57). On the $j\omega$-axis, the border for the maximal frequency, ω_m, where the approximation (Equation 6.58) is still valid, is thus, determined by the inequalities

$$(2\pi)/\tau < \omega_m < (4\pi)/\tau. \tag{6.59}$$

On the $j\omega$-axis, the function (Equation 6.58) becomes

$$F_a(j\omega) = \cos(\omega\pi/2) - j\sin(\omega\pi/2). \tag{6.60}$$

The functions $\cos(\omega\pi/2)$ and $\sin(\omega\pi/2)$ can also be represented as infinite products as in Equations 6.17 and 6.18, one can write that

$$F_a(j\omega) = \prod_{i=0}^{\infty}\left(1 - \frac{\omega^2}{(2i+1)^2}\right) - j\frac{\omega\pi}{2}\prod_{i=1}^{\infty}\left(1 - \frac{\omega^2}{4i^2}\right). \tag{6.61}$$

Using the finite number of terms in the products, one can approximate $F(j\omega)$ as

$$F_a(j\omega) \approx \prod_{i=0}^{n_1}\left(1 - \frac{\omega^2}{(2i+1)^2}\right) - j\frac{\omega\pi}{2}\prod_{i=1}^{n}\left(1 - \frac{\omega^2}{4i^2}\right). \tag{6.62}$$

This result allows one to find a realizable transfer function

$$H_a(s) = \frac{A_1}{\prod_{i=0}^{n_1}\left[s^2 + (2i+1)^2\right] + A_2 s\prod_{i=1}^{n}(s^2 + 4i^2)} \tag{6.63}$$

that approximates $F(s)$ given by Equation 6.56.

The following two possibilities should be considered for n_1 and n. If $n_1 = n$, then the denominator degree is even, if $n_1 = n - 1$ then the denominator degree is odd. Indeed, if these conditions are satisfied, and A_1 and A_2 are positive, then the denominator of $H_a(s)$ is a Hurwitz polynomial [9]. The denominator roots will be located in the left half of the s-plane, and this $H_a(s)$ can be realized as a cascade connection of first- and second-order low-pass stages.

The approximation will be the best if one takes the maximum possible value of n_1 or n in Equation 6.63. Substituting $s = j\omega$ in Equation 6.63, one can find that

$$H_a(j\omega) = \frac{A_1\prod_{i=0}^{n_1}\left[(-\omega^2 + (2i+1)^2\right] - jA_2\omega\prod_{i=1}^{n}(-\omega^2 + 4i^2)}{\left(\prod_{i=0}^{n_1}\left[\left(-\omega^2 + (2i+1)^2\right)\right]^2 + A_2^2\omega^2\left(\prod_{i=1}^{n}(-\omega^2 + 4i^2)\right)^2\right)} \tag{6.64}$$

Comparing Equations 6.64 and 6.62, one can see that to have the maximal order of the transfer function one has to take the maximal value of n_1 or n in Equation 6.62 as well. But the choice of n_1 and n is defined, as it follows from Equation 6.59, by the following inequalities

$$\begin{cases} (2\pi)/\tau < 2n_1 + 1 < (4\pi)/\tau \\ (2\pi)/\tau < 2n < (4\pi)/\tau \end{cases}. \tag{6.65}$$

One has to choose the largest value satisfying one of these inequalities. If this largest value is $n_1 = m$, then one takes $n = m$ as well and obtains the transfer function of even order. If this largest value is $n = m$, then one takes $n_1 = m - 1$, and obtains the transfer function of odd order.

Finally, to find A_1, one equates $H_a(s)$ and $F(s)$ at $s = 0$. This gives $A_1 = \prod_{i=1}^{n_1}(2i+1)^2$. Equating $H_a(s)$ and $F(s)$ at $s = j$ one finds

$$A_2 = \frac{\tau[(2\pi/\tau) - 1]A_1}{8(\pi/\tau)^2\sin(\tau/2)\prod_{i=1}^{n}\left[(2i)^2 - 1\right]}.$$

6.3.3 Example of Transfer Function Design

Assume that we design an amplifier transfer function with $\rho = 1.6$. This requires that $\tau = 5\pi/8$. Substituting this value in Equation 6.65, one obtains the inequalities

$$\begin{cases} 3\frac{1}{5} < 2n_1 + 1 < 6\frac{2}{5} \\ 3\frac{1}{5} < 2n < 6\frac{2}{5} \end{cases}. \tag{6.66}$$

From these inequalities, one finds that $n = 3$ and $n_1 = 2$. Hence, the realizable transfer function

$$H_a(s) = \frac{A_1}{\prod_{i=0}^{i=2}\left[s^2 + (2i+1)^2\right] + A_2\prod_{i=1}^{i=3}\left[s^2 + (2i)^2\right]} \tag{6.67}$$

of the seventh-order approximating $F_a(s)$ (and, hence $F(s)$ given by Equation 6.56) is found. It has the maximal possible order for the required τ. The constants A_1 and A_2 can be calculated equating $F(s)$ given by Equation 6.56 and $H_a(s)$ given by Equation 6.67 at the points $s = 0$ and $s = j$. One finds that $A_1 = 225$ and $A_2 = 0.1522$. Hence, the realizable transfer function

$$H_a(s) = \frac{225}{(s^2+1)(s^2+9)(s^2+25) + 0.1522s(s^2+4)(s^2+16)(s^2+36)} \tag{6.68}$$

is approximating $F(s)$ for $\tau = 5\pi/8$.

The impulse and step responses for this transfer function are shown in Figure 6.8. One can see that the impulse response of Equation 6.68 is close to the response (Equation 6.52) for $t = 5\pi/8$, and the step

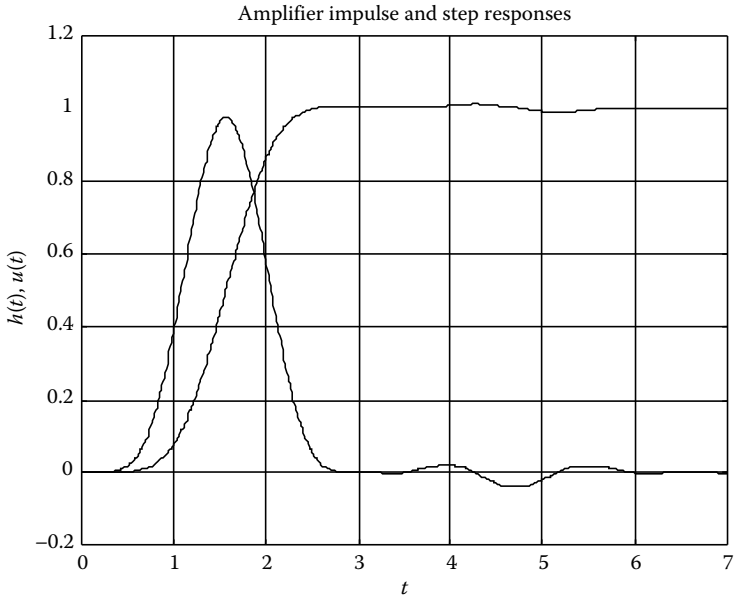


FIGURE 6.8 Example of impulse and step responses.

TABLE 6.2 Step Response Parameters

D	τ	ρ	ρ_s	$Ov(\%)$	$Un(\%)$	ρ_b	$Ov_b(\%)$	ρ_g
4	π	1.00	1.06	1.4	1.9	0.91	0.8	0.87
5	$7\pi/8$	1.15	1.20	0.3	1.5	1.06	0.8	0.98
6	$3\pi/4$	1.33	1.36	1.2	0.8	1.20	0.6	1.11
7	$5\pi/8$	1.60	1.50	1.3	0.9	1.32	0.5	1.21
8	$\pi/2$	2.00	1.63	1.6	0.0	1.42	0.3	1.31
9	$15\pi/32$	2.13	1.74	1.5	0.6	1.52	0.2	1.43
10	$7\pi/16$	2.29	1.86	1.2	0.0	1.63	0.1	1.50

TABLE 6.3 Transfer Functions Poles

D	Poles
4	$-0.664 \pm j2.228$; $-1.103 \pm j0.669$
5	$-0.604 \pm j3.174$; $-1.032 \pm j1.508$; -1.140
6	$-0.532 \pm j4.119$; $-0.949 \pm j2.293$; $-1.259 \pm j0.730$
7	-0.487 ± 5.095; $-0.860 \pm j3.206$; $-1.215 \pm j1.526$; -1.320
8	$-0.460 \pm j6.083$; $-0.806 \pm j4.172$; $-1.148 \pm j2.446$; $-1.230 \pm j0.857$
9	$-0.404 \pm j7.060$; $-0.678 \pm j5.096$; $-0.969 \pm j3.167$; $-1.344 \pm j1.304$; -1.880
10	$-0.400 \pm j8.059$; $-0.677 \pm j6.100$; $-0.980 \pm j4.204$; $-1.272 \pm j2.498$; $-1.312 \pm j0.874$

response is practically monotonic (the overshoot is 1.3% and the undershoot is 0.9%). The realized delay-to-rise-time ratio obtained in simulation is $\rho_s = 1.5$.

6.3.4 Tabulated Results

The results of our derivation are summarized in two tables. Table 6.2 gives the step-response parameters of the derived transfer functions. In Table 6.2, D is the degree of the transfer function denominator, ρ is the estimate of delay-to-rise-time ratio obtained from Equation 6.54, ρ_s is the delay-to-rise-time ratio obtained in simulations of approximating transfer functions, Ov is the step-response overshoot obtained in simulations, and Un is the undershoot also obtained in simulations.

These overshoot and undershoot columns deserve some discussion. Let us return to the example, the step response of the transfer function with $D = 7$ (Figure 6.8). As one can see, the step response $u(t)$ does not have the overshoot as it usually appears in ''normal'' transfer functions, that is, during application of the pulse signal. Here one has a small postpulse wave located in the interval $\pi < t < 2\pi$. One can verify that this is valid for all proposed transfer functions. Finally, Table 6.2 gives the comparison of the proposed transfer functions with the transfer functions of the Bessel (ρ_b, Ov_b) and Gaussian (ρ_g, no overshoot) filter transfer functions. Table 6.3 gives the poles of the transfer functions.

6.4 Forming a Sinusoidal Pulse

Recently, most of the industrial emphasis in ultra-wideband technology (UWB) has focused on the short range, high-data-rate applications. However, due to its low power properties, impulse UWB is also suitable for low-power, low-data-rate applications [25]. One of the challenges in such low-power systems is how to efficiently generate the pulses. Several common pulse waveforms were studied, with the relation between their spectral characteristics and waveform parameters pointed out [26,27]. The approximation method described here allows one to synthesize the pulse-forming reactance networks loaded by resistors, and these networks may represent, in applications to UWB systems, model of transmitting antennas.

Comparing the complexities of the network forming pulses of different shapes gives additional information for UWB system design. The developed synthesis approach is suitable for many pulse forms used in UWB systems, but for the reasons of space limitations, we consider only an example of synthesis of the network forming sinusoidal pulse with sinusoidal envelope [28]. In [29], one can find information on the pulse-forming network for monocycle pulse.

6.4.1 Required Transfer Function

Let the required output pulse $u(t)$ be a sinusoidal oscillation of a radian frequency ν with a sinusoidal envelope, $\sin \Omega t$ that has the finite duration of $\Omega t_d = \pi$ in the normalized time (Figure 6.9), that is,

$$u(t) = \begin{cases} g(t) \sin \nu t = \sin \Omega t \sin \nu t & 0 \le \Omega t \le \pi \\ 0 & \Omega t > \pi \end{cases}. \tag{6.69}$$

It is assumed here that Equation 6.69 is the impulse or a step response. Then using the shift theorem [8], one can find that the Laplace transform of the envelope $g(t)$ is

$$G(s) = \frac{\Omega(1 + e^{-\pi s})}{s^2 + \Omega^2} \tag{6.70}$$

Considering that $\sin \nu t = (e^{j\nu} - e^{-j\nu})/(2j)$ and using the theorem of complex translation [8], one writes that the Laplace transform of $u(t)$ is equal to

$$F(s) = \frac{\Omega}{2j}\left[\frac{1 + e^{-\pi(s-j\nu)}}{(s - j\nu)^2 + \Omega^2} - \frac{1 + e^{-\pi(s+j\nu)}}{(s - j\nu)^2 + \Omega^2}\right]. \tag{6.71}$$

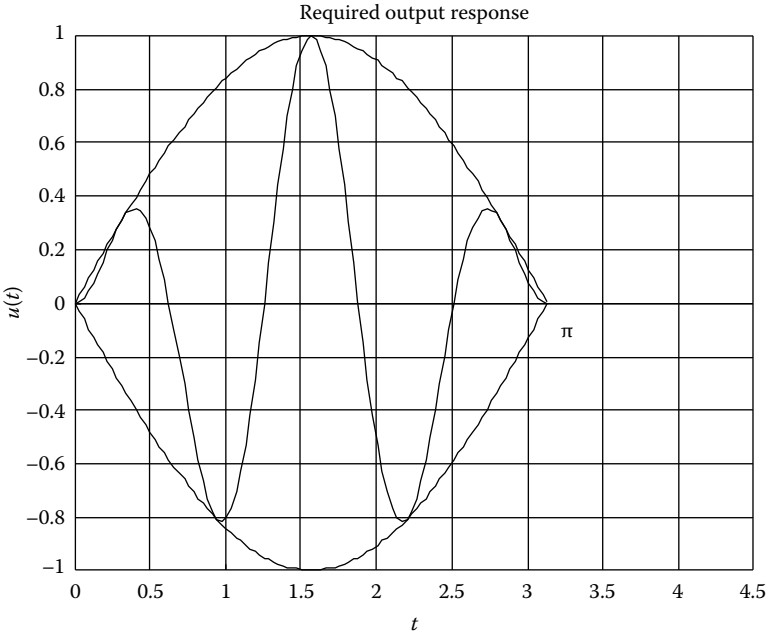


FIGURE 6.9 Required output response with sinusoidal envelope.

After some simple algebra, one obtains that

$$F(s) = \frac{2\nu\Omega s\left[1-(-1)^{\nu+1}e^{-\pi s}\right]}{\left[s^2+(\nu+\Omega)^2\right]\left[s^2+(\nu-\Omega)^2\right]}. \tag{6.72}$$

Considering that $\Omega = 1$ and assuming that ν is an integer, one can rewrite that

$$F(s) = \frac{2\nu s\left[1-(-1)^{\nu+1}e^{-\pi s}\right]}{\left[s^2+(\nu+1)^2\right]\left[s^2+(\nu-1)^2\right]}. \tag{6.73}$$

It is possible to consider two cases. If $\nu+1$ is even, then Equation 6.73 becomes

$$F_{ev}(s) = \frac{4\nu s e^{-\frac{\pi}{2}s}\sinh\left(\frac{\pi}{2}s\right)}{\left[s^2+(\nu+1)^2\right]\left[s^2+(\nu-1)^2\right]}. \tag{6.74}$$

If $\nu+1$ is odd, then Equation 6.73 becomes

$$F_{odd}(s) = \frac{4\nu s e^{-\frac{\pi}{2}s}\cosh\left(\frac{\pi}{2}s\right)}{\left[s^2+(\nu+1)^2\right]\left[s^2+(\nu-1)^2\right]}. \tag{6.75}$$

Then, Equations 6.74 and 6.75 do not belong to a class of realizable transfer functions if they are the transforms of impulse responses. Their multiplication by s (for the case if they are transforms of the step responses) does not give realizable functions either. One has again to truncate the decompositions into products for transcendental functions, and to find the required final approximation.

6.4.2 Approximation for Realization

Now, we find the approximations to the Laplace transforms (Equations 6.74 and 6.75) resulting in the realizable transfer functions. Let us restrict ourselves by the last case ($\nu+1$ is odd) only. The other case can be considered in a similar way. Substituting $s = j\omega$, one can find that

$$\mathrm{Re}F_{odd}(j\omega) = \frac{2\nu\omega\sin\left(\frac{\pi}{2}\omega\right)\cos\left(\frac{\pi}{2}\omega\right)}{\left[-\omega^2+(\nu+1)^2\right]\left[-\omega^2+(\nu-1)^2\right]} \tag{6.76}$$

and

$$\mathrm{Im}F_{odd}(j\omega) = \frac{2\nu\omega\cos^2\left(\frac{\pi}{2}\omega\right)}{\left[-\omega^2+(\nu+1)^2\right]\left[-\omega^2+(\nu-1)^2\right]}. \tag{6.77}$$

Taking Equations 6.17 and 6.18 into consideration, one can see that $\mathrm{Re}F_{odd}(j\omega)$ and $\mathrm{Im}F_{odd}(j\omega)$ have common zeros at $\omega = 0$; and $\omega = \pm1$; ±3; ±5; … (due to the common multiplier $\cos(\pi\omega/2)$), yet the zeros $\omega = \pm(\nu-1)$ and $\omega = \pm(\nu+1)$ of this multiplier are not common zeros (they are canceled by the poles). Then, the $\mathrm{Re}F_{odd}(j\omega)$ has the zeros at $\omega = 0$; ±2; ±4; ±6; … (due to the multiplier $\sin(\pi\omega/2)$), and the $\mathrm{Im}F_{odd}(j\omega)$ has the zeros at $\omega = \pm1$; ±3; ±5; … (due to the second multiplier of $\cos(\pi\omega/2)$). Hence in the last two groups, the zeros are alternating. One can see that the ratio $H_{odd}(s)$ for approximation of $F(s)$ should have the form

$$H_{\text{odd}}(s) = \frac{A_1 s \prod_{k=0,\, k\neq\nu-1,\, k\neq\nu+1}^{k=\rho} \left[s^2 + (2k+1)^2\right]}{\prod_{k=0}^{n} \left[s^2 + (2k+1)^2\right] + A_2 s \prod_{k=1}^{n_1} (s^2 + 4k^2)}. \tag{6.78}$$

Indeed, if one calculates $\text{Re}H_{\text{odd}}(j\omega)$ and $\text{Im}H_{\text{odd}}(j\omega)$ and compare them with $\text{Re}F_{\text{odd}}(j\omega)$ and $\text{Im}F_{\text{odd}}(j\omega)$, one finds that the zeros of $\text{Re}H_{\text{odd}}(j\omega)$ and $\text{Im}H_{\text{odd}}(j\omega)$ have the same properties: there are common zeros (due to the numerator) and alternating zeros (due to even and odd polynomials in the denominator). The number of zeros is, of course, finite, and is defined by the order of $H_{\text{odd}}(s)$. In practice it is sufficient to choose $\rho = \nu + 2$. Then one chooses $n = n_1$ for even and $n = n_1 - 1$ for odd polynomials in the denominator. The maximal of these two numbers $\max(n, n_1) \geq \rho$. The approximation becomes better when the order of $H_{\text{odd}}(s)$ increases. The constants A_1 and A_2 can be found equating $F_{\text{odd}}(s)$ and $H_{\text{odd}}(s)$ at two points of the s-plane. The case of $F_{\text{ev}}(s)$ is treated in a similar way.

6.4.3 Example

As an example, we consider the network that should have the impulse response

$$u(t) = \begin{cases} \sin t \sin 5t & 0 \leq t \leq \pi \\ 0 & t > \pi \end{cases}. \tag{6.79}$$

(It is this pulse that was shown in Figure 6.9.) We have $\nu + 1 = 6$, that is, the even case. It is easy to find that the Laplace transform of Equation 6.79 is equal to

$$F_{\text{ev}}(s) = \frac{20 s e^{-\frac{\pi}{2}s} \sinh\left(\frac{\pi}{2}s\right)}{(s^2 + 4^2)(s^2 + 6^2)}. \tag{6.80}$$

The real and imaginary parts of this function are

$$\text{Re}F_{\text{ev}}(j\omega) = -\frac{20\omega \cos\left(\frac{\pi}{2}\omega\right) \sin\left(\frac{\pi}{2}\omega\right)}{(-\omega^2 + 16)(-\omega^2 + 36)} \tag{6.81}$$

and

$$\text{Im}F_{\text{ev}}(j\omega) = \frac{20\omega \sin^2\left(\frac{\pi}{2}\omega\right)}{(-\omega^2 + 16)(-\omega^2 + 36)}. \tag{6.82}$$

In the interval $0 \leq \omega \leq 8$, the functions $\text{Re}F_{\text{ev}}(j\omega)$ and $\text{Im}F_{\text{ev}}(j\omega)$ have double common zero at $\omega = 0$, and simple common zeros at $\omega = \pm 2$ and $\omega = \pm 8$. The zeros at $\omega = \pm 4$ and $\omega = \pm 6$ are not common zeros, after cancelation of poles at $\omega = \pm 4$ and $\omega = \pm 6$ they disappear from $\text{Re}F_{\text{ev}}(j\omega)$ and left in $\text{Im}F_{\text{ev}}(j\omega)$ only. The function $\text{Re}F_{\text{ev}}(j\omega)$ has zeros at $\omega = \pm 1; \pm 3; \pm 5; \pm 7$. These zeros are alternating with the zeros $\omega = 0\ \pm 2; \pm 4; \pm 6; \pm 8$ that occur in $\text{Im}F_{\text{ev}}(j\omega)$ due to the second $\sin(\pi\omega/2)$ multiplier in the numerator of Equation 6.82. As a result, one can use the ratio

$$H_{\text{ev}}(s) = \frac{A_1 s^2 (s^2 + 4)(s^2 + 64)}{[(s^2 + 1)(s^2 + 9)(s^2 + 25)(s^2 + 49) + A_2 s (s^2 + 4)(s^2 + 16)(s^2 + 36)(s^2 + 64)]}. \tag{6.83}$$

The coefficients A_1 and A_2 can be found, for example, equating Equations 6.81 and 6.84 at $s = j4$ and $s = j5$. One finds that $A_1 = 2.65762$ and $A_2 = 0.13288$.

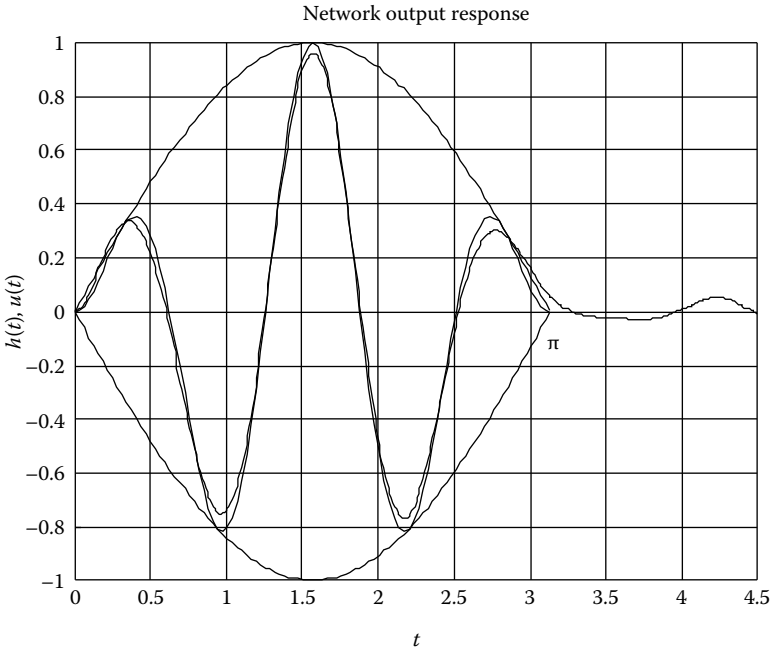


FIGURE 6.10 Network output response.

Because Equation 6.80 is the impulse response, one takes directly

$$H(s) = H_{ev}(s). \tag{6.84}$$

Figure 6.10 shows the inverse Laplace transform $h(t) = \mathbf{L}^{-1}(H(s))$ of this transfer function that is compared with $u(t)$ given by Equation 6.79.

Dividing the numerator and denominator of $H(s) = H_{ev}(s)$ by the odd part of the denominator one writes that

$$y_{21} = -\frac{A_1 s}{A_2(s^2+16)(s^2+36)} \tag{6.85}$$

and

$$y_{22} = \frac{(s^2+1)(s^2+9)(s^2+25)(s^2+49)}{A_2 s(s^2+4)(s^2+16)(s^2+36)(s^2+64)}. \tag{6.86}$$

The LC-realization using these two parameters is shown in Figure 6.11. The realization is done within a constant multiplier for y_{21}. It starts by realization of the private poles of y_{22} at $s = 0$, $s = \pm 2$, and $s = \pm j8$. The residual admittance

$$y'_{22} = \frac{2.850s(s^2+24.936)}{(s^2+16)(s^2+36)} \tag{6.87}$$

is realized, first, by subtraction of the poles of $1/(y'_{22})$ at $s = 0$ and $s = \infty$. This provides the zeros at $s = 0$ and $s = \infty$ common for y_{21} and y'_{22}. Then, after subtraction of the series elements corresponding to these

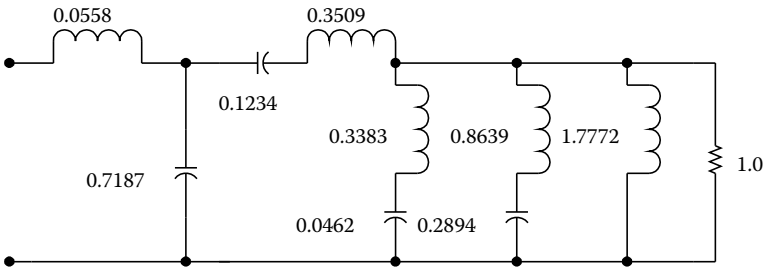


FIGURE 6.11 Realization of the pulse-shaping network.

poles of $1/(y'_{22})$, the final residual conductance is realized to provide two additional zeros of y_{21} at $s = \infty$. One can find that this realization provides the level of y_{21} that is 3.553 times higher than that given by Equation 6.85.

One can see that multiplication of Equation 6.83 by s gives the realizable function as well. The realization of the network providing Equation 6.79 as the step response is not given here for the reason of space limitation.

6.5 Summary

The synthesis of a pulse-forming network with output response that is close to a rectangular shape pulse requires two step approximation procedure. The derivative of this output response is described by positive and delayed negative semi-periods of the sine-squared function. This is the first step of approximation procedure. The real and imaginary parts of Laplace transform of thus approximated output pulse are expanded in infinite products. Then, using a finite number of terms in these products, one obtains an algebraic ratio that approximates this Laplace transform. This is the second step of approximation procedure.

If the input excitation is the unit impulse function, the obtained ratio is directly the required transfer function realizable by a reactance network loaded by resistor. Two closely connected realizable transfer functions (providing the same output) are obtained by simple multiplication of the previously obtained transfer function by s (if the input excitation is the unit step function) or by $(s^2 + \Omega^2)$ (if the input excitation is the sinusoid of unit amplitude and of frequency Ω).

The synthesis of a linear network shaping a nonperiodic (in the considered case quasi-rectangular) pulse from the sinusoidal voltage is using the short period of time after turning on this input voltage. The zeros of transfer function are used to reject the periodic solution and create zero solution during the network steady-state operation. The efficiency of this pulse-shaping circuit may be insufficient; yet, conceptually the synthesis procedure is not very much different from the synthesis of pulse-shaping networks using for this purpose the discharge of a capacitor.

The two-step approximation resulting in the realizable reactance networks may be extended on the wide variety of output pulses including wideband amplifier transient responses and sinusoidal pulses of finite duration and with wide variety of envelop shape.

The method is simple, and the output response is approximated with a small error. The network complexity to obtain better approximations can be easily controlled. The approximation precision as well as the network complexity can be improved using computer methods [30].

The proposed procedure for synthesis of pulse-forming networks is based on approximation of the meaningful part of the output signal spectrum. The zeros of real and imaginary parts of approximated and approximating functions coincide in the meaningful part of the spectrum, and their amplitude values are close to each other. This closeness of spectrums provides good approximation in the time domain.

References

1. G. N. Glasoe, J. V. Lebacqz, eds., The pulse-forming network, in *Pulse Generators*, Chapter 6, Boston Technical Publishers, Boston, MA, pp. 175–224, 1964.
2. P. W. Smith, *Transient Electronics*, Wiley, Chichester, England, 2002.
3. B. E. Thomson, The synthesis of a network to have a sine-squared impulse response, *IEE Proceedings*, 99, Pt. III, 373–376, 1952.
4. J. Jess, H. W. Schussler, On design of pulse-forming networks, *IEEE Transactions on Circuit Theory*, CT-12, 393–400, 1965.
5. P. N. Matkhanov, *Time-Domain Synthesis of Reactance Two-Ports*, Energia Publishing House, Leningrad, USSR, 1970 (in Russian).
6. I. M. Filanovsky, P. N. Matkhanov, Synthesis of reactance networks shaping a quasi-rectangular pulse, *IEEE Transactions on Circuits and Systems*, Part II, 52, 242–245, 2005.
7. V. A. Ditkin, A. P. Prudnikov, *Integral Transforms and Operational Calculus*, Pergamon Press, Oxford, 1965.
8. R. V. Churchill, *Operational Mathematics*, 2nd edn., McGraw-Hill, New York, 1958.
9. N. Balabanian, *Network Synthesis*, Prentice-Hall, Englewood Cliffs, NJ, 1958.
10. L. Weinberg, *Network Analysis and Synthesis*, McGraw-Hill, New York, 1962.
11. P. J. Nahin, *An Imaginary Tale*, Chapter 6, Princeton University Press, Princeton, NJ, 1998.
12. I. M. Filanovsky, P. N. Matkhanov, Synthesis of a pulse-forming reactance network shaping a quasi-rectangular delayed output pulse, *IEEE Transactions on Circuits and Systems*, Part II, 51, 190–194, 2004.
13. W. C. Elmore, The transient response of damped linear networks with particular regard to wideband amplifiers, *Journal of Applied Physics*, 19, 55–63, 1948.
14. D. L. Feucht, *Handbook of Analog Circuit Design*, Chapter 8, Academic Press, Inc., San Diego, CA, pp. 318–379, 1990.
15. T. H. Lee, Risetime, delay and bandwidth, in *The Design of CMOS Radio-Frequency Integrated Circuits*, Chapter 7, Cambridge University Press, New York, pp. 167–174, 1998.
16. P. Staric, E. Margan, *Wideband Amplifiers*, Springer, Dordrecht, The Netherlands, 2006.
17. L. Storch, Synthesis of constant time delay ladder networks using Bessel polynomials, *Proceedings of IRE*, 42, 1666–1675, 1954.
18. M. Dishal, Gaussian-response filter design, *Electrical Communication*, 36, 3–26, 1959.
19. C. S. Lindquist, *Active Network Design*, Steward & Sons, Long Beach, CA, 1977.
20. G. C. Temes, The prolate filter: An ideal low-pass filter with optimum step-response, *Journal of Franklin Institute*, 293(2), 77–103, 1972.
21. I. M. Filanovsky, A new class of wide-band amplifiers with monotonic step response, *IEEE Transactions on Circuits and Systems*, Part I, 50, 569–571, 2003.
22. I. M. Filanovsky, P. N. Matkhanov, On the synthesis of wideband multistage amplifiers with monotonic step response, *Circuits, Systems and Signal Processing*, 21(6), 567–580, 2002.
23. J. Davidse, *Analog Electronic Circuit Design*, Prentice Hall, Englewood Cliffs, NJ, 1991.
24. I. M. Filanovsky, P. N. Matkhanov, Synthesis of time delay networks approximating the pulse response described by an integer power of a sinusoid over its semiperiod, *Analog Integrated Circuits and Signal Processing*, 28, 81–88, 2001.
25. S. A. P. Haddad, Ultra low-power biomedical signal processing, PhD thesis, Delft University of Technology, The Netherlands, 2006.
26. X. Chen, S. Kiaei, Monocycle shapes for ultra wideband system, *IEEE Proceedings of the International Symposium on Circuits and Systems, ISCAS'2002*, 1, 597–600, 2002.
27. B. Parr, ByungLok Cho, K. Wallace, Zhi Ding, A novel ultra-wideband pulse design algorithm, *IEEE Communication Letters*, 7, 219–221, 2003.

28. I. M. Filanovsky, P. N. Matkhanov, On synthesis of a reactance network having the step response described by a sinusoid with a given envelope, *IEEE Proceedings of the International Symposium on Circuits and Systems, ISCAS'2001*, 1, 603–606, 2001.
29. I. M. Filanovsky, X. Dong, On synthesis of reactance network shaping a monocycle pulse, *IEEE Proceedings of the 47th Midwest Symposium on Circuits and Systems,* 1, 101–104, 2004.
30. M. Vucic, G. Molnar, Design of systems with prescribed impulse response based on second-order cone programming, *IEEE Proceedings of the International Symposium on Circuits and Systems, ISCAS'2007*, 3311–3313, 2007.

II

The VLSI Circuits

John Choma, Jr.
University of Southern California

7 **Fundamentals of Digital Signal Processing** *Roland Priemer* **7**-1
Introduction • References

8 **Digital Circuits** *John P. Uyemura, Robert C. Chang, and Bing J. Sheu* **8**-1
MOS Logic Circuits • References • Transmission Gates • References

9 **Digital Systems** *Festus Gail Gray, Wayne D. Grover, Josephine C. Chang, Bing J. Sheu, Roland Priemer, Kung Yao, and Flavio Lorenzelli* **9**-1
Programmable Logic Devices • References • Clocking Schemes • References • MOS Storage Circuits • References • Microprocessor-Based Design • References • Systolic Arrays • References

10 **Data Converters** *Bang-Sup Song and Ramesh Harjani* **10**-1
Digital-to-Analog Converters • References • Analog-to-Digital Converters • Acknowledgment • References • Further Information

7

Fundamentals of Digital Signal Processing

7.1 Introduction 7-1
Fourier Series for Continuous-Time Periodic Signals • Example and Discussion • Discrete-Time Signals • Discrete Fourier Transform • Examples and Discussion • DFT Applications • Fast Fourier Transform • Continuous-Time Fourier Transform • Examples and Discussion • Discrete-Time Fourier Transform • DTFT Model • Sampling Theorem • DTFT Properties • Example and Discussion • Linear and Time Invariant Discrete-Time Systems • Convolution • Stability • Frequency Response • Examples and Discussion • Ideal Digital Filters • z-Transform • Bilateral z-Transform Properties • z-Plane • Transfer Function • Unilateral z-Transform • Conclusion
References 7-36

Roland Priemer
University of Illinois at Chicago

7.1 Introduction

Real-time and off-line processing of continuous-time signals by digital means (digital signal processing [DSP]) has become a viable processing mode over analog means for several reasons, some of which are digital signal processors, microcontrollers, and microprocessors. These are inexpensive, programmable, reproducible, consume low power, have computing speeds suitable for signals with bandwidths beyond base band video, and can operate in extreme environments. Some broad application areas of DSP are automotive industry, consumer electronics, communication systems, and medical systems.

Since real-world signals are continuous in time, the technologist must properly interpret results of processing signals by digital means. This requires an understanding of the origin of and relationship among the basic tools used for DSP. We shall first consider the Fourier series (FS) concept for continuous-time periodic signals. Everything that follows will be based on this concept.

This chapter is intended for those who have some experience with the material generally covered in a first course on continuous-time signals and systems, and would like a brief introduction to the fundamentals of DSP.

7.1.1 Fourier Series for Continuous-Time Periodic Signals

Given is a real and periodic signal $x(t)$, which satisfies

$$x(t) = x(t + T_0)$$

for some period T_0 and all t. One period of $x(t)$ is $x_p(t) = x(t)$, for $t_0 \le t < t_0 + T_0$ and $x_p(t) = 0$ for $t < t_0$ and $t \ge t_0 + T_0$, where t_0 is arbitrary. The periodic signal $x(t)$ can be written as the periodic extension of $x_p(t)$, which is

$$x(t) = \sum_{r=-\infty}^{+\infty} x_p(t - rT_0)$$

Let us approximate $x(t)$ by a sum of sinusoidal functions given by

$$\hat{x}(t) = \hat{a}_0 + \sum_{k=1}^{\infty} \hat{a}_k \cos(k\omega_0 t) + \sum_{k=1}^{\infty} \hat{b}_k \sin(k\omega_0 t)$$

where ω_0, called the fundamental frequency, is found with $\omega_0 = 2\pi/T_0$ rad/s and $f_0 = 1/T_0$ Hz. The frequency ω of each sinusoidal function is $\omega = k\omega_0$. Therefore, the approximation $\hat{x}(t)$ is also periodic with period T_0. The approximation error is

$$e(t) = x(t) - \hat{x}(t)$$

We choose the coefficients $\hat{a}_0$, $\hat{a}_k$, and $\hat{b}_k$ to minimize the mean square error given by

$$\varepsilon^2\left(\hat{a}_0, \hat{a}_k, \hat{b}_k\right) = \frac{1}{T_0} \int_0^{T_0} e^2(t)\mathrm{d}t$$

which is a quadratic function of $\hat{a}_0$, $\hat{a}_k$, and $\hat{b}_k$. Denote the $\hat{a}_0$, $\hat{a}_k$, and $\hat{b}_k$ that minimize ε^2 by a_0, a_k, and b_k, respectively. Setting the partial derivatives of ε^2 with respect to $\hat{a}_0$, $\hat{a}_k$, and $\hat{b}_k$ to zero gives

$$a_0 = \frac{1}{T_0} \int_0^{T_0} x(t)\mathrm{d}t$$

which is the average value of $x(t)$, and

$$a_k = \frac{2}{T_0} \int_0^{T_0} x(t) \cos(k\omega_0 t)\mathrm{d}t$$

$$b_k = \frac{2}{T_0} \int_0^{T_0} x(t) \sin(k\omega_0 t)\mathrm{d}t$$

If $x(t)$ satisfies the Dirichlet conditions, which are that $x(t)$ must

1. Have a finite number of extrema in any given time interval
2. Have a finite number of discontinuities in any given time interval
3. Be absolutely integrable over a period

then setting $\hat{a}_0 = a_0$, $\hat{a}_k = a_k$, and $\hat{b}_k = b_k$ gives $\varepsilon^2(a_0, a_k, b_k) = 0$, and we write

$$x(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(k\omega_0 t) + \sum_{k=1}^{\infty} b_k \sin(k\omega_0 t) \tag{7.1}$$

which is called the FS representation of $x(t)$, and the a_0, a_k, and b_k are the trigonometric FS coefficients of $x(t)$. If $x(t)$ is discontinuous at $t = t_d$, then for $t = t_d$, the FS converges to the average value of $x(t)$ about $t = t_d$, which is

$$\frac{x(t_d^-) + x(t_d^+)}{2} \tag{7.2}$$

Furthermore, if $x(t)$ is everywhere finite and continuous, then we have the stronger result that $e(t) = 0$.

For further development, it is more convenient to apply Euler's identity, which is

$$e^{j\alpha} = \cos(\alpha) + j\sin(\alpha) \tag{7.3}$$

for any real number α, and then $x(t)$ becomes

$$x(t) = a_0 + \sum_{k=1}^{\infty} a_k \frac{e^{jk\omega_0 t} + e^{-jk\omega_0 t}}{2} + \sum_{k=1}^{\infty} b_k \frac{e^{jk\omega_0 t} - e^{-jk\omega_0 t}}{j2}$$

Let

$$X_k = \frac{a_k}{2} - j\frac{b_k}{2}$$

Since $a_k = a_{-k}$ and $b_k = -b_{-k}$, we have $X_{-k} = X_k^*$, and therefore, we can write

$$x(t) = \sum_{k=-\infty}^{+\infty} X_k e^{jk\omega_0 t} \tag{7.4}$$

where $X_0 = a_0$ and

$$X_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} x(t) e^{-jk\omega_0 t} dt \tag{7.5}$$

for any t_0. This representation for $x(t)$ is called the complex FS, and the X_k are called the complex FS coefficients. We say that $x(t)$ and X_k are a FS pair, which is denoted by $x(t) \leftrightarrow X_k$. In general, the FS coefficients X_k are complex. However, if $x(t)$ is an even time function, then X_k is real, which means that $x(t)$ can be represented with only cosine terms, and if $x(t)$ is an odd time function, then X_k is imaginary, which means that $x(t)$ can be represented with only sine terms.

Let us write X_k in polar form to get

$$X_k = \left\| X_k \right\| e^{j\angle X_k}$$

where

$||X_k||$ is the magnitude of X_k

$\angle X_k$ is the angle of X_k

then Equation 7.4 becomes

$$x(t) = X_0 + 2\sum_{k=1}^{\infty} \left\| X_k \right\| \cos(k\omega_0 t + \angle X_k) \tag{7.6}$$

Notice that $||X_k|| = ||X_{-k}||$, an even function of k, and $\angle X_k = -\angle X_{-k}$, an odd function of k. Here we see that $||X_k||$ gives information about the amplitude and $\angle X_k$ gives information about the phase angle of the sinusoidal contribution to $x(t)$ at the frequency $\omega = k\omega_0$. Thus, $||X_k||$ versus k is called the magnitude spectrum of $x(t)$, and $\angle X_k$ versus k is called the phase spectrum of $x(t)$.

As we study cyclical phenomena, we are interested to know the strengths and time displacements of sinusoidal components in $x(t)$ at the frequencies $f = kf_0$ Hz.

7.1.2 Example and Discussion

To be practical, we must truncate the series in Equation 7.4 to use a finite number of terms, giving

$$x(t) \cong \sum_{k=-K}^{+K} X_k e^{jk\omega_0 t} \tag{7.7}$$

To see what happens for different values of K, let us apply Equation 7.7 to the periodic signal shown in Figure 7.1.

The complex FS coefficients for this signal are given by

$$X_k = \frac{1}{3}\int_0^3 x(t)e^{-jk\frac{2\pi}{3}t}\,dt = \frac{1}{3}\int_{0.5}^{2.0} 4e^{-jk\frac{2\pi}{3}t}\,dt = \frac{4}{k\pi}e^{-jk\frac{5\pi}{6}}\,\sin\left(k\frac{\pi}{2}\right), \quad k \neq 0$$

and $X_0 = 2$. Figure 7.2 shows the magnitude spectrum of $x(t)$, which shows the amplitudes of the sinusoidal components of $x(t)$ at the frequencies $f = kf_0$ Hz.

Figure 7.3 shows the application of Equation 7.7 for increasing values of K. Notice the oscillation, called Gibbs oscillation, about points of discontinuity in $x(t)$. Even as K is increased, the oscillation remains with

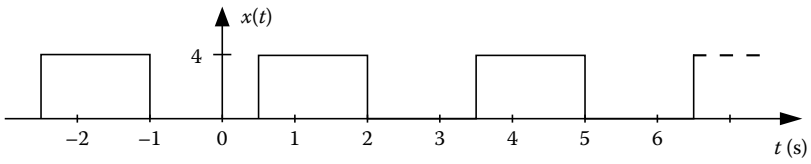


FIGURE 7.1 A periodic signal with discontinuities.

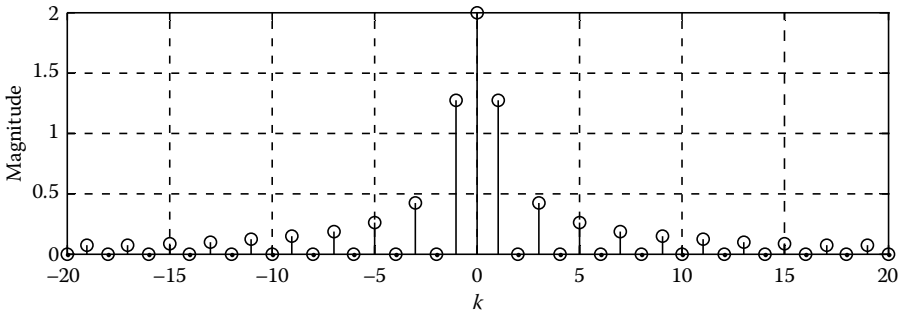


FIGURE 7.2 Magnitude spectrum for $-20 \leq k \leq +20$.

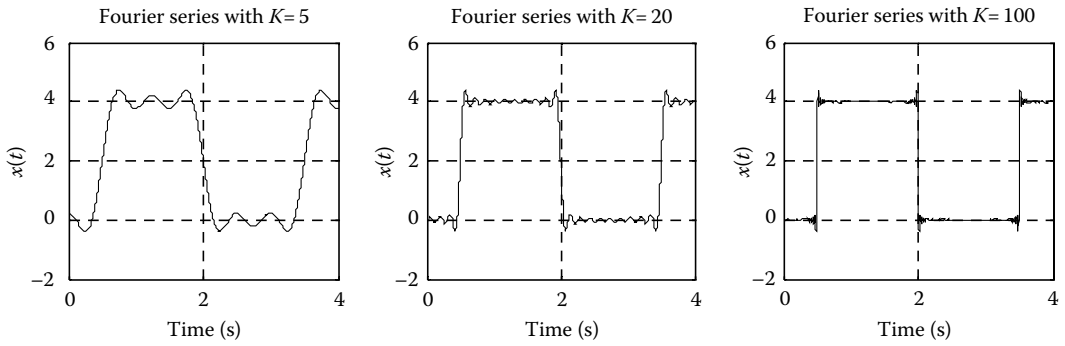


FIGURE 7.3 FS representation with $K = 5$, $K = 20$, and $K = 100$.

an amplitude exceeding $x(t)$ by about 7% of the change in $x(t)$ from one side of the discontinuity to the other side. Also notice that Gibbs oscillation becomes more and more concentrated about the discontinuities as K is increased.

7.1.3 Discrete-Time Signals

We obtain a discrete-time signal by uniformly sampling a continuous-time signal at some sampling rate f_s samples/s (Hz), which is expressed by

$$x(nT) = x(t)|_{t=nT}$$

where n, an integer, is called the discrete-time index and $T = 1/f_s$ is the sample time increment. If T is known and fixed, we may write the discrete-time signal as $x(n)$ instead of $x(nT)$.

Some discrete-time signals are a matter of definition. A few standard signals are

1. Unit step function,
   $u(n - n_0) = \begin{cases} 1, & n - n_0 \geq 0 \\ 0, & n - n_0 < 0 \end{cases}$
2. Unit pulse function (also called the Kronecker delta function),
   $\delta(n - n_0) = \begin{cases} 1, & n - n_0 = 0 \\ 0, & n - n_0 \neq 0 \end{cases}$
3. Exponential function,
   γ^n, for some real or complex constant γ
4. Sinusoidal function,
   $\cos(\theta n + \varphi)$

In terms of a continuous timescale, these discrete-time signals are only defined at the discrete time points given by $t = nT$.

The unit step function is commonly used to start (or stop) a given signal. For example, a sinusoidal pulse can be expressed as $\cos((\pi/4)n)(u(n) - u(n - 8))$, which is one cycle of the sinusoid. The unit pulse function is used to position a value at a time point. For example, $x(n) = -5\delta(n - 2)$ positions the value -5 of x to occur at the time $n = 2$, while for all other n, x has zero value. Through Euler's identity (Equation 7.3), a sinusoidal function can be expressed as the sum of two complex conjugate exponential functions.

We can write any discrete-time signal $x(n)$ as a linear combination of unit pulse functions given by

$$x(n) = \sum_{i=-\infty}^{+\infty} x(i)\delta(n - i) \tag{7.8}$$

7.1.4 Discrete Fourier Transform

For practical signals a function expression for $x(t)$ is usually not available, and therefore we cannot implement Equation 7.5 to find the spectrum X_k of $x(t)$. However, $x(t)$ can be sampled to obtain

$$x(nT) = x(t)|_{t=nT}$$

where T is determined with

$$T = \frac{T_0}{N} \tag{7.9}$$

for some integer N, which makes $x(n)$ a periodic discrete-time signal. We now develop an algorithm to find information about X_k, given $x(n)$.

To process $x(n)$, which can have values in a continuous range, by digital means, we must input each sample $x(n)$ into an analog-to-digital converter to obtain $x_{\mathrm{d}}(n)$, a digital signal, where each sample can have only one of a finite number of possible values. We approximate $x(n)$ with $x_{\mathrm{d}}(n)$, and

$$x(n) = x_{\mathrm{d}}(n) + e_{\mathrm{q}}(n)$$

where $e_{\mathrm{q}}(n)$ is the quantization error. In the following work $x(n)$ will be used for DSP. This may be acceptable if quantization error is negligible.

With the choice for T in Equation 7.9, $x(n)$ is periodic with period N, and with Equation 7.4, $x(n)$ becomes

$$x(n) = \sum_{k=-\infty}^{\infty} X_k \mathrm{e}^{jk\omega_0 nT}$$

Since $\omega_0 = \frac{2\pi}{T_0} = \frac{2\pi}{NT}$, we get for $x(n)$

$$x(n) = \sum_{k=-\infty}^{\infty} X_k \mathrm{e}^{jk\frac{2\pi}{NT}nT} = \sum_{k=-\infty}^{\infty} X_k \mathrm{e}^{j\frac{2\pi}{N}kn}$$

Let us write $k = m + rN$, where $m = 0, 1, \ldots, N-1$ and $r = \ldots, -2, -1, 0, 1, 2, \ldots$ are integers, and then $x(n)$ can be written as

$$x(n) = \sum_{r=-\infty}^{\infty} \sum_{m=0}^{N-1} X_{m+rN} \mathrm{e}^{j\frac{2\pi}{N}(m+rN)n} = \sum_{m=0}^{N-1} \sum_{r=-\infty}^{\infty} X_{m+rN} \mathrm{e}^{j\frac{2\pi}{N}mn}$$

where $\mathrm{e}^{j(2\pi/N)rNn} = 1$. Let $\hat{X}\ (k)$ be determined with

$$\hat{X}(k) = \sum_{r=-\infty}^{\infty} X_{k+rN}$$

and then $x(n)$ becomes

$$x(n) = \sum_{k=0}^{N-1} \hat{X}(k) \mathrm{e}^{j\frac{2\pi}{N}kn} \tag{7.10}$$

Notice that $\hat{X}(k)$ and X_k are related because $\hat{X}(k)$ is the periodic extension of X_k, and therefore $\hat{X}(k)$ is periodic, so that $\hat{X}(k) = \hat{X}(k + iN)$ for any integer i and all k. Knowing X_k, we can find $\hat{X}(k)$, $k = 0, 1, \ldots, N-1$. Since $\hat{X}(k)$ is periodic, we have $\hat{X}(k) = \hat{X}(k-N)$, and since the FS coefficients satisfy $X_k^* = X_{-k}$, we have $\hat{X}^*(k) = \hat{X}(N-k)$, and therefore

$$\hat{X}(k) = \hat{X}^*(N-k), \quad k = 0, 1, \ldots, N-1$$

This means, for example, that $\hat{X}^*(-1) = \hat{X}(1) = \hat{X}^*(N-1)$.

It would be very useful if we could obtain $\hat{X}(k)$ directly from $x(n)$. To do this, multiply both sides of Equation 7.10 by a particular exponential function, giving

$$x(n)e^{-j\frac{2\pi}{N}nm} = e^{-j\frac{2\pi}{N}nm}\sum_{k=0}^{N-1}\hat{X}(k)e^{j\frac{2\pi}{N}kn} = \sum_{k=0}^{N-1}\hat{X}(k)e^{j\frac{2\pi}{N}(k-m)n}$$

where m is any integer in the range $0, 1, \ldots, N-1$. Summing the left-hand and right-hand sides of this equation for $n = 0, 1, \ldots, N-1$ gives

$$\sum_{n=0}^{N-1}x(n)e^{-j\frac{2\pi}{N}nm} = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1}\hat{X}(k)e^{j\frac{2\pi}{N}(k-m)n} = \sum_{k=0}^{N-1}\hat{X}(k)\sum_{n=0}^{N-1}e^{j\frac{2\pi}{N}(k-m)n}$$

Applying the formula for the sum of a geometric series gives

$$\sum_{n=0}^{N-1}e^{j\frac{2\pi}{N}(k-m)n} = \begin{cases} N, & k = m \\ \dfrac{1 - e^{j\frac{2\pi}{N}(k-m)N}}{1 - e^{j\frac{2\pi}{N}(k-m)}} = 0, & k \neq m \end{cases}$$

and therefore

$$\sum_{n=0}^{N-1}x(n)e^{-j\frac{2\pi}{N}nm} = N\hat{X}(m)$$

This results in the discrete Fourier transform (DFT) $X(k)$ of $x(n)$, $n = 0, 1, \ldots, N-1$, which is given by

$$X(k) = \sum_{n=0}^{N-1}x(n)e^{-j\frac{2\pi}{N}nk}, \quad k = 0, 1, \ldots, N-1 \tag{7.11}$$

and therefore

$$X(k) = N\hat{X}(k) = N\sum_{r=-\infty}^{\infty}X_{k+rN} \tag{7.12}$$

Equation 7.10 becomes

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{j\frac{2\pi}{N}kn} \tag{7.13}$$

which is called the inverse DFT (IDFT). Like $\hat{X}(k)$, $X(k)$ is periodic with period N and $X(k) = X^*(N-k)$. We write that $X(k) = \text{DFT}\{x(n)\}$ and $x(n) = \text{IDFT}\{X(k)\}$.

Equations 7.11 and 7.13 are very similar. If $x(n)$ is real, then conjugating Equation 7.13 gives

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k)e^{-j\frac{2\pi}{N}kn} = \frac{1}{N}\text{DFT}\{X^*(k)\}$$

Therefore, if we have an algorithm to compute Equation 7.11, the DFT, then that same algorithm, instead of Equation 7.13, can be used to compute the IDFT.

While our interest is to investigate the spectral properties of a continuous-time periodic signal with Equations 7.11 and 7.12, Equation 7.13 is a discrete-time FS representation for a discrete-time periodic signal, and Equation 7.11 is used to find the discrete-time FS coefficients. We say that $x(n)$ and $X(k)$ are a discrete Fourier series (DFS) pair, which is denoted by $x(n) \leftrightarrow X(k)$. Notice that for a discrete-time periodic signal, the FS uses a finite number of discrete-time sinusoids, while for a continuous-time periodic signal, the FS in Equation 7.4 uses a countable infinite number of sinusoids. Furthermore, $X(k)$ is a periodic function in the frequency domain with period N.

7.1.5 Examples and Discussion

Given $x(n)$, the result of sampling a continuous-time periodic signal $x(t)$, we apply Equation 7.11 to obtain $X(k) = \text{DFT}\{x(n)\}$, because, in view of Equation 7.12, over one period of $X(k)$ we expect to obtain a function that potentially behaves like X_k, the spectrum of the continuous-time periodic signal. An important property of $x(t)$ that strongly influences the relationship between X_k and $X(k)$ is whether or not $x(t)$ is bandlimited. If $x(t)$ is bandlimited, then there is some positive integer M such that $X_k = 0$ for $|k| > M$. We say that the bandwidth of $x(t)$ is $\text{BW} = Mf_0$ Hz. Denote the BW by f_c.

To see the effect of selecting an appropriate sampling frequency f_s, let us work with the bandlimited signal given by

$$x(t) = 6\cos\left(10\pi t + \frac{\pi}{4}\right) + 4\sin\left(30\pi t - \frac{\pi}{3}\right) \tag{7.14}$$

Here, $f_0 = 5$ Hz, and $f_c = 15$ Hz, with $M = 3$. Figure 7.4 shows the magnitude spectrum of $x(t)$. The spectral points occur at the frequencies given by $f = k(1/T_0) = kf_0$ Hz, $k = -M, \ldots, 0, 1, 2, M$.

Now, let us sample this $x(t)$ over one period to obtain $N = 10$ time domain samples $x(n)$, $n = 0, 1, \ldots, N-1$. The magnitude of the DFT of $x(n)$ is shown in Figure 7.5. Here we see two periods of the magnitude of $X(k)$ obtained from $||X_k||$ and its first translation $||X_{k-N}||$. Depending on N, the translations of X_k to all integer multiples of N may overlap. In Figure 7.5, N is large enough to prevent the overlap of the translations of X_k, and therefore one period of $X(k)/N$ gives X_k without error. Notice that for $k = 0, 1, \ldots, M$ we have $||X_k||$ for positive frequencies, and for $k = N-1, N-2, \ldots, N-M$ we have $||X_k||$ for negative frequencies.

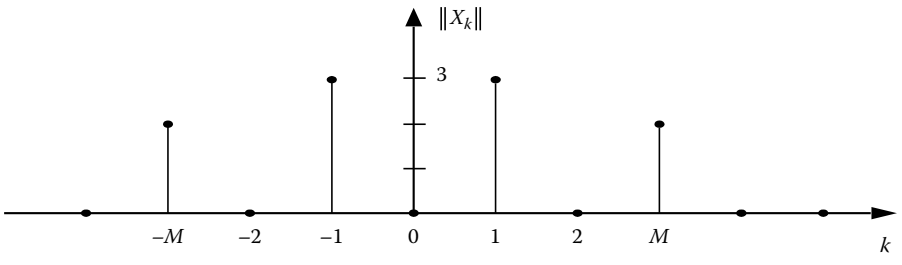


FIGURE 7.4 Magnitude spectrum of a bandlimited signal.

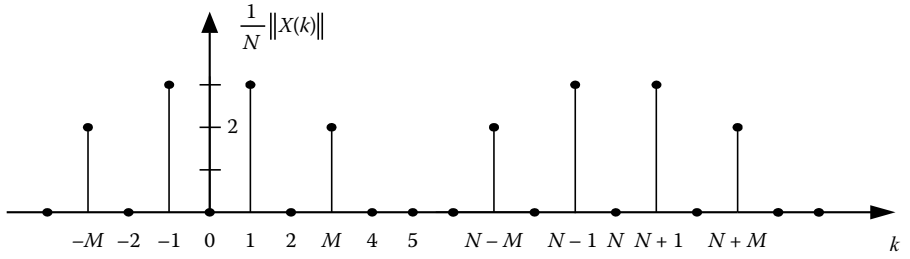


FIGURE 7.5 Two periods of the magnitude of $X(k)$.

In general, if $x(t)$ is bandlimited and $N - M > M$, then the translations of X_k to integer multiples of N cannot overlap. This condition can be written as

$$\frac{N-M}{T_0} > \frac{M}{T_0} \rightarrow \frac{N}{NT} - Mf_0 > Mf_0$$

Therefore, if $x(t)$ is bandlimited to f_c and the sampling frequency f_s satisfies

$$f_s > 2f_c \tag{7.15}$$

such that T_0/T is an integer, which means that f_s must be an integer multiple of f_0, then the DFT of $\{x(n)\}$ gives the spectrum of $x(t)$ without error. This is called the sampling theorem.

In Figure 7.5, the sampling frequency is given by $f_s = \frac{1}{T} = \frac{N}{T_0} = 50$ Hz, and therefore Equation 7.15 is satisfied. Moreover, $x(t)$ can have sinusoidal components with frequencies in the range $0 \leq f < f_s/2$, and the DFT gives X_k without error.

Now, let us try an f_s that does not satisfy Equation 7.15, and such that the frequency points again occur at integer multiples of $f_0 = 5$ Hz. For example, let $f_s = 25$ Hz, giving $N = 5$. Figure 7.6 shows the magnitude of $X(k)$. Like in Figure 7.5, we use the DFT to give the spectrum of $x(t)$ in the frequency range $0 \leq f < f_s/2$, and here $f_s/2 = 12.5$ Hz, which corresponds to $k = 0, 1, 2$. From Figure 7.6, we conclude that $x(t)$ has sinusoidal components at the frequencies 5 and 10 Hz. However, only the spectral point at 5 Hz is correct. The spectral point at 10 Hz, which does not occur in $x(t)$, is called aliasing error, and we cannot even determine that $x(t)$ has a sinusoidal component at 15 Hz. If the original $x(t)$ did consist of sinusoids only at the frequencies 5 and 10 Hz, then Figure 7.6 would be correct.

To obtain correct results with the DFT, we must know the bandwidth and period of a given periodic signal $x(t)$. However, it may not be realistic to assume that we can know this information about a given

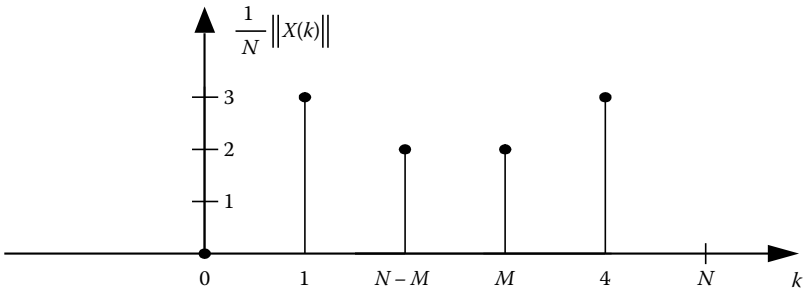


FIGURE 7.6 Magnitude of $X(k)$ for $k = 0, 1, 2, \ldots, N-1$, $N = 5$, and $f_s = 25$ Hz.

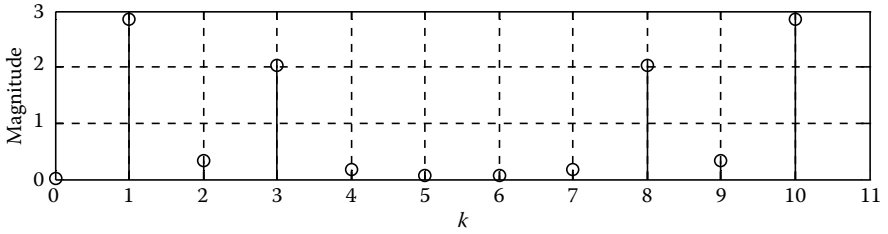


FIGURE 7.7 Magnitude spectrum for $k = 0, 1, \ldots, N-1$, $N = 11$, and $f_s = 53$ Hz.

signal. Furthermore, $x(t)$ may not even be bandlimited (e.g., see Figure 7.2.), in which case translations of X_k to form $X(k)$ will overlap and aliasing error cannot be avoided.

In comparison to the results shown in Figure 7.5, let us try an f_s that does satisfy the sampling theorem of Equation 7.15 for the signal given by Equation 7.14. For example, let $f_s = 53$ Hz. For this sampling frequency, $T_0/T = 10.6$ is not an integer. Therefore, let us set N to $N = 11$, which means that effectively we consider one period $x_p(t)$ to be defined for $0 \leq t < T_0$, where $T_0 = NT = 0.210325$ s. Therefore, the DFT will obtain spectral points at integer multiples of $f_0 = 1/T_0 = 4.7545$ Hz, instead of 5 Hz, as in Figure 7.5. While the signal given by Equation 7.14 is bandlimited, a periodic extension of $x_p(t)$ is a periodic signal with discontinuities at period boundaries, and therefore we have a signal that is not bandlimited. Figure 7.7 shows the magnitude of $X(k)$, and we see that, while it is similar to the magnitude spectrum given in Figure 7.5, this result is an approximation of the spectrum of the signal given by Equation 7.14.

The spectral points in Figure 7.5 at 5 and 15 Hz have spread out about these frequencies in Figure 7.7. This is called leakage error. Leakage error occurs when we use a sample time range other than an integer multiple of the period of the given periodic signal.

If it is practically possible, it is useful to sample the continuous-time signal over a longer time range. For example, with the sampling frequency used to produce the result shown in Figure 7.7, let us obtain $N = 50$ samples of $x(t)$. In this case, $x_p(t)$ becomes the segment of $x(t)$ for $0 \leq t < T_0$, where $T_0 = NT = 0.9434$ s. It is likely that we will still have discontinuities at the period boundaries of the periodic extension of $x_p(t)$, which means that the resulting periodic signal is not bandlimited, and aliasing error cannot be avoided. Since T_0 is not an integer multiple of the period of the given $x(t)$, we will incur leakage error. However, we now have a higher frequency resolution determined by $f_0 = 1/T_0 = 1.06$ Hz.

Figure 7.8 shows the magnitude of the DFT of $x(n)$, $n = 0, 1, \ldots, 49$. Notice that the spectral point at $k = 0$ is not zero, which means that the average value of the periodic signal is not zero, even though the

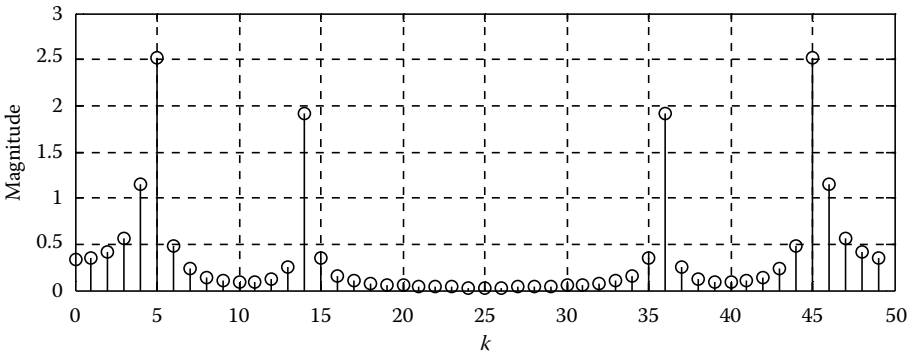


FIGURE 7.8 Magnitude spectrum for $k = 0, 1, \ldots, N-1$, $N = 50$, and $f_s = 53$ Hz.

given $x(t)$ has a zero average value. Also, there is leakage error, as expected, and there is significant aliasing error at higher frequencies, i.e., as $k \rightarrow 25$. Nevertheless, like in Figure 7.7, this magnitude spectrum is an approximation of the magnitude spectrum shown in Figure 7.5.

We can reduce the aliasing error shown in Figure 7.8 by reducing the discontinuities that occur at the period boundaries of the periodic extension of $x_p(t)$. Let us view $x_p(t)$ to be given by

$$x_p(t) = x(t)w(t) \tag{7.16}$$

where $w(t)$, which is called a window function, is given by $w(t) = w_R(t)$, the rectangular window, and

$$w_R(t) = \begin{cases} 1, & 0 \leq t < T_0 \\ 0, & \text{otherwise} \end{cases}$$

We can write $w_R(t) = u(t) - u(t - T_0)$, where $u(t)$ is the unit step function.

To eliminate the discontinuities at period boundaries, let the window function be given by $w(t) = w_H(t)$, where

$$w_H(t) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi t}{T_0}\right)\right) w_R(t)$$

which is called the Hann window. Notice that $w_H(t = 0) = w_H(t = T_0) = 0$. Using the Hann window in Equation 7.16 gives the magnitude spectrum shown in Figure 7.9. Notice the reduced magnitudes about the frequencies of sinusoids in the given $x(t)$. There are numerous other widow functions that could also have been used.

7.1.6 DFT Applications

The DFT has many properties that account for its wide application. Here, we examine how the DFT can be involved to compute the linear convolution of two discrete-time signals. Let $x_1(n)$ and $x_2(n)$ be two periodic discrete-time signals, each having period N. The circular convolution of $x_1(n)$ and $x_2(n)$, which is denoted by $x_1(n) \otimes x_2(n)$, is another periodic signal $x_3(n)$ with period N given by

$$x_3(n) = x_1(n) \otimes x_2(n) = \sum_{i=0}^{N-1} x_1(i)x_2(n-i) = \sum_{i=0}^{N-1} x_1(n-i)x_2(i) \tag{7.17}$$

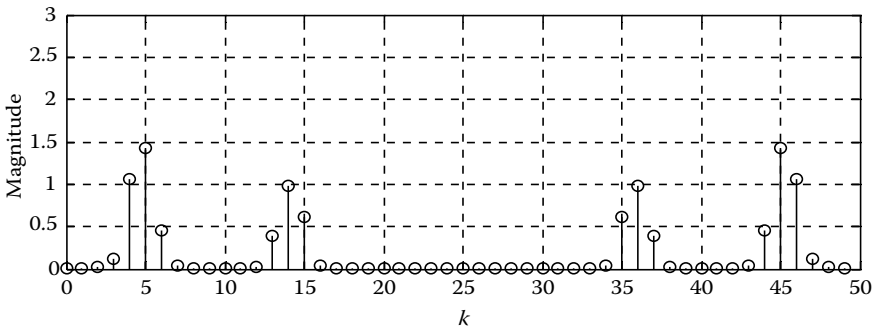


FIGURE 7.9 Magnitude spectrum for $N = 50$, $f_s = 53$ Hz, and $w(t) = w_H(t)$.

It is not difficult to show that $X_3(k) = \text{DFT}\{x_3(n)\}$ is given by

$$X_3(k) = X_1(k)X_2(k) \tag{7.18}$$

which is called the circular convolution theorem. In other words, circular convolution in the time domain can be transformed to multiplication in the frequency domain. Therefore, instead of using Equation 7.17 to find the circular convolution of $x_1(n)$ and $x_2(n)$, we find $x_3(n)$ with the IDFT of $X_3(k)$ found with Equation 7.18. This is the indirect convolution method.

Now consider the linear convolution of $y_1(n)$ and $y_2(n)$, which are not periodic, to obtain $y_3(n)$ given by

$$y_3(n) = \sum_{i=-\infty}^{+\infty} y_1(n-i)y_2(i) = \sum_{i=-\infty}^{+\infty} y_1(i)y_2(n-i) = y_1(n)^* y_2(n) \tag{7.19}$$

Let us assume that $y_1(n)$ and $y_2(n)$ have finite durations, and $y_1(n)$ is given for $n = 0, \ldots, N_1 - 1$ and $y_2(n)$ is given for $n = 0, \ldots, N_2 - 1$. The duration N_3 of $y_3(n)$ will be $N_3 = N_1 + N_2 - 1$.

To employ the DFT, let N be any integer such that $N > N_3$, and define periodic signals $x_1(n)$ and $x_2(n)$ with periods given by

$$x_1(n) = \begin{cases} y_1(n), & n = 0, \ldots, N_1 - 1 \\ 0, & n = N_1, \ldots, N - 1 \end{cases} \quad \text{and} \quad x_2(n) = \begin{cases} y_2(n), & n = 0, \ldots, N_2 - 1 \\ 0, & n = N_2, \ldots, N - 1 \end{cases}$$

We now find the circular convolution $x_3(n)$ of $x_1(n)$ and $x_2(n)$ with the indirect convolution method, and then over one period of $x_3(n)$ we get $y_3(n) = x_3(n)$ for $n = 0, \ldots, N_3 - 1$. At this point, we may expect that the indirect convolution method will require more computation than the direct convolution method using Equation 7.17. However, we will develop a computationally efficient algorithm, called the fast Fourier transform (FFT) to compute the DFT that will make the indirect convolution method more computationally efficient than the direct convolution method.

An operation similar to linear convolution is the linear correlation operation given by

$$r_{12}(l) = \sum_{n=-\infty}^{+\infty} y_1(n)y_2(n-l) \tag{7.20}$$

where l, $l = 0, 1, \ldots$, is called the lag index. However, the correlation $r_{12}(l)$, which is useful to investigate the similarity of two signals, can be obtained with

$$r_{12}(l) = y_1(l)^* y_2(-l) \tag{7.21}$$

Again, we assume that $y_1(n)$ and $y_2(n)$ have finite durations of N_1 and N_2, respectively, and the duration of $r_{12}(l)$ will be $N_3 = N_1 + N_2 - 1$.

To employ the DFT, we define $x_1(n)$ and $x_2(n)$ as before, and by indirect convolution compute

$$x_3(l) = x_1(l) \otimes x_2(-l)$$

to get $r_{12}(l) = x_3(l)$ for $l = 0, \ldots, N_3 - 1$. The DFT of $x_2(-n)$ is given by $X_2^*(k)$, where $X_2(k)$ is the DFT of $x_2(n)$.

7.1.7 Fast Fourier Transform

The FFT is a collection of algorithms that can compute the DFT, $X(k)$, $k = 0, \ldots, N-1$, of a discrete-time signal, $x(n)$, $n = 0, \ldots, N-1$, much more efficiently than with Equation 7.11. The efficiency of the FFT occurs when N can be expressed as a product of many small integers, $N = N_1\, N_2 \ldots N_I$. An ideal case occurs when N is a power of 2, $N = 2^L$, for some integer L, and then the FFT algorithm is called the radix-2 FFT algorithm. Other possibilities are, for example, $N = 3^K$, resulting in a radix-3 FFT, or N is a product of many different small integers, resulting in a mixed-radix FFT. Here, we will outline the development of the radix-2 FFT, and even more specifically, the decimation in time radix-2 FFT.

Before we start the FFT development, let us assess the computational requirements of Equation 7.11, the DFT operation. It is generally assumed that sinusoidal values are found by table lookup, which requires time that is negligible compared to multiplication time. If the DFT input is a set of N complex numbers, then the DFT will require a total of N^2 complex multiplications to obtain the set of N outputs, and we will use this for assessment of DFT computational requirements. We will see how the radix-2 FFT works to require only $NL/2$ complex multiplications.

We start by assuming that the number of points is $N = 2^L$, for some integer L. If for a given discrete-time signal $x(n)$ the number of points N is not a power of two, then we can augment the given $x(n)$ with a sufficient number of zero points for a total number of points that is a power of 2.

Since N is even, we can split the DFT summation in Equation 7.11 into two parts having an equal number of summation terms. Let the first part come from the even indexed terms in Equation 7.11, and let the second part come from the odd indexed terms, and we get

$$X(k) = \sum_{n=0}^{N-1} x(n) \mathrm{e}^{-j\frac{2\pi}{N}nk} = \sum_{n,\text{even}} x(n) \mathrm{e}^{-j\frac{2\pi}{N}nk} + \sum_{n,\text{odd}} x(n) \mathrm{e}^{-j\frac{2\pi}{N}nk}$$

Even n can be written as $n = 2i$, and odd n can be written as $n = 2i + 1$, for $i = 0, 1, \ldots, N_1 - 1$, where $N_1 = N/2$, which is an even integer. Now we have

$$X(k) = \sum_{i=0}^{N_1-1} x(2i) \mathrm{e}^{-j\frac{2\pi}{N}(2i)k} + \sum_{i=0}^{N_1-1} x(2i+1) \mathrm{e}^{-j\frac{2\pi}{N}(2i+1)k}$$

Let $x_0(n)$ and $x_1(n)$ be the N_1 point number sequences defined by

$$\begin{aligned} x_0(n) &= x(2n) \\ x_1(n) &= x(2n+1), \quad n = 0, 1, \ldots, N_1 - 1 \end{aligned}$$

and therefore we get

$$X(k) = \sum_{n=0}^{N_1-1} x_0(n) \mathrm{e}^{-j\frac{2\pi}{N_1}nk} + \mathrm{e}^{-j\frac{2\pi}{N}k} \sum_{n=0}^{N_1-1} x_1(n) \mathrm{e}^{-j\frac{2\pi}{N_1}nk}$$

Now let

$$X_0(k) = \sum_{n=0}^{N_1-1} x_0(n) \mathrm{e}^{-j\frac{2\pi}{N_1}nk} \tag{7.22}$$

$$X_1(k) = \sum_{n=0}^{N_1-1} x_1(n) \mathrm{e}^{-j\frac{2\pi}{N_1}nk}, \quad k = 0, 1, \ldots, N_1 - 1 \tag{7.23}$$

and therefore we get

$$X(k) = X_0(k) + e^{-j\frac{2\pi}{N}k}X_1(k), \quad k = 0, 1, \ldots, N-1 \tag{7.24}$$

An N-point DFT, $X(k)$, is periodic with period N. We see that Equations 7.22 and 7.23 are each N_1-point DFTs, and therefore $X_0(k)$ and $X_1(k)$ are periodic with period N_1. An N_1-point DFT requires N_1^2 complex multiplications. To find $X(k)$, let us use Equation 7.24 only for $k = 0, 1, \ldots, N_1$. However, for $k = N_1, N_1 + 1, \ldots, N-1$, let $m = k - N_1$, so that m varies from $m = 0$ to $m = N_1 - 1$, and Equation 7.24 becomes

$$X(N_1 + m) = X_0(N_1 + m) + e^{-j\frac{2\pi}{N}(N_1+m)}X_1(N_1 + m)$$

Or, since $X_0(k)$ and $X_1(k)$ are periodic with period N_1, we have

$$X(N_1 + k) = X_0(k) - e^{-j\frac{2\pi}{N}k}X_1(k), \quad k = 0, 1, \ldots, N_1 - 1 \tag{7.25}$$

The only difference between Equations 7.24 and 7.25 is addition of two terms in Equation 7.24 and subtraction of the same two terms in Equation 7.25.

Figure 7.10 combines Equations 7.24 and 7.25 for $k = 0, 1, \ldots, N_1 - 1$. This structure is called a butterfly operation. To find $X(k)$ for $k = 0, 1, \ldots, N-1$ given $X_0(k)$ and $X_1(k)$ for $k = 0, 1, \ldots, N_1 - 1$ requires $N/2$ complex multiplications, one complex multiplication for each butterfly operation. The $N/2$ butterfly operations are the first stage of reducing the computational requirements of the DFT operation.

The sequence of steps toward the first stage started with splitting the summation of Equation 7.11 by decimating $x(n)$ and finished with Equations 7.24 and 7.25, which is illustrated by Figure 7.10. We continue to obtain the second stage by decimating $x_0(n)$ into $x_{00}(n)$ and $x_{10}(n)$ from the even and odd, respectively, indexed points in $x_0(n)$ and by decimating $x_1(n)$ into $x_{01}(n)$ and $x_{11}(n)$ from the even and odd, respectively, indexed points in $x_1(n)$ to reduce the computational requirements for $X_0(k)$ and $X_1(k)$. This second stage requires $N_1/2 + N_1/2 = N/2$ complex multiplications. We continue through L stages, where in the last stage, each of the $N/2$ DFT operations is a 2-point DFT consisting of a single butterfly operation. Therefore, the last stage also requires $N/2$ complex multiplications. The overall algorithm is called the decimation in time radix-2 FFT, and it requires $NL/2$ complex multiplications to compute the DFT, compared to a direct DFT computation, which requires N^2 complex multiplications.

Suppose a signal $x(t)$ with BW = 3.6 kHz is sampled at the rate $f_s = 8$ kHz to obtain $N = 2^{13}$ points (about a 1 s time interval). A direct DFT requires approximately 67,000,000 complex multiplications, while an FFT requires approximately 53,000 complex multiplications, a reduction in time by a factor of 1264. The efficiency of the FFT improves as N is increased. With the FFT, indirect convolution can be demonstrated to be much more efficient than direct convolution.

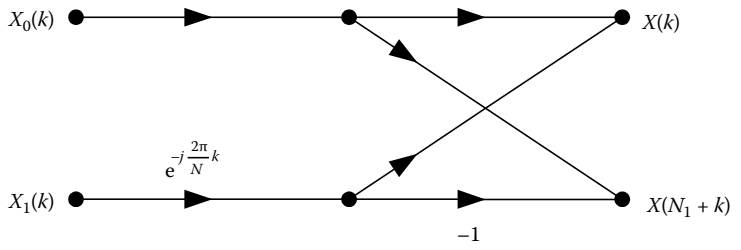


FIGURE 7.10 Butterfly operations of the first stage.

7.1.8 Continuous-Time Fourier Transform

By sampling a periodic continuous-time signal we developed the DFT operation and found a relationship between the result of a DFT operation $X(k)$ and the spectrum X_k of the continuous-time periodic signal. We also came to understand how $X(k)$ can be different from X_k, the FS coefficients. However, real-world continuous-time signals are not necessarily periodic. Here, we extend the concept of a FS approximation of a periodic continuous-time signal to an aperiodic continuous-time signal.

Consider the arbitrary aperiodic signal shown in Figure 7.11, where T_0 is large enough to contain the signal within the time range $-T_0/2 < t < T_0/2$. Over this time range the given signal can be represented with the FS of Equation 7.4, which is repeated here for convenience

$$x(t) = \sum_{k=-\infty}^{+\infty} X_k e^{jk\omega_0 t} \tag{7.26}$$

where

$$X_k = \frac{1}{T_0} \int_{-T_0/2}^{+T_0/2} x(t) e^{-jk\omega_0 t} dt \tag{7.27}$$

If in Figure 7.11 we let T_0 go to infinity, then the FS in Equation 7.26 represents the given signal for all t. Let us see what happens as T_0 is increased. If the integral in Equation 7.27 remains finite, then the X_k will go to zero. Since the spectral points occur at the frequencies given by $\omega = k\omega_0$, where $\omega_0 = 2\pi/T_0$, the spectrum tends toward a continuous function of frequency. Therefore, in view of Equation 7.27, let us work with

$$X(j\omega) = \lim_{T_0 \to \infty} X_k T_0 \tag{7.28}$$

which gives

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \tag{7.29}$$

where $k\omega_0$ has become the continuous frequency ω. And then, Equation 7.26 becomes

$$x(t) = \frac{1}{2\pi} \lim_{T_0 \to \infty} \sum_{k=-\infty}^{+\infty} X_k T_0 e^{jk\omega_0 t} \frac{2\pi}{T_0}$$

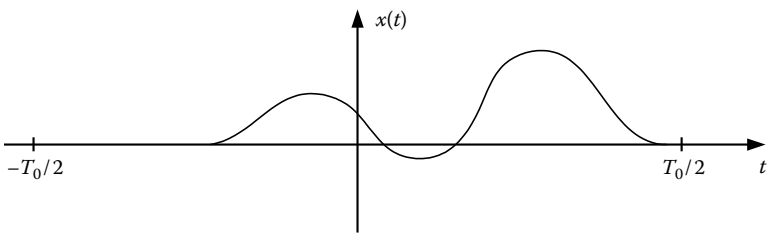


FIGURE 7.11 An arbitrary aperiodic signal.

TABLE 7.1 FT Properties

Linearity	$c_1x_1(t) + c_2x_2(t) \leftrightarrow c_1X_1(j\omega) + c_2X_2(j\omega)$
Time shift	$x(t - t_0) \leftrightarrow X(j\omega)e^{-j\omega t_0}$
Modulation	$x(t)e^{j\omega_0 t} \leftrightarrow X(j(\omega - \omega_0))$
Linear convolution	$x_1(t)^*x_2(t) = \int_{-\infty}^{+\infty} x_1(t-\tau)x_2(\tau)d\tau \leftrightarrow X_1(j\omega)X_2(j\omega)$
Derivative	$\frac{d}{dt}x(t) \leftrightarrow j\omega X(j\omega)$
Multiplication	$x_1(t)x_2(t) \leftrightarrow \frac{1}{2\pi}X_1(j\omega)^*X_2(j\omega)$
Energy (Parseval's theorem)	$\int_{-\infty}^{+\infty} \lvert x(t)\rvert^2 dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \lVert X(j\omega)\rVert^2 d\omega$

which gives

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t}d\omega \tag{7.30}$$

We should view Equation 7.30 similar to Equation 7.26, which is that Equation 7.30 expresses an aperiodic continuous-time signal in terms of a linear combination of an uncountable infinite number of sinusoids. Equation 7.29 gives the Fourier transform (FT) $X(j\omega)$, which is a complex function of the continuous real frequency variable ω, of the continuous-time function $x(t)$, and it describes how the strengths of sinusoidal components of $x(t)$ are distributed over frequency. In view of Equation 7.28, $X(j\omega)$ is the spectral density of $x(t)$. Equation 7.30 gives the inverse FT of $X(j\omega)$, and we say that $x(t)$ and $X(j\omega)$ are a FT pair, denoted by $x(t) \leftrightarrow X(j\omega)$.

The FT has many properties that are widely applied in continuous-time signal and system analysis and design. Assume that we have the FT pair $x(t) \leftrightarrow X(j\omega)$. Table 7.1 gives a few important properties.

7.1.9 Examples and Discussion

The FT of a signal may not exist. For example, for $x(t) = u(t)$, the unit step function, we must utilize an impulse function to express its FT. In view of Equation 7.28, the FT of a sinusoidal signal also requires using impulse functions. A sufficient condition for the existence of the FT is that

$$\lVert X(j\omega)\rVert \le \int_{-\infty}^{+\infty} \left\lVert x(t)e^{-j\omega t}\right\rVert dt = \int_{-\infty}^{+\infty} \lVert x(t)\rVert dt < \infty$$

which means that the signal $x(t)$ must be absolutely integrable.

Consider the signal $x(t)$ given by $x(t) = \sin(\omega_0 t)w_R(t)$, where $w_R(t)$ is a rectangular window extending from $-\tau/2$ to $+\tau/2$, and $x(t)$ is a sinusoidal pulse. To understand the reasons for the features of $X(j\omega)$, the FT of $x(t)$, let us first consider $w_R(t)$. The FT of $w_R(t)$ is given by

$$W_R(j\omega) = \int_{-\tau/2}^{+\tau/2} e^{-j\omega t}dt = \frac{1}{-j\omega}(e^{-j\omega\tau/2} - e^{+j\omega\tau/2}) = \tau\frac{\sin(\omega\tau/2)}{\omega\tau/2}$$

which is a real function of ω, since $w_R(t)$ is an even function, and its magnitude is shown in Figure 7.12. The first zeros of $W_R(j\omega)$ occur at $\omega = \pm 2\pi/\tau$, and within this frequency range we have a feature called the main lobe of $W_R(j\omega)$. Outside this frequency range are the side lobes of $W_R(j\omega)$. We see that as

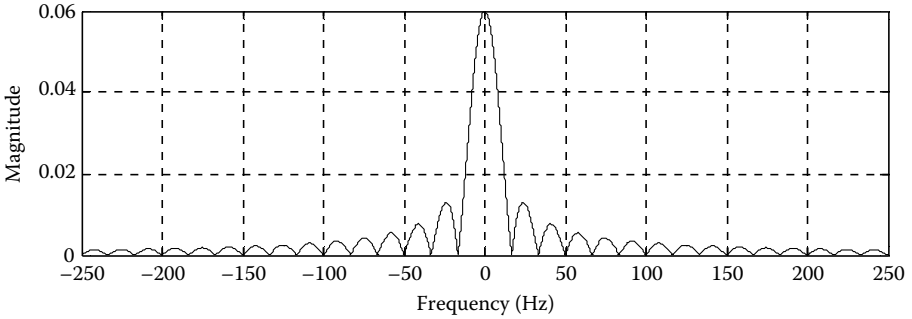


FIGURE 7.12 Spectral density magnitude, $\tau = 0.06$ s.

τ is increased, which makes $w_R(t)$ occur over a longer time range, the main lobe narrows and more energy of $w_R(t)$ is concentrated about $\omega = 0$ and within a smaller frequency range. Conversely, as τ is decreased, which makes $w_R(t)$ occur over a shorter time range, the main lobe broadens and the energy of $w_R(t)$ is spread out over a wider frequency range.

In $x(t)$, let $\omega_0 = 100\pi$ rad/s. The spectral density $X(j\omega)$ of $x(t)$ can be found with the modulation property, which translates $W_R(j\omega)$ up to $\omega = +\omega_0$ and down to $\omega = -\omega_0$. This is shown in Figure 7.13, where we see the main lobes positioned about $\omega = \pm\omega_0$.

In view of the properties given in Table 7.1, the FT is very well suited to investigate the spectral nature of a signal. It is also useful to study the frequency selective behavior of a linear and time invariant (LTI) continuous-time system. Let $h(t)$ denote the impulse response of an LTI system. If the system is initially at rest, then the response $y(t)$ to an input $x(t)$ is given by

$$y(t) = h(t)^* x(t) = \int_{-\infty}^{+\infty} h(t-\tau)x(\tau)\mathrm{d}\tau \tag{7.31}$$

and if the FT $H(j\omega)$ of the impulse response $h(t)$ exists, then by the convolution property we have

$$Y(j\omega) = H(j\omega)X(j\omega)$$

where $H(j\omega)$ is called the system transfer function. Therefore, with $H(j\omega)$ we can find how the system modifies the spectral density of the input to obtain the spectral density of the output. Furthermore, if the input is a sinusoid given by

$$x(t) = A\cos(\omega t + \phi)u(t)$$

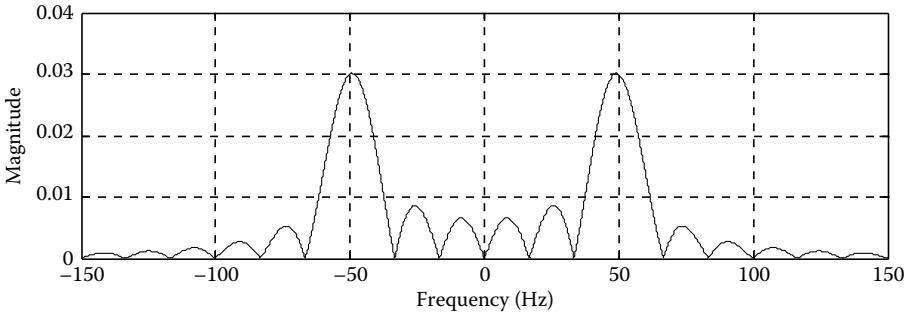


FIGURE 7.13 Spectral density of a sinusoidal pulse.

then the steady-state response of the system will be

$$y(t) = A\|H(j\omega)\| \cos(\omega t + \phi + \angle H(j\omega))$$

which shows the frequency-selective behavior of the system. We plot $||H(j\omega)||$ and $\angle H(j\omega)$ versus ω to see the frequency response of the system.

7.1.10 Discrete-Time Fourier Transform

As for continuous-time periodic signals, we want to study continuous-time aperiodic signals and continuous-time LTI systems by digital means. We start by sampling $x(t)$, the inverse FT of $X(j\omega)$, at the sampling rate $f_s = 1/T$ Hz, and Equation 7.30 becomes

$$x(nT) = \int_{-\infty}^{+\infty} X(j2\pi f) e^{j2\pi f nT} \mathrm{d}f$$

Breaking the frequency range into intervals of length f_s results in

$$x(nT) = \sum_{r=-\infty}^{+\infty} \int_{(2r-1)\frac{f_s}{2}}^{(2r+1)\frac{f_s}{2}} X(j2\pi f) e^{j2\pi f nT} \mathrm{d}f$$

Let us change the integration variable from f to $\lambda = f - rf_s$, and then we have

$$x(nT) = \sum_{r=-\infty}^{+\infty} \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} X(j2\pi(\lambda + rf_s)) e^{j2\pi(\lambda + rf_s)nT} \mathrm{d}\lambda$$

Interchanging summation and integration and simplifying the exponential term results in

$$x(nT) = \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} \sum_{r=-\infty}^{+\infty} X(j2\pi(\lambda + rf_s)) e^{j2\pi\lambda nT} \mathrm{d}\lambda$$

Reverting to f for the frequency variable gives

$$x(nT) = \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} \sum_{r=-\infty}^{+\infty} X(j2\pi(f - rf_s)) e^{j2\pi f nT} \mathrm{d}f \tag{7.32}$$

Let $\hat{X}(j2\pi f)$ be determined with

$$\hat{X}(j2\pi f) = \sum_{r=-\infty}^{+\infty} X(j2\pi(f - rf_s)) \tag{7.33}$$

which is the periodic extension of $X(j2\pi f)$, and therefore $\hat{X}(j2\pi f)$ is a periodic function of frequency with period f_s. And, Equation 7.32 becomes

$$x(nT) = \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} \hat{X}(j2\pi f) e^{j2\pi f nT} df \tag{7.34}$$

If we know $X(j2\pi f)$, the FT of the real-time function $x(t)$, then we can find $\hat{X}(j2\pi f)$ with Equation 7.33. It would be useful if we could obtain $\hat{X}(j2\pi f)$ directly from the samples $x(nT)$ of $x(t)$. Since $\hat{X}(j2\pi f)$ is a periodic function, having an even magnitude and odd phase, we can write it as a FS, which is

$$\hat{X}(j2\pi f) = \sum_{n=-\infty}^{+\infty} \hat{x}_n e^{-jn\frac{2\pi}{f_s}f} \tag{7.35}$$

where the $\hat{x}_n$ are the FS coefficients given by

$$\hat{x}_n = \frac{1}{f_s} \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} \hat{X}(j2\pi f) e^{jn\frac{2\pi}{f_s}f} df \tag{7.36}$$

In view of Equation 7.34, we have $\hat{x}_n = Tx(nT)$. Based on Equation 7.35, let

$$X(e^{j\omega T}) = \sum_{n=-\infty}^{+\infty} x(nT) e^{-j\omega nT} \tag{7.37}$$

which is called the discrete-time Fourier transform (DTFT) of $x(nT)$. Then, $\hat{X}(j\omega) = TX(e^{j\omega T})$, and Equation 7.34 becomes

$$x(nT) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{+\frac{\omega_s}{2}} X(e^{j\omega T}) e^{j\omega nT} d\omega \tag{7.38}$$

which is called the IDTFT. A sufficient condition for the existence of the DTFT is

$$\left\| X(e^{j\omega T}) \right\| = \left\| \sum_{n=-\infty}^{+\infty} x(nT) e^{-j\omega nT} \right\| \leq \sum_{n=-\infty}^{+\infty} \|x(nT)\| < \infty$$

which means that $x(nT)$ must be absolutely summable. We say that $x(nT)$ and $X(e^{j\omega T})$ are a DTFT pair, denoted by $x(nT) \leftrightarrow X(e^{j\omega T})$.

With Equation 7.33, we have

$$X(e^{j\omega T}) = \frac{1}{T} \sum_{r=-\infty}^{+\infty} X(j(\omega - r\omega_s)) \tag{7.39}$$

With Equation 7.39 we can find information about the spectral density of $x(t)$ from the DTFT of $x(nT)$.

7.1.11 DTFT Model

The DTFT is often defined based on the block diagram shown in Figure 7.14, where the impulse sampler is an impulse train $\delta_T(t)$ given by

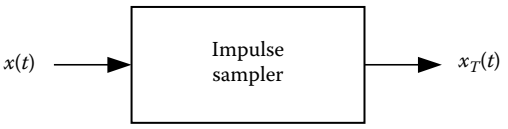


FIGURE 7.14 Model of the sampling process.

$$\delta_T(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$

and $\delta(t)$ is the Dirac impulse function. The impulse sampler output is

$$x_T(t) = x(t)\delta_T(t) = \sum_{n=-\infty}^{+\infty} x(nT)\delta(t - nT) \tag{7.40}$$

and the FT of $x_T(t)$ gives Equation 7.37, which is the DTFT of $x(nT)$.

7.1.12 Sampling Theorem

Suppose a signal $x(t)$ is bandlimited. Then, there is some ω_c such that $X(j\omega) = 0$ for $|\omega| > \omega_c$. Let us use the spectral density shown in Figure 7.15 to illustrate what happens when we apply the DTFT.

If we sample $x(t)$, then according to Equation 7.39 the DTFT of $x(nT)$ is the periodic extension of $X(j\omega)$, as shown in Figure 7.16. Here, the sampling frequency is high enough to prevent translations of $X(j\omega)$ from overlapping, which means that the spectral density $X(j\omega)$ is contained entirely within the frequency range $(-\omega_s/2, +\omega_s/2)$ of the DTFT of $x(nT)$. Therefore, if

$$\omega_s > 2\omega_c \tag{7.41}$$

then $X(j\omega)$ can be recovered from one period of $X(e^{j\omega T})$ without error.

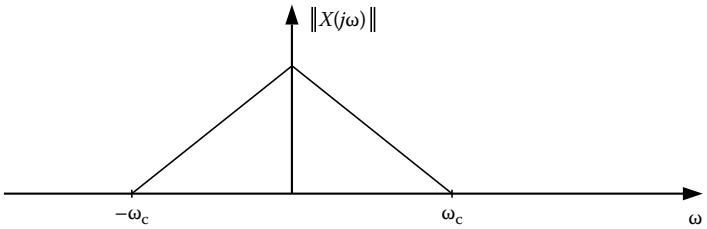


FIGURE 7.15 Spectral density magnitude of a bandlimited signal.

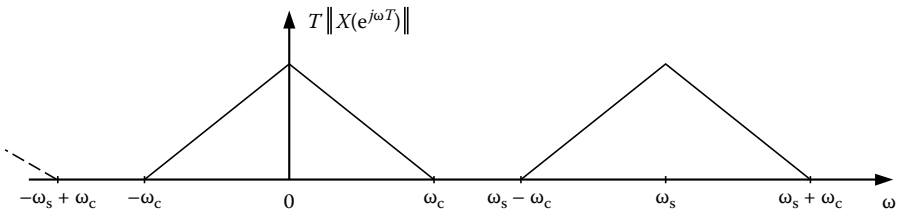


FIGURE 7.16 Magnitude of the DTFT.

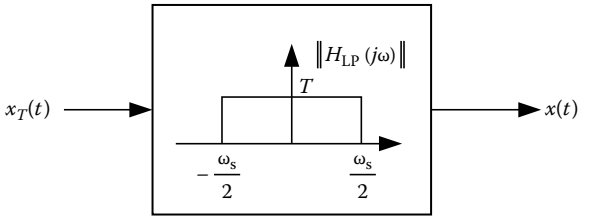


FIGURE 7.17 Ideal low-pass filter magnitude frequency response.

Since, for example, Figure 7.16 also shows the magnitude of the FT of $x_T(t)$, according to Equation 7.40, we can obtain the signal $x(t)$, which has the spectral density shown in Figure 7.15, if we process $x_T(t)$ with an analog low-pass filter as shown in Figure 7.17.

We will find $x(t)$ by convolving the input with the impulse response of the ideal low-pass filter. The impulse response $h_{LP}(t)$ is given by

$$h_{LP}(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} H_{LP}(j\omega)e^{j\omega t}d\omega = \frac{1}{2\pi}\int_{-\omega_s/2}^{+\omega_s/2} Te^{j\omega t}d\omega = \frac{\sin\left(\frac{\omega_s}{2}t\right)}{\frac{\omega_s}{2}t}$$

and then we get

$$x(t) = h_{LP}(t)^* x_T(t) = \sum_{n=-\infty}^{+\infty} x(nT)\frac{\sin\left(\frac{\omega_s}{2}(t-nT)\right)}{\frac{\omega_s}{2}(t-nT)} \tag{7.42}$$

which is called the data reconstruction formula, to obtain $x(t)$, for all t, from the samples $x(nT)$. Together, Equations 7.41 and 7.42 are called the time sampling theorem.

If the sampling frequency ω_s does not satisfy Equation 7.41, then the DTFT produces results like those shown in Figure 7.18. Here we see translations of $X(j\omega)$ overlap. In the frequency range $\omega_s - \omega_c$ to $\omega_s/2$, where $\omega_c > \omega_s/2$, the DTFT of $x(nT)$ is not the same as $X(j\omega)$. This difference is called aliasing error. If we reproduce an analog signal with Equation 7.42, then this analog signal will have energy in the frequency range $\omega_s - \omega_c$ to $\omega_s/2$ that is not the same as that of the given signal $x(t)$. In this situation, if the sampling frequency ω_s cannot be increased to avoid aliasing error, then the given signal should be processed by a low-pass filter, called an antialiasing filter, with bandwidth equal to $\omega_s/2$, which becomes the effective bandwidth of the signal that will be sampled. If the resulting samples are used to reconstruct an analog signal, then the resulting analog signal will not have the very high frequency spectral content of the given signal $x(t)$ in the frequency range $\omega_s/2$ to ω_c.

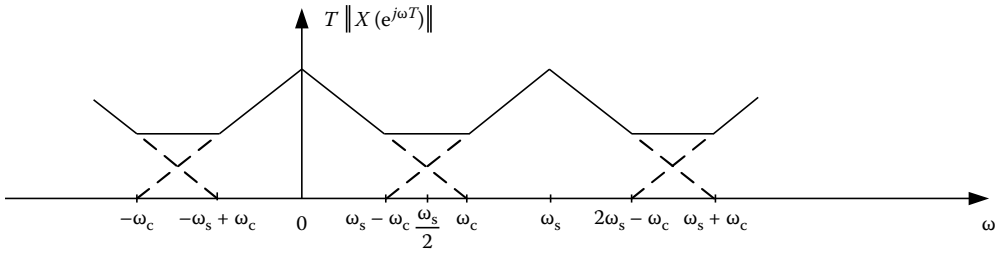


FIGURE 7.18 Magnitude of the DTFT, $\omega_s < 2\omega_c$.

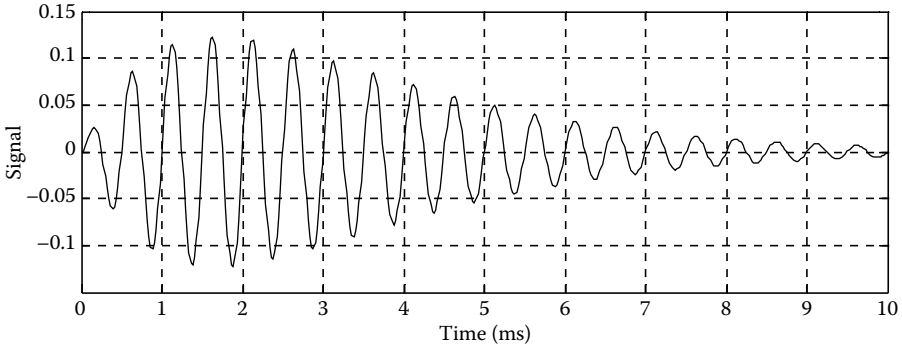


FIGURE 7.19 An exponentially weighted sinusoidal pulse.

If an analog signal $x(t)$ is not bandlimited, then aliasing error cannot be avoided. An antialiasing filter could be used at the cost of eliminating the high frequency sinusoidal components of the signal. Let us use the signal $x(t) = (e^{-at} - e^{-bt})\sin(2\pi f_0 t)u(t)$, where $a = 500$, $b = 700$, and $f_0 = 2000$ Hz, to illustrate what happens. This signal is shown in Figure 7.19. We see that $x(t)$ is a tone that gains strength and then loses strength. The FT of $e^{-at}u(t)$ is $1/(j\omega + a)$, and then the modulation property can be used to find the FT of $x(t)$.

The DTFT of $e^{-anT}u(n)e^{-j\omega_0 nT} = (e^{-aT}e^{-j\omega_0 T})^n u(n)$ is given by

$$\sum_{n=0}^{+\infty} e^{-(aT+j\omega_0 T)n}e^{-j\omega nT} = \lim_{N\to\infty}\sum_{n=0}^{N-1}(e^{-(a+j\omega_0+j\omega)T})^n = \lim_{N\to\infty}\frac{1-(e^{-(a+j\omega_0+j\omega)T})^N}{1-e^{-(a+j\omega_0+j\omega)T}}$$
$$= \frac{1}{1-e^{-aT}e^{-j(\omega+\omega_0)T}} \tag{7.43}$$

The magnitudes of $X(j\omega)$ and $X(e^{j\omega T})$ are shown in Figure 7.20, where they are shown normalized to show them on the same scale. Within the frequency range 0 to $f_s/2$ we see aliasing error that increases as the frequency increases toward $f_s/2$. If we increase the sampling frequency, then the aliasing error will decrease.

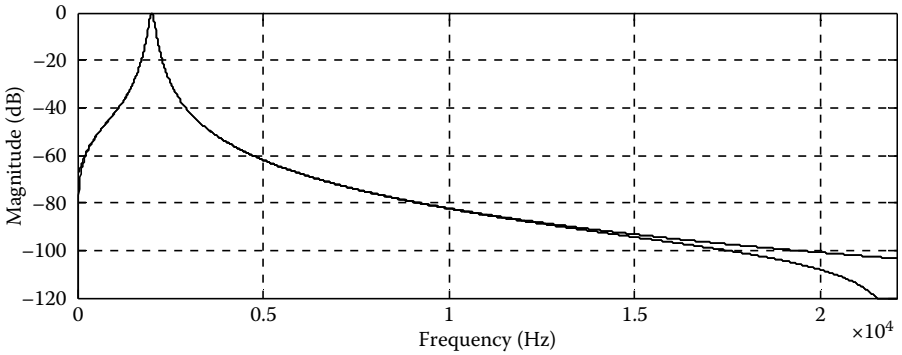


FIGURE 7.20 Magnitude of the FT and the DTFT, for $f_s = 44.1$ kHz.

TABLE 7.2 DTFT Properties

Linearity	$c_1x_1(nT)+c_2x_2(nT) \leftrightarrow c_1X_1(e^{j\omega T})+c_2X_2(e^{j\omega T})$
Time shift	$x(nT-n_0T) \leftrightarrow X(e^{j\omega T})e^{-j\omega n_0 T}$
Modulation	$x(nT)e^{j\omega_0 nT} \leftrightarrow X(e^{j(\omega-\omega_0)T})$
Linear convolution	$x_1(nT)^*x_2(nT)=\sum_{i=-\infty}^{+\infty} x_1((n-i)T)x_2(iT) \leftrightarrow X_1(e^{j\omega T})X_2(e^{j\omega T})$
Difference	$x(nT)-x((n-1)T) \leftrightarrow (1-e^{j\omega T})X(j\omega)$
Multiplication	$x_1(nT)x_2(nT) \leftrightarrow \frac{1}{2\pi}X_1(e^{j\omega T}) \otimes X_2(e^{j\omega T})$
Energy (Parseval's theorem)	$\sum_{n=-\infty}^{+\infty} \lvert x(nT)\rvert^2 = \frac{1}{2\pi}\int_0^{\omega_s} \lVert X(e^{j\omega T})\rVert^2 d\omega$

7.1.13 DTFT Properties

The DTFT has properties, most of which are analogous to properties of the FT. In Table 7.2 are given a few important properties of the DTFT.

7.1.14 Example and Discussion

For practical signals we only have a finite number of samples of $x(nT)$, say N samples, and therefore we must truncate the summation in Equation 7.37, and compute $X(e^{j\omega T})$ by digital means. This means that we can find the DTFT for only a finite number of frequencies. Let us evaluate the truncated DTFT at the frequencies given by

$$\omega = k\frac{\omega_s}{N}, \quad k = 0, 1, \ldots, N-1$$

and then the DTFT becomes

$$X(e^{j\omega T})\big|_{\omega=k\frac{\omega_s}{N}} = \sum_{n=0}^{N-1} x(nT)e^{-jk\frac{\omega_s}{N}nT} = \sum_{n=0}^{N-1} x(nT)e^{-j\frac{2\pi}{N}kn} = X(k) \tag{7.44}$$

which is the DFT of $x(nT)$, $n=0, 1, \ldots, N-1$.

Let us window the signal shown in Figure 7.19. Regardless of how wide the window is, the windowed signal $x_p(t)$ is one period of a periodic signal to which we are applying the DFT. Therefore, we should consider using a window such as $w_H(t)$ over the time range $0 \le t < T_0$. From Figure 7.19 we see that using $T_0 = 5.0$ ms does not incur discontinuities at period boundaries of the periodic extension of $x_p(t)$. Let us obtain $N = 226$ samples of $x(t)$ at the same rate $f_s = 44.1$ kHz. Then the duration of $x_p(t)$ becomes $T_0 = 5.1247$ ms. Figure 7.21 shows the magnitude of the FT of $x(t)$ and its approximation that was found with the DFT. While the DFT produces a peak near f_0, there is aliasing error above and below this frequency. This can be improved if $N(T_0)$ is increased.

7.1.15 Linear and Time Invariant Discrete-Time Systems

The discrete-time system (DTS) that we shall study is described by the block diagram shown in Figure 7.22, where $x(n)$ is the input to the system and $y(n)$ is the response or output of the system. We assume that the output and input are related by the difference equation given by

$$y(n)+a_1y(n-1)+\cdots+a_Ny(n-N)=b_0x(n)+b_1x(n-1)+\cdots+b_Mx(n-M) \tag{7.45}$$

where the a_i, $i=1,\ldots,N$ and b_i, $i=0,1,\ldots,M$ are real constants.

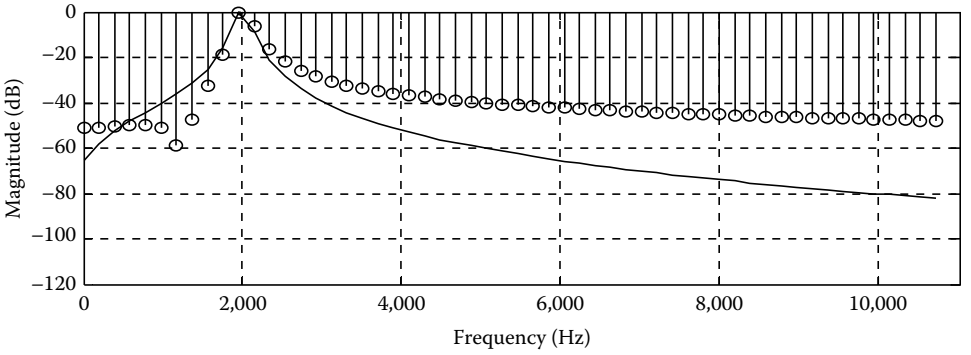


FIGURE 7.21 DFT approximation of the FT of $x(t)$.

To operate the DTS, reorganize Equation 7.45 to express the present output in terms of past outputs and present and past inputs, and we get

$$y(n) = \sum_{i=0}^{M} b_i x(n-i) - \sum_{i=1}^{N} a_i y(n-i) \qquad (7.46)$$

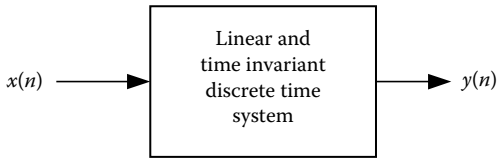


FIGURE 7.22 Block diagram of a DTS.

Suppose an input is applied starting at some time, say $n=0$. Then, to compute $y(n=0)$ given the first input $x(n=0)$, we must know

$$y(-1), y(-2), \ldots, y(-N)$$

which are the outputs prior to applying the input. These outputs are called the initial conditions of the DTS. Having computed $y(0)$, we can then compute $y(n=1)$, once we receive $x(n=1)$, and this can be continued for as long as we want to operate the DTS.

If in Equation 7.45 the right-hand side includes terms such as $b_{-1}x(n+1), b_{-2}x(n+2), \ldots$, then to compute a present output $y(n)$, knowledge of future inputs is required, and DTS cannot be operated in real time. Such a system is said to be noncausal. A system is said to be causal if a present output does not depend on a future value of the input. To operate a system in real time, it must be a causal system. However, when an entire history of an input is known, then at any time within this history, future values of the input are known, and a noncausal system can be operated. Both a causal and a noncausal system can be operated this way, which is referred to as off-line processing.

The DTS described by Equation 7.45 is a LTI system. Suppose all initial conditions are zero; we say the system is initially at rest. Let $y(n)=y_1(n)$ denote the response when the input is $x(n)=x_1(n)$, and similarly for $y_2(n)$ and $x_2(n)$. Then the DTS is a linear system if when the input is $x(n)=c_1x_1(n)+c_2x_2(n)$, then the response is given by $y(n)=c_1y_1(n)+c_2y_2(n)$ for any constants c_1 and c_2. The DTS is a time invariant system if when $y(n)=y_1(n)$ is the reponse to $x(n)=x_1(n)$, then $y(n)=y_2(n)=y_1(n-i)$ is the response to $x(n)=x_2(n)=x_1(n-i)$, for any time shift i.

A very useful function to know about an LTI DTS is its unit pulse response. This is the solution of Equation 7.45, which is denoted by $h(n)$, when $x(n)=\delta(n)$ and all initial conditions are zero. If the system is causal, then for $n<0$, we must have $h(n)=0$. To find $h(n)$, we first solve the simpler problem given by

$$v(n) + a_1 v(n-1) + \cdots + a_N v(n-N) = x(n) \qquad (7.47)$$

where initial conditions are zero and $x(n) = \delta(n)$. Therefore, we already have $v(0) = 1$. If we can determine the solution $v(n)$ of Equation 7.47, then, since the DTS is LTI and in view of the right-hand side of Equation 7.45, $h(n)$ is given by

$$h(n) = b_0 v(n) + b_1 v(n-1) + \cdots + b_M v(n-M) \tag{7.48}$$

For $n > 0$, Equation 7.47 becomes

$$v(n) + a_1 v(n-1) + \cdots + a_N v(n-N) = 0 \tag{7.49}$$

which is a homogeneous equation. Therefore, we can instead solve Equation 7.49 for initial conditions given by $v(0) = 1$, $v(-1) = 0, \ldots, v(N-1) = 0$. We want to find a function that can satisfy Equation 7.49. Let us try $v(n) = K\gamma^n$, for some nonzero constants K and γ. Substituting this $v(n)$ into Equation 7.49 gives

$$K\gamma^n + a_1 K\gamma^{n-1} + a_2 K\gamma^{n-2} + \cdots + a_N K\gamma^{n-N} = 0$$

and after multiplying by K^{-1} and γ^{-n} we get

$$Q(\gamma^{-1}) = 1 + a_1\gamma^{-1} + a_2\gamma^{-2} + \cdots + a_{N-1}\gamma^{-(N-1)} + a_N\gamma^{-N} = 0 \tag{7.50}$$

which is called the characteristic equation of the DTS, and $Q(\gamma^{-1})$ is called the characteristic polynomial. The characteristic equation has N roots, γ_i, $i = 1, 2, \ldots, N$, which can be real or complex, and Equation 7.50 can be factored to become

$$Q(\gamma^{-1}) = \prod_{i=1}^{N} (1 - \gamma_i\gamma^{-1}) = 0$$

Notice that these roots are not determined by the input. They are a characteristic of the system. Let us assume that they are distinct. Since each function γ_i^n can satisfy Equation 7.49, the solution is a linear combination of these exponential functions, and we have

$$v(n) = \sum_{i=1}^{N} K_i\gamma_i^n$$

The N constants K_i can be found by setting up N equations in N unknowns by applying the initial conditions of Equation 7.49. With $v(n)$ and Equation 7.48 we can then obtain the unit pulse response $h(n)$.

It is important to note that the unit pulse response is a linear combination of exponential functions with behavior that is determined by the roots of the characteristic equation. These roots can be real or they can occur in complex conjugate pairs. For example, if a root γ is real, then $h(n)$ will contain the term $K\gamma^n$ that may decrease exponentially to zero if $||\gamma|| < 1$, remain at unity if $\gamma = 1$, or increase exponentially if $||\gamma|| > 1$. Or, if a complex conjugate pair of roots occurs, then $h(n)$ will contain the sum of terms $K\gamma^n + K^*(\gamma^*)^n = 2||K||\ ||\gamma||^n \cos(n\angle\gamma + \angle K)$ that will oscillate and increase or decrease exponentially. If we require that $\lim_{n\to\infty} h(n) = 0$, then all roots of the characteristic equation must satisfy $||\gamma_i|| < 1$, $i = 1, 2, \ldots, N$.

A DTS can be categorized into two very different classes. Suppose that in Equation 7.45, $a_i = 0$ and $i = 1, 2, \ldots, N$. Then, Equation 7.46 becomes

$$y(n) = \sum_{i=0}^{M} b_i x(n-i) \tag{7.51}$$

and the present output depends on present and past inputs. If the input has a finite duration, then the response will have a finite duration. In fact, if $x(n) = \delta(n)$, then the unit pulse response is given by

$$h(n) = \sum_{i=0}^{M} b_i \delta(n-i)$$

which has a finite duration. Therefore, the DTS described by Equation 7.51 is called an FIR (finite impulse response) system, because the unit pulse response has a finite duration. However, if at least one a_i in Equation 7.45 is not zero, then in Equation 7.46 the present output will depend on the past outputs, which means there is feedback in the system. In this case the response to $x(n) = \delta(n)$ can never become and remain zero indefinitely. Such a DTS is called an IIR (infinite impulse response) system, because the response to a finite duration input will have an infinite duration.

7.1.16 Convolution

We can find the response of LTI DTS initially at rest to any input by writing the input as in Equation 7.8. Since the DTS is LTI, the response to $x(i)\delta(n-i)$ is $x(i)h(n-i)$, and the response to $x(n)$ is given by

$$y(n) = \sum_{i=-\infty}^{+\infty} h(n-i)x(i) = h(n)^* x(n) = \sum_{i=-\infty}^{+\infty} h(i)x(n-i) \tag{7.52}$$

which is called the linear convolution operation. Since the DTS is time invariant, the response to $x(n-n_0)$ is

$$y(n-n_0) = h(n)^* x(n-n_0)$$

7.1.17 Stability

A DTS is said to be bounded-input-bounded-output (BIBO) stable if for a bounded input the output is bounded. If the input is bounded, then for some real and positive number $x_{\max}$, we have $||x(n)|| \leq x_{\max}$ for all n. For a bounded output, Equation 7.52 becomes

$$\|y(n)\| = \left\| \sum_{i=-\infty}^{+\infty} h(i)x(n-i) \right\| \leq \sum_{i=-\infty}^{+\infty} \|h(i)x(n-i)\| \leq \sum_{i=-\infty}^{+\infty} \|h(i)\| x_{\max} = x_{\max} \sum_{i=-\infty}^{+\infty} \|h(i)\| < \infty$$

A sufficient condition for BIBO stability of an LTI DTS is that the unit pulse response must be absolutely summable.

Recall that the nature of the unit pulse response is determined by the roots of the characteristic equation (Equation 7.50). If all roots satisfy the condition $||\gamma_i|| < 1$, then we can show that $h(n)$ is absolutely summable. Moreover, we then have $\lim_{n\to\infty} h(n) = 0$. This means that if we apply a finite duration input to an initially at rest system, then eventually, the system will approach the rest state. An important distinction between FIR and IIR systems is that an FIR system is unconditionally stable, while an IIR system can be stable or unstable, depending on the roots of the characteristic equation.

7.1.18 Frequency Response

A major concern is the steady-state response of the DTS described by Equation 7.46 when the input is a sinusoid given by

$$x(n) = A\cos(\omega nT + \phi)$$

If the DTS is stable, then, in steady state, the response will be a sinusoid. It will be easier to determine the steady-state response if we work with an input $v(n)$ defined by

$$v(n) = e^{j\omega nT}$$

and let $w(n)$ denote the response to $v(n)$. Since

$$x(n) = \frac{A}{2}e^{j\phi}v(n) + \frac{A}{2}e^{-j\phi}v^{*}(n)$$

and since the DTS is a linear system, then the response $y(n)$ to $x(n)$ must be

$$y(n) = \frac{A}{2}e^{j\phi}w(n) + \frac{A}{2}e^{-j\phi}w^{*}(n)$$

With Equation 7.52, $w(n)$ is given by

$$\begin{aligned} w(n) &= \sum_{i=-\infty}^{+\infty} h(i)v(n-i) = \sum_{i=-\infty}^{+\infty} h(i)e^{j\omega(n-i)T} = e^{j\omega nT}\sum_{i=-\infty}^{+\infty} h(i)e^{-j\omega iT} \\ &= H(e^{j\omega T})e^{j\omega nT} \end{aligned}$$

where $H(e^{j\omega T})$ is the DTFT of the unit pulse response $h(n)$, and therefore $y(n)$ becomes

$$y(n) = \frac{A}{2}e^{j\phi}H(e^{j\omega T})e^{j\omega nT} + \frac{A}{2}e^{-j\phi}H^{*}(e^{j\omega T})e^{-j\omega nT}$$

Expressing $H(e^{j\omega T})$ in polar form, then with Euler's identity we get

$$y(n) = A\left|H(e^{j\omega T})\right|\cos\left(\omega nT + \phi + \angle H(e^{j\omega T})\right) \tag{7.53}$$

Let us consider another approach to find $y(n)$. First, let us substitute $v(n)$ into Equation 7.45 to get

$$\begin{aligned} w(n) + a_1 w(n-1) + \cdots + a_N w(n-N) &= b_0 e^{j\omega nT} + b_1 e^{j\omega(n-1)T} + \cdots + b_M e^{j\omega(n-M)T} \\ &= (b_0 + b_1 e^{-j\omega T} + \cdots + b_M e^{-j\omega MT})e^{j\omega nT} \end{aligned}$$

Since the right-hand side of this equation is an exponential function, then the left-hand side must also be an exponential function. Let us try $w(n) = We^{j\omega nT}$, for some unknown constant W, and then we get

$$W(1 + a_1 e^{-j\omega T} + \cdots + a_N e^{-j\omega NT})e^{j\omega nT} = (b_0 + b_1 e^{-j\omega T} + \cdots + b_M e^{-j\omega MT})e^{j\omega nT}$$

Therefore, we can solve for W, and $w(n)$ is

$$w(n) = \frac{(b_0 + b_1 e^{-j\omega T} + \cdots + b_M e^{-j\omega MT})}{(1 + a_1 e^{-j\omega T} + \cdots + a_N e^{-j\omega NT})}e^{j\omega nT}$$

Then, $H(e^{j\omega T})$ is given by

$$H(e^{j\omega T}) = \frac{(b_0 + b_1 e^{-j\omega T} + \cdots + b_M e^{-j\omega MT})}{(1 + a_1 e^{-j\omega T} + \cdots + a_N e^{-j\omega NT})} \tag{7.54}$$

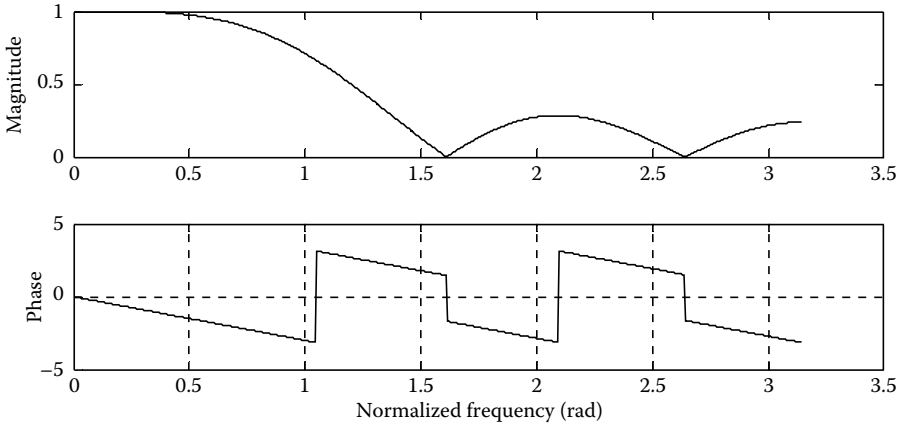


FIGURE 7.23 Magnitude and phase frequency response of an FIR filter.

which we can find by inspection of the given difference equation. In view of Equation 7.53 we have that $H(e^{j\omega T})$ gives the frequency response of the DTS. Notice that $H^*(e^{j\omega T}) = H(e^{-j\omega T})$, and therefore $||H(e^{j\omega T})||$ is an even periodic function of ω and $\angle H(e^{j\omega T})$ is an odd periodic function of ω.

7.1.19 Examples and Discussion

With Equation 7.54 we can assess the frequency selective behavior of an LTI DTS. Consider the FIR DTS given by

$$y(n) = 1/21[-2x(n) + 3x(n-1) + 6x(n-2) + 7x(n-3) + 6x(n-4) + 3x(n-5) - 2x(n-6)]$$

The frequency response is

$$H(\mathrm{e}^{j\omega T}) = 1/21[-2 + 3\mathrm{e}^{-j\omega T} + 6\mathrm{e}^{-j2\omega T} + 7\mathrm{e}^{-j3\omega T} + 6\mathrm{e}^{-j4\omega T} + 3\mathrm{e}^{-j5\omega T} - 2\mathrm{e}^{-j6\omega T}]$$

To see the frequency selective behavior of this DTS we could plot the magnitude of the frequency response for $0 \leq \omega \leq \omega_s/2$. Instead, let us plot the magnitude versus $\theta = \omega T$, where $0 \leq \theta \leq \pi$. Figure 7.23 shows the magnitude and phase frequency response, and we see that this DTS is a low-pass filter. Furthermore, this FIR filter has a phaseshift characteristic that varies linearly with frequency. The step changes in phase are due to wraparound when the phase changes from, for example, $-\pi$ to $+\pi$.

As a comparison, let us assess the frequency selective behavior of the IIR LTI DTS described by

$$y(n) = 0.09780\ x(n) + 0.19560\ x(n-1) + 0.09780\ x(n-2) + 0.94175\ y(n-1) - 0.33296\ y(n-2)$$

with $H(\mathrm{e}^{j\omega T})$ given by

$$H(\mathrm{e}^{j\omega T}) = \frac{0.09780 + 0.19560\mathrm{e}^{-j\omega T} + 0.09780\mathrm{e}^{-j2\omega T}}{1 - 0.94175\mathrm{e}^{-j\omega T} + 0.33296\mathrm{e}^{-j2\omega T}}$$

Figure 7.24 shows magnitude and phase frequency response of this IIR DTS. This is a low-pass filter, and the transition from the passband to the stopband is not very sharp. The phaseshift does not vary linearly with frequency.

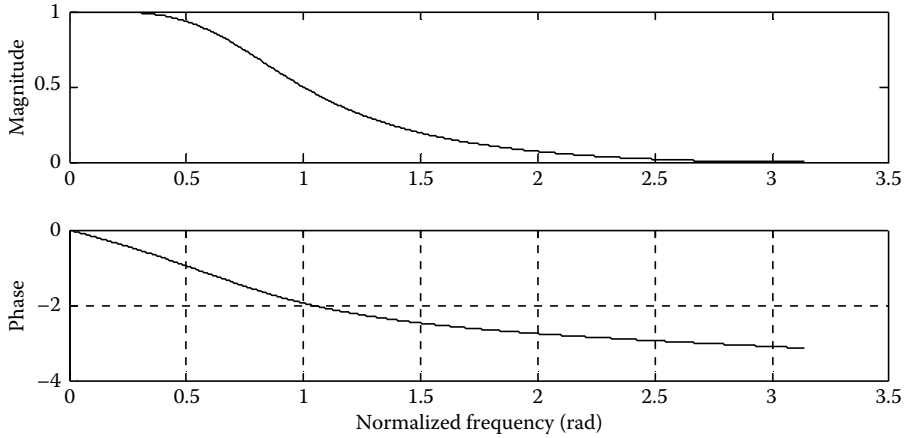


FIGURE 7.24 Magnitude and phase frequency response of an IIR filter.

These two filters have similar performance. Each has advantages and disadvantages compared to the other filter. For example, the FIR filter is unconditionally stable and it has a linear phase characteristic, and the IIR filter has a sharper magnitude frequency response with less computation than required by the FIR filter.

7.1.20 Ideal Digital Filters

We can design the coefficients of the digital filter described by Equation 7.45 to achieve filter performance that comes arbitrarily close to an ideal filter performance. Figure 7.25 shows the ideal performance of several standard filter types. The frequency response for each filter is specified over the frequency range, $-\omega_s/2 \leq \omega \leq \omega_s/2$, which is the frequency range of the filter input. Then, this frequency response is extended periodically to integer multiples of ω_s.

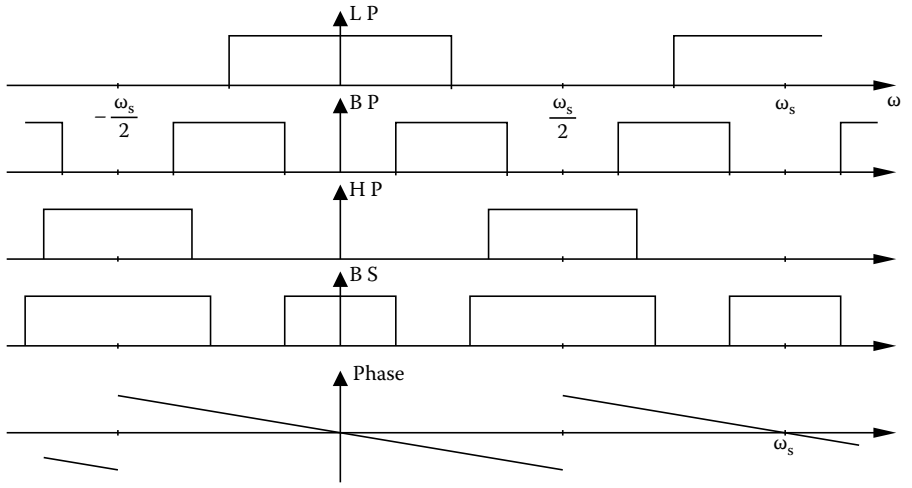


FIGURE 7.25 Ideal frequency response of a low-pass (LP), band-pass (BP), high-pass (HP), and band-stop (BS) digital filter.

To justify an ideal linear phase characteristic, let us consider an input $x(t) = A_1 \cos(\omega_1 t + \phi_1) + A_2 \cos(\omega_2 t + \phi_2)$ of an analog filter, where ω_1 and ω_2 are within the passband of the analog filter. To preserve the input wave shape, the output $y(t)$ can be a delayed version of the input. Now we have

$$\begin{aligned} y(t) = x(t - t_0) &= A_1 \cos(\omega_1(t - t_0) + \phi_1) + A_2 \cos(\omega_2(t - t_0) + \phi_2) \\ &= A_1 \cos(\omega_1 t + \phi_1 - t_0\omega_1) + A_2 \cos(\omega_2 t + \phi_2 - t_0\omega_2) \end{aligned}$$

and to preserve the input wave shape, the input to output phase change must be proportional to frequency.

7.1.21 *z*-Transform

While the DTFT is useful for spectral analysis, its wider application to system analysis is limited, because, for example, not even the DTFT of a discrete-time sinusoidal signal exists. Given a discrete-time signal $x(nT)$, let us consider instead the DTFT given by

$$Y(\sigma, e^{j\omega T}) = \text{DTFT}\{e^{-\sigma nT} x(nT)\} = \sum_{n=-\infty}^{+\infty} e^{-\sigma nT} x(nT) e^{-j\omega nT} \tag{7.55}$$

for some real number σ. Assuming that $Y(\sigma, e^{j\omega T})$ exists, taking its IDTFT gives

$$e^{-\sigma nT} x(nT) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{+\frac{\omega_s}{2}} Y(\sigma, e^{j\omega T}) e^{j\omega nT} d\omega$$

and after multiplying both sides of this equation by $e^{\sigma nT}$ we get

$$x(nT) = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{+\frac{\omega_s}{2}} e^{\sigma nT} Y(\sigma, e^{j\omega T}) e^{j\omega nT} d\omega \tag{7.56}$$

For convenience, let

$$z = e^{\sigma T} e^{j\omega T} = e^{(\sigma + j\omega)T} = \|z\| e^{j\angle z} \tag{7.57}$$

and let $X(z)$ denote $Y(\sigma, e^{j\omega T})$, and therefore, Equation 7.55 becomes

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n} \tag{7.58}$$

which is called the bilateral z-transform (BZT) of $x(n)$. We write $X(z) = Z\{x(n)\}$. In Equation 7.56, let us change the integration variable from the real variable ω to the complex variable z. When $\omega = -\omega_s/2$, $z = e^{\sigma T} e^{-j\pi}$, and when $\omega = +\omega_s/2$, $z = e^{\sigma T} e^{j\pi}$, and therefore z follows a circle of radius $||z|| = e^{\sigma T}$. We also have

$$\frac{dz}{d\omega} = jT e^{\sigma T} e^{j\omega T} = jTz$$

and therefore Equation 7.56 becomes

$$x(nT) = \frac{1}{\omega_s} \oint X(z) z^n \frac{dz}{jTz} = \frac{1}{j2\pi} \oint X(z) z^{n-1} dz \tag{7.59}$$

which is called the inverse z-transform, denoted by $x(n) = Z^{-1}\{X(z)\}$. Notice that the integration path is a circle with a radius determined by σ.

7.1.22 Bilateral z-Transform Properties

The BZT has many useful properties. For our development we will use the properties given in Table 7.3.

7.1.23 z-Plane

To see the utility of introducing the factor $e^{-\sigma nT}$ in Equation 7.55, let us obtain the BZT of $x(n) = \gamma^n u(n)$ for any real or complex number γ. With Equation 7.58 we have

$$X(z) = \sum_{n=-\infty}^{+\infty} \gamma^n u(n) z^{-n} = \sum_{n=0}^{+\infty} \left(\frac{\gamma}{z}\right)^n = \lim_{N\to\infty} \sum_{n=0}^{N-1} \left(\frac{\gamma}{z}\right)^n = \lim_{N\to\infty} \frac{1 - \left(\frac{\gamma}{z}\right)^N}{1 - \left(\frac{\gamma}{z}\right)} = \frac{1}{1 - \gamma z^{-1}}, \quad \left\|\frac{\gamma}{z}\right\| < 1$$

If $||\gamma|| > 1$, then $x(n)$ increases exponentially, and it does not have a DTFT. However, $x(n)$ does have a BZT, because we restrict z with σ to satisfy the convergence condition, $||z|| = e^{\sigma T} > ||\gamma||$, which requires $\sigma > \frac{1}{T} \ln\|\gamma\|$. If σ satisfies this condition, then the product $e^{-n\sigma T}\ \gamma^n$ decreases exponentially. The function $e^{-n\sigma T}$ is called the convergence factor.

We say that $x(n)$ and $X(z)$ are a BZT pair, denoted by

$$\gamma^n u(n) \leftrightarrow 1/(1 - \gamma z^{-1}), \quad \|z\| > \|\gamma\| \tag{7.60}$$

This transform pair includes signals such as

$$x(n) = u(n), \quad \gamma = 1$$
$$x(n) = \cos(\theta n) u(n) = \frac{1}{2} e^{j\theta n} u(n) + \frac{1}{2} e^{-j\theta n} u(n), \quad \gamma = e^{j\theta} \quad \text{and} \quad \gamma = e^{-j\theta}$$

and many other signals by applying BZT properties. For example, notice that Equation 7.60 can be obtained with $Z\{u(n)\}$ and the modulation property.

If $||\gamma|| < 1$, then $x(n)$ decreases exponentially, and it has a DTFT. It also has the BZT given in Equation 7.60. With $||\gamma|| < 1$, the condition $\sigma > \frac{1}{T} \ln\|\gamma\|$ allows σ to be negative, and with a negative σ the convergence factor increases exponentially, while the product $e^{-n\sigma T}\gamma^n$ still decreases exponentially. We can even set σ to $\sigma = 0$, which with Equation 7.57 converts the BZT to the DTFT.

TABLE 7.3 BZT Properties

Linearity	$c_1 x_1(n) + c_2 x_2(n) \leftrightarrow c_1 X_1(z) + c_2 X_2(z)$
Time shift	$x(n - n_0) \leftrightarrow X(z) z^{-n_0}$
Modulation	$x(n)\alpha^n \leftrightarrow X(z\alpha^{-1})$
Linear convolution	$x_1(n) * x_2(n) = \sum_{i=-\infty}^{+\infty} x_1(n-i) x_2(i) \leftrightarrow X_1(z) X_2(z)$

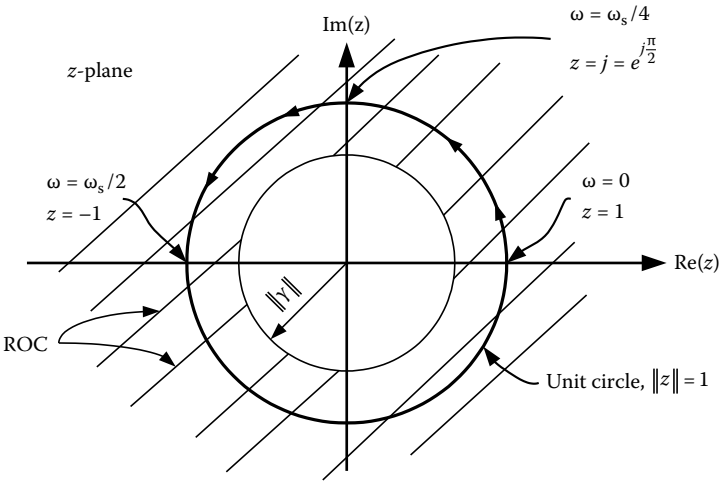


FIGURE 7.26 The z-plane, showing the unit circle and an ROC.

These different situations can be nicely pulled together by describing them with respect to the unit circle in the z-plane, as shown in Figure 7.26. The convergence condition, $||z|| > ||\gamma||$, does not impose any restriction on the angle of z. Therefore, the BZT of $x(n) = \gamma^n u(n)$ converges for all z outside a circle in the z-plane having a radius equal to $||\gamma||$, and this region of the z-plane is called the region of convergence (ROC). If $||\gamma|| < 1$, which means that $x(n)$ has a DTFT, then the ROC is outside a circle that is inside the unit circle in the z-plane (see Figure 7.26). However, if $||\gamma|| > 1$, which means that the DTFT of $x(n)$ does not exist, then the ROC is outside a circle that is outside the unit circle. This means that if the ROC of the BZT includes the unit circle, then $X(z)$ can be evaluated for $z = e^{j\omega T}$ ($||z|| = 1$), and a BZT can be converted to a DTFT by setting z to $z = e^{j\omega T}$.

The ROC is an important information about the BZT of a signal, because there can be an ambiguity between $X(z)$ and its inverse z-transform. To see why this is true, let us find the BZT of $x(n) = -\gamma^n u(-n-1)$, which is zero for $n \geq 0$. Applying Equation 7.58 gives

$$X(z) = \sum_{n=-\infty}^{+\infty} -\gamma^n u(-n-1) z^{-n} = -\sum_{n=-\infty}^{-1} \gamma^n z^{-n} = -\sum_{n=1}^{+\infty} (\gamma^{-1} z)^n = -\sum_{n=0}^{+\infty} (\gamma^{-1} z)^n + 1$$
$$= 1 - \frac{1}{1-\gamma^{-1} z} = \frac{1}{1-\gamma z^{-1}}, \left\|\gamma^{-1} z\right\| < 1 \rightarrow \|z\| < \|\gamma\| \tag{7.61}$$

Here the ROC is the interior of a circle in the z-plane. If $||\gamma|| > 1$, then the ROC includes the unit circle and $x(n)$ decreases exponentially as $n \rightarrow -\infty$, which means that the DTFT of $x(n)$ exists. However, if $||\gamma|| < 1$, then the ROC does not include the unit circle and $x(n)$ increases exponentially as $n \rightarrow -\infty$, which means that the DTFT of $x(n)$ does not exist.

If the BZT of Equation 7.60 or 7.61 is given, we cannot know the inverse z-transform unless we also know the convergence condition. In Equation 7.60, $x(n)$ is said to be a right-handed signal, and its ROC is a region in the z-plane outside a circle, and in Equation 7.61, $x(n)$ is said to be a left-handed signal, and its ROC is a region in the z-plane inside a circle. Consider how the ROC must change if we time reverse a signal. For either kind of signal, right-handed or left-handed, if the ROC includes the unit circle, then it has a DTFT. We will work with right-handed signals unless explicitly stated otherwise, in which case the ROC is the type given in Equation 7.60 and illustrated in Figure 7.26.

7.1.24 Transfer Function

Given an input $x(n)$ of an LTI DTS, the convolution Equation 7.52 of the input with the unit pulse response $h(n)$ gives the response of the DTS under zero initial conditions. Applying the convolution property in Table 7.3 to Equation 7.52 gives

$$Y(z) = H(z)X(z) \tag{7.62}$$

where $H(z) = Z\{h(n)\}$, which is called the transfer function of the system. We assume that the DTS is causal, and therefore $h(n)$ is a right-handed signal. If $x(n) = \delta(n)$, then $X(z) = 1$, and, as expected, $y(n) = h(n)$.

Let us apply the linearity and time-shift properties of the BZT, and take the BZT term-by-term of Equation 7.45 to get

$$Y(z) + a_1 Y(z)z^{-1} + \cdots + a_N Y(z)z^{-N} = b_0 X(z) + b_1 X(z)z^{-1} + \cdots + b_M X(z)z^{-M}$$

Having converted the difference equation to an algebraic equation permits us to find

$$Y(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} X(z) \tag{7.63}$$

and therefore, $H(z)$ in Equation 7.62 is given by

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \frac{P(z^{-1})}{Q(z^{-1})} = \frac{b_0 \prod_{i=1}^{M} (1 - z_i z^{-1})}{\prod_{i=1}^{N} (1 - p_i z^{-1})} \tag{7.64}$$

The z_i, $i = 1, \ldots, M$ are called the zeros of $H(z)$, and the p_i, $i = 1, \ldots, N$ are called the poles of $H(z)$.

Comparing Equations 7.64 and 7.50, we see that the denominator of $H(z)$ is the characteristic polynomial, and therefore the poles of $H(z)$ are the roots of the characteristic equation, which determine the behavior of the unit pulse response. Therefore, $h(n)$ is a linear combination of the exponential functions, p_i^n, $i = 1, 2, \ldots, N$. This means that the DTS is stable if all the poles of $H(z)$ are inside the unit circle in the z-plane. Furthermore, the ROC of $H(z) = Z\{h(n)\}$ is outside a circle with a radius equal to the magnitude of the pole having the largest magnitude among all the poles of $H(z)$. Therefore, if the DTS is stable, $H(z)$ can be converted to the DTFT with $z = e^{j\omega T}$.

Comparing Equations 7.64 and 7.54, we see that the frequency response of the DTS can be found by evaluating $H(z)$ on the unit circle, where $z = e^{j\omega T}$. This is depicted in Figure 7.26, where for $\omega = 0$, $z = 1$. As ω is increased from $\omega = 0$ to $\omega = \omega_s/2$, the angle of z, $\angle z = \omega T$, changes from $\angle z = 0$, where $z = 1$, to $\angle z = \pi$, where $z = -1$.

7.1.25 Unilateral z-Transform

Consider a signal $x(n)$, which is defined over $-\infty < n < +\infty$. Generally, $x(n)$ is not a right-handed or a left-handed signal. Let us express $x(n)$ as

$$x(n) = x_l(n) + x_r(n) = x(n)u(-(n+1)) + x(n)u(n)$$

where $x_l(n) = 0$, $n \geq 0$ and $x_r(n) = 0$, $n < 0$, so as to have $x_l(n)$ account for the left-handed part of $x(n)$, while $x_r(n)$ accounts for the right-handed part of $x(n)$. The BZT of $x(n)$ is

$$X(z) = \sum_{n=-\infty}^{-1} x(n)z^{-n} + \sum_{n=0}^{+\infty} x(n)z^{-n} = Z\{x_l(n)\} + Z\{x_r(n)\} = X_l(z) + X_r(z) \tag{7.65}$$

Suppose we are only interested to observe a signal $x(n)$ for $n \geq 0$, even if it is not strictly a right-handed signal. Then, with

$$Z^{-1}\{X(z)\} = x(n)$$

we can find $x(n)$ for $n \geq 0$, and we also get $x(n)$ for $n < 0$. However, we can also find $x(n)$ for $n \geq 0$ with

$$Z^{-1}\{X_r(z)\} = x_r(n)$$

Therefore, we will now only work with

$$X(z) = \sum_{n=0}^{+\infty} x(n)z^{-n} \tag{7.66}$$

which is called the unilateral z-transform, or just z-transform (ZT). This does not imply that $x(n)$ for $n < 0$ must be zero. Instead, we prefer to avoid involving $X_l(z)$.

The bilateral and unilateral z-transforms have many properties in common, while for some properties the difference is very useful. Consider the ZT of $y(n) = x(n-1)$. Notice that $y(0) = x(-1)$, and therefore

$$Z\{y(n) = x(n-1)\} = \sum_{n=0}^{+\infty} y(n)z^{-n} = \sum_{n=0}^{+\infty} x(n-1)z^{-n} = x(-1) + z^{-1}\sum_{n=0}^{+\infty} x(n)z^{-n}$$

Generally, we have

$$Z\{x(n-i)\} = x(-i) + z^{-1}Z\{x(n-(i-1))\}, \quad i \geq 1 \tag{7.67}$$

which is not the same as the time-shift property given in Table 7.3 for the BZT.

Unlike BZT, ZT can be used to find the complete response of an LTI DTS as described by Equation 7.45. We assume that the input $x(n)$ is zero for $n < 0$. Taking the term-by-term ZT of Equation 7.45 gives

$$\begin{aligned} Y(z) + a_1\big(y(-1) + z^{-1}Y(z)\big) + \cdots + a_N\big(y(-N) + z^{-1}Z\{y(n-(N-1))\}\big) \\ = b_0X(z) + b_1z^{-1}X(z) + \cdots + b_Mz^{-M}X(z) \end{aligned}$$

which can be rearranged to become

$$(1 + a_1z^{-1} + a_2z^{-2} + \cdots + a_Nz^{-N})Y(z) - Y_{IC}(z^{-1}) = (b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_Mz^{-M})X(z)$$

where $Y_{IC}(z^{-1})$ accounts for all initial condition terms. If all initial conditions are zero, then $Y_{IC}(z^{-1}) = 0$. Solving for $Y(z)$ results in

$$Y(z) = \frac{Y_{\mathrm{IC}}(z^{-1})}{Q(z^{-1})} + H(z)X(z) \tag{7.68}$$

We can factor the denominator of $X(z)$, and write

$$X(z) = \frac{P_x(z^{-1})}{\prod_{i=1}^{N_x}(1 - p_{x,i}z^{-1})}$$

where the $p_{x,i}$, $i = 1, \ldots, N_x$, are the N_x poles of $X(z)$. Therefore, Equation 7.68 becomes

$$Y(z) = \frac{Y_{\mathrm{IC}}(z^{-1})}{\prod_{i=1}^{N}(1 - p_i z^{-1})} + \frac{P(z^{-1})}{\prod_{i=1}^{N}(1 - p_i z^{-1})} \frac{P_x(z^{-1})}{\prod_{i=1}^{N_x}(1 - p_x z^{-1})}$$

Let us assume that the poles of $H(z)$ and the poles $X(z)$ are distinct, and then a partial fraction expansion of the right-hand side of this equation has the form

$$Y(z) = \sum_{i=1}^{N} \frac{K_{\mathrm{IC},i}}{1 - p_i z^{-1}} + \sum_{i=1}^{N} \frac{K_i}{1 - p_i z^{-1}} + \sum_{i=1}^{N_x} \frac{K_{x,i}}{1 - p_{x,i} z^{-1}}$$

The complete solution $y(n)$ of Equation 7.45 is given by

$$y(n) = \sum_{i=1}^{N} K_{\mathrm{IC},i}(p_i)^n u(n) + \sum_{i=1}^{N} K_i(p_i)^n u(n) + \sum_{i=1}^{N_x} K_{x,i}(p_{x,i})^n u(n) \tag{7.69}$$

If the input $x(n)$ is zero, then $y(n)$ in Equation 7.69 consists of only the first sum term, which is called the zero-input response. If all initial conditions are zero, then $y(n)$ in Equation 7.69 consists of the second and third sum terms, which together are called the zero-state response. Furthermore, the first and second sum terms are due to the poles of $H(z)$, and together they are called the transient response. There can be a transient response even if all initial conditions are zero.

The third sum term in Equation 7.69 is a discrete-time function like the input, and it is called the steady-state response. In digital filtering a signal, this is the part of the complete response that is of main concern. We prefer that the transient response quickly decays to negligible levels. If the input is, for example, a single sinusoid, then $N_x = 2$, and the third sum term in Equation 7.69 will consist of the sum of two complex conjugate exponential functions that combine via Euler's identity to become a sinusoid, as given in Equation 7.53.

7.1.26 Conclusion

The concept of a FS was developed as the minimum mean square error solution for the problem of approximating a continuous-time periodic signal with a linear combination of sinusoidal functions. If the continuous-time periodic signal is everywhere continuous, then the approximation error is zero. By sampling FS, DFT was obtained, and its relationship to the spectral nature of the continuous-time signal was found. Depending on the bandwidth of the continuous-time signal, the sampling rate and the sampling duration, we came to understand the kinds of errors that can occur when the DFT is used to study the spectral nature of a signal. The DFT is a widely applied method of signal and system analysis, because the DFT can be efficiently computed with the FFT, which was demonstrated by the development of the decimation in time radix-2 FFT.

The FS concept was extended to aperiodic continuous-time signals, resulting in the FT. By sampling the inverse FT, the DTFT was obtained, and its relationship to the spectral nature of the continuous-time signal was found. The sampling theorem came by studying this relationship. The DTFT was applied to signal and system analysis, and for practical reasons, the DFT again played an important role.

To broaden the class of signals to which the DTFT can be applied, a convergence factor was introduced, resulting in the BZT. The relationship between the DTFT and BZT was thoroughly investigated to understand the role of the BZT in spectral analysis. The BZT was modified to work with right-handed signals resulting in the unilateral z-transform, or just ZT. The ZT was applied to find the complete response of a DTS, a main advantage of the ZT.

The next step for the reader may be to investigate methods for the design of digital filters, as can be found in the references and elsewhere, where the DFT, DTFT and ZT, which are all based on the concept of a FS, play major roles.

References

Ambardar, A., 2007, *Digital Signal Processing: A Modern Introduction*, Thomson, Publishing, Washington, DC.

Antoniou, A., 1993, *Digital Filters*, McGraw-Hill, New York.

Bracewell, R.N., 1978, *The Fourier Transform and Its Applications*, McGraw-Hill, New York.

McClellan, J.H., Schafer, R.N., and Yoder, M.A., 1997, *DSP First: A Multimedia Approach*, Prentice-Hall, Upper Saddle River, NJ.

Oppenheim, A.V. and Schafer, R.W., 1975, *Discrete Time Signal Processing*, Prentice-Hall, Upper Saddle River, NJ.

Orfanidis, S.J., 1996, *Introduction to Signal Processing*, Prentice-Hall, Upper Saddle River, NJ.

Priemer, R., 1991, *Introductory Digital Signal Processing*, World Scientific, Singapore.

Proakis, J.G. and Manolakis, D.G., 1996, *Digital Signal Processing: Principles, Algorithms and Applications*, Prentice-Hall, Upper Saddle River, NJ.

Rabiner, L.R. and Gold, B., 1974, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Upper Saddle River, NJ.

Tan, T., 2008, *Digital Signal Processing, Fundamentals and Applications*, Academic Press, New York.

8

Digital Circuits

John P. Uyemura
Georgia Institute of Technology

Robert C. Chang
University of Southern California

Bing J. Sheu
Taiwan Semiconductor Manufacturing Company

8.1 MOS Logic Circuits **8**-1
Introduction • MOSFET Models for Digital Circuits • The Digital Inverter • nMOS Logic Gates • CMOS Inverter • Static CMOS Logic Gates • Dynamic CMOS Logic Gates
References **8**-15
8.2 Transmission Gates **8**-15
Digital Processing • Analog Processing
References **8**-33

8.1 MOS Logic Circuits

John P. Uyemura

8.1.1 Introduction

MOS-based technology has become the default standard for high-density logic designs for several reasons. The most obvious is that MOSFETs can be made with side dimensions of <0.1 μm (10^{-7} m), allowing for complex logic functions to be constructed in small areas. The section is an investigation of the basics of designing and characterizing logic gates in an MOS technology.

8.1.2 MOSFET Models for Digital Circuits

The properties of digital logic gates are derived from a large-signal analysis of the circuits. Because transistor characteristics are intrinsically nonlinear, accurate analytic modeling becomes quite complicated, and closed-form solutions can be difficult to come by. To overcome this problem, simplified MOSFET models are used to estimate the circuit operation in first-cut designs. Once the basic operation is established, computer simulations are used to obtain more accurate information.

Square-law models are useful for understanding the operation of MOS logic circuits. Consider first an *n*-channel, enhancement-mode MOSFET that has a threshold voltage $V_{Tn} > 0$. As shown in Figure 8.1, the primary device voltages are V_{DS}, V_{GS}, and V_{SB}. The value gate–source voltage V_{GS} relative to the threshold voltage V_{Tn} determines if drain current I_D flows. If $V_{GS}, < V_{Tn}$ then $I_D \approx 0$, establishing the condition of ''cutoff.'' Elevating the gate–source voltage to a value $V_{GS}, > V_{Tn}$ places the MOSFET into the ''active region'' where I_D will be nonzero if a drain–source voltage V_{DS} is applied; the value of I_D depends on the values of the device voltages.

To describe active operation, we introduce the drain–source saturation voltage $V_{DS,sat}$ defined by

$$V_{DS,sat} = V_{GS} - V_{Tn} \tag{8.1}$$

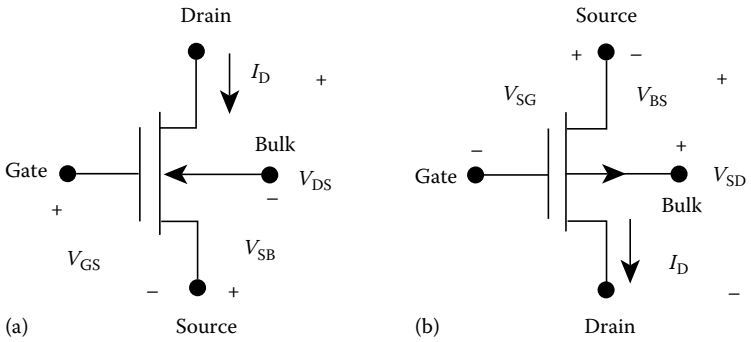


FIGURE 8.1 MOSFET symbols. (a) *n*-channel MOSFET; (b) *p*-channel MOSFET.

The threshold voltage is affected by the affected by the source–bulk (body) voltage V_{SB} by

$$V_{Tn} = V_{T0n} + \gamma\left(\sqrt{2|\phi_F| + V_{SB}} - \sqrt{2|\phi_F|}\right) \tag{8.2}$$

where

V_{T0n} is the nFET zero-body-bias threshold voltage
γ is the body bias (or, body effect) coefficient
ϕ_F is the bulk Fermi potential

When $V_{DS} \cdot V_{DS,sat}$, the MOSFET is nonsaturated with

$$I_D \approx \left(\frac{\beta_n}{2}\right)\left[2(V_{GS} - V_{Tn})V_{DS} - V_{DS}^2\right] \tag{8.3}$$

In this equation β_n is the device transconductance given by $\beta_n = k'_n(W/L)$, with k'_n the process transconductance in units of [A/V^2], W the channel width, and L the channel length; the width-to-length (W/L) is called the "aspect ratio" of the transistor. The process transconductance is given by $k'_n = \mu_n C_{ox}$ where μ_n is the electron surface mobility and C_{ox} the oxide capacitance per unit area. For an oxide layer with thickness t_{ox}, the MOS capacitance per unit area is calculated from

$$C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} \tag{8.4}$$

where ε_{ox} is the oxide permittivity. In the current technologies t_{ox} is smaller than about 60 Å. If $V_{DS} \geq V_{DS,sat}$, the MOSFET is saturated with

$$I_D \approx \left(\frac{\beta_n}{2}\right)(V_{GS} - V_{Tn})^2 \tag{8.5}$$

This ignores several effects, most notably that of "channel-length modulation," but is still a reasonable approximation for estimating basic performance parameters.

The structure of the MOSFET gives rise to several parasitic capacitances that tend to dominate the circuit performance. Two types of capacitors are contained in the basic model shown in Figure 8.2. The contributions C_{GS} and C_{GD} are due to the MOS layering of the gate–oxide semiconductor, which is the origin of the field effect. The total gate capacitance G_G is calculated from

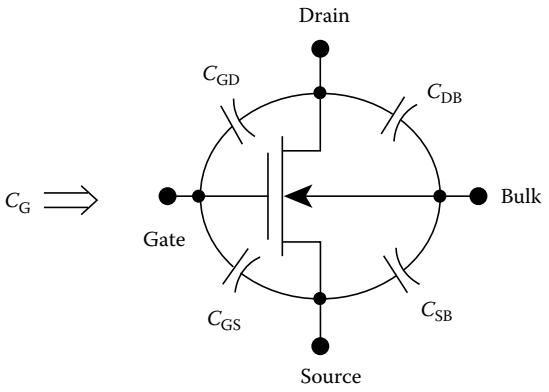


FIGURE 8.2 MOSFET capacitances.

$$C_G = C_{ox} WL \tag{8.6}$$

and the gate–source and gate–drain contributions can be approximated to first order by

$$C_{GD} \approx \frac{C_G}{2} \approx C_{GS} \tag{8.7}$$

Capacitors C_{DB} and C_{SB} are depletion contributions from the reverse-biased *pn* junction at the drain and source. These are nonlinear, voltage-dependent elements that decrease with increasing reverse voltage.

A *p*-channel MOSFET (pMOS or pFET) is the electrical complement of an *n*-channel device. An enhancement-mode pFET is defined to have a negative threshold voltage, i.e., $V_{Tp} < 0$. It is common to use device voltage of V_{SG}, V_{SD}, and V_{BS}, as shown in Figure 8.1 to describe the operation. Cutoff occurs if $V_{SG} < |V_{Tp}|$, while the device is active if $V_{SG} \geq |V_{Tp}|$. The saturation voltage of the pFET is defined by

$$V_{SD,sat} = V_{SG} - |V_{Tp}| \tag{8.8}$$

With $V_{SG} \geq |V_{Tp}|$ and $V_{SG} < V_{SD,sat}$, the transistor is nonsaturated with

$$I_D \approx \left(\frac{\beta_p}{2}\right)\left[2\left(V_{SG} - |V_{Tp}|\right)V_{SD} - V_{SD}^2\right] \tag{8.9}$$

For the pFET, β_p is the device transconductance $\beta_p = k'_p$ (W/L); where $k'_p = \mu_p C_{ox}$ is the process transconductance, and (W/L) is the aspect ratio of the device. In complementary metal–oxide–semiconductor (CMOS) inverters nFETs and pFETs are used in the same circuit, and it is important to note that $k'_n > k'_p$ due to the fact that the electron mobility is larger than the hole mobility, typically by a factor of 2 to 8.

It is often convenient to use the simplified MOSFET symbols shown in Figure 8.3. The polarity of the transistor (nMOS or pMOS) is made explicit by the absence or presence of the gate inversion "bubble," as shown. These symbols do not show the bulk electrode explicitly, but it is important to remember that all nFETs have their bulks connected to the lowest voltage in the circuit (usually ground), while all pFET bulks are connected to the highest voltage (usually the power supply V_{DD}).

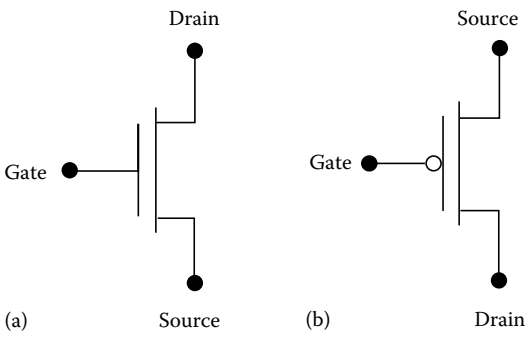


FIGURE 8.3 Simplified MOSFET symbols. (a) nMOSFET; (b) pMOSFET.

In digital circuit design, it is useful to model MOSFETs as voltage-controlled switches, as shown in Figure 8.4. The MOSFET switches are controlled by a gate input signal *G*, which is taken to be a Boolean variable. Employing a positive logic convention, $G = 0$ corresponds to a low voltage (below V_{Tn}), while $G = 1$ is a high voltage. The operation of the FET switches is straightforward. An input of $G = 0$ places the nFET into cutoff, corresponding to an OPEN switch; $G = 1$ implies active operation, and the switch is CLOSED. The pFET has a complementary behavior, with $G = 0$ giving a CLOSED switch, and $G = 1$ giving an OPEN switch.

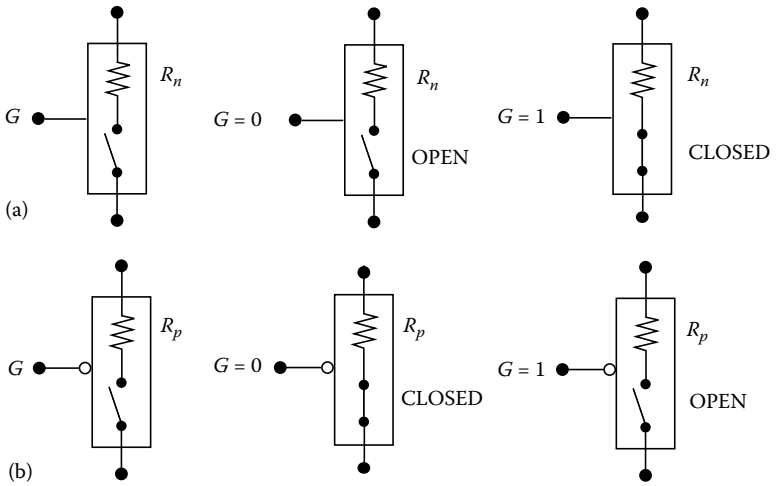


FIGURE 8.4 MOSFET switching models. (a) nFET switch model; (b) pFET switch model.

The switch models include parasitic drain-to-source resistance R_n and R_p, which are usually estimated using

$$R_n = \frac{1}{k'_n(\mathrm{W/L})_n(V_{\mathrm{DD}} - V_{\mathrm{T}n})}$$
$$R_p = \frac{1}{k'_p(\mathrm{W/L})_p(V_{\mathrm{DD}} - V_{\mathrm{T}p})} \tag{8.10}$$

These equations illustrate the general dependence that the drain–source resistance R is inversely proportional to the aspect ratio (W/L). However, the MOSFET is at best a nonlinear resistor, so that these are only rough estimates. It is important to note the MOSFET parasitic capacitances C_{GS}, C_{GD}, C_{SB}, and C_{DB} must be included in the switching models when performing transient analysis.

8.1.3 The Digital Inverter

An ideal digital inverter is shown in Figure 8.5. In terms of the Boolean variable A, the inverter accepts A and produces the complement $\overline{A}$. Electronic implementation of the inverter requires assigning voltage ranges for V_{in} and V_{out} to represent logic 0 and logic 1 states. These are chosen according to the DC

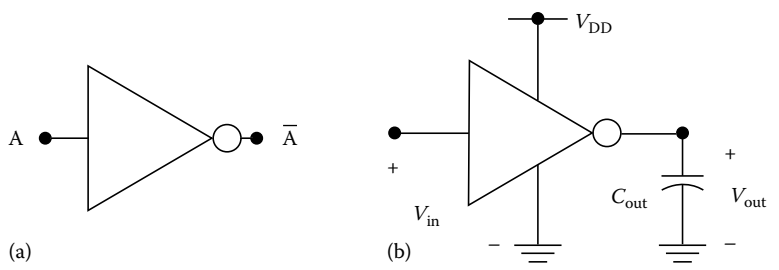


FIGURE 8.5 Basic inverter. (a) Ideal inverter symbol. (b) Electronic parameters.

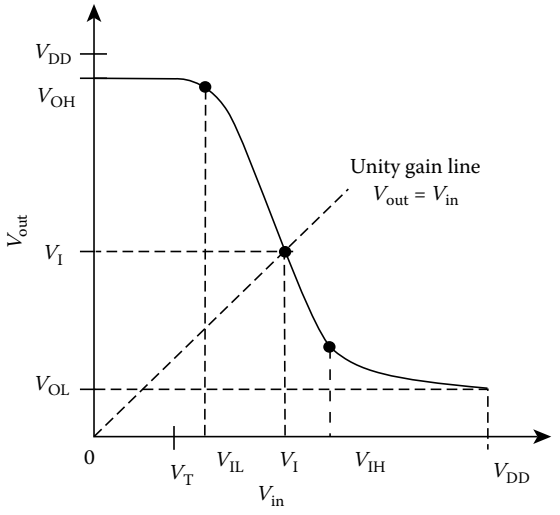


FIGURE 8.6 Inverter VTC.

voltage transfer characteristics (VTC) of the electronic circuit. A VTC is simply a plot of the output voltage V_{out} as a function of V_{in}; a general VTC is shown in Figure 8.6.

Consider first the output voltage V_{out}. The maximum value of V_{out} is denoted by V_{OH}, and is called the output high voltage. This is used to represent an ideal logic 1 output voltage. Conversely, the smallest value of V_{out} is denoted as V_{OL}, and is called the output low voltage. V_{OL} is the output logic 0 voltage. The logic swing of the inverter is then defined by $(V_{OH} - V_{OL})$. The range of input voltage V_{in} used to represent logic 0 and logic 1 input states is usually determined by points on the VTC at which the slope has a value of $(dV_{out}/dV_{in}) = -1$. Logic 0 voltages are those with values between 0 V and V_{IL}, the input low voltage. Similarly, voltages in the range from the input voltage V_{IH} to V_{OH} represent logic 1 input levels. The intersection of the VTC with the unity gain line defined by $V_{out} = V_{in}$ gives the inverter threshold voltage V_I; this represents the switching point of the circuit. The numerical value of V_{OH}, V_{OL} V_{IL}, V_{IHt}, and V_I are determined by the circuit and topology and the characteristics of the devices used in the circuit.

The transient characteristics of the gate are defined by two basic transition times. Figure 8.7 shows V_{in} and V_{out}, and the most important output switching intervals. The input voltage has been taken as an ideal step-like pulse. In a more realistic situation the input voltage is better approximated by a ramp or an exponential. The idealized pulse is used here because it allows a comparison among various circuits.

The most important switching properties of the inverter are the low-to-high time t_{LH}, and the high-to-low time t_{HL} as shown in Figure 8.7. These represent the minimum response times of the inverter.

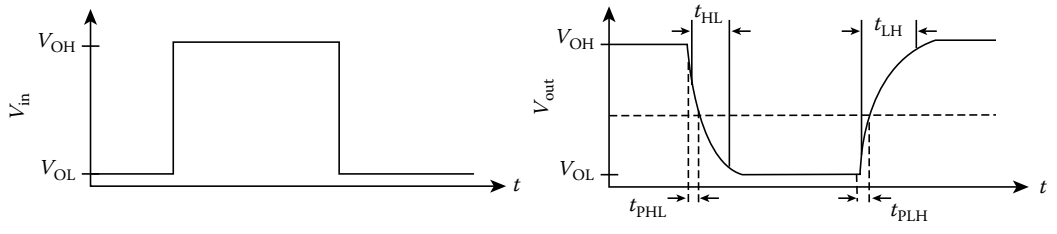


FIGURE 8.7 Inverter switching times.

Note that these time intervals are usually defined between the 10% and 90% voltages instead of the full logic swing. The maximum switching frequency is computed from

$$f_{\max} = \frac{1}{t_{\mathrm{LH}} + t_{\mathrm{HL}}} \tag{8.11}$$

The propagation delay t_{P} for the gate is the average time required for a change in the input to be seen at the output. It is computed using the time intervals t_{PHL} and t_{PLH} shown in the diagram from

$$t_{\mathrm{P}} = \left(\frac{1}{2}\right)(t_{\mathrm{PHL}} + t_{\mathrm{PLH}}) \tag{8.12}$$

Note that the transition times for this parameter are measured to the 50% voltage.

8.1.4 nMOS Logic Gates

Early generations of MOS logic circuits were based on a single type of MOSFET. The Intel 4004, for example, used only pMOS transistors, while subsequent microprocessor chips such as the Intel 8088, the Zilog Z80, and the Motorola 6800 used only *n*-channel MOSFETs. Although all current MOS-based designs are implemented in CMOS, which employs both nMOS and pMOS devices, it is worthwhile to examine nMOS-only logic circuits. This provides an introduction to the basic characteristics of MOS logic circuits, many of which are used in even the most advanced CMOS techniques.

Several types of inverter circuits can be constructed using *n*-channel MOSFETs. Three configurations are shown in Figure 8.8. Each circuit uses a switching transistor MD, known as the "driver," which is controlled by the input voltage V_{in}. The VTC is determined by the load device that connects the drain of MD to the power supply V_{DD}. The MD can be viewed as the switched "pull-down" device, while the load serves as the "pull-up" device. In Figure 8.8a, a simple linear resistor with a value R_{L} is used as the load. The circuits in Figure 8.8b use an enhancement-mode (defined with $V_{\mathrm{T}n} > 0$) nMOSFET biased into saturation, while Figure 8.8c has a depletion-mode nMOSFET (where $V_{\mathrm{T}n} < 0$) as an active load. Active loads provide better switching characteristics due to the nonlinearity of the device. In addition, MOSFETs are much smaller than resistors and process variations are not as critical because the circuit characteristics depend on the ratio of driver-to-load dimensions.

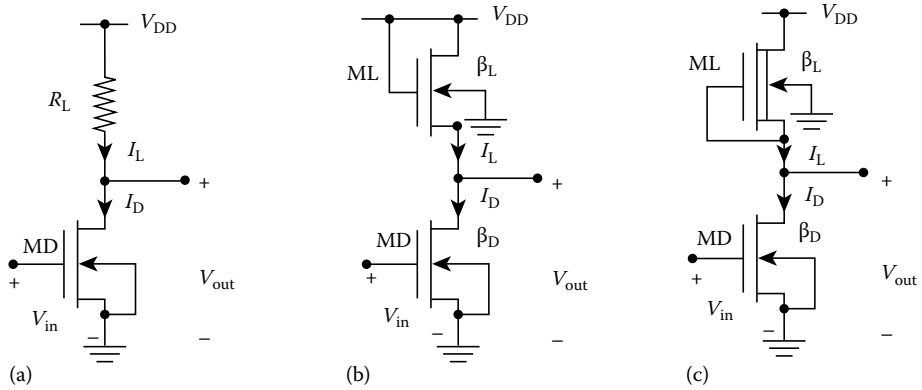


FIGURE 8.8 nMOS inverter circuits: (a) resistor load; (b) saturated enhancement model MOSFET load; and (c) depletion mode MOSFET load.

Although the three nMOS inverter circuits are similar in structure, they have distinct switching properties. Consider the output voltage swing. Figure 8.8a and c both have $V_{OH} \approx V_{DD}$, but the active load in Figure 8.8b gives

$$V_{OH} = V_{DD} - V_{TL} \tag{8.13}$$

with the threshold voltage computed from

$$V_{TL} = V_{T0n} + \gamma\left(\sqrt{2|\phi_F| + V_{OH}} - \sqrt{2|\phi_F|}\right) \tag{8.14}$$

This is referred to as a ''threshold voltage loss,'' and is due to the fact that the load must have a minimum gate–source voltage of $V_{GSL} = V_{TL}$ to be biased into the active mode. Obviously, $V_{OH} < V_{DD}$ for this circuit.

The value of $V_{OL} > 0$ is determined by a ratio of driver parameters to load parameters. In Figure 8.8a, this ratio is given by $R_L\beta_D$, which is inversely proportional to V_{OL}. This means a small V_{OL} requires that both the load resistance and the driver dimensions are large. In Figure 8.8b and c, V_{OL} is set by the driver-to-load ratio $\beta_R = (\beta_D/\beta_L) = (W/L)_D/(W/L)_L$; increasing β_R decreases V_{OL}. For the depletion MOSFET load circuit in Figure 8.8c, the design equation is given by

$$\beta_R = \frac{|V_{TL}|^2}{2(V_{DD} - V_{TD})V_{OL} - V_{OL}^2} \tag{8.15}$$

A condition of $\beta_R > 1$ is generally required to achieve a functional inverter, implying that the driver MOSFET is always larger than the load device. Also, note that is not possible to achieve $V_{OL} = 0$ because this requires an infinite driver-to-load ratio.

The transient switching characteristics are obtained by including the output capacitance C_{out} as shown in Figure 8.5 at the output. C_{out} consists of the input gate capacitance seen looking in the MOSFET of the next stage, and also has parasitic contributions from the MOSFETs and interconnects. By using a switch model for the driver, it is seen that the transient characteristics are determined by the time required to charge and discharge C_{out}. The high-to-low time t_{HL} represents the time it takes to discharge the capacitor through the driver MOSFET with a device transconductance value of β_D. A rough estimate is obtained using the RC time constant such that $t_{HL} \approx 2R_D\ C_{out}$, with R_D the equivalent resistance. Similarly, the low-to-high time t_{LH} is the time interval needed to charge C_{out} through the load device. With respect to Figure 8.8c has the best transient response such that $t_{LH} \approx 2R_L\ C_{out}$, where R_L represents the equivalent resistance of the load MOSFET. nMOS circuits in the mid-1980s had inverter transition times on the order of a few nanoseconds. Because the DC design requires that $\beta_R = (\beta_D/\beta_L) > 1$, and the drain–source resistance of a MOSFET is inversely proportional to β, these circuits exhibit nonsymmetrical switching times with $t_{LH} > t_{HL}$. The propagation delay times can be estimated using $t_{PHL} \approx R_D\ C_{out}$ and $t_{PLH} \approx R_L\ C_{out}$ because these are measured relative to the 50% voltage levels.

MOS-based logic design allows one to easily construct other logic functions using the inverter circuit as a guide. For example, adding another driver MOSFET in parallel gives the NOR operation, while adding a series connected driver yields the NAND operation; these are shown in Figure 8.9.

Complex logic gates for AOI (AND-OR-INVERT) and OAI (OR-AND-INVERT) canonical logic functions can be constructed using the simple rules

- nMOSFETs (or groups) in parallel provide the NOR operation
- nMOSFETs (or groups) in series provide the NAND operation

Examples are provided in Figure 8.10. It should be noted that this type of circuit structuring is possible because the drain and source are interchangeable. The main problem that arises in design complex

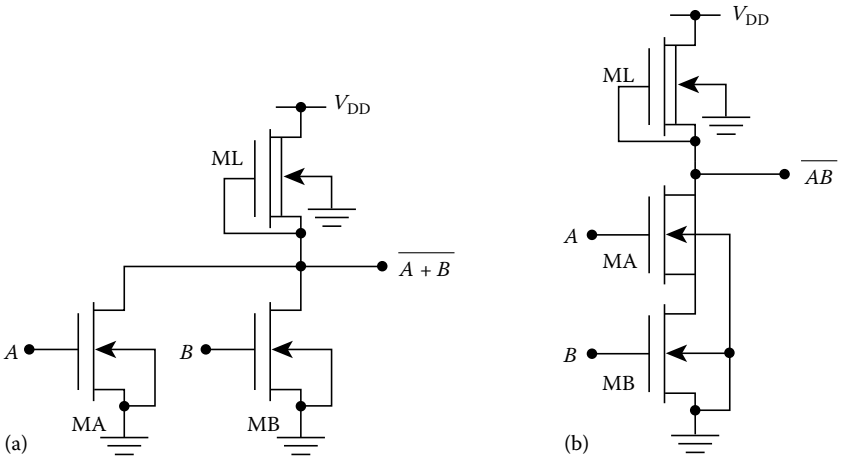


FIGURE 8.9 nMOS NOR and NAND gates: (a) two-input NOR gate and (b) two-input NAND gate.

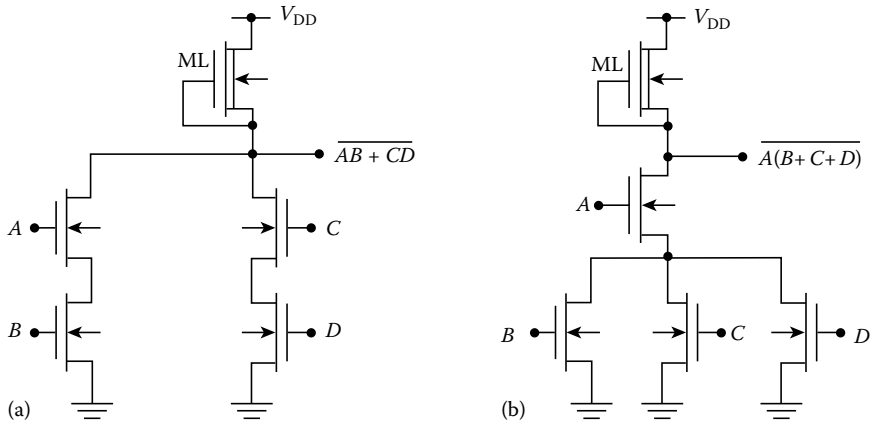


FIGURE 8.10 nMOS AOI logic gates.

nMOS logic gates is that the circuit requires large driver-to-load ratios to achieve small V_{OL} values. The switching FET arrays collectively act like a driver network that must be designed to have a large overall effective β value. Although parallel-connected MOSFETs are not a problem, the pull-down resistance of series-connected MOSFETs can be large unless the individual aspect ratios are increased. Satisfying this condition requires additional chip area, decreasing the logic density.

8.1.5 CMOS Inverter

A CMOS inverter is shown in Figure 8.11. This circuit uses a pair of transistors, one nMOS and one pMOS, connected with their gates together. When $V_{in} < V_{Tn}$, the pFET is active and the nFET is in cutoff. Conversely, when $V_{in} > (V_{DD} - V_{Tn})$, the nFET is active while the pFET is in cutoff. The two MOSFETs are said to form a complementary pair.

The complementary arrangement of the MOSFETs gives the circuit a full rail-to-rail output range, i.e., $V_{OL} = 0$ V and $V_{OH} = V_{DD}$. The devices are connected in such a way that terminal voltages satisfy

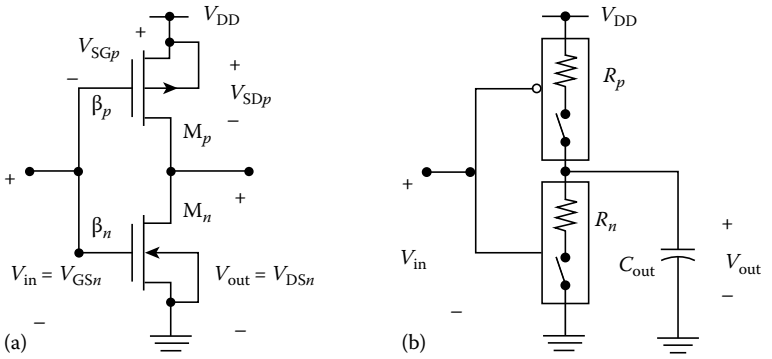


FIGURE 8.11 CMOS inverter: (a) circuit and (b) switch model.

$$\begin{aligned} V_{GSn} + V_{SGp} &= V_{DD} \\ V_{DSn} + V_{SDp} &= V_{DD} \end{aligned} \tag{8.16}$$

Note in particular the relationship between the gate–source voltages. Increasing the voltage on one transistor automatically decreases the voltage applied to the other. This provides the VTC with a very sharp transition, as shown in Figure 8.12. Moreover, the shape of the VTC is almost insensitive to the power supply value V_{DD}, which allows CMOS circuits based on this construction to be used with a range of values. The minimum value of V_{DD} is set by the device the device threshold voltages, and is usually estimated as being about 3 V_T. This is based on the input switching voltage V_{in}, and allows one V_T to switch the nFET, one V_T to switch the pFET, and one V_T for separation. Currently, V_T values equal ~0.5 V, so that the minimum V_{DD} is about 1.5 V. Because V_T is set in the fabrication, the minimum power supply used in low-voltage designs depends upon the process specifications. The maximum value of the power supply voltage is limited by the reverse breakdown voltages of the drain–bulk junctions. This is typically around 14–17 V.

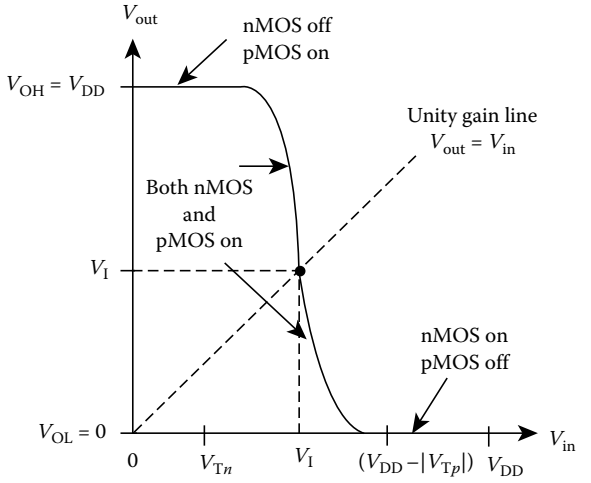


FIGURE 8.12 CMOS VTC.

Because the structure of the CMOS circuit automatically gives a full-rail output logic swing, the DC design of the gate centers around setting the inverter threshold voltage V_I. At this point, both FETs are saturated, and equating currents gives the expression

$$V_I = \frac{\sqrt{\beta_n/\beta_p}V_{Tn} + (V_{DD} - |V_{Tp}|)}{1 + \sqrt{\beta_n/\beta_p}} \tag{8.17}$$

This equation shows that V_I can be set by adjusting the ratio β_n/β_p. If $\beta_n = \beta_p$, and $V_{Tn} \approx |V_{Tp}|$, then $V_I \approx (V_{DD}/2)$. Increasing this ratio decrease the inverter switching voltage. If the nFET and pFET are of equal size, then $\beta_n > \beta_p$ (as $k'_n > k'_p$), and $V_I < (V_{DD}/2)$.

The transient characteristics are obtained by analyzing the charge and discharge current flow paths through the transistors. By using the switch model in Figure 8.11b, the primary time constants are

$$\begin{aligned} \tau_n &= R_n C_{out} = \frac{C_{out}}{\beta_n(V_{DD} - V_{Tn})} \\ \tau_p &= R_p C_{out} = \frac{C_{out}}{\beta_p(V_{DD} - |V_{Tp}|)} \end{aligned} \tag{8.18}$$

Analyzing the transitions with a step input voltage yields

$$\begin{aligned} t_{HL} &= \tau_n \left[\frac{2(V_{Tn} - V_0)}{(V_{DD} - V_{Tn})} + \ln\left(\frac{2(V_{DD} - V_{Tn})}{V_0} - 1 \right) \right] \\ t_{LH} &= \tau_p \left[\frac{2(|V_{Tp}| - V_0)}{(V_{DD} - |V_{Tp}|)} + \ln\left(\frac{2(V_{DD} - |V_{Tp}|)}{V_0} - 1 \right) \right] \end{aligned} \tag{8.19}$$

where $V_0 = 0.1\ V_{DD}$ is the 10% voltage. Noting once again that $k'_n > k'_p$, equal size transistors will give $t_{LH} > t_{HL}$. To obtain symmetrical switching, the pMOSFET must have an aspect ratio of $(W/L)_p = (k'_n > k'_p)\ (W/L)_n$. This illustrates that while the ratio of β-values sets the DC switching voltage V_I, the individual choices for β_n and β_p determine the transient switching times. In general, fast switching requires large transistors, illustrating the speed vs. area trade-off in CMOS design. The propagation delay time exhibits the same dependence.

Another interesting characteristic of the CMOS inverter is the power dissipation. Consider an inverter with stable logic 0 or logic 1 inputs. Because one MOSFET is in cutoff, the DC power supply current I_{DD} is very small, being restricted to leakage levels. The standby DC power dissipation is $P_{DC} = I_{DD}V_{DD} \approx 0$, so that static logic circuits do not dissipate much power under static conditions. Appreciable I_{DD} from the power supply to ground flows only during a transition. Dynamic power dissipation, on the other hand, occurs due to the charging and discharging of the output capacitance C_{out}. The dynamic power dissipation can be estimated by

$$P_{Dynamic} = C_{out}V_{DD}^2 f \tag{8.20}$$

where f is the switching frequency of the signal. Qualitatively, this is understood by noting that this is just twice the average stored energy multiplied by the frequency. This illustrates the important result that the power dissipation of a CMOS circuit increases with the switching frequency.

8.1.6 Static CMOS Logic Gates

Static logic gates are based on the inverter. The term ''static'' means that the output voltages (logic levels) are well defined as long as the inputs are stable. The nFET rules discussed for nMOS logic gates still apply

to CMOS. However, static logic gates provide an nFET and a pFET for every input. Proper operation requires that rules be developed for the pMOSFET array as follows:

- pMOSFETs (or groups) in parallel provide the NAND operation
- pMOSFETs (or groups) in series provide the NOR operation

When these rules are compared to the nMOS rules, it is seen that the nFET and pFET arrays are logical duals of one another (i.e., OR goes to AND, and vice versa).

An N-input static CMOS logic gate requires 2 N transistors. NAND and NOR gates are shown in Figure 8.13 using the rules; this type of logic is termed series-parallel, for obvious reasons. Examples of complex logic gates are shown in Figure 8.14. Note in particular the circuit in Figure 8.14b. This implements the XOR function by means of

$$A \oplus B = \overline{AB + \overline{A}\,\overline{B}} = A\overline{B} + \overline{A}B \tag{8.21}$$

Reductions of this type are often performed to work the AOI or OAI equation into a more familiar form.

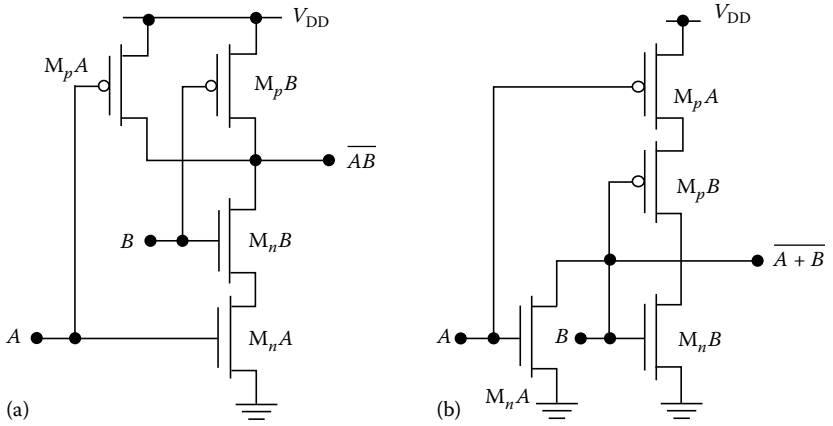


FIGURE 8.13 CMOS: (a) NAND and (b) NOR gates.

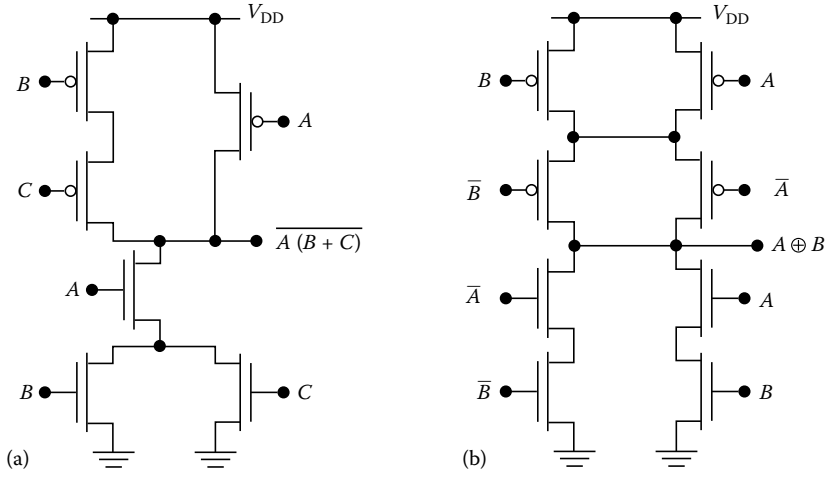


FIGURE 8.14 CMOS AOI logic examples: (a) AOI gate and (b) XOR circuit.

As seen from these examples, the logic function is determined by the placement of the nFETs and pFETs in their respective arrays. Electrically, the design problem centers around choosing the aspect ratios to achieve acceptable switching times. Because a MOSFET has a parasitic drain–source resistance that varies as $(1/\beta)$, series-connected transistor chains exhibit larger time constants than parallel-connected arrangements. Recalling that $R_n < R_p$ shows that for equal size devices, series chains of nFETs are preferable to the same number of series-connected pFETs. Consequently, NAND gates are used more frequently than NOR gates, and AOI logic functions with a small number of OR operations are better. It is also possible to expand to transistor arrays that are not of the series-parallel type, such as a delta configuration, but it is difficult to devise general design guidelines for these circuits.

Canonical CMOS static logic design is based on using pairs of nMOS and pMOS transistors. In modern very large scale integration (VLSI) design, the design complexity is limited by the interconnect (as opposed to the number of transistors), so that the need to connect every input to two transistors may result in problems in the chip layout. Pseudo-nMOS circuits provide an alternative to standard CMOS circuits. These logic gates implement logic using nFET arrays; however, the pMOS array is replaced by a single *p*-channel MOSFET that acts as a load device. Figure 8.15 shows an inverter and an AOI circuit implement based on pseudo-nMOS structuring. In both circuits, the load pMOSFET is biased active with $V_{SGp} = V_{DD}$ by grounding the gate. Although the circuits are simplified, two main problems arise with this type of circuit. First, the output low voltage V_{OL} is determined by the driver-to-load ratio $(\beta_n/\beta_p) > 1$, so that large driver nFETs are required. Second, if the input voltage is high, then the circuit dissipates DC power. Despite these drawbacks, pseudo-nMOS circuits may be useful in certain situations.

Transmission gates (TGs) provide another approach to implementing logic functions in CMOS. The properties of TGs are discussed in more detail in Section 8.2. However, because they are useful for certain types of logic gates, a short discussion has been included here. A basic TG consists of an nMOSFET and a pMOSFET in parallel, as shown in Figure 8.16a; the symbol in Figure 8.16b represents the composite structure. TGs act like voltage-controlled switches: logically, a condition of $C = 0$ gives an open switch, while $C = 1$ gives a closed switch. TGs can pass the full range of voltages (from 0 V to V_{DD}) in either direction; this is not possible with a single device, due to the threshold voltage characteristic discussed earlier in this section.

Figure 8.17 illustrates a simple 2:1 multiplexer (MUX) with two input lines, D_0 and D_1, and a control bit *S*. When $S = 0$, the upper TG is closed, and the output is $F = D_0$. Conversely, $S = 1$ closes the bottom TG, so $F = D_1$. The operation of this circuit is expressed by

$$F = \overline{S}D_0 + SD_1 \tag{8.22}$$

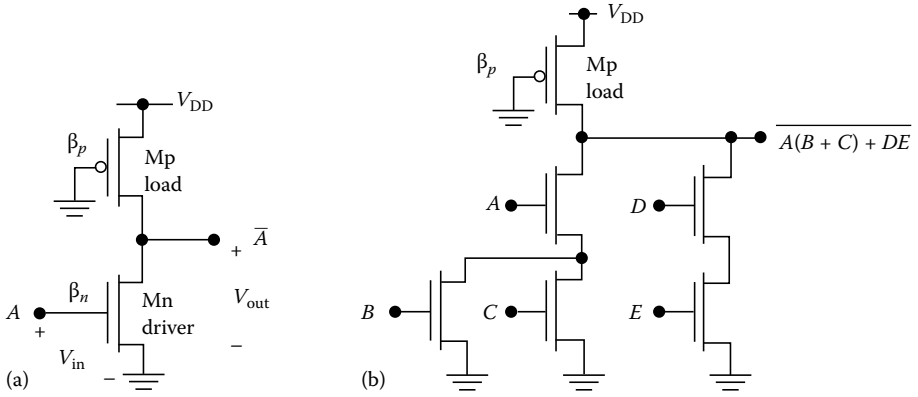


FIGURE 8.15 Pseudo-nMOS logic circuits: (a) Inverter and (b) AOI logic gate.

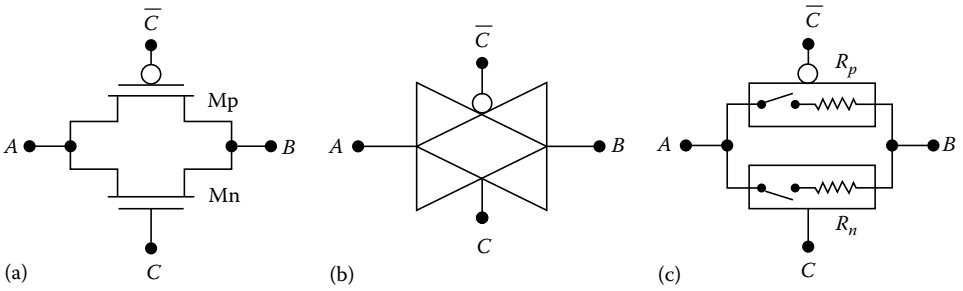


FIGURE 8.16 Transmission gate: (a) circuit; (b) symbol; and (c) switching model.

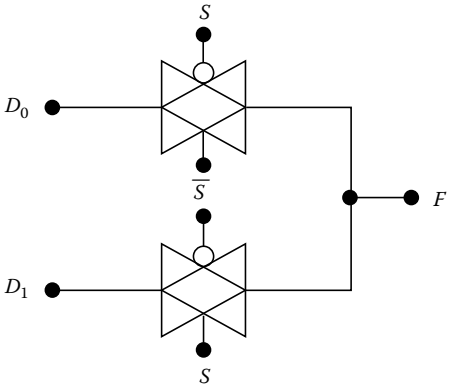


FIGURE 8.17 TG-based 2:1 multiplexer.

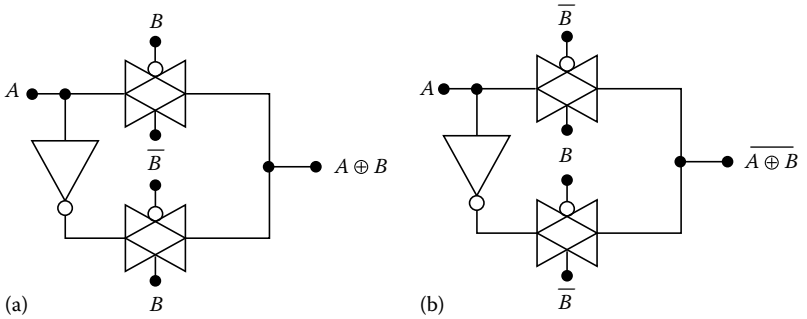


FIGURE 8.18 TG-based (a) XOR and (b) XNOR logic gates.

The circuit can be expanded easily to create larger multiplexers. For example, an 8:1 requires three select bits, and each of the eight lines will be a switching network using three TGs. Several other TG-based logic functions are popular in CMOS design. Figure 8.18 shows the exclusive-OR (XOR) and exclusive-NOR (XNOR) circuits. The primary drawbacks of TG-based logic circuits are that (1) the TG does not have a connection to the power supply, and acts as a parasitic RC element to the stage that drives it, and (2) the chip layout may become large, complicated, or both. In particular, (1) implies that TG circuits may be slower than equivalent functions designed using basic static CMOS techniques.

8.1.7 Dynamic CMOS Logic Gates

Dynamic CMOS logic gates are characterized as having outputs that are valid only for a limited time interval. Although this property inherently makes the circuit design more challenging, dynamic logic circuits can potentially achieve fast switching speeds. In general, dynamic circuits use parasitic capacitors in the MOS circuit to store charge Q. Because $Q = CV$, the presence or absence of charge corresponds to a logic 1 or logic 0 level, respectively, MOSFETs are used as voltage-controlled switches to "steer" the charge on and off the logic nodes. Several dynamic logic families have appeared in the literature, each having distinct characteristics. We now merely touch on some characteristics of a basic circuit which illustrates the important points.

Consider the dynamic logic circuit in Figure 8.19 for a three-input NAND gate. The transistors labeled MP and MN are controlled by the clock $\phi(t)$, and provide synchronization of the data flow. Note the presence of capacitors C_{out}, C_1, C_2, and C_3. These represent parasitic capacitances due to the transistors and interconnect, and are crucial to the operation.

The circuit is controlled by the timing provided by $\phi(t)$. When $\phi = 0$, MP is ON and MN is OFF. During this time, C_{out} is charged to a voltage $V_{out} = V_{DD}$, which is called a "precharge event." When ϕ changes to a level $\phi = 1$, MP is driven into cutoff, but MN is biased ON; the operation of the circuit during this time is termed a "conditional discharge event." If the inputs are set to $(A, B, C) = (1, 1, 1)$, then all three logic transistors, MA, MB, and MC, are ON, and C_{out} can discharge through these transistors and MN to a final voltage of $V_{out} = 0$ V. If at least one input is a logic 0, then C_{out} does not have a direct discharge path to ground. Ideally, V_{out} would stay at V_{DD}. However, charge leakage occurs across the reverse-biased drain–bulk *pn* junctions in the MOSFETs, eventually leading to a value of $V_{out} = 0$ V. Typically, the output voltage can be held only for a few milliseconds, thus leading to the name "dynamic circuit."

Another problem that arises in dynamic logic circuits is that of charge sharing. Consider the three-input NAND gate with inputs of $(A, B, C) = (0, X, X)$ during the precharge, where X is a do not care condition. The total charge transferred to the circuit from the power supply is

$$Q_T = C_{out} V_{DD} \tag{8.23}$$

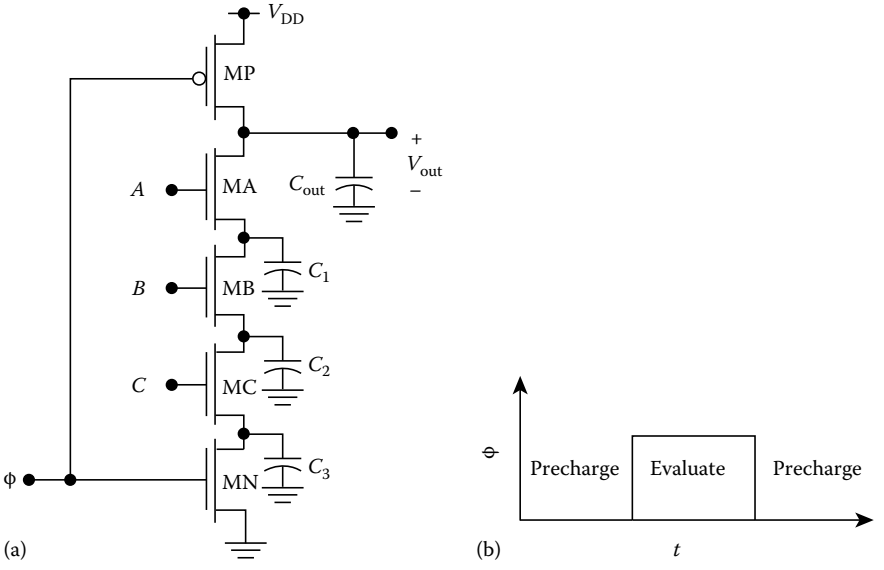


FIGURE 8.19 Dynamic CMOS logic gate: (a) three-input NAND gate and (b) timing intervals.

Now suppose that the inputs are switched to $(A, B, C) = (1, 1, 0)$ during the evaluation phase. MOSFETs MA and MB are ON, but MC is OFF, blocking the discharge path. Charge sharing occurs because the charge originally stored on C_{out} is now shared with C_1 and C_2. After the transients have decayed, the three capacitors are in parallel, Ignoring any threshold drop, they will share the same final voltage V_f such that

$$Q_T = (C_{out} + C_1 + C_2)V_f \tag{8.24}$$

Equating the two expressions for charge gives

$$V_f = \frac{C_{out}}{C_{out} + C_1 + C_2} V_{DD} < V_{DD} \tag{8.25}$$

To ensure that the output voltage remains at a logic 1 high voltage, the capacitors must satisfy the relation

$$C_{out} \gg C_1 + C_2 \tag{8.26}$$

The capacitance values are proportional to the sizes of the contributing regions, so that the performance is closely tied to the layout of the chip.

References

The material in this section is quite general. The references listed below are books in the field of digital MOS integrated circuits that provide further reading on the topics discussed here.

1. L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Reading, MA: Addison-Wesley, 1985.
2. H. Haznedar, *Digital Microelectronics*, Reading, MA: Addison-Wesley, 1991.
3. J. P. Uyemura, *Circuit Design for CMOS VLSI*, Norwell, MA: Kluwer Academic, 1992.
4. J. P. Uyemura, *Fundamentals of MOS Digital Integrated Circuits*, Reading MA: Addison-Wesley, 1988.

8.2 Transmission Gates

Robert C. Chang and Bing J. Sheu

A signal propagates through a transmission gate (TG) in a unique manner. In conventional logic gates, the input signal is applied to the gate terminal of an MOS transistor and the output signal is produced at the drain or the source terminal. In a TG, the input signal propagates between the source and the drain terminals through the transistor channel, while the gate voltage is held at a constant value. The TG is turned off if the voltage applied to the gate terminal is below the threshold voltage. The TG approach can be used in digital data processing to implement special switching functions with high performance as well as a small transistor count [1]. It also can be used in analog signal processing to act as a compact voltage-controlled resistor.

8.2.1 Digital Processing

8.2.1.1 Single Transistor Version

A TG can be constructed by a single nMOS or pMOS transistor, as shown in Figure 8.20. For an nMOS TG to pass a signal V_{in} to the output terminal, the selection signal S is set to the logic 1 value, i.e., the gate

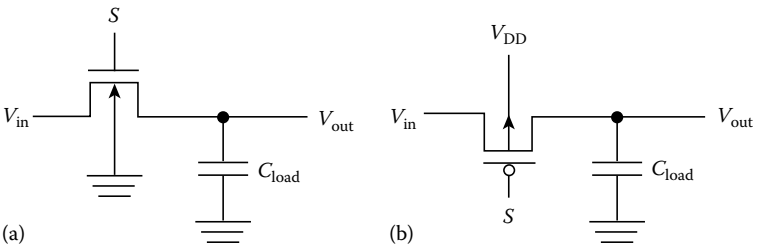


FIGURE 8.20 (a) nMOS TG and (b) pMOS TG.

voltage V_G is set to a high voltage value V_{DD}. If the input signal is also the V_{DD} value, the output voltage V_{out} is determined by [2],

$$V_{out}(t) = (V_{DD} - V_{thn})\left[\frac{t/\tau_{nc}}{1 + (t/\tau_{nc})}\right] \tag{8.27}$$

where

V_{thn} is the threshold voltage of the nMOS transistor with the body effect
τ_{nc} is the charging time constant which can be expressed as

$$\tau_{nc} = \frac{2C_{out}}{\mu_n C_{OX}(W/L)(V_{DD} - V_{thn})} \tag{8.28}$$

where

μ_n is the carrier mobility
C_{OX} is the per-unit-area capacitance value
W/L is the transistor aspect ratio

If time t goes to ∞, then V_{out} will approach $V_{DD} - V_{thn}$, which indicates that a threshold voltage loss occurs in the signal from the input node to the output node. This is due to the fact that V_{GS} must be greater than the threshold voltage to turn on the nMOS transistor. Owing to this voltage reduction, an nMOS TG can only transmit a "weak" logic 1 value. However, a logic 0 can be transmitted by an nMOS TG without penalty. In order to analyze this case, we set $V_{in} = 0$ and $V_{out}(t = 0) = V_{DD} - V_{thn}$. The output voltage V_{out} is determined by

$$V_{out}(t) = (V_{DD} - V_{thn})\left[\frac{2e^{-(t/\tau_{nd})}}{1 + e^{-(t/\tau_{nd})}}\right] \tag{8.29}$$

where the discharge time constant can be expressed as

$$\tau_{nd} = \frac{C_{out}}{\mu_n C_{OX}(W/L)(V_{DD} - V_{thn})} \tag{8.30}$$

Notice that V_{out} will approach zero as time goes to infinity. Input–output (I–O) characteristics of an nMOS TG are shown in Figure 8.21.

The schematic diagram of a pMOS TG is shown in Figure 8.20b. For a pMOS TG to pass a signal V_{in} to the output terminal, the selection signal S is set to the logic 0 value. To transmit a logic 0 value with the initial V_{out} value being V_{DD}, the expression for V_{out} is given as

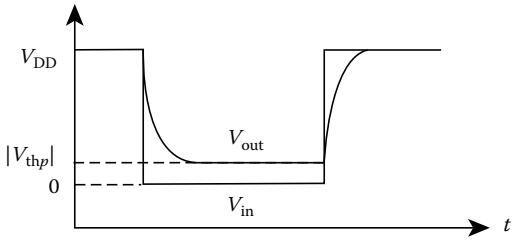


FIGURE 8.21 Characteristics of nMOS TG.

$$V_{out}(t) = |V_{thp}| + \frac{V_{DD} - |V_{thp}|}{1 + (V_{DD} - |V_{thp}|)(t/2\tau_{pd})} \tag{8.31}$$

where τ_{pd} is the discharging time constant for the pMOS TG. As time goes to infinity, V_{out} will approach $|V_{thp}|$, so that the pMOS TG can only transmit a "weak" logic 0 value. On the other hand, the pMOS TG can perfectly transmit a logic 1 value. To analyze this case, we set $V_{in} = V_{DD}$ and assume the initial V_{out} value as $|V_{thp}|$. The expression for V_{out} is given as

$$V_{out}(t) = V_{DD} - (V_{DD} - |V_{thp}|)\left(\frac{2e^{-(t/\tau_{pc})}}{1 + e^{-(t/\tau_{pc})}}\right) \tag{8.32}$$

where τ_{pc} is the charging time constant for the pMOS TG. The output voltage will approach V_{DD} as time goes to ∞. The transfer characteristics of the pMOS TG is shown in Figure 8.22.

8.2.1.2 Complementary Transistor Version

Figure 8.23 is the schematic diagram of a complementary transistor version of the TG which can be constructed by combining the characteristics of nMOS and pMOS TGs. The CMOS TG can transmit both the logic 0 and logic 1 values without any degradation. The voltage transmission properties of the single transistor and CMOS TGs are summarized in Table 8.1. The overall behavior of the CMOS TG can

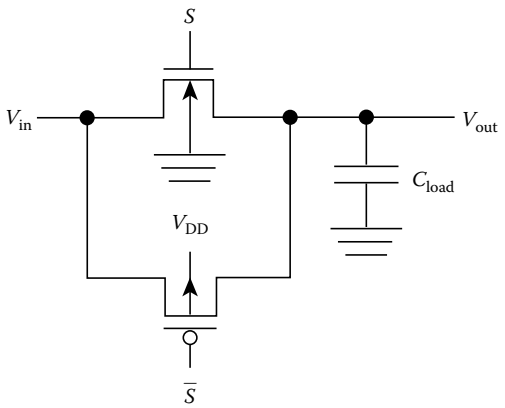


FIGURE 8.22 Characteristics of pMOS TG.

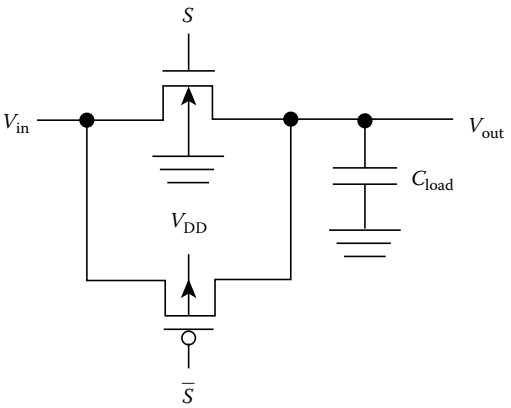


FIGURE 8.23 CMOS TG.

TABLE 8.1 Transmission Gate Characteristics

V_{out} V_{in} Type	$V_{in} = 0$ (Logic 0)	$V_{in} = V_{DD}$ (Logic 1)
nMOS	0	$V_{DD} - V_{thn}$
pMOS	$\|V_{thp}\|$	V_{DD}
CMOS	0	V_{DD}

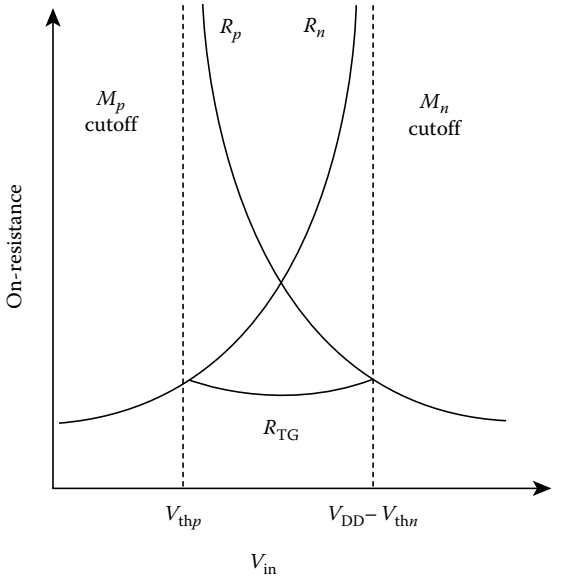


FIGURE 8.24 TG resistances.

be described as follows. When the selection signal S is low, both the nMOS and pMOS transistors are cut off. The output voltage V_{out} will remain at a high impedance state. When the selection signal S is high, both the nMOS and pMOS transistors are turned on and the output voltage will be equal to the input voltage.

Three regions of operation exist for a CMOS TG. In region 1 $V_{in} < |V_{thp}|$, then nMOS transistor is in the triode region and the pMOS transistor is in the cutoff region. Because the pMOS transistor is turned off, the total current, I_{tot}, is supplied by the nMOS transistor and I_{tot} decreases as V_{in} increases. In region 2 $|V_{thp}| < V_{in} < V_{DD} - V_{thn}$ both the nMOS and pMOS transistors are in the triode region. In this region, the nMOS transistor current decreases and the pMOS transistor current increases as V_{in} increases. Thus, I_{tot} is approximately a constant value. In region 3, $V_{in} > V_{DD} - V_{thn}$ the nMOS transistor is turned off and the pMOS transistor is in the triode region. The plot of the TG on-resistance is shown in Figure 8.24.

8.2.1.3 Pass-Transistor Logic

Pass-transistor logic is a family of logic which is composed of TG. Methods for deriving pass-transistor logic using nMOS TGs have been reported [3]. Figure 8.25 shows the schematic diagram of the pass-transistor logic in which a set of pass signals, P_is, are applied to the sources of the nMOS transistors and another set of control signals, C_is, are applied to the gates of the nMOS transistors.

The desired logic function F can be expressed as $F = C_1 \cdot P_1 + C_2 \cdot P_2 + \cdots + C_n \cdot P_n$. When C_is are high, P_is are transmitted to the output node. P_is can be logic 0, logic 1, true, or complement of the ith input variable X_i, or the high-impedance state Z. Constructing a Karnaugh map can help one to design the pass-transistor circuit. The pass function rather than the desired output values is put to the corresponding locations in the Karnaugh map. Then any variables that may act as a control variable or a pass variable are grouped.

For example, consider the design of a two-input XOR function. The truth table and the modified Karnaugh map of the XOR function are given in Tables 8.2 and 8.3, respectively. By grouping the A column when B is 0, and the $\overline{A}$ column when B is 1, the function can be expressed as

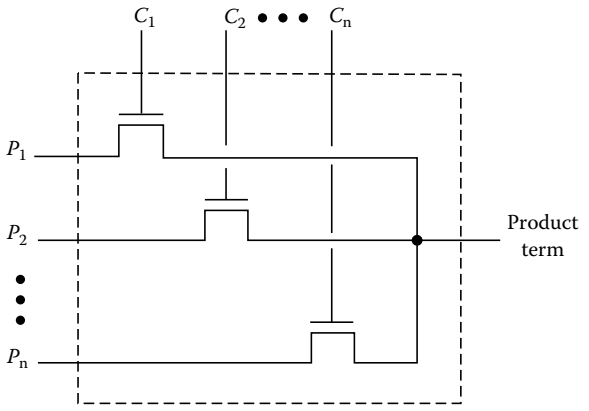


FIGURE 8.25 Model for pass transistor logic.

TABLE 8.2 Truth Table of XOR Function

A	B	A_B	Pass Function
0	0	0	$A+B$
0	1	1	$\overline{A}+B$
1	0	1	$A+\overline{B}$
1	1	0	$\overline{A}+\overline{B}$

TABLE 8.3 Modified Karnaugh Map for XOF Function

		B			
		0		1	
A	0	A		$\overline{A}$	
			B		B
	1	A		$\overline{A}$	
			$\overline{B}$		$\overline{B}$

$$F = \overline{B} \cdot A + B \cdot \overline{A} \tag{8.33}$$

where

B is a control variable

A is a pass variable

Figure 8.26a and b shows the schematic diagrams of nMOS and CMOS implementations of the XOR function. When the control variable B is with a logic 0 value, the pass variable A is transmitted to the output. When the control variable B is with a logic 1 value, the pass variable $\overline{A}$ is transmitted to the output. Another implementation of the XOR function is shown in Figure 8.26c.

It is not permitted to have groupings that transmit both true and false values of the input variable to the output simultaneously. The final expression must contain all the cells in the Karnaugh map. Note that

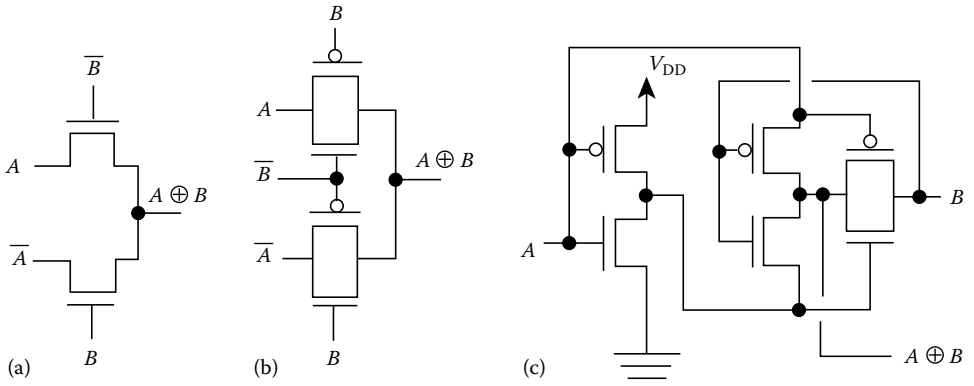


FIGURE 8.26 XOR gates: (a) nMOS version; (b) complementary version I; and (c) complementary version II.

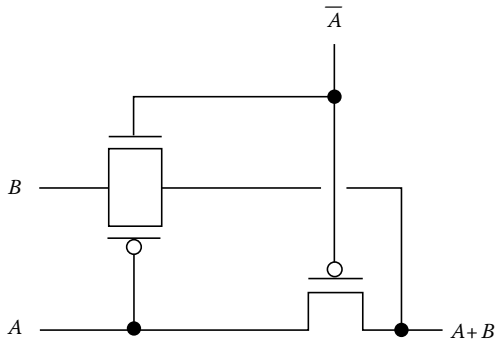


FIGURE 8.27 OR gates.

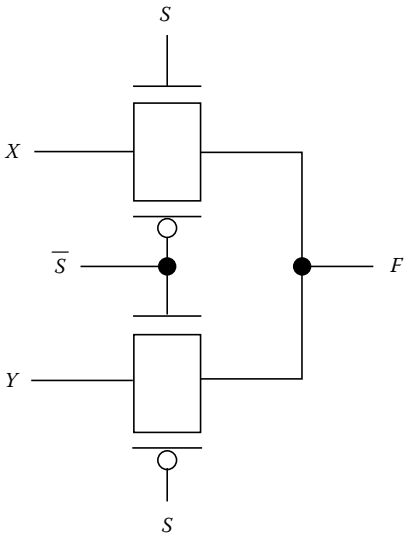


FIGURE 8.28 A two-input multiplexer.

the *p*-transistor circuit is the dual of the *n*-transistor circuit. Thus, the *p*-pass function must be constructed when a complementary version is required. In addition, the pass variable with logic 0 value is transmitted by the nMOS network in a complementary implementation while the pass variable with logic 1 value is transmitted by the pMOS network.

The OR function can be constructed by one pMOS transistor and one CMOS TG, as shown in Figure 8.27. When the input signal *A* is with the logic 0 value, the CMOS TG is turned on and the input signal *B* is passed to the output node. On the other hand, if the input signal *A* is with the logic 1 value, the pMOS TG is turned on and the logic 1 value of input signal *A* is transmitted to the output node. Because the pMOS TG can propagate a "strong" logic 1 value it is not necessary to use another CMOS TG.

TGs can be used to construct a multiplexer which selects and transmits one of the inputs to the output. Figure 8.28 is the circuit schematic diagram of a two-input multiplexer, which is composed of CMOS TGs. The output function of the two-input multiplexer is

$$F = X \cdot S + Y \cdot \overline{S} \tag{8.34}$$

If the selection signal *S* is at a logic 1 value, the input signal *X* is transmitted to the output. On the other hand, if the selection signal *S* is at a logic 0 value, the input signal *Y* is transmitted to the output. Multiplexers are important components in CMOS data manipulation structures and memory elements.

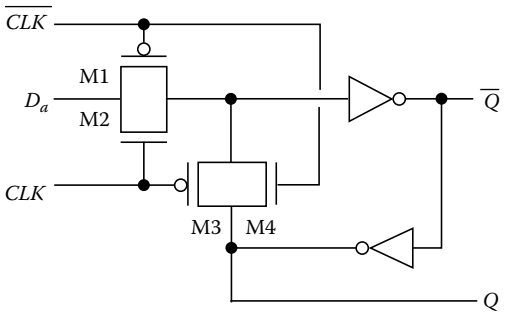


FIGURE 8.29 A CMOS *D* latch.

A basic *D* latch can be constructed by two TGs and two invertes, as shown in Figure 8.29. When the *CLK* signal is at a logic 0 value, pass transistors M1 and M2 are turned off so that the input signal D_a cannot be transmitted to the outputs Q and $\overline{Q}$. In addition, pass transistors M3 and M4 are turned on so that a feedback path around the inverter pair is established and the current state of Q is stored. When the *CLK* signal is at a logic 1 value, M1 and M2 are turned on and M3 and M4 are turned off. Thus, the output signal Q is set to the input signal D_a and $\overline{Q}$ is set to $\overline{D}_a$. Because the output signal Q will follow the change of input signal D_a when the *CLK* signal is high, this circuit is a positive level-sensitive *D* latch. A positive edge-trigger register or so-called *D* flip-flop can be designed by combining one positive level-sensitive *D* latch and one negative level-sensitive *D* latch. By cascading *D* flip-flops, a shift register can be constructed.

TGs can be used in the design of memory circuits. A typical random access memory (RAM) architecture consists of one row/word decoder, one column/bit decoder, and memory cells. The memory cells used in RAMs can be categorized into static cells and dynamic cells. Memory data/charges are stored on the latches in static cells, while on the capacitors in dynamic cells. The static random access memories (SRAMs) are not forced to include the refresh circuitry and are faster than the dynamic random access memories (DRAMs). However, the size of SRAM cells is much larger than that of DRAM cells. The most commonly used circuit in the design of SRAM cells is the six-transistor circuit shown in Figure 8.30. Four transistors are used to form two cross-coupled inverters. The other two transistors, M1 and M2, are TGs to control the read/write operation of the memory cell. If the word line is not selected, the data stored on the latch will not change as long as the leakage current is small. If the word line is selected, the transistors M1 and M2 are turned on. Through the *bit* and $\overline{bit}$ lines, data can be written into the latch or the stored data can be read out by the sense amplifier. TGs can also be found in the four-transistor DRAM cell circuit, as shown in Figure 8.30b. When the *Read* line is selected, pass transistor M1 is turned on and the data stored on the capacitor C_1 are read out. When the *Write* line is selected, pass transistor M2 is turned on and the data from *data_W* line are written into the cell.

Figure 8.31 is the circuit schematic diagram of a TG adder, which consists of four TGs, four inverters, and two XOR gates [4]. The *SUM* output, which represents $A \oplus B \oplus C$, is constructed by a multiplexer controlled by $A \oplus B$ and its complement. Notice that when $A \oplus B$ is false, the *CARRY* output equals *A* or *B*. Otherwise, *CARRY* output takes the value of input signal *C*. Although the TG adder has the same

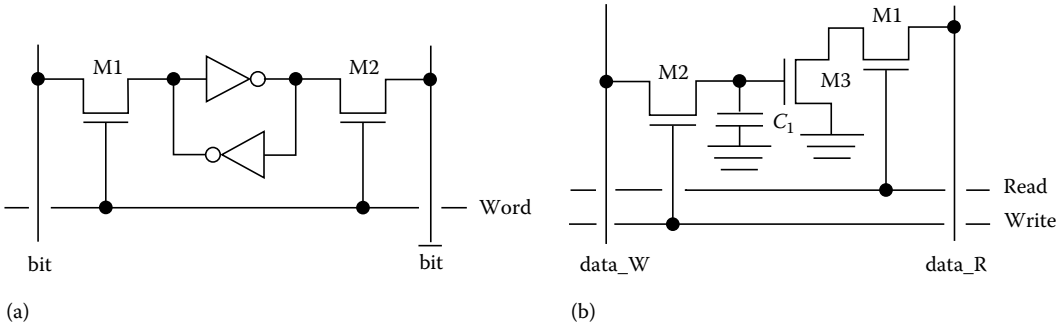


FIGURE 8.30 (a) SRAM cell and (b) DRAM cell.

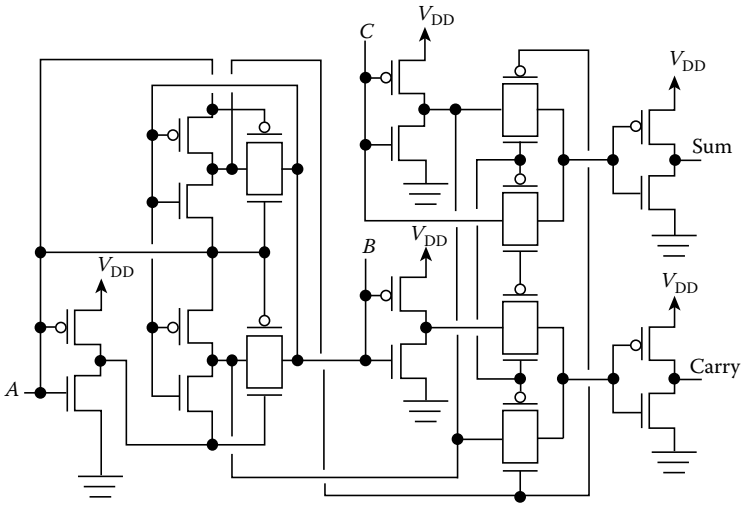


FIGURE 8.31 TG adder.

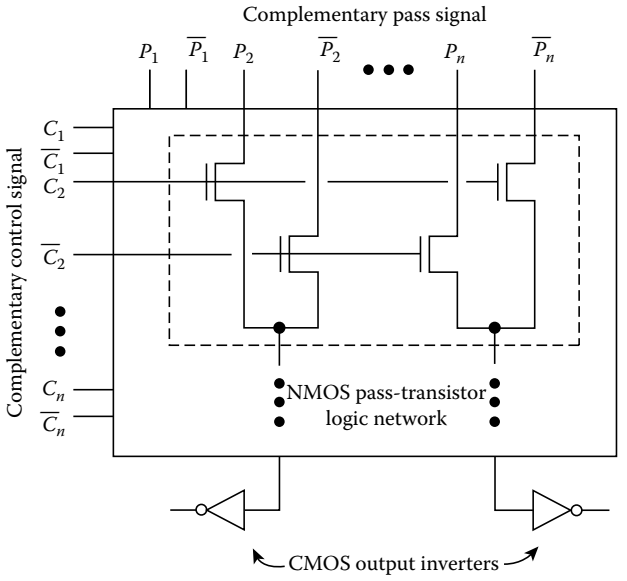


FIGURE 8.32 Schematic structure of the basic CPL circuit.

number of transistors as the combinational adder, it has the advantage of having noninverted *SUM* and *CARRY* output signals and an equal delay time for the *SUM* and *CARRY* output signals.

Another form of differential CMOS logic, complementary pass-transistor logic (CPL), has been developed and utilized on the critical path to achieve very high speed operation [5]. Figure 8.32 is the circuit schematic diagram of the basic CPL structure using an nMOS pass-transistor logic organization. The CPL is constructed by an nMOS pass-transistor logic network, complementary inputs and outputs, and CMOS output inverters. As the nMOS pass transistor will transmit a logic 1 signal with one threshold voltage reduction, the output signals must be amplified by the CMOS inverters which can shift the logic threshold voltage and drive a large capacitive load. One attractive feature of the CPL design

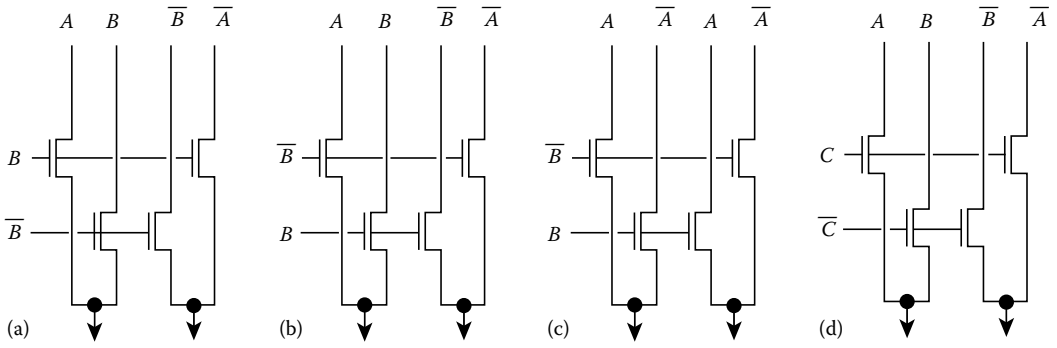


FIGURE 8.33 CPL circuit modules: (a) AND/NAND; (b) OR/NOR; (c) XOR/XNOR; and (d) Wire-AND/NAND.

is that complementary outputs are generated by the simple four-transistor circuits. Because inverters are not required in CPL circuits, the number of critical-path gate stages can be reduced.

Figure 8.33 shows the schematic diagrams of four basic CPL circuit modules: an AND/NAND module, an OR/NOR module, an XOR/XNOR module, and a wired-AND/NAND module [5]. By combining these four circuit modules, arbitrary Boolean functions can be constructed. These modules have an identical circuit schematic and are distinguished by different arrangements of input signals. This property of CPL is quite suitable for master-slice design.

The schematic diagram of a CPL full adder is shown in Figure 8.34. Both the circuitry to produce the *SUM* output signal and the circuitry to produce the *CARRY* output signal are constructed from basic CPL modules. The *SUM* circuitry consists of two XOR/XNOR modules, while the *CARRY* circuitry consists of three wired-AND/NAND modules. The CMOS output inverters are fixed ''overhead'' because they are required whether the circuit has one, two, or many inputs. Thus, designing with a complex Boolean function in a CPL gate is preferred to minimize the delay time and overall device count.

Figure 8.35 is the block diagram of a 16×16 bit multiplier, which is constructed by using a parallel multiplication architecture. A carry-look-ahead (CLA) adder and a Wallace-tree adder array are used to minimize the critical-path gate stages. The number of transistors in the CPL multiplier is less than that in a full CMOS counterpart [6].

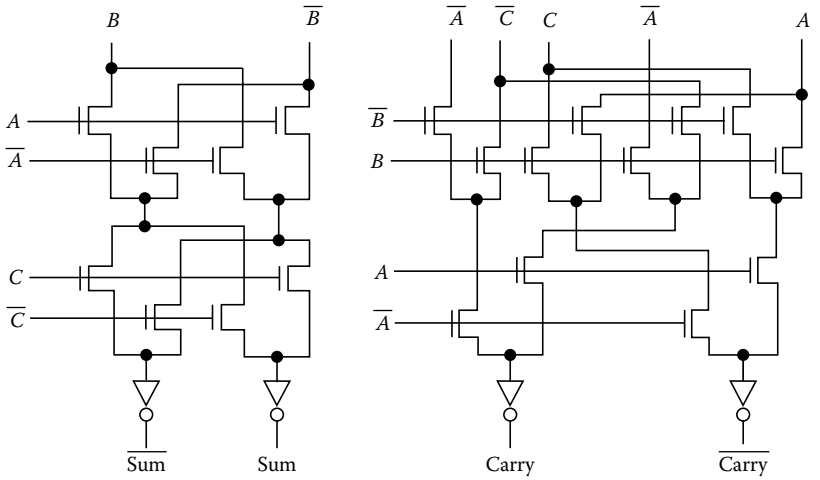


FIGURE 8.34 CPL full adder circuit.

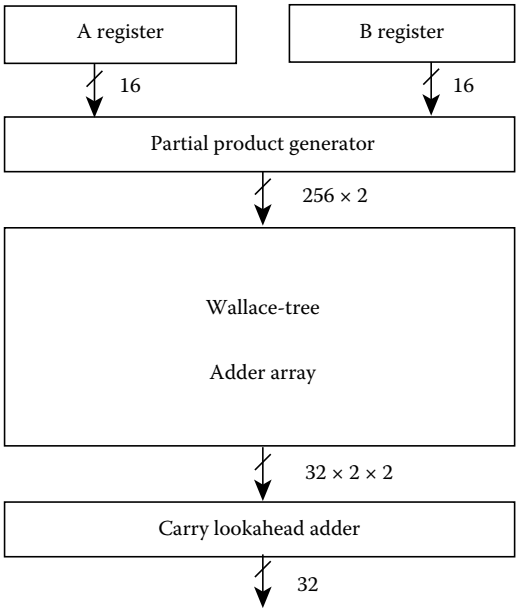


FIGURE 8.35 Block diagram of the 16 × 16 bit multiplier.

Due to the continued device miniaturization and the recent drive of portable systems, VLSI systems have been pushing toward low-voltage, low-power operation. Various techniques from system level to device level were developed to reduce the operating voltage and the power consumption of the VLSI circuits [7,8]. The low-power design can be addressed at four levels: algorithm, architecture, logic style, and integration. At the logic design level, capacitive loads are to be reduced and the number of charging/discharging operations are to be minimized. A CPL is one of the most attractive logic families that can achieve very low power consumption. The input capacitance in CPL is about half that of the CMOS configuration because pMOS can be eliminated in logic organization. Therefore, CPL can achieve a higher speed and dissipate less power. Experimental results [5] show that for the same delay time of the CMOS full adder operating at 5 V, the CPL adder requires only a 2 V supply. As the supply voltage decreases, the delay time will increase, but the power-delay product will decrease. Hence, it is desirable to operate at the slowest allowable speed to reduce power dissipation. Experimental results indicate that performance of CPL logic style is better than the conventional CMOS logic style from the viewpoint of power consumption.

8.2.2 Analog Processing

8.2.2.1 MOS Operational Amplifier Compensation

The frequency stabilization of a basic two-stage CMOS amplifier can be achieved by using a pole-splitting capacitor C_C [9]. The pole p_1 due to the capacitive loading of the first stage is pushed down to a very low frequency, and the pole p_2 due to the capacitance at the output node of the second stage is pushed to a very high frequency. However, a right-half-plane zero is introduced by the feedthrough effect of the compensation capacitor C_C. It will degrade the stability of the op-amp and make the second stage behavior like a short-circuited configuration at high frequencies. In order to remove the effects of the zero, a source follower can be inserted in the path from the output back through the compensation capacitor. Another approach is to insert a nulling resistance, R_Z, in series with the compensation capacitor. If R_Z is set to $1/g_{M2}$, where g_{M2} is the transconductance of the second stage, the zero

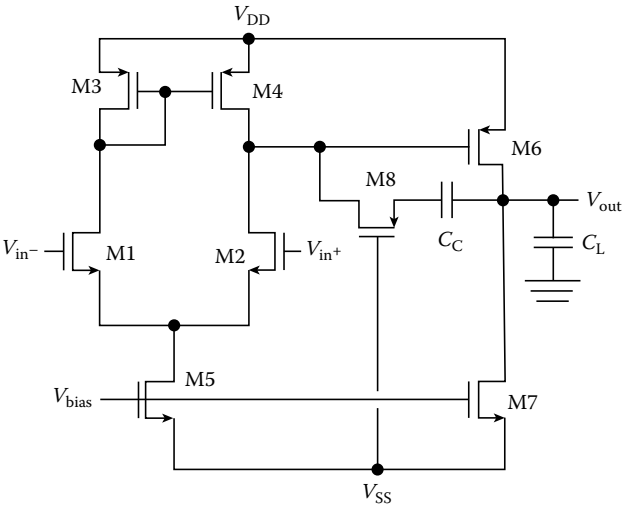


FIGURE 8.36 CMOS op-amp.

vanishes and the feedthrough effect is cancelled out. A single transistor or complementary version of the TG can be used to implement R_Z. Figure 8.36 is the schematic diagram of a basic two-stage op-amp supplemented by a feedback branch (M8, C_C) for compensation [10]. Capacitance C_L is the load capacitance to be driven by the amplifier. The pMOS TG M8 is biased in the triode region and provides the equivalent resistance.

TGs can be used to construct the cascode configuration of an op-amp. A fully differential folded-cascode op-amp is shown in Figure 8.37 [10]. The output cascode stage consists of TGs M5 to M10.

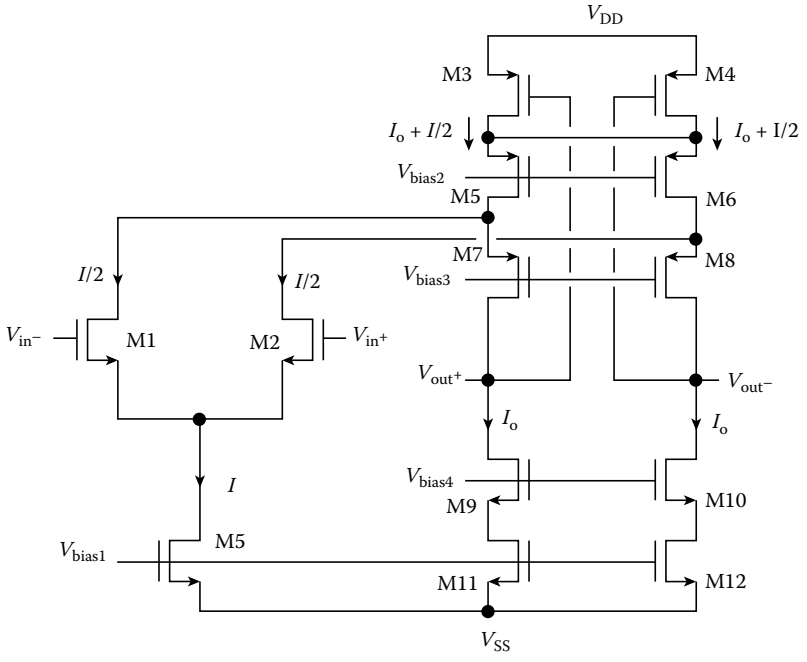


FIGURE 8.37 A fully differential CMOS op-amp with stabilized DC output level.

A high output impedance can be achieved by using the split-load arrangement. The bias voltage V_{bias1} establishes the bias current I of the input stage, and a bias current I_{o} in the output transistors M7 to M12. Thus, each transistor of M3 to M6 has a stabilized current $I_{\text{o}} + I/2$. The source voltages of M5 and M6 are stabilized because they conduct stabilized currents and their gate voltages are fixed at V_{bias2}. This fixes $|v_{\text{DS}\,3}|$ and $|v_{\text{DS}\,4}|$. Let transistors M3 and M4 have the same W/L ratio and bias them in the triode region by choosing a suitable value for V_{bias2}. If the output common-mode voltage $v_{\text{o,c}}$ drops, the resistance of M3 and M4 reduces, which increases $|v_{\text{GS}\,5}|$ and $|v_{\text{GS}\,6}|$. Because the current in M5 and M6 remains unchanged, $|v_{\text{DS}\,5}|$ and $|v_{\text{DS}\,6}|$ are forced to decrease. Then, the drain voltages of M5 and M6 increase, which increases $|v_{\text{GS}\,7}|$ and $|v_{\text{GS}\,8}|$. Therefore, $|v_{\text{DS}\,7}|$ and $|v_{\text{DS}\,8}|$ reduce which forces v_{o}^{+} and v_{o}^{-} to rise. The common-mode voltage $v_{\text{o,c}}$ is thus increased. This approach can increase the common-mode rejection ratio (CMRR) of the op-amp. The negative feedback scheme tends to keep $v_{\text{o,c}}$ at a constant value. It means that the small-signal common-mode output is zero or a very small value. Thus, a high CMRR is achieved.

8.2.2.2 Transimpedance Compensation

The optical receiver is an important component of the optical fiber communication system. One of the basic modules in a high performance optical receiver is the low-noise preamplifier. Several approaches are available to design the preamplifier. One approach is to use the transimpedance design which can avoid the equalization and limited dynamic range problems by using negative feedback. TGs can be used to provide the feedback resistance for a transimpedance amplifier.

A complete preamplifier circuit schematic is given in Figure 8.38 [11]. This circuit consists of three gain stages and two TGs. Each gain stage is composed of a pMOS current source achieving a common-source amplification with a folded nMOS load. One TG transistor, M10, functions as a feedback resistor and the other, M11, functions to implement the automatic gain control function. The gate voltage of M10 is derived from another circuit which minimizes the temperature and power supply dependence of the feedback resistance [11]. Transistor M11 is controlled by the automatic gain control voltage [12] and is normally off. If the input current to the preamplifier forces the output voltage out of its linear range, M11

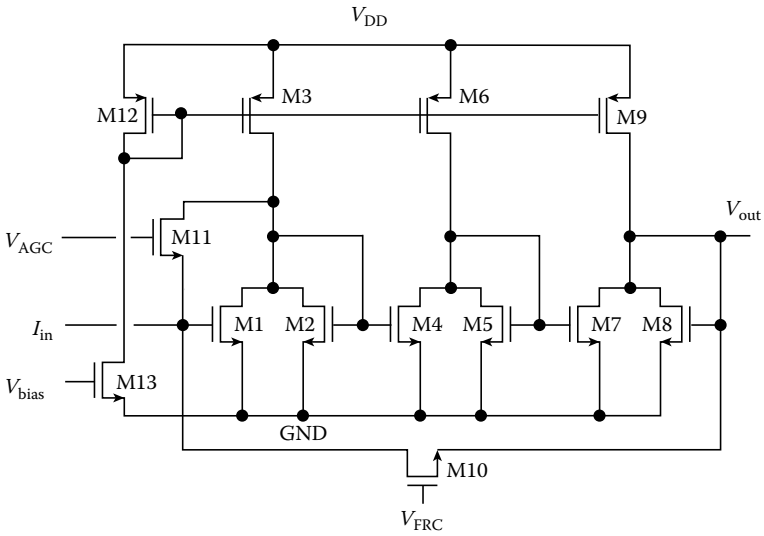


FIGURE 8.38 Circuit schematic of a preamplifier.

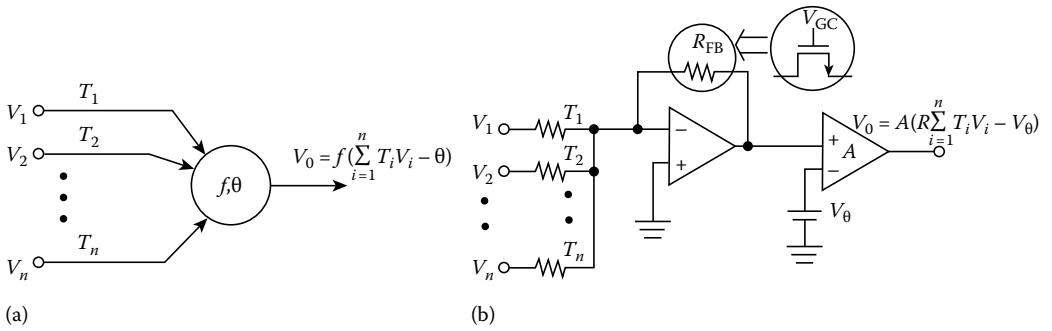


FIGURE 8.39 Neuron and synapse operation: (a) mathematical model and (b) analog circuit model with adjustable gain.

is turned on and begins to shunt current away from the feedback resistor and into the first stage output node.

With recent progress in intelligent information processing, artificial neural networks can be used to perform several complex functions in scientific and engineering applications, including classification, pattern recognition, noise removal, optimization, and adaptive control [13]. Design and implementation of VLSI neural networks have become a very important engineering task. The basic structure of an artificial neural network consists of a matrix of synapse cells interconnecting an array of input neurons with an array of output neurons. The inputs, V_i, are multiplied by weight values, T_i, of the synapses. The results of the multiplication are summed and compared to the threshold value θ_i in the output neurons. Schematic diagrams of a mathematical model of a neuron and its electronic counterpart are shown in Figure 8.39. The circuit in Figure 8.39b uses a gain-controllable amplifier in which the voltage gain is controlled by changing the feedback resistance. The feedback resistor R_{FB} can be constructed by the TG structure so that feedback resistance can be adjusted by the gain-control voltage V_{GC} [14].

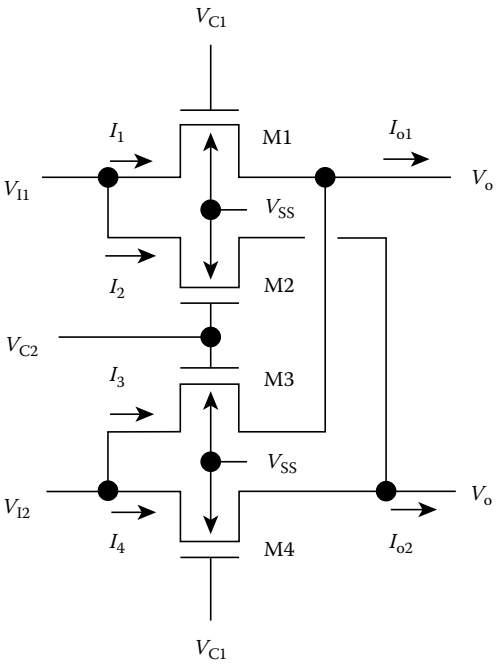


FIGURE 8.40 Double-MOS implementation of a differential AC resistor.

8.2.2.3 Continuous-Time Filters

Resistors are important components in the construction of continuous time filters [15]. However, the implementation of resistors by integrator circuit (IC) fabrication technologies was found to be lacking in several areas of performance. The TG can be used to realize active resistance. For example, a double-MOS differential configuration, shown in Figure 8.40, is used to implement a differential AC resistor [16].

This circuit consists of four nMOS TGs. Not only can it linearize the AC resistor, but it can also eliminate the effects of the bulk-source voltage [17]. To determine the AC resistance, assume that all the transistors are matched and are biased in the triode region.

The current I_{o1} and I_{o2} can be expressed as

$$\begin{aligned} I_{o1} &= I_1 + I_3 \\ &= \mu_n C_{OX}(W/L)\left[(V_{C1} - V_0 - V_{thn})(V_{I1} - V_0) - (1/2)(V_{I1} - V_0)^2\right] \\ &\quad + \mu_n C_{OX}(W/L)\left[(V_{C2} - V_0 - V_{thn})(V_{I2} - V_0) - (1/2)(V_{I2} - V_0)^2\right] \end{aligned} \tag{8.35}$$

$$\begin{aligned} I_{o2} &= I_2 + I_4 \\ &= \mu_n C_{OX}(W/L)\left[(V_{C2} - V_0 - V_{thn})(V_{I1} - V_0) - (1/2)(V_{I1} - V_0)^2\right] \\ &\quad + \mu_n C_{OX}(W/L)\left[(V_{C1} - V_0 - V_{thn})(V_{I2} - V_0) - (1/2)(V_{I2} - V_0)^2\right] \end{aligned} \tag{8.36}$$

Equations 8.35 and 8.36 can be combined to determine the differential current

$$I_{o1} - I_{o2} = \mu_n C_{OX}(W/L)[(V_{C1} - V_{C2})(V_{I1} - V_{I2})] \tag{8.37}$$

Thus, r_{ac} is given by

$$r_{ac} = \frac{V_{I1} - V_{I2}}{I_{o1} - I_{o2}} = \frac{1}{\mu_n C_{OX}(W/L)(V_{C1} - V_{C2})} \tag{8.38}$$

Because all transistors are required to be biased in the triode region, Equation 8.38 holds when

$$V_{I1}, V_{I2} \leq \min\left[V_{C1} - V_{thn}, V_{C2} - V_{thn}\right] \tag{8.39}$$

The double-MOSFET differential resistor is really a transresistance, thus, it can be applied only to differential-in, differential-out op-amps.

8.2.2.4 Switched-Capacitor Circuits

Switched-capacitor circuits make use of TGs in processing the analog signals [10,16]. This approach uses switches and capacitors and is in discrete time. If the clock rate is much higher than the signal frequency, an AC resistor can be implemented by combining switches and capacitors. The equivalent resistance is dependent only on the clock rate and the capacitor. The circuit schematic diagram of the direct digital integrator (DDI) is shown in Figure 8.41. The resistance is realized by two MOS switches and one capacitor. The difference equation can be expressed as

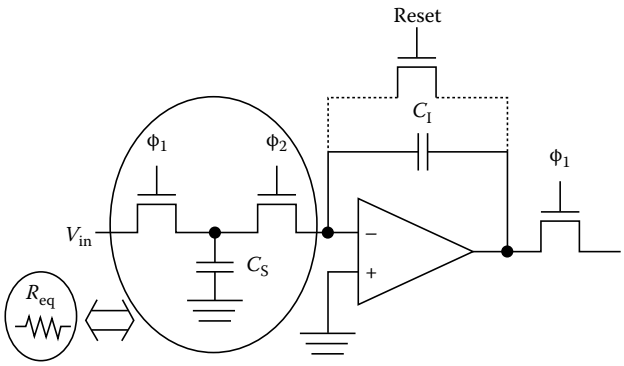


FIGURE 8.41 Direct digital integrator.

$$v_{0,n+1} = v_{0,n} - \frac{C_S}{C_I} v_{in} \tag{8.40}$$

After taking the z-transform, the new expression becomes

$$z \cdot V_0(z) = V_0(z) - \frac{C_S}{C_I} V_{in}(z) \tag{8.41}$$

By rearranging the various terms, the transfer function of the DDI can be expressed as

$$\frac{V_0(z)}{V_{in}(z)} = -\frac{C_S}{C_I} \cdot \frac{z^{-1}}{1 - z^{-1}} \tag{8.42}$$

By setting $z = e^{j\omega T}$, the frequency response can be determined:

$$\frac{V_0}{V_{in}}(j\omega) = -\frac{C_S}{C_I} \frac{1}{j\omega T} \cdot \frac{\omega T/2}{\sin(\omega T/2)} \cdot e^{-j\omega T/2} \tag{8.43}$$

where T is the period of the clock. In Equation 8.43, the first term corresponds to an ideal integrator, the second term contributes to the magnitude error, and the third term is the phase error. Because the frequency response of the ideal integrator is $-[j\omega R_{eq} C_I]^{-1}$, the equivalent resistance value is determined by

$$R_{eq} = \frac{T}{C_S} \tag{8.44}$$

A ladder network can be constructed by cascading the DDI integrators. In the ladder network all cascaded stages sample the input signal at clock Φ_1 and transform the signal at clock Φ_2, where Φ_1 and Φ_2 are nonoverlapping clock signals. This clocking scheme induces the extra half-cycle phase delay. This phase error can cause extra peaking in frequency response and generate cyclic response. In order to remove the excess phase, other integrators, such as lossless digital integrators (LDI) or bilinear integrators, can be used. In an LDI ladder network, the odd-number stages sample the input signal at clock Φ_1 and transform the signal at clock Φ_2, while the even-number stages sample the input signal at clock Φ_2 and transform the signal at clock Φ_1. Thus, the frequency response of an LDI integrator can be expressed by

$$\frac{V_0}{V_{in}}(j\omega) = -\frac{C_S}{C_I} \frac{1}{j\omega T} \cdot \frac{\omega T/2}{\sin(\omega T/2)} \tag{8.45}$$

Figure 8.42 is the circuit schematic diagram of the bottom-plate differential-input LDI. Output of an LDI integrator is more insensitive to parasitic components.

Figure 8.43 is the circuit schematic diagram of a differential bilinear integrator. The transfer function of the bilinear integrator is

$$\frac{V_0^+ - V_0^-}{V_{in}^+ - V_{in}^-} = \frac{C_S}{C_I} \cdot \frac{1 + z^{-1}}{1 - z^{-1}} \tag{8.46}$$

As the output of the bilinear integrator does not change during clock Φ_1, it can be used to feed another identical integrator.

TGs can be used to initialize the switched-capacitor circuits. For example, capacitor C_I in Figure 8.41 is to perform the integration function and to be reset or discharged before operation. An nMOS TG can be

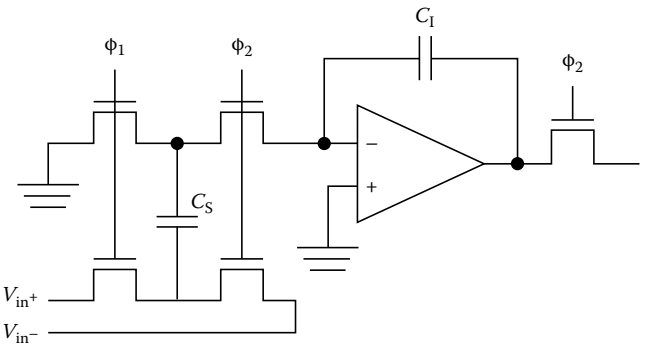


FIGURE 8.42 Bottom-plate differential-input lossless digital integrator.

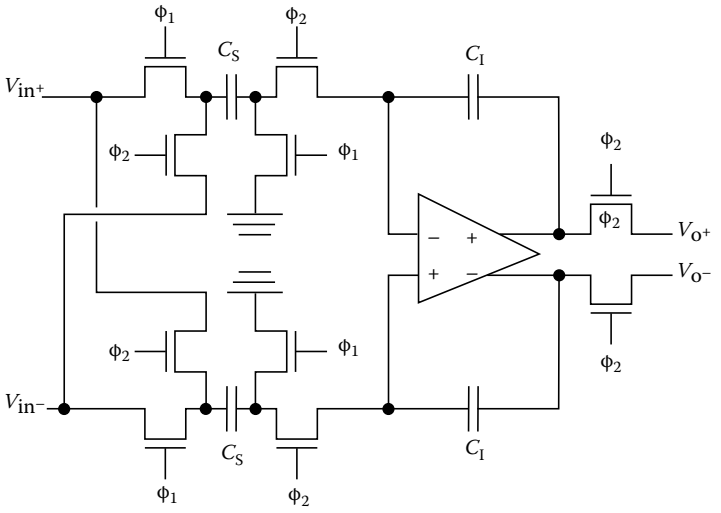


FIGURE 8.43 Differential bilinear integrator.

put in parallel with the capacitor C_I. Before normal operation, the TG is turned on and the capacitor C_I is discharged so that the initial capacitor voltage value is reset to zero.

The accuracy of switched-capacitor circuits is disturbed by charge injection when the controlling switch turns off. The turn-off of an MOS switch consists of two phases. The gate voltage is higher than the transistor threshold voltage V_{th} during the first phase. A conduction channel extends from the source to the drain of the transistor. As the gate voltage decreases, mobile carriers exit through both the drain and the source terminals and the channel conduction decreases. During the second phase, the gate voltage is smaller than V_{th} and the conduction channel no longer exists. The coupling between the gate and the data-holding node is only through the gate-to-diffusion overlap capacitance. The following analysis is focused on the switch charge injection due to the first phase of the switch turn-off.

Figure 8.44 is the circuit schematic corresponding to the general case of switch charge injection. Capacitance C_L is the lumped capacitance at the data-holding node. Capacitance C_S could be the lumped capacitance associated with the amplifier output node, while resistance R_S could be the output resistance of an op-amp. Let C_G represent the total gate capacitance of the switch, including both the channel

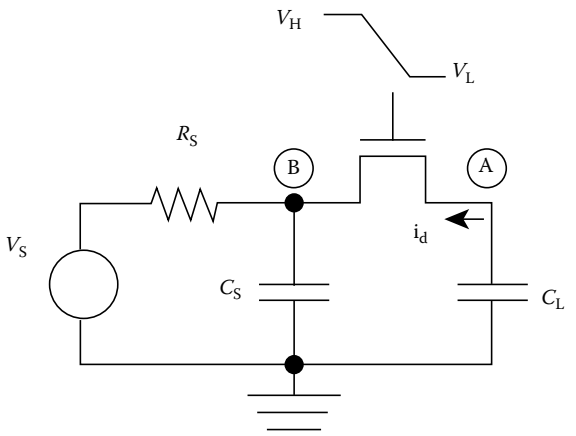


FIGURE 8.44 Circuit for analysis of switch charge injection.

capacitance and gate-to-drain/gate-to-source overlap capacitances. Kirchhoff's current law at node *A* and node *B* requires

$$C_L \frac{dv_L}{dt} = -i_d + \frac{C_G}{2} \frac{d(V_G - v_L)}{dt} \tag{8.47}$$

and

$$\frac{v_S}{R_S} + C_S \frac{dv_S}{dt} = i_d + \frac{C_G}{2} \frac{d(V_G - v_S)}{dt} \tag{8.48}$$

where v_L and v_S are the error voltages at the data-holding node and the signal-source node, respectively. Gate voltage is assumed to decrease linearly with time from the turn-on value V_H:

$$V_G = V_H - \alpha t \tag{8.49}$$

where α is the falling rate. When the transistor is biased in the strong inversion region,

$$i_d = \beta(V_{HT} - \alpha \cdot t)(v_L - v_S) \tag{8.50}$$

where

$$\beta = \mu C_{OX} \frac{W}{L} \tag{8.51}$$

and

$$V_{HT} = V_H - V_S - V_{thn} \tag{8.52}$$

Here, V_{thn} is the transistor effective threshold voltage, including the body effect. For small-geometry transistors, narrow- and short-channel effects should be considered in determining the V_{thn} value. Under the condition $|dV_G/dt| \gg |dV_L/dt|$ and $|dV_S/dt|$, and Equations 8.47 and 8.48 can be simplified to

$$C_L \frac{dv_L}{dt} = -\beta(V_{HT} - \alpha t)(v_L - v_S) - \frac{C_G}{2}\alpha \tag{8.53}$$

and

$$\frac{v_S}{R_S} + C_S \frac{dv_S}{dt} = \beta(V_{HT} - \alpha t)(v_L - v_S) + \frac{C_G}{2}\alpha \tag{8.54}$$

No closed-form solution to this set of equations can be found. Numerical integration can be employed to find final results. Analytical solutions to special cases are given next.

Figure 8.45a is the circuit schematic diagram of the case, with only a voltage sourced at the signal-source node. Because $C_S \gg C_L$, v_S can be approximated as zero and the governing equation reduces to

$$C_L \frac{dv_L}{dt} = -\beta(V_{HT} - \alpha t)v_L - \frac{C_G}{2}\alpha \tag{8.55}$$

When the gate voltage reaches the threshold condition, the error voltage at the data-holding node is

$$v_L = -\sqrt{\frac{\pi\alpha C_L}{2\beta}}\left(\frac{C_G}{2C_L}\right)\text{erf}\left(\sqrt{\frac{\beta}{2\alpha C_L}}V_{HT}\right) \tag{8.56}$$

Notice that the value of the error function erf(·) can be found from mathematical tables.

Another special case is when the source capacitance is negligibly small, as is shown in Figure 8.45b. The governing equations reduce to

$$C_L \frac{dv_L}{dt} = -\beta(V_{HT} - \alpha t)(v_L - v_S) - \frac{C_G}{2}\alpha \tag{8.57}$$

and

$$\frac{v_S}{R_S} = \beta(V_{HT} - \alpha t)(v_L - v_S) + \frac{C_G}{2}\alpha \tag{8.58}$$

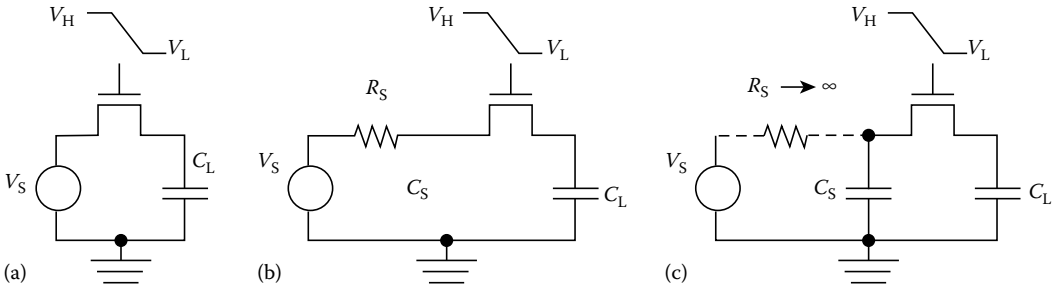


FIGURE 8.45 Special cases of switch charge injection: (a) no source resistance and capacitance; (b) no source capacitance; and (c) infinitely large source resistance.

When the gate voltage reaches the threshold condition, the error voltage at the data-holding node is

$$v_L = -\frac{\alpha C_G}{2C_L}\exp\left(-\frac{V_{HT}}{\alpha C_L R_S}\right)\cdot\int_0^{V_{HT}/\alpha}[\beta R_S(V_{HT}-\alpha\theta)+1]^{1/C_L\beta R_S^2\alpha}$$
$$\cdot\exp\left(\frac{\theta}{C_L R_S}\right)\left(2-\frac{1}{1+\beta R_S(V_{HT}-\alpha\theta)}\right)d\theta \tag{8.59}$$

If a time constant $R_S C_S$ is much larger than the switch turn-off time, then the channel charge will be shared between C_S and C_L, as shown in Figure 8.45c. For the case of a symmetrical transistor and $C_S = C_L$, half of the channel charge will be deposited to each capacitor. Otherwise the following equations can be used to find the results:

$$C_L\frac{dv_L}{dt} = -\beta(V_{HT}-\alpha t)(v_L - v_S) - \frac{C_G}{2}\alpha \tag{8.60}$$

and

$$C_S\frac{dv_S}{dt} = \beta(V_{HT}-\alpha t)(v_L - v_S) + \frac{C_G}{2}\alpha \tag{8.61}$$

We can multiply Equation 8.61 by the ratio C_L/C_S and then subtract the result from Equation 8.60 to obtain

$$C_L\frac{d(v_L - v_S)}{dt} = -\beta(V_{HT}-\alpha t)\left(1+\frac{C_L}{C_S}\right)(v_L - v_S) - \frac{\alpha C_G}{2}\left(1-\frac{C_L}{C_S}\right) \tag{8.62}$$

When the gate voltage reaches the threshold condition, the amount of voltage difference between the data-holding node and the signal-source node becomes

$$v_L - v_S = -\sqrt{\frac{\pi\alpha C_L}{2\beta(1+C_L/C_S)}}\left(\frac{C_G(1-C_L/C_S)}{2C_L}\right)$$
$$\cdot\,\mathrm{erf}\left(\sqrt{\frac{\beta(1+C_L/C_S)}{2\alpha C_L}}V_{HT}\right) \tag{8.63}$$

References

1. N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, 2nd ed. Reading, MA: Addison-Wesley, 1993.
2. J. P. Uyemura, *Fundamentals of MOS Digital Integrated Circuits*, Reading, MA: Addison-Wesley, 1988.
3. D. Radhakrishnan, S. R. Whitaker, and G. K. Maki, Formal design procedures for pass transistor switching circuits, *IEEE J. Solid State Circuits*, 20(2), 531–536, April 1985.
4. Y. Suzuki, K. Odagawa, and T. Abe, Clocked CMOS calculator circuitry, *IEEE J. Solid State Circuits*, 8(6), 734–739, Dec. 1973.
5. K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, A 3.8-ns CMOS 16×16-b multiplier using complementary pass-transistor logic, *IEEE J. Solid State Circuits*, 25(2), 388–395, April 1990.

6. Y. Oowaki et al., A 7.4ns CMOS 16 × 16 multiplier, in *IEEE: International Solid-State Circuits Conference Digest of Technical Papers*, San Francisco, California, 1987.
7. A. P. Chandrakasan, S. Sheng, and R.W. Brodersen, Low-power CMOS digital design, *IEEE J. Solid-State Circuits*, 27(4), 473–484, April 1992.
8. K. Shimohigashi and K. Seki, Low-voltage ULSI design, *IEEE J. Solid-State Circuits*, 28(4), 408–413, April 1993.
9. P. G. Gray and R. G. Meyer, MOS operational amplifier design—A tutorial overview, *IEEE J. Solid-State Circuits*, 17(6), 969–982, Dec. 1982.
10. R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, New York: John Wiley & Sons, 1986.
11. D. M. Pietruszynski, J. M. Steininger, and E. J. Swanson, A 50-Mbit/s CMOS monolithic optical receiver, *IEEE J. Solid-State Circuits*, 23(6), 1426–1433, Dec. 1988.
12. G. Williams, U.S. Patent, 4,574,249, Mar. 4, 1986.
13. P. K. Simpson, Foundations of neural networks, in *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementations*, E. Sánchez-Sinencia and C. Lau, Eds. New York: IEEE Press, 1992, pp. 3–24.
14. S. M. Gowda, B. J. Sheu, J. Choi, C.-G. Hwang, and J. S. Cable, Design and characterization of analog VLSI neural network modules, *IEEE J. Solid-State Circuits*, 28(3), 301–313, Mar. 1993.
15. M. Ismail, S. V. Smith, and R. G. Beale, A new MOSFET-C universal filter structure for VLSI, *IEEE J. Solid-State Circuits*, 23(2), 183–194, Feb. 1988.
16. R. E. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, New York: McGraw-Hill, 1990.
17. M. Banu and Y. Tsividis, Fully integrated active RC filters in MOS technology, *IEEE J. Solid-State Circuits*, 18(6), 644–651, Dec. 1983.

9

Digital Systems

9.1 Programmable Logic Devices **9**-1
PLD Device Technologies • PLD Notation • Programmable Logic Array • Programmable Read Only Memory • Programmable Array Logic • Classification of Combinational Logic PLD Devices • Designing with Combinational Logic PAL Devices • Designing with Sequential PAL Devices • Designing with PALs Having Programmable Macrocell Outputs • FPGA Technologies • FPGA Architectures • Design Process • VHDL Synthesis Style for FPGAs • Synthesis of State Machines
References **9**-32
9.2 Clocking Schemes **9**-32
Introduction • Clocking Principles • Clock Distribution Schemes • Future Directions
References **9**-70
9.3 MOS Storage Circuits **9**-72
Dynamic Charge Storage • Shift Register • Dynamic CMOS Logic
References **9**-82
9.4 Microprocessor-Based Design **9**-82
Introduction • Features of a Microprocessor-Based System • Memory • Microprocessor Architecture • Design with a General Purpose Microprocessor • Interfacing • Design with a Microcontroller • Design Guidelines
References **9**-111
9.5 Systolic Arrays **9**-111
Concurrency, Parallelism, Pipelining, and Systolic Array • Digital Filters • Systolic Word and Bit-Level Designs • Recursive LSs Estimation • Kalman Filtering • Eigenvalue and SVDs
References **9**-142

Festus Gail Gray
Virginia Polytechnic Institute and State University

Wayne D. Grover
University of Alberta

Josephine C. Chang
University of Southern California

Bing J. Sheu
Taiwan Semiconductor Manufacturing Co.

Roland Priemer
University of Illinois at Chicago

Kung Yao
University of Southern California, Los Angeles

Flavio Lorenzelli
University of Milan, Crema

9.1 Programmable Logic Devices

Festus Gail Gray

Traditional programmable logic devices (PLDs) and field programmable gate arrays (FPGAs) allow circuit designers to implement logic circuits with fewer chips relative to standard gate level designs based on primitive gates and flip-flops. As a result, layout and unit production costs are generally reduced. In this chapter, we use the term ''programmable device'' to refer to the class of moderately complex single-chip devices, in which the user in the field can program the function of the device. We include

TABLE 9.1 Complexity Ladder of Devices

Device	Complexity of a Single Chip	Range of Realizable Functions	Initial Design and Cost of Design Change	Unit Product Cost (High Volume)	Time to Market
SSI discrete gate chip	Lowest	All functions	High	High	High
MSI chip		Narrow	High	High	High
PLD[a]		Moderate	Low	Moderate	Low
LSI (PROM)[a]		All combinational functions with n inputs	Very low	Moderate	Low
FPGA		Wide	Low	Moderate	Low
MPGA		Wide	High	Low	High
Standard cell		Wide	High	Low	High
Custom chip	Highest	All functions	Very high	Very low	Very high

[a] Field programmable.

such devices as the programmable logic array (PLA), programmable array logic (PAL), programmable read-only memory (PROM), and the FPGA. Since most commercial vendors provide software design aids for mapping designs to their specific chips, initial design costs and time to market are low. Another important advantage of programmable device designs is ''flexibility.'' Design changes do not require physical changes to the printed circuit board as long as the revised functions still fit onto the same programmable chip. The low cost of design revisions makes programmable chips very attractive for prototype design and low volume production. Designers often move up the design ladder once proven designs move into high volume production.

Table 9.1 shows the position of PLDs, PROMs, and FPGAs on the complexity ladder of device types. In the ''range of realizable functions'' column, we compare the range of realizations for various device types. Discrete gates can implement any function if enough gates are available. MSI chips implement very specialized functions such as shift registers, multiplexers (MUXs), decoders, etc. Table 9.1 compares programmable devices (PLDs, FPGAs, and PROMs) relative to the range of functions that can be implemented on a single chip. A PROM chip with n address inputs can implement any combinational function of n variables. A PLD chip with n inputs can implement only a subset of the combinational functions of n variables. Gate arrays can implement a wide range of both combinational and sequential functions. The programmable devices are characterized by low time to market, low design cost, low cost of modifications, and moderate production costs. Nonfield programmable devices such as mask programmable gate arrays (MPGAs), standard cell devices, and full custom devices are characterized by high initial design costs and longer time to market, but have lower volume production costs. Custom chips are preferred for large volume production because of the very low unit production costs. However, initial design costs and the cost of design changes are very high for custom chip design. Also, the design of custom chips requires highly trained personnel and a large investment in equipment. The low design cost, low cost of design changes, and low time to market make PLDs and FPGAs good choices for lower volume production and for prototype development.

The primary difference between PLDs and FPGAs arise because of a difference in the ratio of the number of combinational logic (CL) gates to the number of flip-flops. PLD devices are better for applications that require complex CL functions that drive a relatively small number of flip-flops, such as finite-state machine (FSM) controllers or purely CL functions. FPGAs are better for devices that require arithmetic operations (adders, multipliers, and arithmetic logic units [ALUs]), or that require a large number of registers and less complex CL functions, such as digital filters.

9.1.1 PLD Device Technologies

Companies produce PLD devices in different technologies to meet varying design and market demands. There are two categories of technologies. ''Process technology'' refers to the underlying semiconductor

structure, which affects device speed, power consumption, device density, and cost. ''Programming technology'' refers to the physics of chip programming and affects ease of programming and the ability to reprogram or to reconfigure chips.

9.1.1.1 Process Technologies

The dominant technologies in PLD devices are bipolar and CMOS. Bipolar devices are faster and less expensive to manufacture, but consume more power than CMOS devices. The higher power requirements of bipolar devices limit the gate density. Typical CMOS devices, therefore, achieve much higher gate densities than bipolar devices. The power consumption of CMOS devices depends on the application because a CMOS device only consumes power when it is switching states. The amount of power consumed increases with the speed of switching. Therefore, the total amount of power consumed depends on the frequency and speed of state changes in the device. Some devices have programmable power standby activation that puts the device in a lower power consumption configuration if no input signal changes for a predefined amount of time. The device then responds to the next input change much slower than normal but switches back to the faster speed configuration and maintains the faster speed as long as input changes continue to occur frequently. When programmed in the ''standby power mode,'' power consumption is reduced on the average at the expense of response time of the device. When programmed to operate in the ''turbo'' mode, the device stays in the faster configuration at all times. The result is higher power consumption, but faster response time. The mode of operation is selected by the user to match the requirements of an application.

To take advantage of the higher densities of CMOS devices and still be compatible with bipolar devices, many CMOS PLAs have special driver circuits at the input and output pins to allow pin compatibility with popular bipolar devices such as the commonly used TTL devices.

ECL is a very high-speed technology used in some PLD devices. Although ECL has the highest speed of the popular technologies, the power consumption is very high which severely limits the gate density.

Security is another issue that is related to process technology. Many PLDs have a programmable option that prevents reading the program. Since the software provided by most manufacturers allows the user to read the program in the chip in order to verify correct programming, it is extremely easy to copy designs. To prevent illegal copying of patented designs, one simply blows the ''security fuse,'' which permanently prevents anyone from reading the program by normal means. However, the program in most bipolar circuits can easily be read by removing the case and examining the programmed fuses under a microscope. CMOS PLDs are much more secure because it is virtually impossible to determine the program by examining the circuit.

9.1.1.2 Programming Technologies

The programming technologies used in PLDs are virtually the same as the programming technologies available for read-only memories (ROMs). Programming technologies are divided into two broad categories: mask programmable devices and field programmable devices.

In mask programmable technologies, identical base chips are produced in mass. The final metallization step is simply omitted. A mask programmable PLD chip is programmed by performing a final metal deposition step that selects the programming options. Clearly, this step must be performed at the manufacturer's plant. The user makes entries on order forms that specify how the chip is to be programmed and sends it to the manufacturer. The manufacturer must then prepare one or more production masks prior to making the chip. Mask programmable devices incur a high setup cost to make the first device, but unit production costs are typically less than half of that for field programmable technologies. The usual practice is to use field programmable devices for prototype work and implement only proven designs in mask programmable technologies when a large production volume is required. Many PLD devices are available in both mask programmable and field programmable versions, which make the conversion easy and reliable.

The user can program field programmable technologies directly. Specialized equipment is needed. Modern programming devices can actually program both ROM and PLD devices. The programmer is typically controlled by a small computer (PC) and uses files prepared in standard format (JEDEC) by software provided by the manufacturer or written by software vendors. Such software can include elegant features such as a programming language (ABEL, VHDL, Verilog, etc.), truth table input, equation input, or state machine input. Selection of a chip vendor should include careful evaluation of the support software for programming the chip.

Field programmable PLD technologies can be classified into three broad categories: fusible link PLDs, ultraviolet erasable PLDs (EPLDs), and electrically erasable PLDs (EEPLDs). Field programmable ROMS come in analogous forms: fusible link ROMs (PROMs), ultraviolet erasable ROMS (EPROMs), and electrically erasable ROMS (EEPROMs).

Fusible link PLDs typically utilize bipolar process technology. The programmer blows selected fuses in the device. Because higher than normal voltages and currents are required to blow the fuses, programming fusible link PLDs can be quite stressful for the device. Overheating is a common problem. However, this technology is quite well developed and the design of programming devices is sufficiently mature so that reliable results can be expected as long as directions are carefully followed. Fusible link technologies provide the convenience of on-site programming, which reduces the time required to develop designs and the time required to make design changes. The trade-off involves at least a twofold increase in per unit cost and a significant reduction in device density relative to mask programmable devices because the fuses take up considerable chip space. A fusible link PLD can be programmed only once because the blown fuses cannot be restored.

''Ultraviolet EPLDs'' have a window on the top of the chip. Programming the chip involves storing charges at internal points in the circuit that control switch settings. Shining ultraviolet light through the window on the chip can dissipate the charges. Therefore, EPLDs provide the convenience of reprogramming as a design evolves. On the downside, EPLDs cost at least three times as much per chip as mask programmable PLDs and operate at much slower speeds. Since EPLDs typically utilize CMOS technology, they are slower than fusible link PLDs, but require less power. Therefore, EPLDs are often used in development work with the final design being implemented in either fusible link technology (for faster speed) or mask programmable technology (for faster speed and lower density). In spite of the fact that EPLDs cost more than fusible link PLDs, the reprogramming feature eventually results in a lower cost for development than using fusible link PLDs. This technology requires an additional piece of hardware to erase the chips.

EEPLDs provide the convenience of reprogramming without the need to erase the previous program because the chip is programmed by setting the states of flip-flops inside the device. It is, therefore, not necessary to purchase an erasing device. The reprogramming also requires less time to accomplish. Of course, EEPLD chips cost more and have a lower gate density than EPLD chips.

9.1.2 PLD Notation

PLDs typically have many logic gates with a large number of inputs. Also, there are often many gates that have the same set of inputs. For example, the PAL22V10 has 132 AND gates, each with the same 44 gate inputs. Obviously, the diagram for such a complex circuit using standard AND gate symbols would be extremely complex and difficult to read.

Figure 9.1 is the conventional diagram for an eight-input AND gate. Clearly, a similar diagram for a 44-input AND gate would be very cumbersome. Figure 9.2 is the same eight-input AND gate in PLD notation. The eight parallel

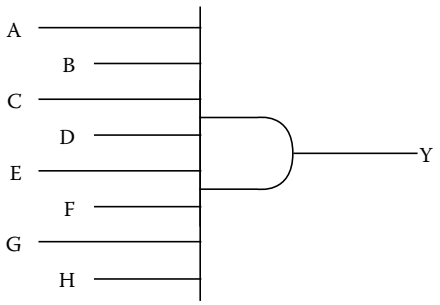


FIGURE 9.1 Conventional diagram for an eight-input AND gate.

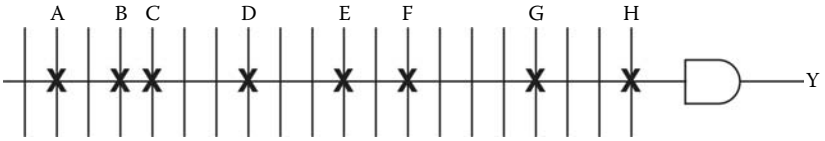


FIGURE 9.2 PLD notation for an eight-input AND gate.

wires that actually occur as inputs to the AND gate are represented by a single horizontal line in PLD notation. The actual inputs to the AND gate are drawn perpendicular to the single line. There are usually more signal lines than just the eight needed for this gate. An *X* is placed at the intersection of the single line with each of the perpendicular lines that provide actual inputs to the AND gate. Keep in mind that the single horizontal line actually represents eight parallel wires that are not physically connected to each other.

By comparing the internal structures of PLAs, PALs, and PROMs, we will describe the capabilities and limitations of each type of PLD. Since the PAL is currently the most popular PLD device, we will describe design methodology for both combinational and sequential PAL devices. By emphasizing the difference between designing with PALs and designing with standard logic gates, we provide practical insights about PLD design.

9.1.3 Programmable Logic Array

Figure 9.3 shows that the basic PLA consists of a programmable AND array followed by a programmable OR array. Vertical lines in the AND array represent the input variables (A, B, C, D). Since each input drives many AND gates, an internal buffer provides high current drive signals in both true and complemented format for each input variable. Initially, there is a connection from each input variable and its complement to each AND gate. In this example circuit, each AND gate initially has eight inputs ($A, \overline{A}, B, \overline{B}, C, \overline{C}, D, \overline{D}$). Each AND gate input line contains a fuse or electronic switch. We program the chip by blowing the fuses in lines that we do not need, or by programming the electronic switches. After

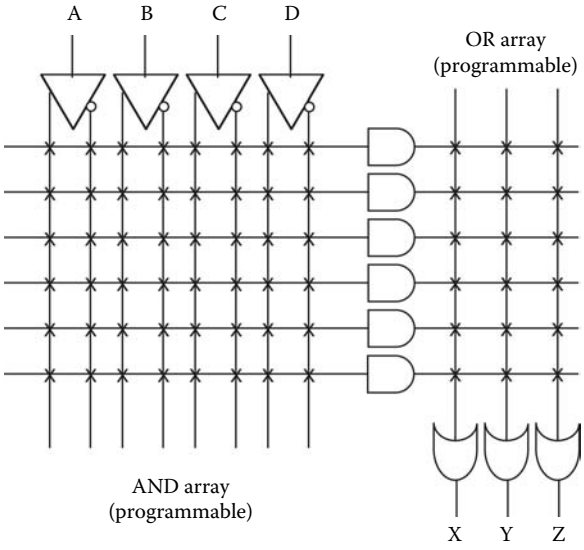
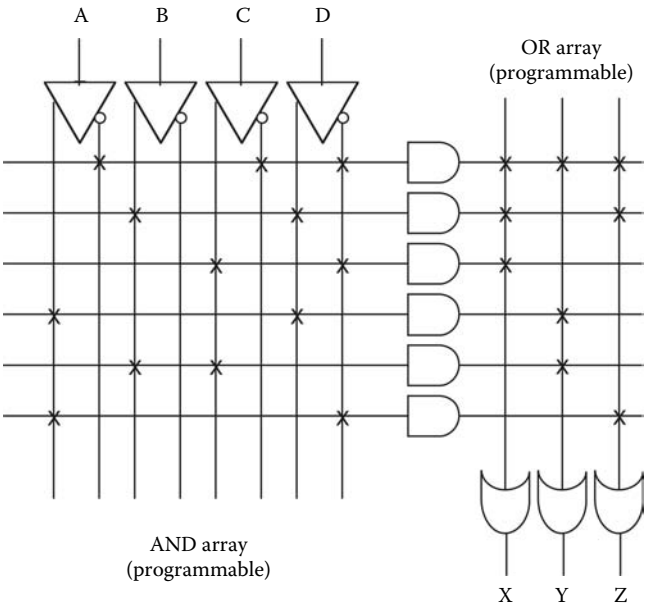


FIGURE 9.3 Basic architecture for a PLA.



$X = \overline{A} \cdot \overline{C} \cdot \overline{D} + B \cdot D + C \cdot \overline{D}$
$Y = \overline{A} \cdot \overline{C} \cdot \overline{D} + A \cdot D + B \cdot C$
$Z = \overline{A} \cdot \overline{C} \cdot \overline{D} + B \cdot D + A \cdot \overline{D}$

FIGURE 9.4 An example of a programmed PLA.

programming, we remove the X's from the lines that are disconnected. For example, in the programmed chip of Figure 9.4, the upper AND gate implements product term $(\overline{A} \cdot \overline{C} \cdot \overline{D})$.

In the OR array of Figure 9.3, there is an initial connection from each AND gate output to every input on each OR gate. Again, the single vertical line connected on the input side of each OR gate represents all six wires. Each of the input lines to the OR gates also contains a fuse or programmable switch. Figure 9.4 shows that, after programming, output X connects to product terms $\overline{A} \cdot \overline{C} \cdot \overline{D}$, $B \cdot D$, and $C \cdot \overline{D}$.

The number of product lines on a chip limits the range of functions that fit onto the chip. The PLA chip in Figure 9.3 can implement any three functions (X, Y, Z) of the same four variables (A, B, C, D) as long as the total number of required product terms is less than or equal to six. However, there are 100 different product terms involving four variables. So, practical chips have many more product lines than this contrived example.

In order to fit functions onto the chip, designers must be able to simplify multiple output functions using gate sharing whenever possible. Finding a minimal gate implementation of multiple output functions with gate sharing is a very complex task. The goal is to minimize the total number of gates used. The size of gates does not matter. For example, whether an AND gate has four inputs or two inputs is not important. All that changes are the number of fuses that are blown. This differs dramatically from the minimization goals when discrete gates are used. For discrete gate minimization, a four-input gate costs more than a two-input gate. Therefore, the classical minimization programs need to be modified to reflect the different goals for PLA development.

Three parameters determine the capacity of a PLA chip. Let n be the number of inputs, p the number of product term lines, and m the number of outputs. Then, the PLA chip can implement any m functions of the same n variables that require a total of p or fewer product terms. The device complexity is proportional to $(m + n)p$.

9.1.4 Programmable Read Only Memory

The PROM is the most general of the CL PLD devices described in this section. However, from a structural viewpoint, the PROM is a special case of the PLA in which the AND array is fixed and the OR array is programmable. Figure 9.5 is a conceptual diagram of a PROM. The filled circles in the AND array represent permanent connections. The X's in the OR array indicate that it is programmable. The number of product lines in a PROM is 2^n; whereas the number of product lines in a typical PLA is much smaller. A PROM has a product line for each combination of input variables. Since any logic function of n variables can be expressed in a canonical sum of minterms form in which each product term is a product of exactly n literals, the PROM can implement any function of its n input variables.

To demonstrate the generality of the PROM, Figure 9.6 shows how the PROM of Figure 9.5 must be programmed so as to implement the same set of logic functions that are programmed into the PLA of Figure 9.4. The PROM program follows directly from the truth table for a logic function. The truth table for the logic functions X, Y, and Z appears in Table 9.2. The correspondence between the truth table and the program in the PROM of Figure 9.6 is straightforward. A logic 1 in the truth table corresponds to an X in the figure and a logic 0 in the table corresponds to the absence of an X.

A PROM with n address lines (serving as n input variable lines) and m data lines (serving as m output variable lines) can implement any m functions of the same n variables. Unlike a PLA, a PROM has no restrictions due to a limited number of product lines. The PROM contains an n input, 2^n output decoder that generates 2^n internal address lines that serve as product lines. Since the decoder grows exponentially in size with n, the cost of a PROM also increases rapidly with n. The justification for a PLA is to reduce the cost of the PROM decoder by providing fewer product terms since many practical functions require significantly fewer than 2^n product terms. As a result, some n variable functions will not fit onto a PLA chip with n input variables, whereas any n variable function will fit onto a PROM with n address lines.

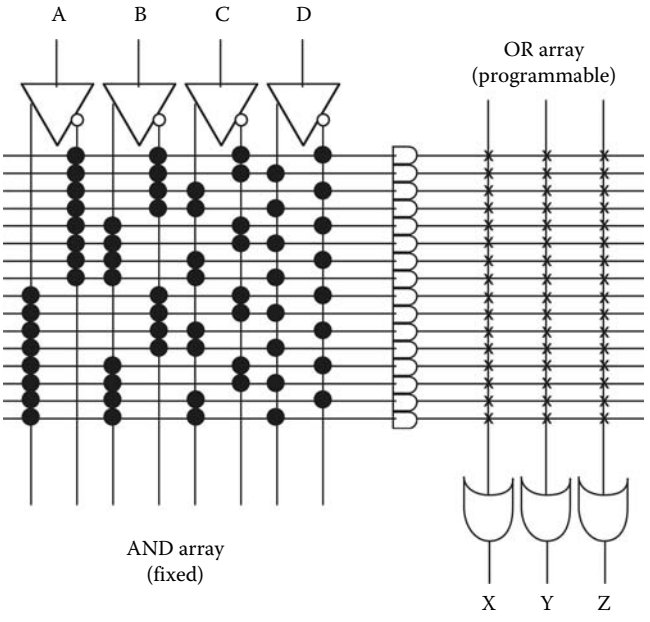


FIGURE 9.5 Conceptual diagram of a PROM.

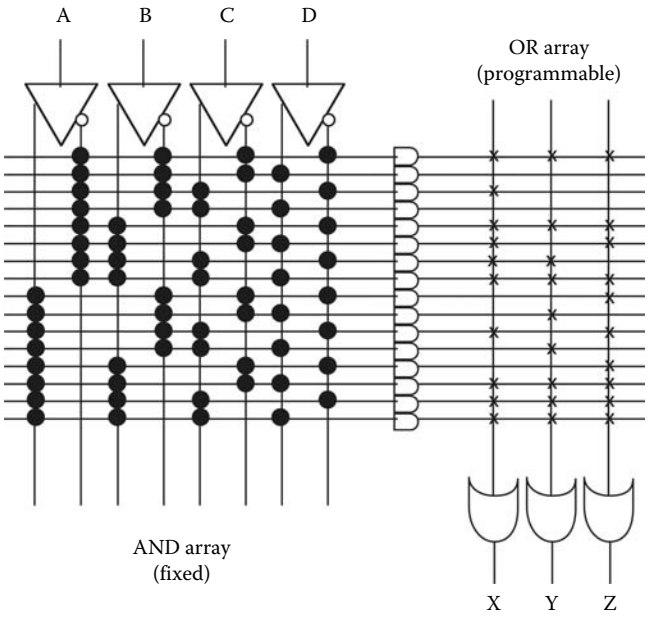


FIGURE 9.6 An example of a programmed PROM.

TABLE 9.2 Truth Table for the Logic Functions Implemented in the PROM

ABCD	XYZ	ABCD	XYZ
0000	111	1000	001
0001	000	1001	010
0010	100	1010	101
0011	000	1011	010
0100	111	1100	001
0101	101	1101	111
0110	110	1110	111
0111	111	1111	111

9.1.5 Programmable Array Logic

PAL is the most popular form of PLD today. Lower price, higher gate densities, and ease of programming all tend to make PAL more popular than PLA. On the negative side, the range of functions that can fit onto a chip with the same number of inputs, outputs, and product lines is less for a PAL than for a PLA.

Figure 9.7 is the basic architecture of a PAL. The PAL architecture is a special case of the PLA architecture in which the OR array is fixed. The filled circles in the OR array indicate permanent connections. Only the AND array is programmable. Compare this PAL architecture with the PLA architecture in Figure 9.3. Since the OR array is not programmable, it is immediately evident that fewer functions will fit onto the PAL. In the PLA, the product terms can be divided among the three

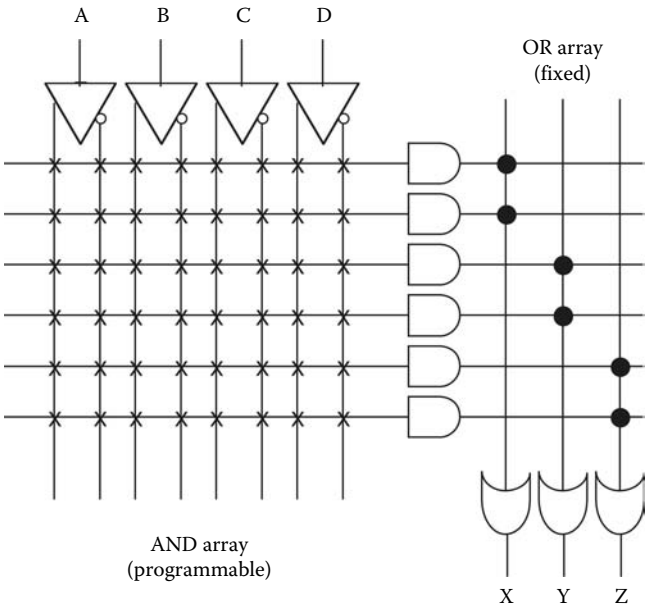


FIGURE 9.7 Basic architecture of a PAL.

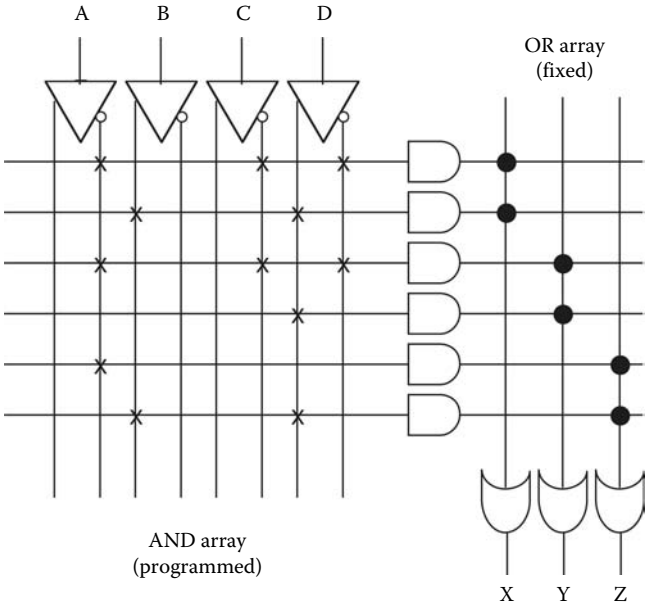
outputs in any way desired and product terms that are used in more than one output can share the same product line. In the PAL, each output is limited to a fixed number of product terms. In Figure 9.7, all outputs are limited to two product terms. In addition, if two output functions both require the same product term in a PAL, two different product lines must be used.

Consider the three functions implemented on the PLA in Figure 9.4. Since the three functions require a total of nine product terms, they will not fit onto the PAL of Figure 9.7. However, any function that would fit onto this PAL would obviously fit onto the PLA since the OR array in the PLA can be programmed to be identical to the OR array of the PAL. Figure 9.8 is an example of three functions that fit onto this PAL. Note that we must use two different product lines to provide the same product term ($\overline{A} \cdot \overline{C} \cdot \overline{D}$) to outputs X and Y.

In order to describe the range of applications for a PAL, we must know the number of inputs, n, the number of outputs, m, and the number of product lines that are permanently connected to each output OR gate. The PAL of Figure 9.7 has four inputs, $n = 4$, three outputs, $m = 3$, and has two product lines connected to each OR gate. This PAL is described as a 2-2-2 PAL with four-input variables. Many PALs have the same number of product terms permanently connected to each output. In this case, the three parameters n, m, and p completely describe the size of the PAL. For PALs, the parameter p usually represents the number of product terms per output instead of the total number of product terms, as was the case for PLAs.

The minimization algorithm for multiple output PALs is significantly less complex than the minimization algorithm for a PLA because gate sharing is eliminated as a possibility by the fact that the OR array is not programmable. This means that each output function can be minimized independently. Minimizing a single output function is much less complex than minimizing a set of output functions where gate sharing must be considered.

Overall, the higher density, less complex minimization algorithms, and lower cost of PALs tend to offset the additional functional capabilities of PLAs.



$X = \overline{A} \cdot \overline{C} \cdot \overline{D} + B \cdot D$

$Y = \overline{A} \cdot \overline{C} \cdot \overline{D} + D$

$Z = \overline{A} + B \cdot D$

FIGURE 9.8 An example of a programmed PAL.

9.1.6 Classification of Combinational Logic PLD Devices

The programmability of the AND and OR array provide a convenient means to classify CL PLD types. The classification of Table 9.3 illustrates comparative features of CL PLD devices. Even though PLAs have the most general structure (i.e., both the AND and OR arrays are programmable), the number of functions that fit onto the chips is limited by the number of product terms per output. ROMs and PROMs have fixed AND arrays, but all possible product terms are provided. Therefore, PROMs and ROMs are the most general devices from a functional viewpoint. Applications are only limited by the size and cost of available devices. PALs are the most restricted devices from both the structural and functional viewpoints. Nevertheless, the lower cost relative to PROMS and PLAs, higher gate densities relative to PLAs, and wider variety of available chip types have contributed to a rapid rise in popularity of PAL devices. For this reason, we will concentrate on PAL devices in the rest of this section.

TABLE 9.3 Classification of Combinational PLD Devices

AND Array	OR Array	Device	Typical Number of Product Terms per Output Gate
Fixed	Mask programmable	ROM	2^n
Fixed	Field programmable	PROM, EPROM, EEPROM	2^n
Field programmable	Fixed	PAL	16
Field programmable	Field programmable	PLA	50–150

9.1.7 Designing with Combinational Logic PAL Devices

Determining if a function will fit onto a PAL device is a complex procedure. To demonstrate the difficulty, we will examine the types of functions that can fit onto a PAL16L8 chip. Figure 9.9 shows that the PAL16L8 chip has eight-output pins and 16 pairs of vertical input lines to the AND array. The "L" in the chip name indicates that the eight outputs are all active low CL outputs. The most important information not provided by the device name is the number of product terms per output, which is seven for this device. An additional product term provides an output enable for each output pin, so there are eight product lines per output pin (a total of 64 product lines). There are 10 primary input pins (pin numbers 1–9 and 11). In terms of our definitions, it would seem that $n = 10$, $m = 8$, and $p = 7$. As we will demonstrate, this simplistic analysis significantly understates the capacity of this chip.

A simplistic analysis would say that the PAL16L8 chip could implement any eight functions of the 10 input variables as long as each function requires no more than seven product terms. As far as it goes, this statement is correct. However, it significantly understates the capacity of the chip because it does not take into account the fact that six of the output pins are internally connected as inputs to the AND array (pins 13–18). This is the source of the additional six inputs to the AND array. These internal feedback connections significantly expand the capacity of the chip.

Consider the following logic function.

$$X = A\overline{B}C + B\overline{C}D + \overline{A}E + D\overline{E}F + \overline{A}C + \overline{D} + \overline{F}GH + F\overline{G}I + BE\overline{H} + \overline{C}H + \overline{I}J + \overline{B}E\overline{J} + \overline{D}H$$

It appears that this logic function will not fit onto the PAL16L8 chip because it requires 13 product terms and each output has only seven product lines. However, if not all chip outputs are needed for the application, we can use one of the feedback lines to fit this function onto the chip.

We first partition the function X as follows:

$$X = A\overline{B}C + B\overline{C}D + \overline{A}E + D\overline{E}F + \overline{A}C + \overline{D} + Y$$
$$Y = \overline{F}GH + F\overline{G}I + BE\overline{H} + \overline{C}H + \overline{I}J + \overline{B}E\overline{J} + \overline{D}H$$

Since Y has only seven product terms, we will map function Y to the macrocell connected to pin 18 and use the feedback line from pin 18 to connect Y to a pair of vertical lines in the AND array. Function Y is now available to all other cells as an input variable. Since function X also has seven product terms, we will map X to the macrocell connected to pin 17. One of the product terms in X is the single variable Y. Figure 9.10 shows the final implementation. To obtain the needed product terms for output X, we used two macrocells in the array. As a result, pin 18 is no longer available as an output.

Two practical matters need to be considered. First, some signals must now pass through the AND array twice as they proceed from an input to an output due to the feedback path. Therefore, the delay of the chip is now twice what it was when the feedback path was not utilized. Second, the buffer inverts the outputs; therefore, we actually obtain X on pin 17. If X is specified to be active low, then the output on pin 17 is exactly what we need. If the output X is specified to be active high, then an inverter is required. Since PALs are available with both active low and active high outputs, a designer should select an appropriate PAL to eliminate the need for inverters.

Another feature that adds to the flexibility of the chip is that pins 13–18 can be used either as inputs or as outputs. For example, the enable for the output buffer at pin 15 can be permanently disabled. Since pin 15 is connected directly into the AND array, it is no different from any other input, say pin 2. Of course, by permanently disabling the buffer at pin 15, the OR array connected to the buffer is also disconnected. In order to use pin 15 as an input, we must give up the use of the macrocell connected to pin 15. However, dual use pins and feedback lines dramatically extend the range of applications for the chip. For example, suppose we need only one output and select pin 19 for that use. Then, pins 13–18 are available either as inputs or as feedback lines. We can therefore fit a wide range of functions onto the chip with varying numbers of inputs and product terms.

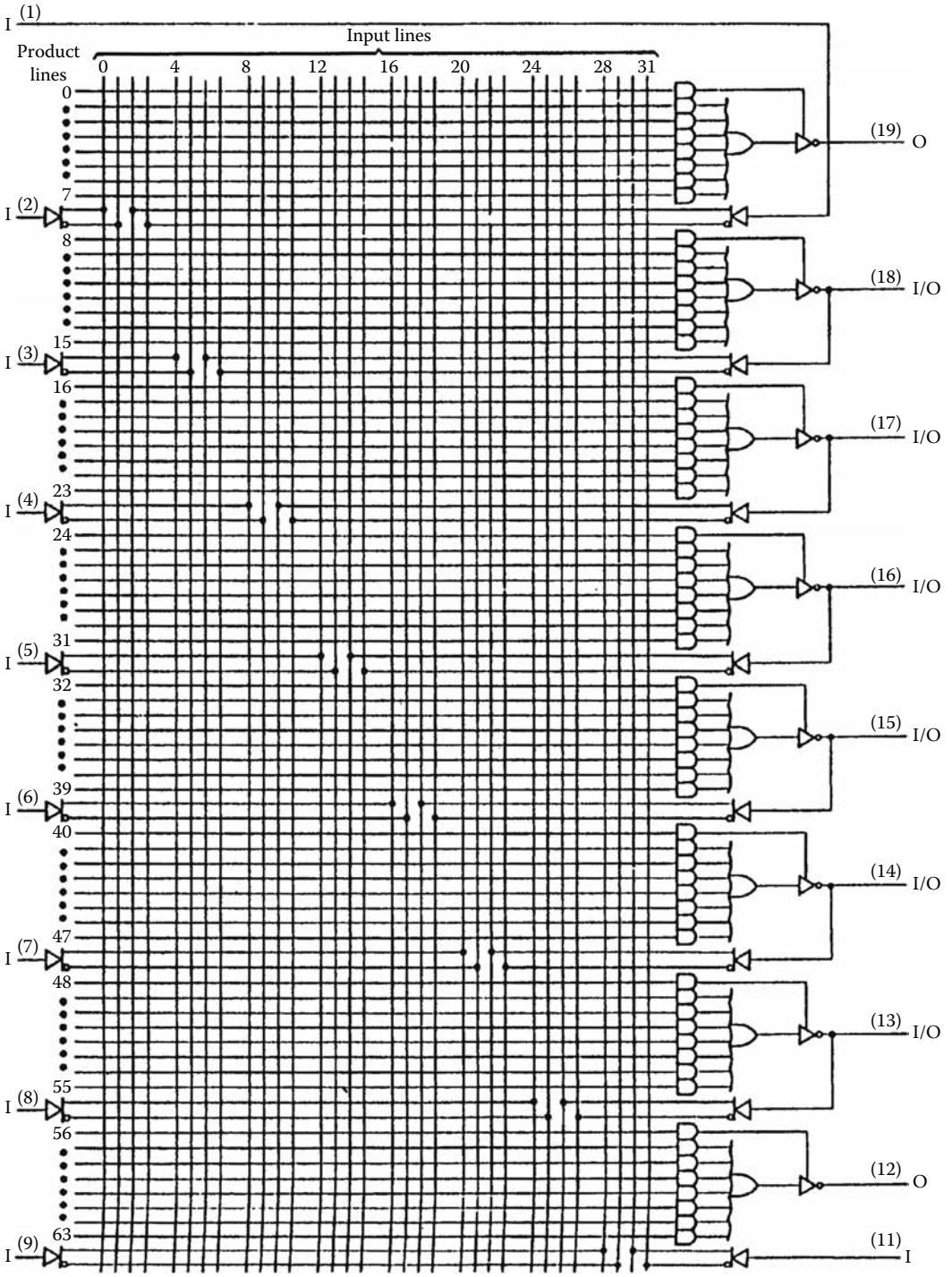


FIGURE 9.9 Logic diagram of PAL 16L8 chip. (Courtesy of Texas Instruments, Dallas.)

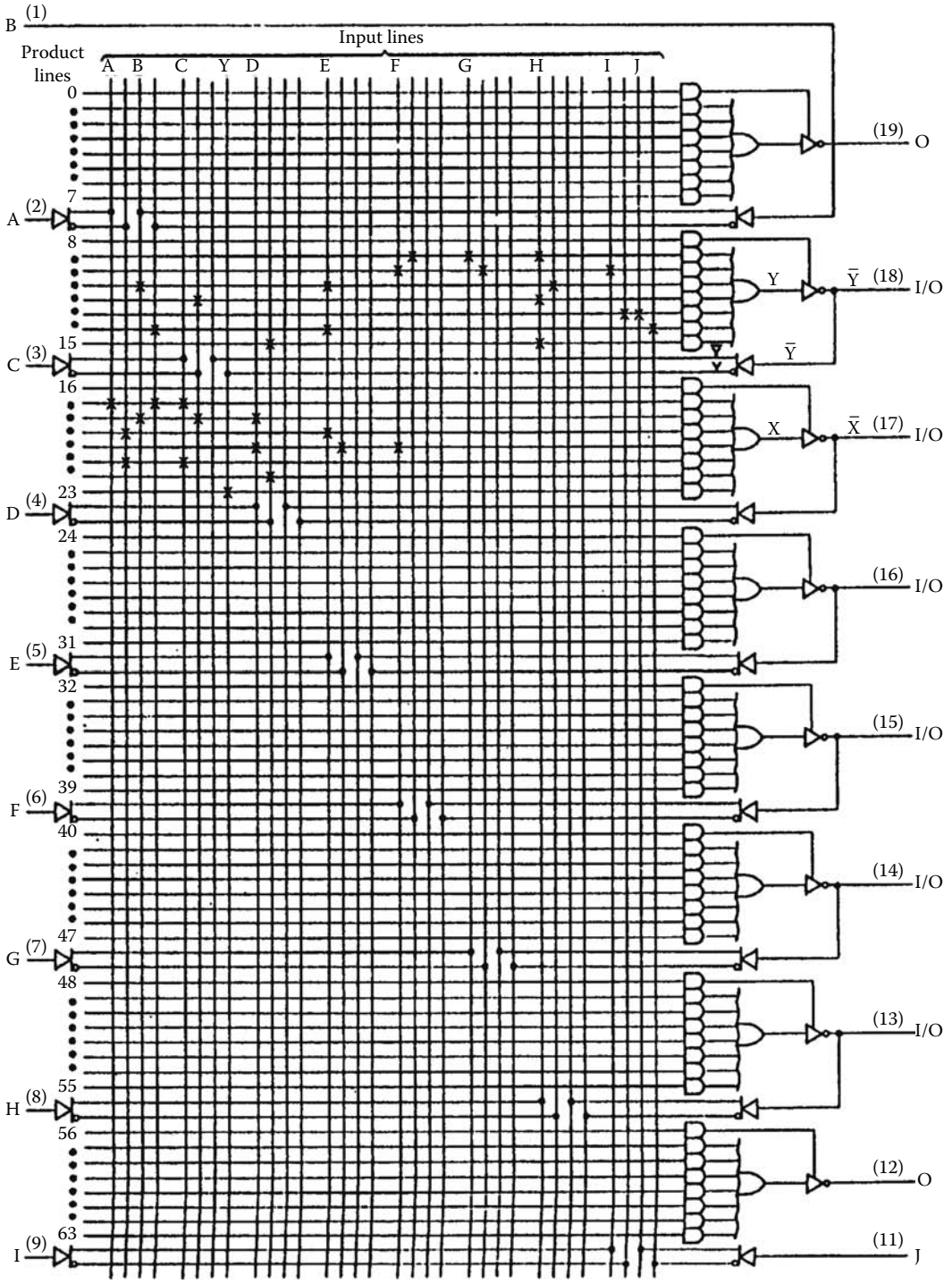


FIGURE 9.10 Implementation of function with 13 product terms on PAL16L8 chip.

Assuming that we need only the one output on pin 19, then we could have up to 16 inputs (pins 19, 11, 13–18). We could, of course, use only the seven product lines in the macrocell connected to pin 19 to implement our function. We, therefore, conclude that the PAL16L8 chip can implement any single function of 16 variables that requires no more than seven product terms.

If we need more product terms but not as many input variables, then we can connect pin 11 to pin 12. This connects the output on pin 12 back into the AND array the same as any other input variable. The output on pin 19 can pick up the seven product terms on pin 12 as an input. This takes up one of the product lines for pin 19, but six product lines remain. Therefore, the single output on pin 19 can now have up to 13 product terms. However, pin 11 is no longer available as an input. Therefore, we must conclude that the PAL16L8 can implement any single function of 15 variables that requires no more than 13 product terms.

If we want to maximize the number of product terms for a single output at pin 19, then we can use pins 13–18 and pin 11 as feedback lines. Each feedback line contributes seven product terms. The AND array for pin 19 can pick up all 49 product terms by using one product line to pick up each feedback variable. All product lines are now busy. The OR gate at pin 19 then sums all 49 product terms. The only pins that are now available for inputs are 1–9. We therefore conclude that the PAL16L8 can implement any single function of nine variables that requires 49 or fewer product terms.

Clearly, there are many combinations of implementations with a variety of values for the number of inputs, the number of outputs, and the number of product terms per output. Table 9.4 shows the full range of possible implementations. For example, from the tables we note that the PAL16L8 can implement any three functions of 10 variables in which the product terms are distributed among the three outputs in any of the following ways: 7–7–37, 7–13–31, 7–19–25, 13–13–25, or 13–19–19. The notation 7–7–37 means that two of the outputs require at most seven product terms and that the third output requires at most 37 product terms. To accomplish these results, five of the output pins must be devoted to feedback lines.

Any implementation that uses a feedback line will have a time delay equal to twice that of a single macrocell. In the delay column of Table 9.4, symbol T_A represents the delay that a signal experiences while passing through the AND–OR array one time. In implementations that do not use feedback lines, signals experience a maximum delay of T_A. For implementations that use one or more feedback lines, the delay is $2T_A$ because some input signals propagate through the AND–OR array twice before reaching an output pin. However, none of the implementations in the table requires more than twice the normal time delay for a single macrocell.

Although the tables cover broad generalizations for classes of functions that will fit onto the PAL16L8 chip, there are certain special types of more complex functions that will fit. For example, suppose that input variables A, B, C, D, E, F, G, H, and I occupy pins 1–9. Further suppose that we implemented functions S, T, V, W, X, Y, and Z using the macrocells connected to pins 12, 13, 14, 15, 16, 17, and 18, respectively. Further suppose that we connect pin 12 to pin 11 so that all of these functions are connected to a pair of vertical input lines in the AND array. Thus, all of these functions are now available to the single output P at pin 19. This approach allows many very complex logic functions to fit onto the chip.

Example 9.1

Let each of the functions S, T, V, W, X, Y, and Z be a sum of products expression involving the nine input variables with at most seven product terms. For example, S might be

$$S = A\bar{B}CD\bar{E}F\bar{G}HI + \bar{A}B\bar{C}DE\bar{F}GH\bar{I} + \bar{B}CD\bar{E}F\bar{G}HI + \bar{F}GH\bar{I} + \bar{A}H\bar{I} + \bar{D}EF\bar{G}HI + ABCDEFGHI$$

Variables T, V, W, X, Y, and Z could be of similar complexity.

TABLE 9.4 Range of Implementations of PAL16L8 Chip

m	*n*	Number of Product Terms per Output	Delay
1	16	7	T_A
1	15	13	$2T_A$
1	14	19	$2T_A$
1	13	25	$2T_A$
1	12	31	$2T_A$
1	11	37	$2T_A$
1	10	43	$2T_A$
1	9	49	$2T_A$
2	16	7–7	T_A
2	15	7–13	$2T_A$
2	14	7–19, 13–13	$2T_A$
2	13	7–25, 13–19	$2T_A$
2	12	7–31, 13–25, 19–19	$2T_A$
2	11	7–37, 13–31, 19–25	$2T_A$
2	10	7–43, 13–37, 19–31, 25–25	$2T_A$
3	15	7–7–7	T_A
3	14	7–7–13	$2T_A$
3	13	7–7–19, 7–13–13	$2T_A$
3	12	7–7–25, 7–13–19, 13–13–13	$2T_A$
3	11	7–7–31, 7–13–25, 7–19–19, 13–13–19	$2T_A$
3	10	7–7–37, 7–13–31, 7–19–25, 13–13–25, 13–19–19	$2T_A$
4	14	7–7–7–7	T_A
4	13	7–7–7–13	$2T_A$
4	12	7–7–7–19, 7–7–13–13	$2T_A$
4	11	7–7–7–25, 7–7–13–19, 7–13–13–13	$2T_A$
4	10	7–7–7–31, 7–7–13–25, 7–7–19–19, 7–13–13–19, 13–13–13–13	$2T_A$
5	13	7–7–7–7–7	T_A
5	12	7–7–7–7–13	$2T_A$
5	11	7–7–7–7–19, 7–7–7–13–13	$2T_A$
6	12	7–7–7–7–7–7	T_A
6	11	7–7–7–7–7–13	$2T_A$
6	10	7–7–7–7–7–19, 7–7–7–7–13–13	$2T_A$
7	11	7–7–7–7–7–7–7	T_A
7	10	7–7–7–7–7–7–13	$2T_A$
8	10	7–7–7–7–7–7–7–7	T_A

Then, output P might be

$$P = A\overline{B}CD\overline{E}FG\overline{H}IS\overline{T}VW\overline{X}Y\overline{Z} + B\overline{C}DEF\overline{G}\overline{S}T\overline{V}WXY\overline{Z} + \cdots$$

where P has at most seven such product terms.

The delay of this implementation is still twice the delay of one basic macrocell.

Example 9.2

This example illustrates embedded factors. Each equation has at most seven product terms involving the listed variables.

$$S = f(A - I) = \overline{A}BC\overline{D}E\overline{F}GH\overline{I} + A\overline{B}\overline{C}DEF\overline{G}HI + \cdots$$

$$T = f(S, A - I) = AB\overline{C}D\overline{E}FG\overline{H}IS + \overline{A}BCD\overline{E}\overline{F}GHI\overline{S} + \cdots$$

$$V = f(S, T, A - I) = \overline{C}DE\overline{F}GH\overline{I}\overline{S}T + BC\overline{D}HIS\overline{T} + \cdots$$

$$W = f(S, T, V, A - I) = \overline{A}BCD\overline{E}F\overline{G}HI\overline{S}T\overline{V} + \overline{D}E\overline{F}HIS\overline{T}V + \cdots$$

$$X = f(S, T, V, W, A - I) = \overline{A}\overline{B}CDE\overline{F}GH\overline{I}ST\overline{V}\overline{W} + \overline{E}T\overline{V}W + \cdots$$

$$Y = f(S, T, V, W, X, A - I) = ABC\overline{D}EFG\overline{H}IS\overline{T}VW\overline{X} + \overline{D}H\overline{I}S\overline{T}V\overline{W}X + \cdots$$

$$Z = f(S, V, T, W, X, Y, A - I) = A\overline{B}CD\overline{E}FGH\overline{I}S\overline{T}V\overline{W}X\overline{Y} + \overline{F}H\overline{I}WXY + \cdots$$

$$P = f(S, V, T, W, X, Y, Z, A - I) = A\overline{B}CD\overline{E}FG\overline{H}IS\overline{T}VW\overline{X}Y\overline{Z} + B\overline{C}DEF\overline{G}ST\overline{V}WXY\overline{Z} + \cdots$$

The delay of this implementation is eight times the delay of a single macrocell because an input signal change might have to propagate serially through all of the macrocells on its way to the output at pin 19.

These examples demonstrate that very complex functions can fit onto the chip. Determining the optimum way to factor the equations is a very complex issue. Chip manufacturers and third-party vendors provide software packages that aid in the fitting process.

9.1.8 Designing with Sequential PAL Devices

The concept of registered outputs extends the range of PAL devices to include sequential circuits. Figure 9.11 is the logic diagram for the PAL16R4 chip. Again, this chip has 16 pairs of inputs to the AND array. The R4 part of the designation means that the chip has four outputs connected directly to D type flip-flops, i.e., the outputs are registered. Let us add another parameter, k, to designate the number of flip-flops on the chip. An examination of Figure 9.11 indicates that the PAL16R4 also has four combinational outputs with feedback connections to the AND array. These pins are I/O pins because they can also be used as inputs if the OR output to the pins is permanently disabled. All outputs are active low. The chip has eight-input pins. Using our parameter system, the PAL16R4 apparently has $n = 8$, $m = 4$, $k = 4$, $p = 7$ for combinational pin outputs and $p = 8$ for registered pin outputs. However, as for the PAL16L8, these numbers significantly understate the capabilities of this chip.

Since the four registered outputs are also connected back into the AND array, this chip can implement a sequential circuit with the registered outputs serving as state variables. Therefore, this chip can implement any eight-input, four-output, sequential circuit that needs no more than four-state variables (16 states) and no more than seven product terms for each output or eight product terms for each state variable. Separate pins provide an enable for the state variables (pin 11) and a clock for the flip-flops (pin 1). Thus, the state variables are also available at output pins.

By an analysis similar to that used in the previous section, we can utilize the feedback lines to significantly expand the types of circuits that will fit onto the PAL16R4 chip. Table 9.5 shows the range of basic possibilities.

For example, the table indicates that the PAL16R4 chip can implement any single output, eight-input, sequential circuit that requires no more than four-state variables (16 states) and in which the available product terms may be divided among the outputs and state variables in seven different distributions. The notation (7)-(8–8–8–26) means that the single output can have up to seven product terms, that one state variable can have up to 26 product terms, and that the other three state variables can have up to eight product terms each.

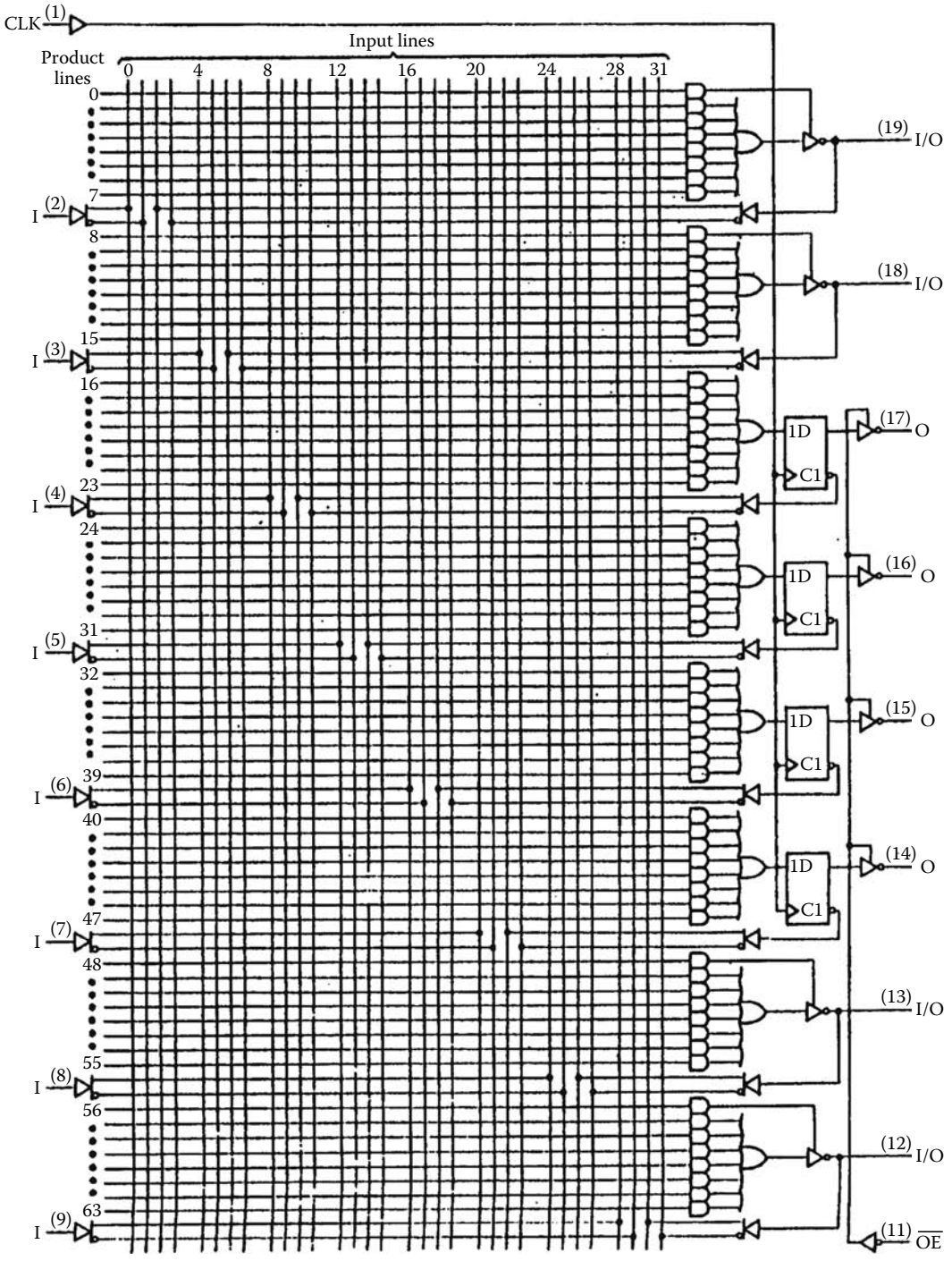


FIGURE 9.11 Logic diagram of PAL16R4 chip.

TABLE 9.5 Range of Basic Implementations of PAL16R4 Chip

M	n	K	Number of Product Terms per (Combinational Output)–(State Variable)	Delay
1	11	4	(7)−(8–8–8–8)	T_A
1	10	4	(7)−(8–8–8–14), (13)−(8–8–8–8)	$2T_A$
1	9	4	(7)−(8–8–8–20), (7)−(8–8–14–14), (13)−(8–8–8–14), (19)−(8–8–8–8)	$2T_A$
1	8	4	(7)−(8–8–8–26), (7)−(8–8–14–20), (7)−(8–14–14–14), (13)−(8–8–8–20), (13)−(8–8–14–14), (19)−(8–8–8–14), (25)−(8–8–8–8)	$2T_A$
2	10	4	(7–7)−(8–8–8–8)	T_A
2	9	4	(7–7)−(8–8–8–14), (7–13)−(8–8–8–8)	$2T_A$
2	8	4	(7–7)−(8–8–8–20), (7–7)−(8–8–14–14), (7–13)−(8–8–8–14), (7–19)−(8–8–8–8), (13–13)−(8–8–8–8)	$2T_A$
3	9	4	(7–7–7)−(8–8–8–8)	T_A
3	8	4	(7–7–7)−(8–8–8–14), (7–7–13)−(8–8–8–8)	$2T_A$
4	8	4	(7–7–7–7)−(8–8–8–8)	T_A

9.1.9 Designing with PALs Having Programmable Macrocell Outputs

The PAL16R4 chip has limited application potential because the outputs from pins 14–17 ''must'' be registered. Most new chips allow a user to decide whether to have registered or combinational outputs at each pin and also allow the user to select either active high or active low outputs.

The PAL22V10 chip (see architecture in Figure 9.12) demonstrates this additional flexibility. Each of 10 macrocells contains a normal PAL AND array and an I/O architecture control block. Each PAL AND array provides a differing number of product terms permanently connected as inputs to an OR gate and an additional product term that enables an output buffer. The number of product terms per output is printed near the OR gate symbol (8, 10, 12, 14, 16, 16, 14, 12, 10, 8). Figure 9.13 shows that this chip is similar in form to the PAL chips described earlier in this chapter. There are 22 vertical pairs of input lines to the AND array. Of these pairs, 11 are connected directly to input pins labeled $I_1 - I_{11}$ (pins 2–11, 13). Ten pairs are feedback lines from the architecture control blocks of the 10 macrocells. Each macrocell is associated with a bidirectional pin (pins 14–23) that can be used either as an input pin, an output pin, or a bidirectional bus pin. If used as a bidirectional bus pin, the designer must control the O/E using a

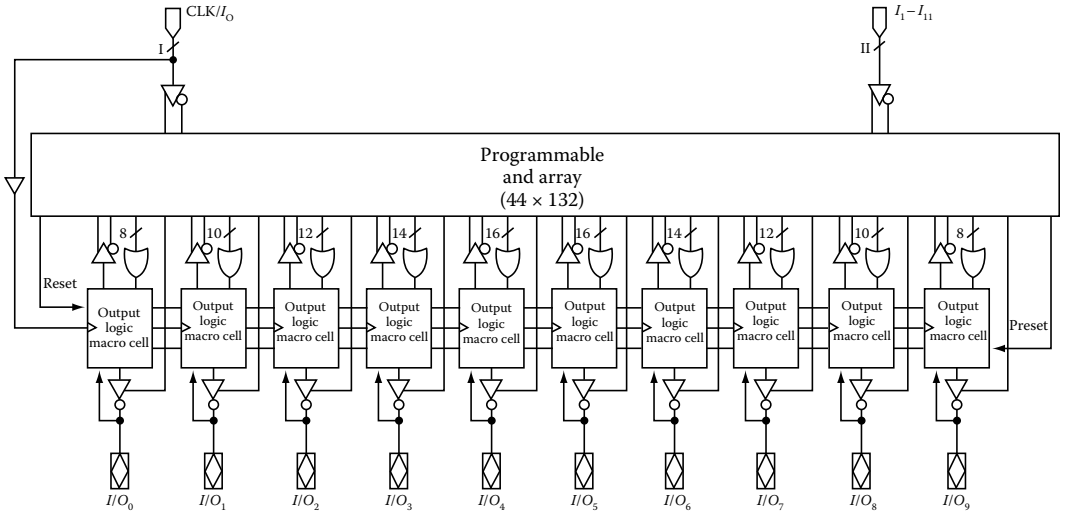


FIGURE 9.12 Architecture of the PAL22V10 chip. (Courtsey of Advanced Micro Devices, Inc., Sunnyvale.)

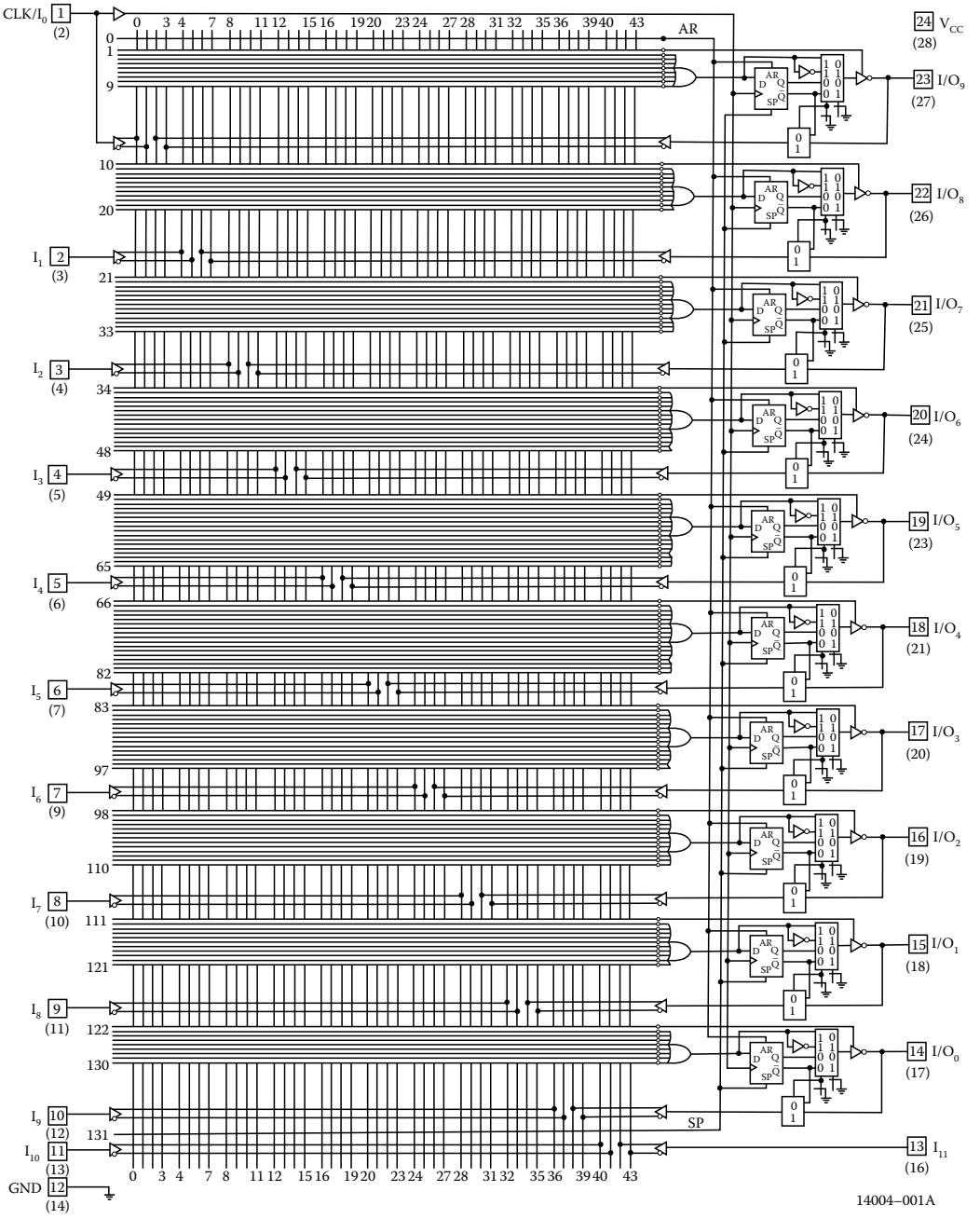


FIGURE 9.13 Complete circuit diagram for the PAL22V10 chip. (Courtesy of Advanced Micro Devices, Inc., Sunnyvale.)

product term from the AND array. The 22nd pair, labeled CLK/I_0, is connected to pin 1. If the chip is being used to implement a purely combinational circuit, pin 1 can be used as an additional input variable. If a sequential circuit is being implemented, pin 1 must be used as the clock signal for the flip-flops.

The architecture control block in each macrocell provides designer control over the signal that is connected to the bidirectional pin and feedback line associated with that macrocell. Figure 9.14 shows

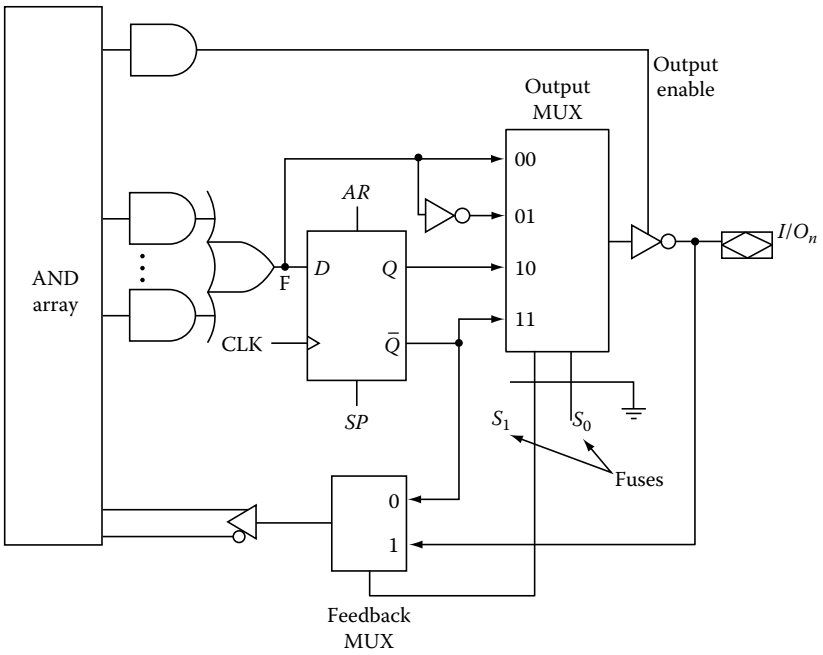


FIGURE 9.14 Macrocell architecture of PAL22V210 chip. (Courtesy of Advanced Micro Devices, Inc., Sunnyvale.)

that the architecture control block contains a D-flip-flop, an inverter, two MUXs with programmable select lines, and an output buffer with an enable. The Output MUX selects either the direct output of the combinational AND array (either active high or active low) or the data value stored in the D-flip-flop (either active high or active low). If the O/E is active, the buffer steers the signal selected by the Output MUX to the pin. An inactive enable causes the buffer to enter the high impedance state, which effectively disconnects the buffer from the pin. The pin can then be an input or can be connected to an external bus.

The feedback signal selected by the Feedback MUX is either the pin signal or the data value stored in the flip-flop (low active). Therefore, the feedback line can be used to expand the number of product terms, to provide a state variable for a sequential circuit, to provide an additional input for the chip, or to implement a bidirectional pin.

Figures 9.13 and 9.14 show that the common clock input to all flip-flops comes directly from pin 1, that a single product (SP) term provides a common synchronous preset for all flip-flops, and that another single product line (AR) provides a common asynchronous reset for all flip-flops. The asynchronous reset occurs when the product line is active independent of clock state. The synchronous preset occurs only on the active edge of the clock when the preset product line is active.

The two programmable MUXs in the architecture control block significantly increase the flexibility of the chip compared to either the PAL16L8 or the PAL16R4. Table 9.6 shows several combinations of switch settings along with typical applications for each setting.

The PAL22V10 is much more versatile than either the PAL16L8 or the PAL16R4. Since pins 14–23 can be used as inputs, combinational outputs, or registered outputs, the following inequalities describe the possibilities.

If the chip is used to implement a CL function, the constraints are

$$n \leq 22, \quad m \leq 10$$

$$(n + m) \leq 22$$

TABLE 9.6 Applications for Combinations of Switch Settings for the PAL22V10 Chip

Name	Output Connection	Feedback Connection	Application
INP	None	Pin	Use pin as input only
COCF	Combinational	Combinational	Combinational output and/or combinational feedback
COIF	Combinational	Pin (input)	Bidirectional pin implementing a combinational output
RORF	Register	Register	Typical state machine controller
ROIF	Register	Pin (input)	Bidirectional pin with registered output. Bus applications

For sequential circuits, the constraints are

$$n \leq 21, \quad m \leq 10, \quad k \leq 10$$

$$(m + k) \leq 10, \quad (n + m + k) \leq 21$$

because the clock signal for the flip-flops uses one of the input pins (pin 1).

Table 9.7 shows representative sizes for circuits that will fit onto the PAL22V10 chip.

TABLE 9.7 Representative Circuit Sizes That Will Fit onto a PAL22V20 Chip

Representative CL Circuits

M	n	Number of Product Terms per Output	Delay
1	21	16	T_A
1	20	31	$2T_A$
1	12	111	$2T_A$
2	20	16–16	$2T_A$
2	12	16–96, 29–83, 40–72, 49–63, 56–56	$2T_A$
3	19	14–16–16	T_A
3	12	16–16–81	$2T_A$
3	12	37–38–38	$2T_A$
5	12	15–19–23–27–31	$2T_A$
5	17	12–14–14–16–16	T_A
10	12	8–10–12–14–16–16–14–12–10–8	T_A

Representative Sequential Circuits

M	K	n	Number of Product Terms per (Output)–(State Variable)	
1	3	17	(16)−(14–16–16)	T_A
1	3	13	(25)−(25–25–25)	$2T_A$
1	3	11	(88)−(8–8–10)	$2T_A$
1	3	11	(31)−(27–28–28)	$2T_A$
2	3	16	(16–16)−(12–14–14)	T_A
2	3	13	(16–16)−(19–23–23)	$2T_A$
2	3	11	(44–45)−(8–8–10)	$2T_A$
2	3	11	(23–23)−(23–23–23)	$2T_A$
4	4	13	(14–14–16–16)−(10–10–12–12)	T_A
4	4	11	(12–12–29–29)−(8–8–10–10)	$2T_A$
5	5	11	(12–14–14–16–16)−(8–8–10–10–12)	T_A
2	8	11	(16–16)−(8–8–10–10–12–12–14–14)	T_A

9.1.10 FPGA Technologies

Due to the relatively high complexity of FPGAs, almost all FPGAs use CMOS process technology because of its high density and low power characteristics. Currently, there are two popular FPGA programming technologies, static RAM (SRAM), and anti-fuse.

The anti-fuse device gets its name from the fact that its electrical properties are the dual of the electrical properties of a fuse. The anti-fuse is a pair of conducting plates separated by a dielectric insulator, similar to a small capacitor. By contrast, the fuse is a pair of terminals separated by a thin conducting wire. A fuse is programmed by passing a high current through the thin wire causing the wire to heat up and melt, producing an open circuit where a short circuit previously existed. Fusible link technology is used in many PLA and EPROM devices. By contrast, the anti-fuse is programmed by applying a high voltage across the dielectric insulator that permanently breaks down the dielectric insulator, producing a short circuit where an open-circuit previously existed. Both fusible-link and anti-fuse devices are nonvolatile which makes them particularly well-suited for use in extreme environments, such as space and other high radiation environments. Anti-fuse technology also provides higher speed operation than other technologies.

SRAM chips are volatile (i.e., they lose their program when power is removed) and have lower density and slower speed than anti-fuse chips. On the positive side, SRAM chips are lower cost, re-programmable and, therefore, dynamically reconfigurable. In the current marketplace, SRAM chips have captured most of the popular commercial market.

9.1.11 FPGA Architectures

Figure 9.15 shows a high-level layout of an FPGA chip. Each chip contains a two-dimensional array of identical configurable logic blocks (CLBs). The FPGA in Figure 9.15 has 64 CLBs arranged in an 8×8 array. The user can program the CLBs to implement specific combinational or sequential logic functions. A programmable interconnect structure is permanently placed in the space between CLBs. The user programs switches that make desired connections between his programmed CLBs either by setting SRAM bits or by permanently closing the anti-fuses. Programmable I/O blocks are located around the perimeter of the chip that allows the user to connect signals to pins on the chip.

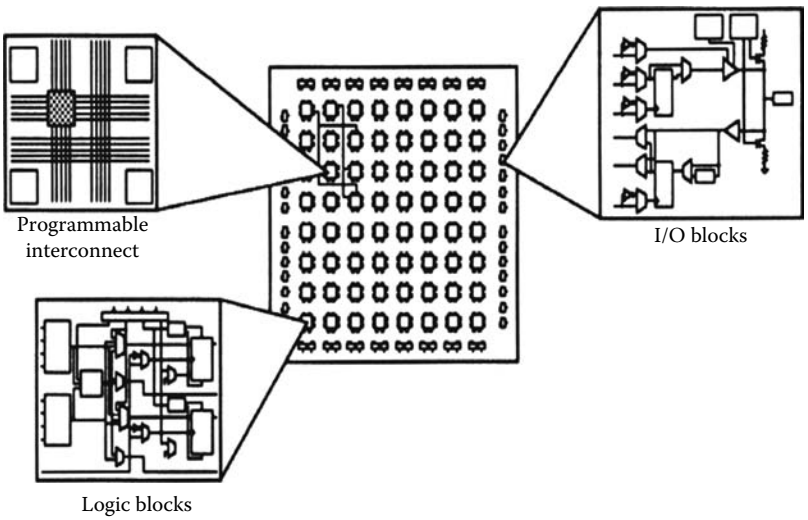


FIGURE 9.15 FPGA architecture. (Courtesy of Xilinx, San Jose.)

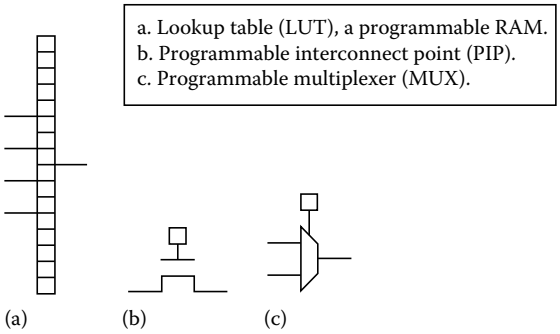


FIGURE 9.16 Programmable FPGA elements. (Courtesy of Xilinx, San Jose.)

For convenience, in this paragraph we describe FPGA elements using SRAM terminology. Anti-fuse devices have similar components. FPGAs have three basic programmable elements, illustrated in Figure 9.16. The lookup table (LUT) is a programmable RAM. The LUT shown in Figure 9.16a is a 16×1 RAM. It has four address inputs and one data output. It is programmed by storing either a logic 1 or a logic 0 in each RAM location. The value at a specific location is read out (looked up) by applying the address at the RAM inputs. A programmable interconnect point (PIP) is simply a CMOS pass transistor with a programmable SRAM bit controlling the gate signal. If a connection between the two transistor terminals for the PIP in Figure 9.16b is desired, a logic 1 is stored in the SRAM control bit. If no connection is desired, a logic 0 is stored in the SRAM control bit. SRAM bits also control the address lines of programmable MUXs. The MUX in Figure 9.16c has two input lines and therefore can be controlled by one SRAM bit. Programming the SRAM control bit to be logic 0 connects the upper MUX input to the MUX output. Programming the SRAM control bit to be logic 1 connects the lower MUX input to the MUX output.

To illustrate the principles, consider the minimal CLB shown in Figure 9.17. This CLB has three input signals (A, B, and C) and one output signal (X). The CLB has one 8×1 LUT and one D flip-flop with a reset input (R). It has three 2×1 programmable MUXs with SRAM control bits labeled M1, M2, and M3, respectively. The MUX controlled by M3 connects either the LUT output (F) or the bit stored in the flip-flop (Q) to the CLB output (X). If $M3 = 0$, the D flip-flop is bypassed and the CLB will implement the CL function stored in the LUT. If $M3 = 1$, the CLB will implement a sequential function. The MUX controlled by SRAM bit M2 selects either the LUT output (F) or the input signal (C) as the reset signal for the flip-flop. The MUX controlled by SRAM bit M1 selects either input signal C or the bit stored in the flip-flop (Q) as the third address input (E) to the LUT. Inputs *A* and *B* are permanently connected to two of the LUT address lines.

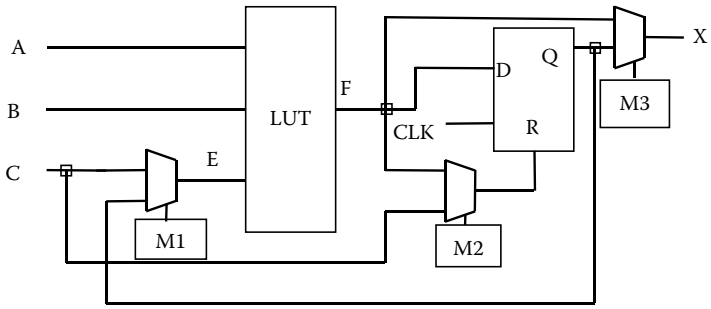


FIGURE 9.17 A minimal CLB.

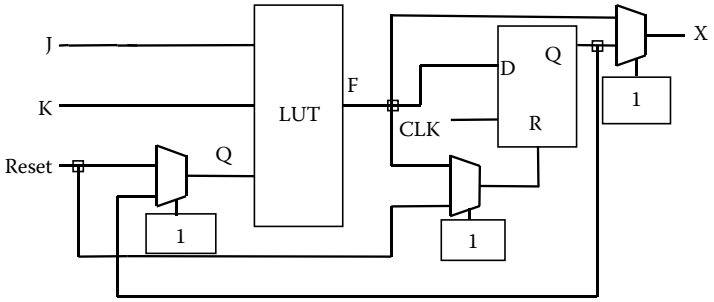


FIGURE 9.18 Minimal CLB programmed to be a JK flip-flop.

TABLE 9.8 Contents of LUT for Programmed CLB

J	K	Q	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

To illustrate how the minimal CLB in Figure 9.17 could be programmed, we will show how to program it to implement a JK flip-flop. M3 will be 1 to select the flip-flop output as the CLB output. M2 will be 1 to select input C as the flip-flop reset signal. M1 will be 1 to select the flip-flop output (Q) as the E input to the LUT. Input A will be designated as the J input and input B will be designated as the K input for the flip-flop. Figure 9.18 shows the programmed CLB with all signal names related to their use in the JK flip-flop. Table 9.8 shows how the LUT must be programmed to implement the function of a JK flip-flop.

Figure 9.19 shows an actual CLB in the XC4010XL chip, a small FPGA chip manufactured by Xilinx. This CLB uses the same small set of elements that we used in the minimal CLB. This CLB contains 2 D-flip-flops, 3 LUTs, and 16 programmable MUXs.

If the user had to directly program each CLB and each PIP in the interconnect structure, the task would be formidable. However, most chip manufacturers provide software packages that allow the user to specify the device function using a variety of high-level abstractions. In the next section, we will discuss this process in more detail.

9.1.12 Design Process

From the previous discussion, it is clear that fitting designs to PLD chips is a complex process. PLD manufacturers and third-party vendors market software packages that help engineers map designs onto chips. Selecting a package appropriate for a particular design environment is a critical decision that will significantly affect the productivity of the design group.

There are basically three types of development system packages: user designed packages, vendor designed packages, and universal packages. Since these programs are very complex and require many

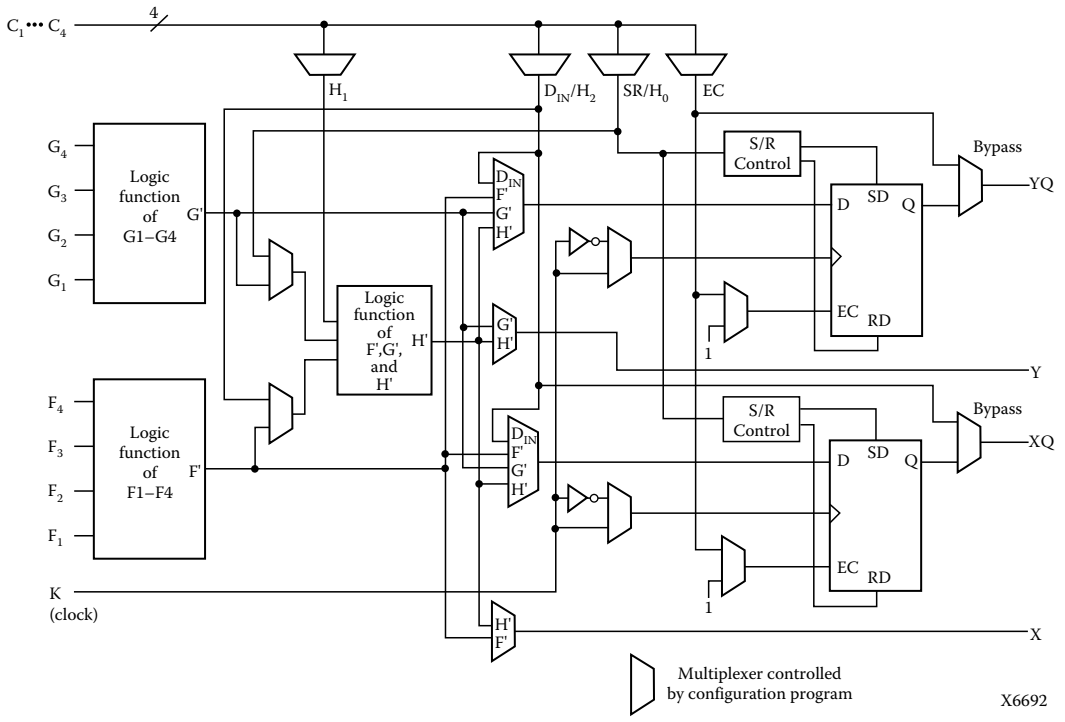


FIGURE 9.19 Actual CLB for the XC4010XL chip. (Courtesy of Xilinx, San Jose.)

years of effort to develop, most design groups will lack the time and resources to develop their own. Many vendors provide design aids that are specific to a particular product line. There is a great danger in becoming dependent upon one vendor's products because new products in this field appear frequently. Clearly, a universal design package that supports a wide variety of product lines is most desirable. A variety of development systems with different features, capabilities, and price is available.

Figure 9.20 shows a flow diagram for the process typically used to design PLAs and FPGAs. ''Design entry'' refers to the use of an editor to create a source file that specifies the functional behavior of the device. High-level simulation verifies correct functional behavior of the device. ''Logic synthesis'' refers to the process of implementing the design using the primitive elements present on a specific chip, such as gates, flip-flops, registers, etc. Most development systems support prelayout simulation at this level to verify that the design still functions correctly. ''System partitioning'' and ''mapping'' refers to the process of grouping blocks of primitive elements into sets that map directly into major chip structures, such as CLBs in FPGAs or AND–OR arrays in PLDs. ''Place and route'' refers to mapping the structures into specific locations on the chip and making connections between them. The software package then performs a timing analysis on the final design to verify that design timing specifications are met. Finally, the chip is configured by generating an output file that can be read by the chip programmer.

9.1.12.1 Design Entry

It is essential for a universal development system to have a variety of design entry modes. Many vendors market the more complex design entry modes as optional features. This section describes some of the more common design entry modes and their value to PLD and FPGA designers.

"Boolean equations" are an important method of design entry. A PLD design system must support Boolean equation entry because the AND–OR arrays on PLD chips directly implement Boolean equations. Almost all PLD designers will use Boolean equation entry extensively. Boolean equation entry is also useful for FPGA designs.

"Truth table" entry allows specification of a CL function by defining the output for each of the 2^n input combinations. This form is particularly valuable if "don't care" entries exist. Truth table entry is most commonly used for functions with a small number of input variables that are not easily described by Boolean equations. Code converters, decoders, and LUTs are examples. A good design tool will support truth table entry.

"Symbolic state machine" entry is crucial for PLD and FPGA designers because both PLDs and FPGAs are often used to implement state machine controllers. Current tools have features described as state machine entry that vary dramatically in form and usefulness. Before selecting a tool, the specifics of the state machine entry format should be carefully investigated. The most useful formats allow symbolic representation of states and specification of state transitions using some form of conditional statement such as "if_then_else," or "case." Relational operators are also useful in this context. The tool should perform automated state assignment and should fit the state variable equations to the target chip.

"State diagrams" using graphics are useful, but not essential. This feature is mainly a convenience, provided that symbolic state machine entry is available.

"Schematic" entry is a widely accepted way to describe logic systems. To be useful, it must be combined with a powerful partitioning and mapping tool that can fit the circuit onto chips. Schematic entry is useful to convert existing gate level designs into PLD or FPGA implementations.

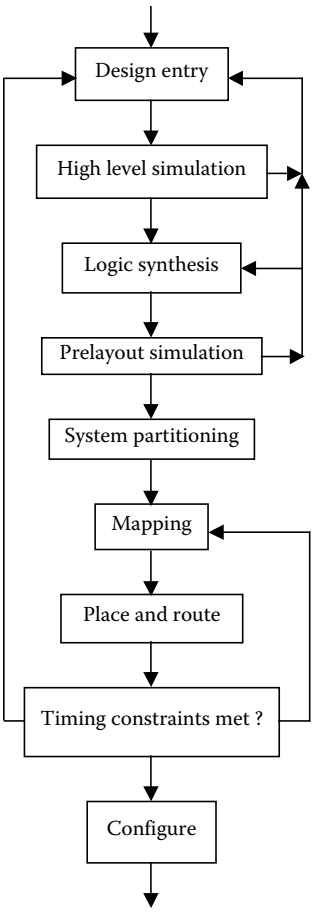


FIGURE 9.20 Design process.

"Hardware description language (HDL)" entry is potentially the most useful of all methods. Popular HDL languages are VHDL, Verilog, and System C. Using these languages, a designer can specify an executable specification of his device. Mature simulators exist for all of these languages that allow functional verification of the high level HDL design.

9.1.12.2 Logic Synthesis

Logic synthesis is the process of transforming a given description of a device produced by one of the design entry methods described in the previous section into an equivalent netlist using primitive components. For example, the process of transforming a symbolic state machine description or an HDL description into a netlist is an example of logic synthesis. The power of the synthesis algorithms in a development system is perhaps the most important feature of the system.

HDL synthesis tools are beginning to be mature enough for use in both PLD and FPGA designs. Very good synthesis tools exist for subsets of the popular HDL languages. Full synthesis tools for all language constructs are still in the research phase. In the next section, we will illustrate how to synthesize FPGA designs using the VHDL language.

"Logic minimization" is obviously an essential process in a PLD development system because the number of product terms per output gate on PAL chips is limited. Recall that the goal of logic minimization for PLD designs is to reduce the number of product terms, not the size of the product terms. Classical logic minimization algorithms use cost functions that reward reduction of the number of gate inputs. This is important for TTL gate implementations, for example, because an eight-input gate costs

about four times as much as a four-input gate. In PLD designs, the number of gate inputs does not matter. Each product term consumes one product line in the chip. A one-literal product term, such as X, costs exactly the same as a ten-literal product term, such as ABCDEFGHIJ. Therefore, traditional logic minimization programs, such as Espresso, need to be modified for PLD development. If product terms can be shared among different outputs, then multiple output minimization is necessary. However, for most PAL devices, the product terms cannot be shared; therefore, single output minimization algorithms are sufficient. Single output minimization algorithms are much less complex and take much less time to execute than multiple output minimization algorithms. Therefore, systems that do single output minimization result in higher productivity. Therefore, be careful of systems that advertise well-known traditional logic minimization algorithms to market their products, especially if multiple output minimization is stressed.

''Equation factoring,'' which is sometimes called ''multiple level minimization,'' is essential in order to fit large functions onto PLD chips using multiple cells combined with feedback lines inside the chips. This feature is missing from most vendor PLD development systems. However, in order to provide effective automated PLD synthesis, this operation is absolutely necessary. In most current PLD development systems, the designer must interact with the synthesis program to implement multiple level minimization. Such interaction requires extensive skill from the user of the software package.

9.1.12.3 Simulation of Designs

All good development systems include some form of simulation capability. The simulators vary widely in scope, user interface, and general usefulness.

Behavioral simulation allows high-level design descriptions to be simulated independent of implementation. Behavioral simulators verify the input–output behavior of the device. Correct behavioral simulation verifies the correctness of the algorithms prior to mapping the design to specific hardware components.

Device simulators verify the function of the design after mapping the design to a specific chip but before actually programming the chip. This is the most common type of simulator in current PLD development systems. A device simulator will construct a software model of the target PLD architecture, map the design to that architecture, and then simulate the behavior of the specific PLD. The better simulators will provide timing information as well as functional information.

9.1.12.4 Mapping Designs to Chips

System partitioning, mapping, place and route, and configure functions are usually performed by vendor-specific development software. These software packages usually accept a netlist as input and produce an output file that can be read by the programming tool.

```
SynchronousRegProcess: process (CLK)
 begin
  -- No other statements here
  if (CLK'event and CLK='1') then
   if RESET ='0' then
    AREG <='0';
   else
    AREG <= A;
   end if;
  end if;
  -- No other statements here
 end process;
```

FIGURE 9.21 VHDL code for register with synchronous reset.

9.1.13 VHDL Synthesis Style for FPGAs

Since HDL synthesis is one of the most popular ways to design FPGAs, we will show representative synthesis techniques for VHDL, one of the common HDL languages. The user may use these examples as templates to write code that will synthesize successfully.

9.1.13.1 Registers and Flip-Flops

Figure 9.21 shows a VHDL template for a register, AREG, with synchronous reset signal (RESET) and data input, A. In VHDL, entities are called processes. This code defines a process named

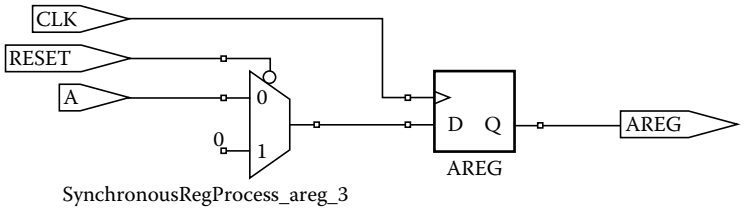


FIGURE 9.22 Synthesis of VHDL code in Figure 9.21.

SynchronousRegProcess. The list of signal names in the parentheses following the word process is called the sensitivity list for the process. The sensitivity list for a register process must contain the clock signal for the process. The notation, CLK'event, is a Boolean expression that is TRUE when signal CLK changes value. This change may either be a rising edge or a falling edge. Combining CLK'event with CLK = '1' specifies a rising edge triggered clock. The notation *AREG <= A* specifies that the current value of input signal A is assigned to register AREG on the rising edge of signal CLK. Similarly, the notation, AREG <= '0', implies that signal AREG is reset to 0 on the rising edge of CLK when RESET = '0'. Current synthesis semantics dictate the following constraints on the VHDL code.

1. The "if" statement that identifies the clock signal must be the only top level statement in the process. No other statements may come before or after this "if" statement, as indicated by comments in the VHDL code.
2. The condition that identifies the clock signal for the register (CLK'event and CLK = '1'), must contain no other conditions. For example, we could not implement the synchronous reset by writing (if CLK'event and CLK = '1' and RESET = '0' then...).
3. Only one clock edge may occur in each process. That is, it is unacceptable to have both CLK1'event ad CLK2'event in the same process.

Figure 9.22 shows how a synthesis tool will synthesize this VHDL code into a clocked register. Note that the RESET signal is low active (a logic 0 causes a reset).

Figure 9.23 shows a VHDL template for a register, AREG, with an asynchronous reset. The same restrictions that were listed for the synchronous register also apply here. Notice that the primary differences are that, in the asynchronous register, signal RESET must be in the sensitivity list and the RESET test occurs before the test for the clock edge, whereas the test for RESET in the synchronous register occurs after the test for the clock edge. Figure 9.24 shows how a synthesis tool synthesizes the VHDL code in Figure 9.23 into a register with an asynchronous reset signal.

```
Asynchronous_Reset: process (CLK, RESET)
 begin
  if RESET = '0' then
   AREG <= '0';
  elsif (CLK'event and CLK = '1') then
   AREG <= A;
  end if;
 end process;
```

FIGURE 9.23 VHDL code for a register with asynchronous reset.

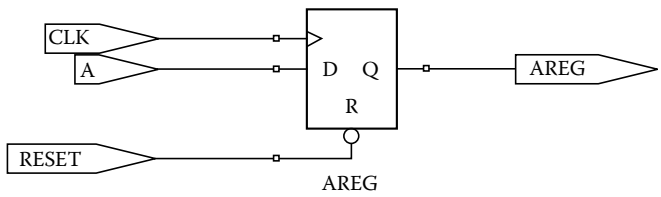


FIGURE 9.24 Synthesis of VHDL code in Figure 9.23.

TABLE 9.9 Ones Counter Truth Table

A(2)	A(1)	A(0)	C(1)	C(0)
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

```
process(A)
begin
  case A is
    when "000" => C<= "00";
    when "001"|"010"|"100" => C<= "01";
    when "011"|"101"|"110" => C<= "10";
    when "111" => C<= "11";
    when others => null;
  end case;
end process;
```

FIGURE 9.25 VHDL code for combinational logic.

9.1.13.2 Combinational Logic

Consider the ones counter truth table of Table 9.9. The input is a 3-bit vector, A, and the output is a 2-bit vector, C. The output reflects the number of ones in the input vector. For example, if A = 111, then C = 11 indicating that A has 3 ones.

There are many ways to represent CL using VHDL. The most direct way is to use a case statement, as shown in Figure 9.25. This type of code is directly related to the truth table format. It simply specifies the output for each combination of the inputs. Figure 9.26 shows how a synthesis tool synthesizes the code in Figure 9.25. Note that it takes the unusual approach of using a MUX as part of the circuit. The reason for this is that the target FPGA chip has many MUXs in its CLBs (see Figure 9.19).

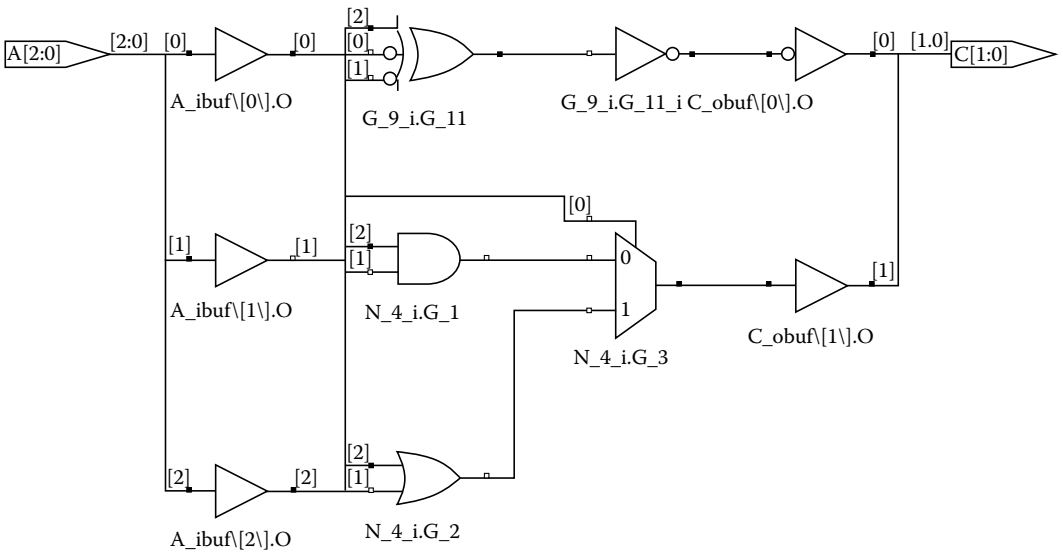


FIGURE 9.26 Synthesis of VHDL code Figure 9.25.

9.1.13.3 Latches

Figure 9.28 shows that the VHDL code in Figure 9.27 synthesizes as a latch with output signal name Q. Note that the process sensitivity list must include both the latch input data signal name, DATA, and the latch enable signal name, ENABLE. Also, the code for a latch may not contain an expression referring to an edge, such as ENABLE'event. The reason that the synthesis tool produces a latch is that a new value is not assigned to signal Q during every call to the process. A new value is only assigned when ENABLE = '1'. Otherwise, the signal Q must not change, i.e., it must retain its old value when ENABLE = '0'. This action requires a latch.

Figure 9.30 shows that the VHDL code in Figure 9.29 synthesizes as CL and does not produce a latch. In this VHDL code, signal Q is assigned a new value every time the process is called. Therefore, the synthesized circuit is CL instead of a latch.

9.1.14 Synthesis of State Machines

Figure 9.31 shows a VHDL template for a state machine. First, the code includes a declaration of a data type called STATE_TYPE. This data type is simply a list of the names of the states. The names should be chosen to reflect the purpose of the state, such as INITIAL, IDLE, TRANSMIT, RECEIVE, etc. Next, a signal, STATE, is declared to be of type STATE_TYPE. Signal STATE keeps track of the current state of

```
LATCH: process (ENABLE, DATA)
  begin
    if (ENABLE = '1') then
      Q <= DATA;
    end if;
  end process;
```

FIGURE 9.27 VHDL code for a latch.

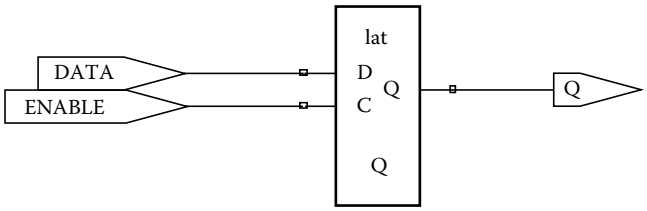


FIGURE 9.28 Synthesis of VHDL code in Figure 9.27.

```
CLP: process (ENABLE, DATA)
  begin
    if ENABLE = '1' then
      Q <= DATA;
    else
      Q <= '0';
    end if;
  end process;
```

FIGURE 9.29 VHDL code for combinational logic.

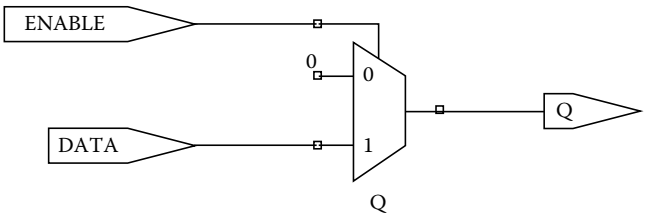


FIGURE 9.30 Synthesis of VHDL code in Figure 9.29.

```
architecture FSM of NAME is
    type STATE_TYPE is (S0, S1,..., Sn);
    signal STATE: STATE_TYPE ;
        -- other_signal_declarations
begin
STATE_PROCESS: process (CLK, RESET)
  begin
    if RESET='0' then
      STATE <= S0; -- The initial state
      -- Insert Reset statements
    elsif CLK'event and CLK = '1' then
      case STATE is
        when S0 = >
             -- Data_Section
             -- Control_Section
        when S1 =>
             -- Data Section
             -- Control Section
        when others =>
             -- Actions
      end case;
    end if;
  end process;
OUTPUT_PROCESS: process (STATE) begin
    case STATE is
      when S0 = >
           -- Output_Signal_Assignments
      when S1 =>
           -- Signal Assignments
      when others =>
           -- Signal Assignments
    end case;
  end process;
end FSM;
```

FIGURE 9.31 VHDL template for a state machine.

the state machine. The architecture consists of two processes. Process STATE_PROCESS updates the current state on each positive transition of clock signal CLK. Process OUTPUT_PROCESS updates the state machine outputs whenever there is a change in state. As written, these processes implement a Moore state machine. To design a Mealy state machine, simply add the machine input signals to the OUTPUT_PROCESS sensitivity list. The state machine has a low-active asynchronous RESET signal that initializes the state machine to state S0. A case statement performs data transfer statements and computes the next state based on the current state. For other approaches to using high-level languages to design digital systems, see Ref. [8].

References

1. *Programmable Logic Data Book*, Texas Instruments, Dallas, TX.
2. *Programmable Logic*, Intel Corporation, Mt. Prospect, IL.
3. C. Alford, *Programmable Logic Designer's Guide*, Howard W. Sams & Company, Indianapolis, IN, 1989.
4. H. Katz, *Contemporary Logic Design*, Benjamin/Cummings Publishing Company, Redwood City, CA, 1994.
5. L. Pappas, *Digital Design*, West Publishing Company, St. Paul, MN, 1994.
6. D. Pellerin and M. Holley, *Practical Design Using Programmable Logic*, Prentice Hall, Englewood Cliffs, NJ, 1991.
7. J. F. Wakerly, *Digital Design, Principles & Practices*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 1994.
8. J. R. Armstrong and F. G. Gray, *VHDL Design: Representation and Synthesis*, 2nd edn., Prentice Hall, Englewood Cliffs, NJ, 2000.
9. V. P. Nelson, H. T. Nagle, B. D. Carroll, and J. D. Irwin, *Digital Logic Circuit Analysis and Design*, Prentice Hall, Englewood Cliffs, NJ, 1995.

9.2 Clocking Schemes

Wayne D. Grover

9.2.1 Introduction

Advances in very large-scale integrated (VLSI) processing technology, particularly CMOS, have resulted in nanometer-scale processes with applications at clock speeds of several gigahertz. For example, at the time of this revision the state of the art is fairly well represented by the Intel Core2 Duo processor chip, which is implemented in 65 nm CMOS, consists of 376 million transistors, and clocks at 2.66 GHz. New design challenges must be mastered to realize systems at an ever-increasing clocking rates and circuit sizes. In particular, clocking-related issues of skew, delay, power dissipation, and switching noise can be design-limiting factors. In large synchronous designs, the clock net is typically the largest contributor to on-chip power dissipation and electrical noise generation, particularly ''ground bounce,'' which reduces noise margin. Ground bounce is a rise in ground potential due to surges of current returning through a nonzero (typically inductive) ground path impedance. At the board and shelf level, clock distribution networks can also be a source of electromagnetic emissions, and may require considerable delay tuning for optimization of the clock distribution network.

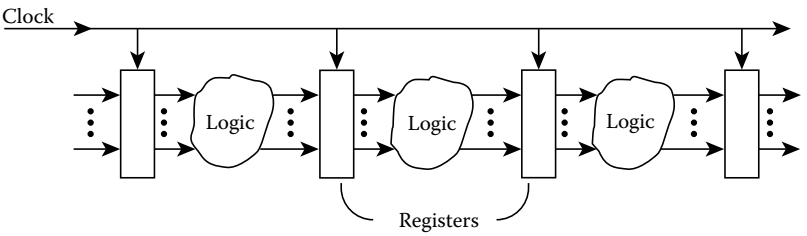
In the past, multiphase clocking schemes and dynamic logic structures helped minimize transistor count, but this is now less important than achieving low skew, enhancing routability, controlling clock-related switching noise, and providing effective CAD tools for clock net synthesis and documentation. For these reasons, a shift has occurred toward single-phase clocking and fully static logic in all but the largest custom designs today. In addition, phase-feedback control schemes using phase-locked loops (PLLs) are becoming common, as are algorithmic clock-tree synthesis methods.

This section focuses on the issues and alternatives for on-chip and multichip clocking, with the primary emphasis on CMOS technology. We first review the fundamental nature and sources of skew and the requirements for the clocking of storage elements. We then outline and compare a number of ''open-loop'' clock distribution approaches, such as the single-buffer, clock trunk, clock ring, H-tree, and balanced clock tree approaches. PLL synchronization methods and PLL-based clock generation are then outlined. In closing, we look at future technologies and developments for high-speed clocking. The concepts and methods of this section apply to many circuit technologies on- and off-chip. However, we emphasize CMOS because CMOS processes (including bi-CMOS) presently represent the vast majority of digital VLSI designs and are expected to continue to do so.

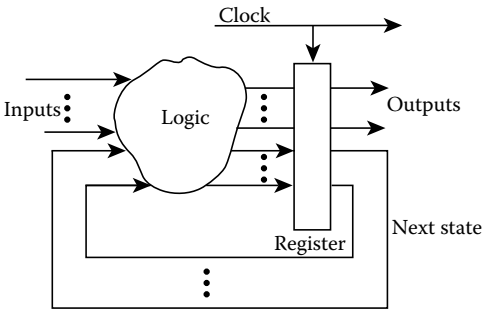
Asynchronous, self-timed, and wavefront array systems are outside the scope of this chapter. These approaches aim to minimize the need for low-skew synchronous clocking. However, truly asynchronous modules tend to require a large overhead in logic for interaction with each other, so that speed, size, and power often suffer relative to synchronous design. Nonetheless, self-timing can be an effective approach for random-access memory (RAM and ROM) cells, to which considerable optimization effort can be invested for reuse in many designs. Self-timed methods should be considered the alternative to fully synchronous design in large, highly modularized systems, particularly where well-defined autonomous modules have relatively infrequent interactions. The main issues in self-timed systems are the possibly high delay required to avoid metastability problems between self-timed modules, and the circuit costs of the synchronization protocol for intermodule communication.

9.2.2 Clocking Principles

Most of us accept the clocked nature of digital systems without question, but what, fundamentally, is the reason for clocking? Any digital system can be viewed either as a pipeline or as an FSM architecture, as outlined in Figure 9.32. In the pipelined architecture clocked sections are cascaded, each section comprising an asynchronous CL block followed by a latch or storage element that samples and holds the logic state at the clock instant. In the FSM, the only difference is that the next state input and the system outputs are determined by the asynchronous logic block, and the sampled next state value (S) are fed back into the CL. The FSM can therefore be conceptually unfolded and also represented in a pipeline fashion. The fundamental reason for clocking digital systems is seen in this pipelined abstraction of a digital system: it is to bring together and retain coordination among asynchronously evolved intermediate results. With physical delays that are temperature, process, and input dependent in CL, we need to create ''agreed-upon time instants'' at which all analog voltages in a system are valid when interpreted as Boolean logic states. Clocking deals with delay uncertainty in logic circuit paths by holding



(a) Pipelined architecture



(b) Finite-state machine atchitecture

FIGURE 9.32 Architecture of digital systems.

up the fast signals and waiting for the slower signals so that both are valid before they are again combined or interact with each other. Without this coordination purely asynchronous logic would develop severe propagation path differences, and be slow in repetitive operations. Ultimately, a valid state would evolve, but all inputs would have to be stable for the entire time required for this evolution. On the other hand, when the overall CL function is appropriately partitioned between clocked storage latches, system speed can approach the limit given by the delay of a single gate because each logic subblock is reused in each clock period.

From this, we obtain several insights: (1) only the storage elements of a digital system need become loads on the clock net (assuming state logic gates); (2) the system cannot be clocked faster than the rate set by the slowest combinational signal path delay between clocked storage elements; (3) any uncertainty in clock timing (skew) is indistinguishable from uncertainty in the settling time of the intervening CL; and (4) for a logic family to work, its storage elements must: (a) at no time be transparent (i.e., simultaneously connect input to output), (b) have a setup time less than $(T - t_{clk-Q})$ where T is the clock period and t_{clk-Q} is the clock-to-Q output delay of the same type of flop, and (c) have a hold time less than their clock-to-output delay. The last points may be better appreciated by considering that the CL block may be null, i.e., a zero-delay wire, such as in a shift-register.

An implication of [4(a)] is that two-phase nonoverlapping clocks, or an equivalent sequencing process, are fundamentals for the storage elements of a digital system. This may sound unusual to readers who have already designed entire systems with SSI and MSI parts, or in gate-array design systems, without having seen anything but single-phase edge-triggered flip-flops, latches, counters, etc. However, at least two clock phases (or clock-enabling phases) are internally required in any clocked storage device. An analogy is of a ship descending in elevation through a lock [27]. During the first clock phase, sluice gates ''charge'' the lock up to the incoming water level and open the input gate to bring a ship in. Throughout this phase, it is essential that the output gate is closed, or water will race destructively right though the lock. Only when the ship is entirely in the lock and the input gate is closed (isolating the input) can the output sluice gates be opened to equalize the water level to that on the outgoing side, allowing the ship to leave (producing a new output). Similarly, in a flip-flop or latch, the currently stored value, which appears at the output, must be isolated from the input while the input evolves to its next value.

9.2.2.1 Skew and Delay

Clock ''skew'' is defined, most generally, as the difference in time between the actual and the desired instant of active clock edge at a given clocked storage element. In the majority of designs in which the desired instant of clocking is the same at all storage elements, skew is the maximum difference in clock waveform timing at different latches. Clock skew is of concern because it ultimately leads to the violation of setup or hold times within latches, or to clock race problems in multiphase clocking. Furthermore, from a design viewpoint, uncertainty in clock timings must be treated as equivalent to actual clock skew. Skew or timing uncertainty are therefore equivalent to an increase in critical path logic delay. In either case, the clock period must be extended to ensure valid logic levels and proper setup/hold requirements relative to the clock time.

To illustrate the equivalence of skew (either actual of design timing uncertainty) to a loss of system speed, consider a 200 MHz process used in a design which has 25% skew (i.e., actual clock edge dispersion or, equivalently, uncertainty in clock timing is 1.25 ns). If a competitor uses the same process at 200 MHz and achieves 5% skew (0.25 ns), then the latter design has 20% more of each 5 ns clock cycle for settling CL paths. Alternatively, for the same logic functions, the low-skew system could be clocked at 250 MHz with the same timing margins as the high-skew system. Skew, therefore, represents a loss of performance relative to basic process capabilities developed at the great expense. However, skew reduction costs relatively little, and is in the logic designer's control, not the process developer's and yet it is directly equivalent to a basic enhancement in process speed.

Skew is usually of primary concern on-chip or within any module that is designed on the presumption of a uniform clock phase throughout the module. Clock ''delay,'' on the other hand, is the difference

between the nominal clock edge time at an internal flip-flop and the system clock, or timing reference, external to the chip. While skew is typically managed internal to the die, delay is of concern at the system level to ensure external setup and hold time requirements. Skew and delay may be independent in any given clock distribution scheme. For example, an on-chip clocking tree that yields essentially zero skew may, nonetheless, impart a high clock delay, which will be of importance at the system level. The ''early clock'' technique and some PLL methods (presented later) can be used to address on-chip clock delay problems.

9.2.2.2 Isochronic or Equipotential Regions

The clock distribution problem arises at all scales of system design, from on-chip clock distribution in VLSI and wafer-scale integration (WSI) to the synchronization of circuit packs tens of meters apart. These applications are unified as a generic problem: synchronous clocking of ''electrically large'' systems, i.e., systems in which propagation time across the system is significant relative to the clock period. In such systems:

$$D/v > k/f_{\text{app}} \tag{9.1}$$

where

D is the characteristic scale or distance of the system
v is the propagation velocity
f_{app} is the application clock frequency
k is the skew requirement as a fraction of the clock period

For all locations around a clock entry point at which Equation 9.1 is false, we can consider events to be essentially simultaneous (or, equivalently, the region is a single electrical node), and the clock can be distributed within such regions without delay equalization. The region over which clock can be distributed without any significant skew is also known as an equipotential [27] region, or an isochronic region [1]. In this section, we are concerned only with cases in which Equation 9.1 is true, but it is implicit that the clocked end nodes may be either clocked loads directly or a buffer that feeds a local isochronic region.

The diameter (and shape) of an isochronic region on-chips depends on the wire type employed for interconnection. To control skew on chip, we need to consider delay differences due both to wire lengths and to the lumped capacitive efforts of the driven loads. Where the RC time constant of the wiring interconnect τ_w is much less than the RC combination of the driving source resistance and the lumped capacitance of N clocked loads on the net ($\tau_{\text{net}} = R_s C_{\text{gate}} N = N\tau_g$), we can consider all points on a net to be isochronic, meaning that the net acts like one electrical node characterized by the total lumped capacitance of gates on the net. Wires on a chip are often modeled as distributed R_0C_0 sections, where R_0 and C_0 are the resistance and capacitance per unit length, respectively (see Figure 9.33). In such cases, the propagation delay for a wire of length l follows the diffusion equation [31].

$$\tau_w = R_0 C_0 l^2/2 \tag{9.2}$$

Therefore, if we consider a net of length l with N standard loads, we can consider the net to be isochronic if $\tau_w \ll N\tau_g$. From Equation 9.2, this implies

$$l \ll \sqrt{\frac{2NR_sC_{\text{gate}}}{R_0C_0}} \tag{9.3}$$

This relationship provides a guideline for the maximum length over which wire delays may be neglected relative to gate-charging delays. Based on typical values for a 1 μm process, $\tau_g < 500$ ps, isochronic regions for lightly loaded lines ($N = 1$) are up to 10,000λ for lines in third layer metal, 5,000 and 8,000λ

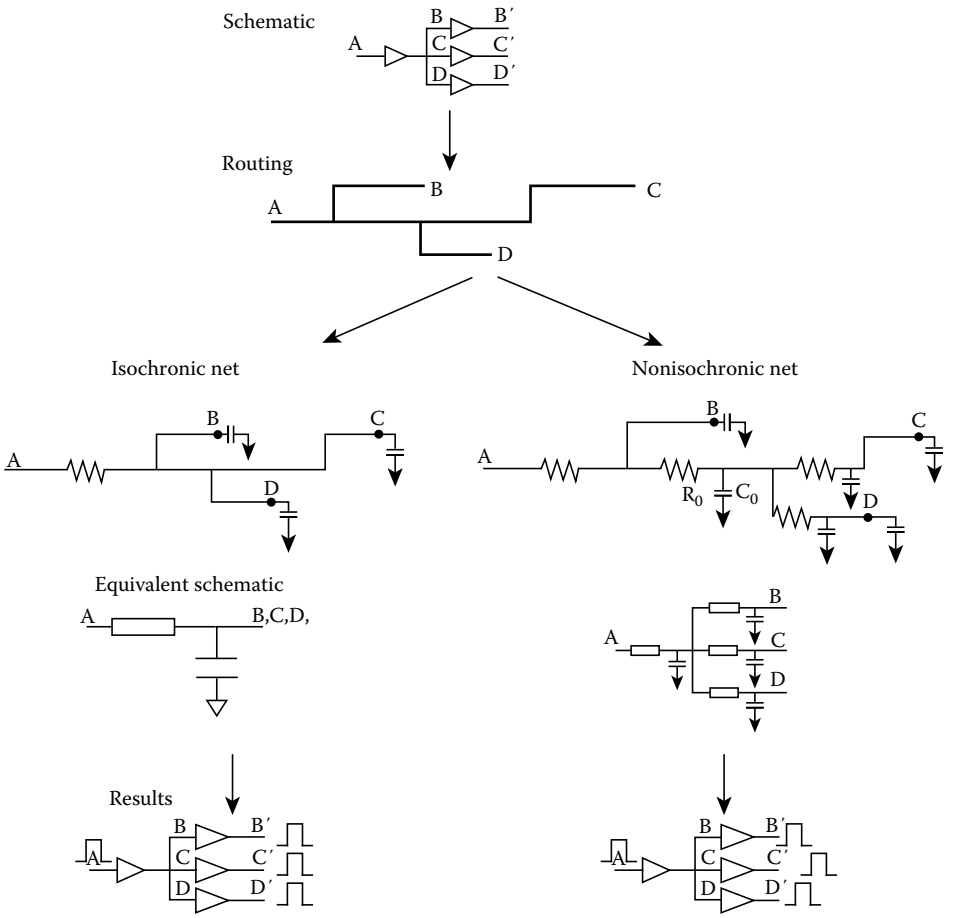


FIGURE 9.33 Isochronic and nonisochronic passive wiring nets.

for first and second layer metal, respectively, and 200λ for polysilicon wires, where λ is the minimum feature size of the process [31]. This illustrates the importance of distributing clock within metal layers to the greatest extent possible. Even a few short polysilicon links may introduce sufficient series resistance to drastically reduce the isochronic region for the given clock line. This also illustrates that if clock is distributed in metal layers, and is always buffered before exceeding the isochronic distance, it will be primarily differences in lumped capacitive loads and not wire length that determine clock signal delays, and hence relative clock skews.

9.2.2.3 Nature of Skew On-Chip

The concept of isochronic regions helps us understand the nature of clock skews in VLSI and helps explain why on-chip skews may be greater than those between off-chip points that are physically many more times distant. A key realization is that signals do not propagate at the "speed of light." If they did, absolute delays across even the largest chips (2 cm edges) would be subnanosecond and the isochronic diameter would easily encompass an entire die at clock speeds up to 200 MHz. Rather, on-chip propagation delay depends much more on the time needed for output drivers to charge the total lumped capacitance associated with all the gate inputs of the driven net. In other words, fanout and driver current

abilities have more to do with delay than path lengths. This is especially true when clock distribution is exclusively via metal layers, as is the norm in a modern design. On the other hand, off-chip, we route signals via impedance-controlled coaxial or microstrip lines, or via optical fiber, and these media typically do exhibit propagation velocities of 0.6–0.8 c. Therefore, off-chip, differences in physical propagation distances are the dominant source of skew, while on-chip, it is imbalances in driver loads that are the most common source of skew.

In on-chip cases in which wire diffusion delays and lumped capacitive effects are both significant, a difference in line length can also result in skew due to a different total wiring capacitance. In addition, equal length lines that go through different metallization layers or through polysilicon links will have different delays due to different levels of capacitive coupling to V_{ss} and different series resistances, especially in the case of polysilicon links. Accordingly, an important principal to simplify clock net design is to aim for buffering levels and fanouts that yield isochronic conditions for the passive wiring nets between buffered points on the clock net. This simplifies skew control in clock net design because attention then only need be paid to balancing loads on each buffer and to matching the number of buffers in each clock path. The alternative, in which passive wiring nets are not isochronic, requires detailed delay modeling of each wire path, taking into account the actual routing, the R_0C_0 of the wire type, the temperature, and the exact position of each lumped load on the wiring path. The important concept, however, is that by the choice of metal layers, line widths, and/or loadings, one can establish formally defined isochronic conditions on some or all portions of a complete clock net, which, in its entirety, is far too large to be isochronic. When fully isochronic subregions (such as a wide clock trunk) can be established, or even when a defined region is not isochronic but has delay that is simply and reliably predicted from position (such as on a clock ring), the remaining clock net layout and skew control problem is simplified, design risk is lowered, and pre- and postlayout simulations are more consistent because final routing of clock paths from these reference regions is shortened and overall uncertainty reduced. We shall see and use this principle in analyzing the popular clock distribution schemes that follow.

The skew that intrinsically arises from differences in RC time constants of either lines or gate loads is aggravated by threshold variations in buffers and clocked loads due to minute differences in electronic parameters and lithographic variation in line widths and lengths at different devices. Time-constant and threshold effects interact to give a worst-case skew, which is the difference between the time at which the voltage response of line with the slowest time constant, τ_{max}, crosses the threshold of the logic element with the highest threshold, V_{Tmax}, until switching of the device with the lowest threshold driven by the line with fastest RC time constant. Taking the difference of the earliest and latest switching times we have [32]

$$\delta = \tau_{min}\ \ln\left(1 - \frac{V_{Tmin}}{V_{DD}}\right) - \tau_{max}\ \ln\left(1 - \frac{V_{Tmin}}{V_{DD}}\right) \tag{9.4}$$

Equation 9.4 implies that a clock system design in which buffered electrical segments of the clock net have 10% variation in τ about τ_{nom}, and 10% variation of V_T about $V_{DD}/2$, will have an estimated skew of at least 17% of τ_{nom}.

9.2.2.4 Single-Phase Clocking

Clocks ultimately always drive a storage register or latch of some type. The form of clock signal(s) required in a system therefore depends on the type of latch or flip-flop element used and on properties of the CL circuits used. True single-phase clocking is the most complex clocking principles with which to design systems, and has traditionally not been used, although recent work has assessed some truly single phase logic families [2]. The reason for caution with single-phase clocking is that invalid states may be passed to the output in two ways, as shown in Figure 9.34a if the CL delay is less than T_h (i.e., too fast) or (Figure 9.34b) the CL delay is

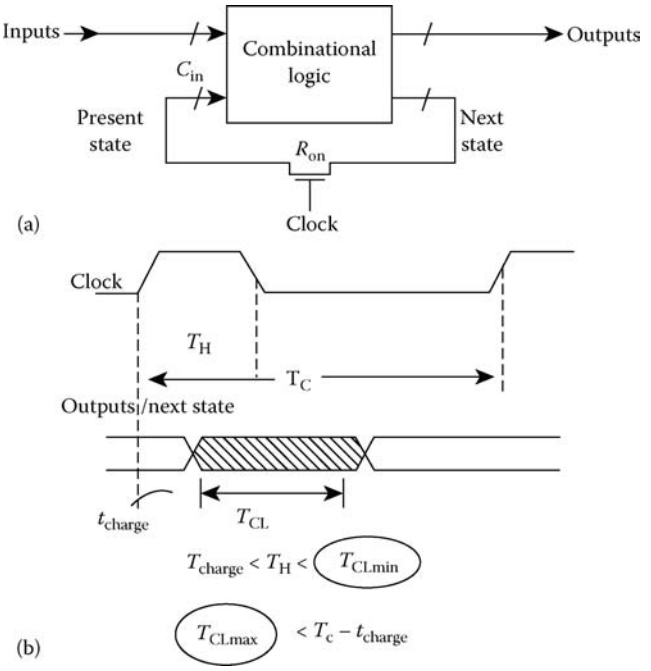


FIGURE 9.34 In single-phase clocking the CL path must be neither too slow nor too fast.

greater than $T_C - t_{charge}$ (i.e., too slow). In other words, a two-sided (min and max) constraint on logic path delay exists for single-phase clocking [27]. This means that although attractive to minimize total interconnect, buffer counts, and interphase skew is avoided, truly single-phase clocking involves a greater design risk and timing analysis complexity. Because of this, the most common overall clocking scheme is single-phase clock distribution with local generation of a two-phase clock.

9.2.2.5 Two-Phase Clocking

With two nonoverlapping clock phases, we can eliminate one of the risks of single-phase clocking, that of a logic path being too fast. On the first phase the input is made transparent and allowed to affect the CL, charging the inputs through R_{on} C_{in} in time t_{charge}. During this time, the CL outputs are isolated from the input latch. On the second phase, the new CL output values are stored by the second phase latch while the input latch is opaque, isolating inputs from the new values until the next phase one clock time. A nonoverlapping period between phases ensures that at no time does direct transparency occur from input to output. With two-phase nonoverlapping clocks, as shown in Figure 9.35a, we need to ensure only that the maximum delay in the CL is less than $T_C - t_{charge} - T_3 - t_{preset}$. It is essential that the nonoverlapping interval, T_2, be greater than zero, but T_3 can be arbitrarily short. When present, however, T_3 acts as an extra timing margin against skew.

It is obviously desirable to make T_2 as small as possible, but we can do so only when distributing two-phase clock directly if the interphase skew is less than T_2. In the worst case, the interphase skew may be twice the skew of each of the two clock phase nets individually. Skew, therefore, necessitates at least a 1:1 derating in speed for two-phase clocking in addition to the basic loss of clock cycle time for logic settling, to ensure correct operation of storage devices. If skews in the two clock phase nets are uncorrelated, however, the extra penalty could be as high as 2:1. Every nanosecond of skew in the clock net for each phase then not only reduces the basic critical path logic timing margin by 1 ns, but also adds 2 ns to the T_2 requirement.

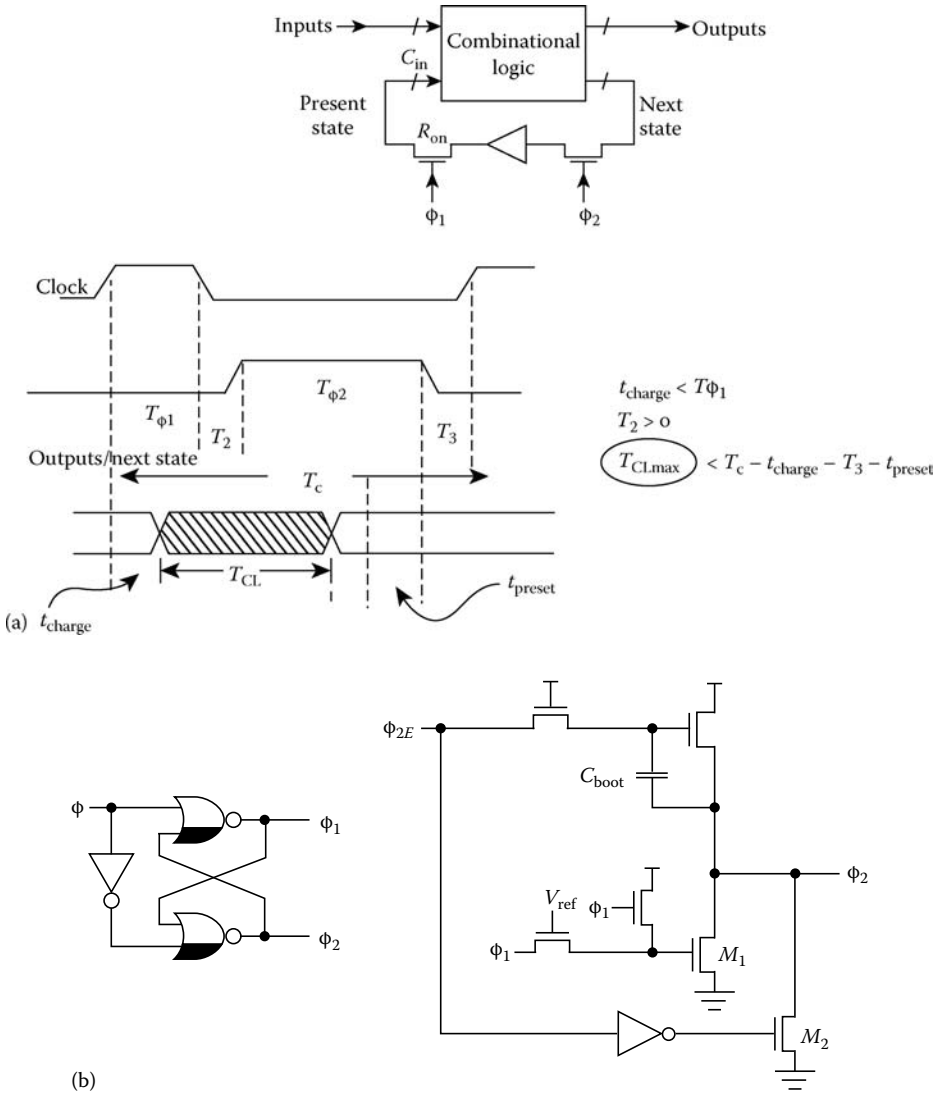


FIGURE 9.35 (a) With nonoverlapping two-phase clocks, no lower limit exists on the CL delay; and (b) generator for two-phase nonoverlapping clock and buffer circuit to ensure nonoverlap period. (From Glasser, L.A. and Dobberpuhl, D.W., *The Design and Analysis of VLSI Circuits*, Addison-Wesley, Reading, MA, 1985, 349.)

Therefore, in high-performance systems we have quite an incentive to distribute a single clock phase throughout the design and accept the extra logic required to generate two-phase clocks locally at each device (or small group of similar devices) that requires them.

9.2.2.6 Two-Phase Clock Generator Circuit

The canonical form of circuit to generate the local two-phase nonoverlapping clocks from a single phase clock is shown in Figure 9.35b. The feedback of ϕ_2 into NOR1 ensures that ϕ_2 must be low before ϕ_1 can go high after the single-phase ϕ_{in} input has gone low, and vice versa. A special clock buffer circuit is shown in Figure 9.35b, which helps ensure that a period of nonoverlap exists in the presence of the threshold variations in the driven loads [12]. It does this by using transistor M_1 to clamp the ϕ_2 output

low until far into the fall of ϕ_1. ϕ_2 is held low until ϕ_1 has fallen below ($V_{ref} - V_{thresh}$) to finally cut off M_1. V_{ref} might be set at 2 V in a 5 V process, thereby ensuring that ϕ_1 is well below the logic threshold of all clocked loads before ϕ_2 begins to rise, while at the same time minimizing but guaranteeing the existence of a nonoverlap period, which is lost processing time.

9.2.2.7 Multiple-Phase Overlapping Clocks

Generating and/or distributing nonoverlapping clocks with a minimal T_2 can be quite difficult in large systems. An alternative is to define three or more primary functional logic steps and use a similar number of overlapping clock phases. In this case, the multiple stages of clocking removes the need for the guaranteed nonoverlap period in two-phase clocking. Let us consider three-phase overlapping clocking. The principles generalize to any higher number of phases.

In three-phase clocking, the middle phase can be thought of as providing the nonoverlap time, which ensures time isolation between I/O activation for the module enabled on each clock phase. In fact, each phase plays a similar isolating role with respect to operations performed on its adjacent phases. Figure 9.36 illustrates the concept. The number of phases and the role for each phase typically reflects some natural cycle or step sequence of the basic system being designed; for example: bus input, add to accumulator, bus output.

In WSI systems, in which uncertainty in clock delays, circuit speeds, and interconnect impedances may be high [10], overlapping clock logic can give high tolerance to clock skew, and is compatible with self-timing in selected subcircuits. Three-phase overlapping clocking has a distinct advantage: no hazard exists unless all three clock phases overlap in time. In the presence of severe clock skew, this can be a major advantage. Although called overlapping clocks, the circuits still function if successive phases do not actually overlap, although speed is sacrificed if overlap is lost.

9.2.2.8 Overlapping Clock Phase Generator

Figure 9.37 illustrates a circuit for generating three-phase overlapping clocks. Phase overlap is ensured because it is the onset of each phase that kills its predecessor. A Johnston counter comprised of three static *D*-flip-flops generates the phase-enabling signals which sequence the actual generator stage, which is comprised of the three cross-coupled NOR gates. A deliberately limited positive-going drive ability of the enable input ensures that the Johnston counter exercises underlying rate and sequence control, while the output waveforms are determined by the interactions between the actual clock phase signals. While the enable inputs to each NOR are logically sufficient to drive the output high when the other input is low, they are arranged not to be able to drive the NOR output low on their own when the enable signal returns high. The output of phase *i* therefore stays high after its own enable signal has

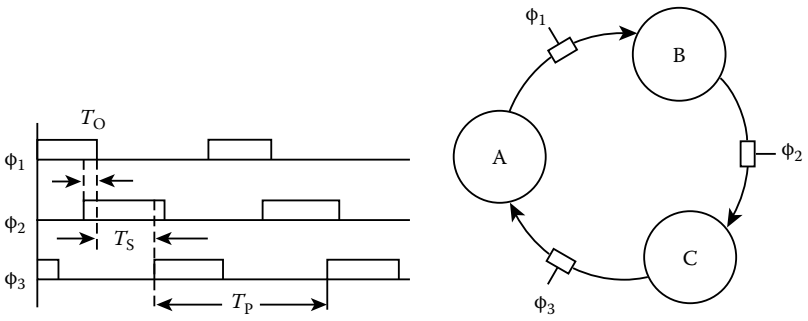


FIGURE 9.36 Principle of multiphase clocking. Outputs are isolated from inputs by other stages of nonactive logic even though any two active clock waveforms may overlap. (From Glasser, L.A. and Dobberpuhl, D.W., *The Design and Analysis of VLSI Circuits*, Addison-Wesley, Reading, MA, 1985, 352.)

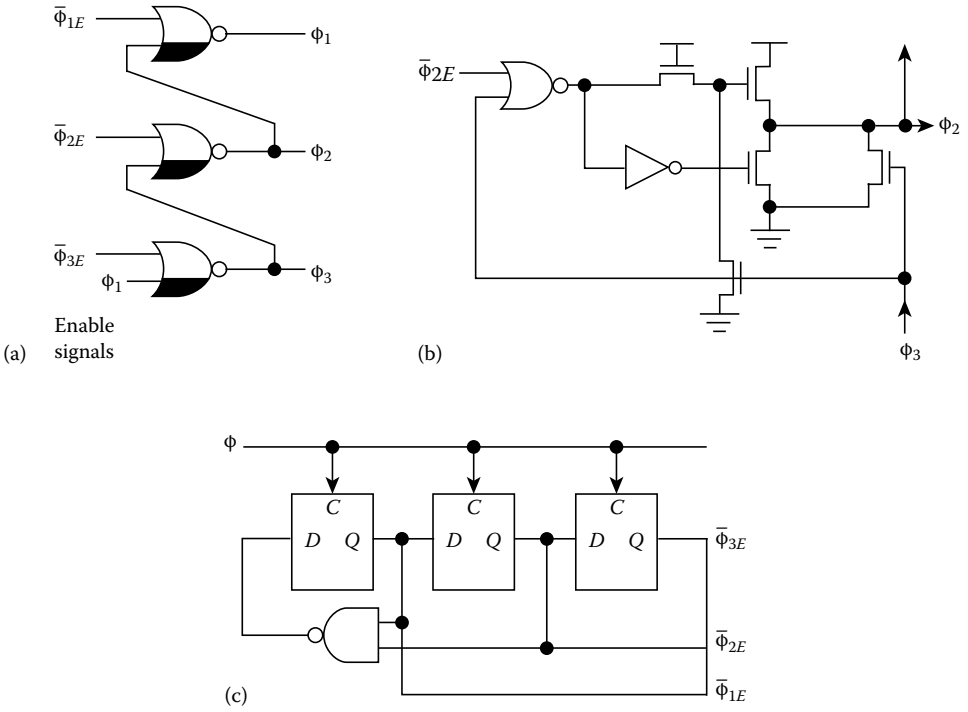


FIGURE 9.37 Three-phase clock generator logic and buffer design to ensure overlap. (From Glasser, L.A. and Dobberpuhl, D.W., *The Design and Analysis of VLSI Circuits*, Addison-Wesley, Reading, MA, 1985, pp. 348 and 354.)

disappeared (gone high) until the phase $i+1$ output is also high in response to the low-going phase $i+1$ enable. Figure 9.37 shows this logic and a NORing clock buffer circuit in which the phase $i+1$ signal is necessary to assist in returning the phase i output to zero.

9.2.2.9 Clocking Latches

A latch is a storage element which is level sensitive to the clock waveform. The latch output conforms to the input that is present while the clock waveform is at its active level for that latch, and then continues to hold that value when the clock level falls below the active level. Latches have setup and hold time requirements analogous to those in the flip-flops that follow. Circuit designs for high and low active latches are given in [31]. By combining latches of opposite active polarity, with logic between the stages, there can be two logic operations per clock period. In this type of operation, however, skew adds to the needed clock period as usual, but in addition any imbalance in the clock duty cycle requires a further margin because the minimum duty cycle half-width must remain greater than the worst-case logic delay. The clock edges also must be kept sharp enough that transparency never occurs between two successive latches working on opposite clock phases simultaneously, or that some minimum logic delay always exists between latches that exceeds the possible overlap time. The DEC ALPHA microprocessor was an example of a two-phase latch machine in which both phases drive latches that are active in the respective phases permitting logic evaluation twice per cycle. This is one case in which to control the very high transistor count, two-phase clock is distributed globally rather than generated at each module. The entire clock net on each phase is driven from a single large buffer placed at the center of the die where a PLL is also fabricated to advance the on-chip clocking phase relative to external bus timing, thereby compensating for buffer delay.

Two principles for maintaining 50% clock duty cycle in a latch machine are (1) whenever generating or phase-locking to a system clock, do so at twice the rate needed, then (frequency) divide by two. This results in 50/50 clock waveform, regardless of the original clock source waveform. (2) When repeatedly buffering clock in a chain, or when distributing clock through a hierarchy of clock buffers, use inverting clock buffers at each stage. Inverting the clock at every buffering stage inherently counteracts the effects of different rise and fall times in buffers. Otherwise, these can accumulate to extend or shorten the ON period of the clock waveform. For example, if a noninverting buffer has greater fall time than rise time, a clock path transiting several of these buffers will develop a broadened ON period. This effect is self-compensating in a chain of inverting buffers.

9.2.2.10 Clocking Flip-Flops

Flip-flops are more complex storage circuits than latches, but have no dependency on clock duty cycle because they are sensitive only during an active edge of the clock waveform. A rising edge *D*-flip-flop (for instance) updates its output to match its *D* input on the rising edge of the clock waveform. The *Q* output retains the updated value thereafter, regardless of further changes in the input or clock waveform (with the exception of another rising clock transition). A latch is a more fundamental circuit element than the *D*-flip-flop in that edge-triggered behavior is attained only by implementing two latches and generating a pair of two-phase nonoverlapping clock pulses internally, in response to the active clock transition at the edge-triggered input.

For instance, Figure 9.38 shows a typical *D*-flip-flop in which inverters I_1 and I_2 generate the internal two-phase clock signals for level-sensitive latches L_1 and L_2. In specialized applications, it may be advantageous to design a custom module within which multiphase clocks are distributed directly, without incurring greatly increased skew problems. For example, an error-correcting codec ASIC prototype for 45 Mb/s digital communications includes a 2613 stage tapped delay line comprised of seven-gate single-phase *D*-flip-flop modules. The use of single-phase clock flip-flops in this context is relatively expensive, but appropriate for fast validation of the system design. For cost- and power-reduced production in volume, a more detailed latch-based design using directly distributed two-phase nonoverlapping clocks may be worthwhile. In general, while it is most common to conduct system design based on the single-phase clocking model, two-phase or multiphase clocking may be advantageous within specialized substructures.

Although edge triggered, a minimum clock pulse width is still typically required to deliver enough switching energy on the clock line. For correct operation (specifically, to avoid uncertain outputs due to metastability) of an edge-triggered flip-flop, data must be stable at the input(s) for a minimum setup time before the clock edge, and the data must remain stable at the input for the hold time, after the clock edge. The time until the *D*-flip-flop output is valid after the clock edge occurs is the clock-to-*Q* delay. For hazard-free transfer of data from one stage to another with *D*-flip-flops, without assuming a minimum logic delay constraint between stages, the clock-to-*Q* delay must exceed the hold time. Typical values for a range of *D*-flip-flop types in a 1.5 μm CMOS process are $t_{\text{setup}} = 0.8$–1.5 ns, $t_{\text{hold}} = 0.2$–0.4 ns, and $t_{\text{clk}-Q} = 1.3$–3.5 ns for *Q* output fanouts of 1–16, respectively. With the extra input logic delays in a JK flip-flop, many JK flip-flop cell implementations exhibit $t_{\text{hold}} = 0.0$ ns. By comparing magnitudes of typical setup and hold time requirements, it is apparent that skew is more likely to cause a setup time violation on critical delay logic paths than it is to result in a hold time violation.

9.2.2.11 Role of Clocks in Dynamic Logic

Clock signals are also used to implement a variety of logic gate functions in a dynamic circuit style, i.e., based on short-term charge storage, not static logic. This typically involves precharging on one phase and logic evaluation steered by the inputs on the second phase. The ''Domino'' logic approach combines a dynamic NMOS gate with a static CMOS buffer [8]. In ''NORA'' (no-race), logic dynamic logic blocks are combined with clocked CMOS latch stages. A variety of other dynamic logic circuits, using up to four clock signals to structure the precharge and to evaluate timing, are covered by [12,31]. In all of these gate

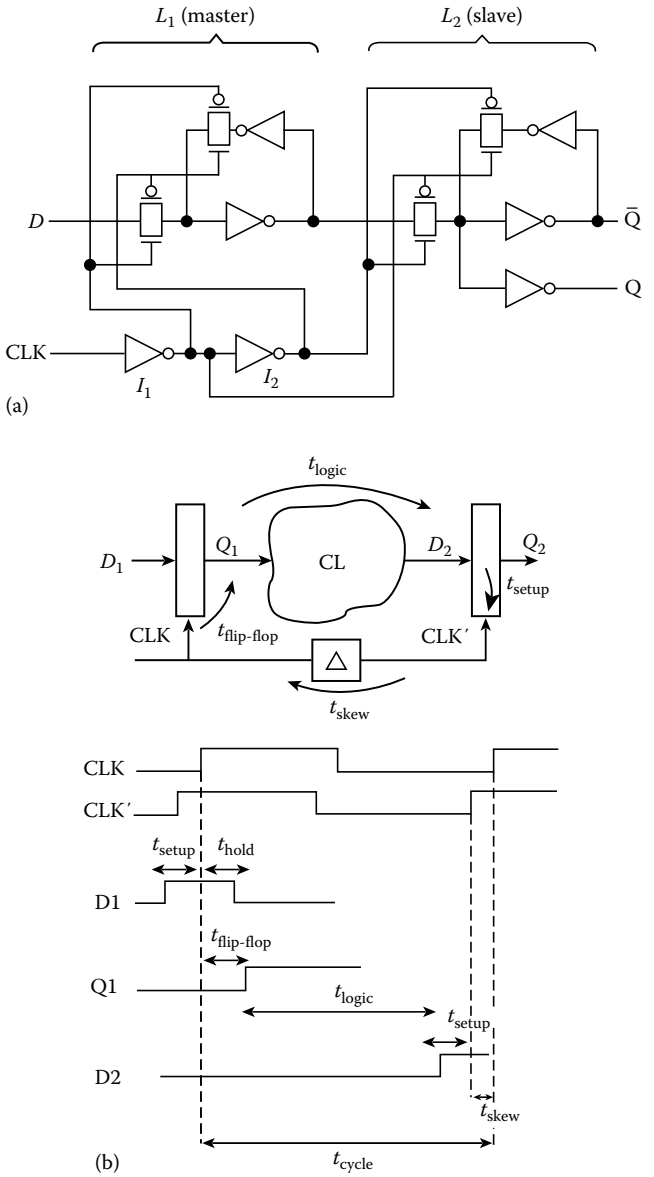


FIGURE 9.38 (a) Two-phase latch structure of a typical CMOS positive edge-triggered *D*-flip-flop; and (b) setup, hold, and delay times for a *D*-flip showing how skew is equivalent to a shorter clock period and threatens setup time margin. (From Bakoglu, H.B., *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley, Reading, MA, 1990, 345.)

level circuit implementations, the clocking-related issues are ultimately manifestations of the basic principles already seen for two-phase clocking; i.e., of never simultaneously enabling a direct path from input (or precharge source) to output. These logic styles were developed to reduce transistor counts. However, modern designers will most often be faced with a greater challenge in managing system-level problems of skew in a single clock phase distributed to static register than the challenge of reducing transistor count.

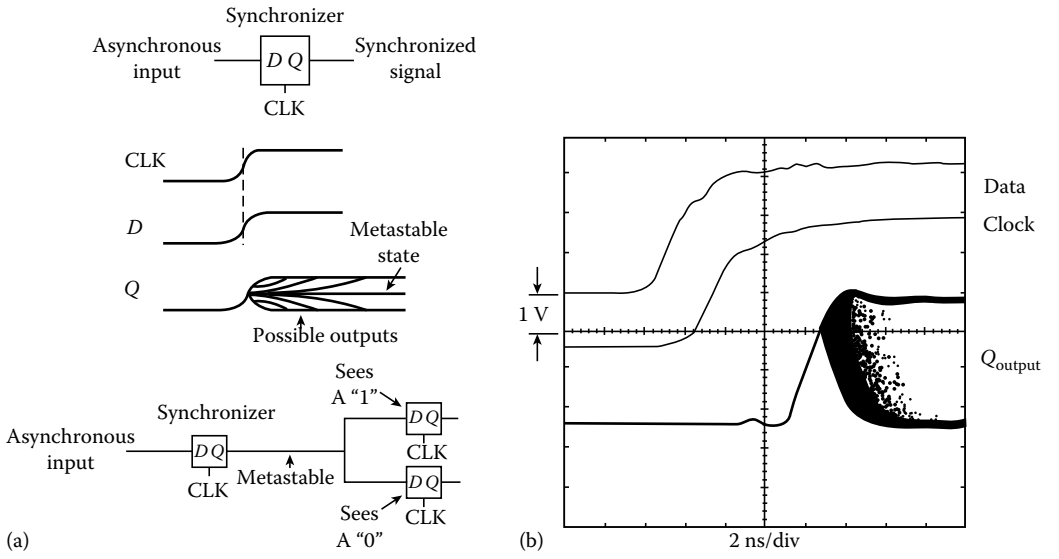


FIGURE 9.39 (a) Metastability in a synchronizer circuit. (From Bakoglu, H.B., *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley, Reading, MA, 1990, 357.) (b) Experimental illustration of metastability in 74F/74 (TTL) *D*-flip-flop showing output rise before indeterminate final response (10 s point accumulation). (From Johnson, M.W. and Graham, M., *High-Speed Digital Design: A Handbook of Black Magic*, Prentice Hall, Englewood Cliffs, NJ, 1993, 130.)

9.2.2.12 Synchronizers and Metastability

Many systems need to sample external inputs which may be timed independently of the synchronous system clock, such as switch-based control inputs, keyboard states, or external process states in a real-time controller. The external state needs to be synchronized with the system time base for processing. Metastability leading to synchronizer failure is a fundamental possibility that can never be entirely eliminated.

Figure 9.39 is a basic synchronizer circuit. Our concern is that it is possible for the synchronizer output to take an arbitrarily long time to settle to one or the other valid logic states if the input signal voltage is sampled in the intermediate voltage range, i.e., $V_{iL} < V_{in}(t) < V_{iH}$. In this range, it is possible to find the input voltage at a value that leaves the cross-coupled latches internal to a flip-flop in an intermediate state, with insufficient positive feedback to snap the output to either high or low valid states. System noise or quantum fluctuation will ultimately perturb such a precarious balance point and the output runs to one direction or the other, but it can take an arbitrarily long time for the synchronizer to reach a valid output state. As shown in Figure 9.39b, some flip-flop outputs may also tend to rise at least halfway toward the positive output level before deciding the input was really a zero. This glitch may trigger edge-sensitive circuits following the synchronizer.

Fortunately, the probability of an indeterminate latch output falls exponentially with the time T after sampling the possibility indeterminate input

$$P(t > T) = f_{clk} f_{in} \Delta e^{-T/\tau_{sw}} \tag{9.5}$$

where

f_{clk} is the synchronous sampling frequency
f_{in} is the frequency of external transitions to be synchronized
Δ is the time taken for the input voltage in transition to cross from V_{iL} to V_{iH} (or vice versa)
τ_{sw} is the time constant characterizing the bandwidth of the latch device

Having recognized the strict possibility of a metastable logic state resulting from synchronizer input, the designer can address the issue in several practical ways:

1. Use a high gain fast comparator to minimize Δ by minimizing the voltage range V_{iL} to V_{iH}.
2. Ensure or specify fast transition in external sensors or other devices to be sampled, if design control extends to them.
3. If there is no real-time penalty from an additional clock period of input response delay, the synchronizing latch should be followed by one or two more identical synchronizer latch stages, thereby increasing T in Equation 9.5 to reduce the chance of a metastable state being presented to internal circuitry to an acceptably low probability.

The effect of input metastability on the system also should be analyzed for its impact. If it is extremely crucial to avoid a metastability hazard, then phase-locking the external system to the system clock may be considered, or if the system and external timebases are free running but well characterized (e.g., in terms of a static frequency offset or known phase modulation), then the anticipated times of synchronization hazard may be calculated and avoided or otherwise resolved.

As a practical matter, the way in which design software handles metastability should be considered. Potentially metastable conditions should be flagged as a warning to the user, but not necessarily treated as a violation prohibited by the design environment. Some applications, particularly in VLSI for telecommunications, need design support for plesiochronous (near-synchronous), phase-modulated, or jittered signal environments. This implies test vector support to represent clocks interacting through logic at slightly different long-term or instantaneous free-running frequencies with design and simulation rules that permit the metastable conditions inherent as such clocks walk relative to one another. Circuit simulations must be allowed to continue with a random value resulting from the simulated ''synchronizer failure'' to be useful in such applications.

9.2.2.13 Controlled Introduction of Skew

Skew is not necessarily all bad. In fact, from the viewpoint of the system power supply, and power and grounded-related noise current surges, it is undesirable to have all logic transitions occurring exactly simultaneously. In a CMOS IC with 20 K register stages at 0.1 pF load each and a 1 ns clock rise time, 10 A of peak current can be drawn by the clock net. This can present a serious $L\,dI/dt$ problem through power and ground pins and can even lead to electromigration problems for the metallic clock lines. Chip clocking strategies should take this into account early in the design by seeking ways to deliberately stagger or slightly disperse the timing of some modules with respect to others. Also, the system clock waveform may not necessarily need the fastest possible rise time. Consistent with avoiding slow-clock problems, and controlling threshold-related skew in buffers, the clock edge should not be made faster than this as an end in itself. Excessively fast clock edges clock edges only aggravate power and ground noise problems as well as ringing and potentially causing electromagnetic radiation problems in the chip-to-chip interconnect. These principles motivate the widely used 10 K ECL logic family, which is based on the much-faster 100 K series with explicit measures to slow down the rise and fall times of the basic 100 K logic gates.

When considering random skew, it may or may not be beneficial to pursue skew reduction below a certain level in the design. In the case of a microprocessor or a design in which the fastest possible IC speed is always useful skew reduction does mean a performance improvement. In some other applications, however, the clock speed is set by the application. For instance, a VLSI circuit for a telecommunications MUX may be required to operate at a standard line rate of 45 MHz. In this case there may be no premium for a design that can perform the same functions at a higher speed. A working design with skew of 5–7 ns (out of a 22 ns clock period) may then be more desirable than a functionally equivalent design with 0.5 ns skew because dI/dT effects are eased by distributing the total switching current over time in the former. This principle may be important in practice as automated clock synthesis tools become more widely used and effective at achieving low skew, possibly creating unnecessary system-level noise and

EMI emission problems. Future clock-related CAD tools should possibly aim to disperse clock timing at various loads while satisfying a target worst-case skew, rather than absolutely minimizing skew.

Strictly speaking, skew can also be beneficial when allowed to build up in a controlled way in certain regular logic structures. For instance, by propagating clock in the opposite direction to data in a shift register, one enhances the effective setup time of the data transfer from register to register. In general, however, it is not feasible or advisable to try to design every clock path with a desired (nonsimultaneous) clocking time at each register, taking into account the individual logic paths of signals leading each clocked latch input. Especially when designing larger systems mediated by CAD tools for placement, routing, and delay estimation, the most practical and low-risk approach is to consider any deviations from a common nominal clock time as undesired skew. Indeed, for any one latch, timing margin may be enhanced by the actual skew that arises, but with thousands of logic paths, it is impossible to analyze the relative data and clock timing for each latch. Only one instance in which the skew works against the assumed timing margin is enough to fail a design. Therefore, the ''customized skew'' approach is recommended only for small and very high speed specialized circuit design.

9.2.2.14 Clock Signal Manipulation

As a matter of design discipline, some commercial ASIC and cell-based layout systems may prohibit a designer from directly gating or manipulating a clock signal. Any needed clock qualification is done through defined enable or reset inputs on register structures. As in software development, in which structured design disciplines have been developed, gating the clock may be riskier than its apparent efficiency warrants. In addition, clock gating within random logic designs can interfere with test pattern generation. The risk also exists of creating clock glitches or even logical lockups when clocked logic functions decode conditions that gap its own clock. On the other hand, in high performance and in large system-level designs, clock gating for power down and ''clock tuning'' may be unavoidable.

Wagner [30] discusses clock pulse-width manipulation, defining four canonical subcircuits that can be used to ''chop,'' ''shrink,'' or ''stretch'' the clock waveform for either delay tuning or duty cycle maintenance. The effect of these circuits on the positive pulse portion of a clock waveform is shown in Figure 9.40, where AND gates have delay d_a, OR gates have delay d_0, inverters have delay d_i and the delay elements have delay D. Aside from a single gate delay, the chopper and stretchers leave the rising edge unaltered and tune the trailing edge. These can be used to maintain a balanced clock duty cycle or to tune the nonoverlap period in two-phase clocking. The shrinker delays the rising edge of the clock as might be helpful to specifically delay clocking a latch or a flip-flop that is known to follow a particularly long logic path delay. This is not generally a preferred design approach, especially when manufacturing repeatability and temperature dependence of delay elements are considered.

By ''clock gating,'' we mean selectively removing or masking active phases or edges from the clock signal at one or more latches. One valid reason to gate the clock in CMOS is to reduce power consumption. Many circuit designs possess large modules or subsystems which it makes sense to stop cycling in certain application states. Gating the clock off is therefore the simplest form of power management, because CMOS has negligible power dissipation in a static state. However, even for this simple use of clock gating, the main issue is avoiding glitches when gating the clock.

Before gating any clock, the designer should see if gating can be avoided with an alternate design style. For example, if it is desired to hold the data on one register for a number of cycles while other registers on the same clock proceed, a preferred approach is to use a 2:1 MUX at the register input. Rather than gate the clock, the MUX is steered to select the register's own output for those cycles in which gating would have occurred. Ultimately, if it is appropriate to gate out one or more clock pulses, a recommended way of doing so in rising edge active logic is to OR out the undesired clock edges, decoding the clock gapping conditions on the same clock polarity as the one being qualified (see Figure 9.41). A natural tendency seems to be to AND out the gapped clock edge and/or to decode the gapping condition on CLK, but these approaches are more apt to generate clock line glitch than the OR-based approach. In the AND approach

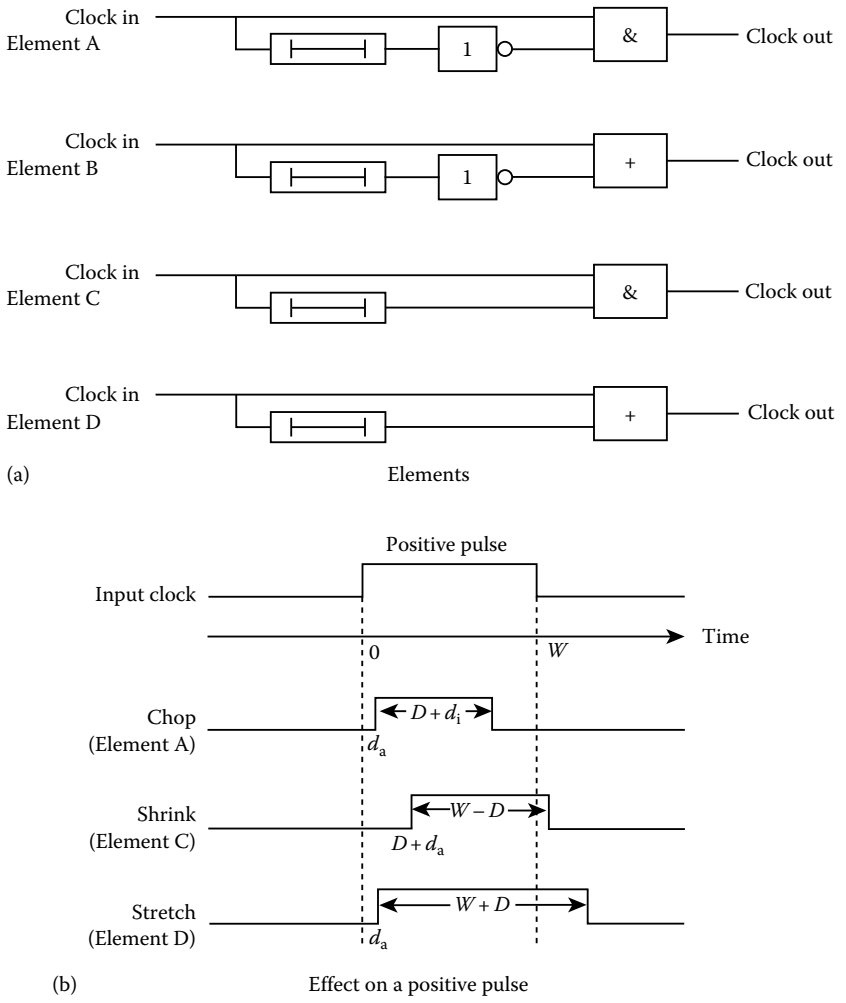


FIGURE 9.40 Standard circuits for chopping, stretching, and shrinking a clock waveform to adjust duty cycle or timing margins. (Adapted from Wagner, K.D., *A Survey of Clock Distribution Techniques in High Speed Computer Systems*, Report CRC 86-20, Stanford University Center for Reliable Computing, Stanford, CA, December, 1986, 15.)

the gating line returns high at the same time as the falling edge after the last-gapped active edge. In the case of minimum delay through the gapping logic the risk is that both AND inputs are momentarily above threshold.

9.2.2.15 Minimizing Delay Relative to an External Clock

In a large system, skew can build up between clock and data at the system level, even if the clock is skew-free everywhere within the ICs because data paths through ICs can build up delay relative to the system clock. For instance, if an ECL or TTL system clock line is distributed to a large CMOS IC, then the system clock must be level shifted and a large clock buffer may be required in each IC to drive its internal clock net. The delay through the on-chip clock interface and buffer can mean that even if the chip timing is internally skew-free, the on-chip clock is significantly delayed relative to the external system timing. Short of using the phase-lock methods described later, a simple technique to minimize this form of system-level skew is either to retime chip outputs with a separate copy of the system clock that has not

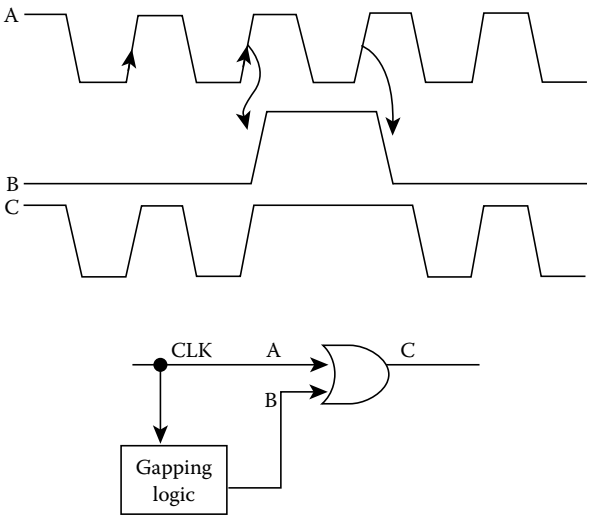


FIGURE 9.41 OR-ing out a clock edge when clock gating is essential.

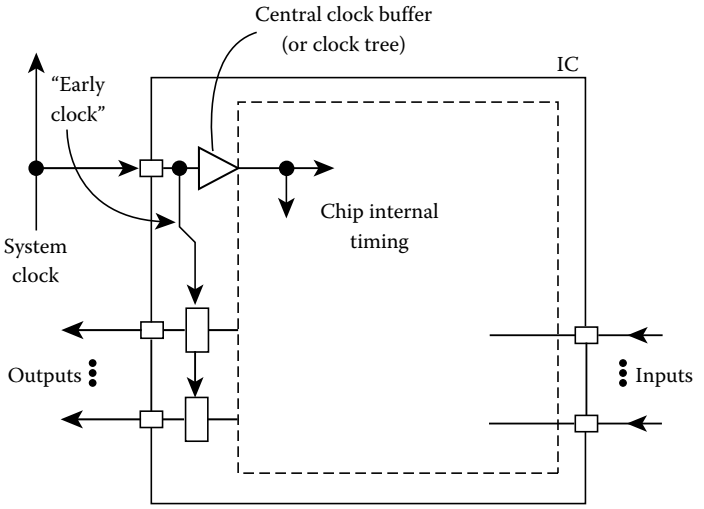


FIGURE 9.42 The "early clock" technique for reducing chip delay relative to external system timing.

gone through the internal clock buffer, or, if electrically compatible, to use the external system clock to directly retime the output signals from each IC (Figure 9.42). This is called the "early clock" concept. Note that this assumes an adequate timing margin exists in the final stage of internal logic to permit the relatively early sampling of logic states.

9.2.3 Clock Distribution Schemes

9.2.3.1 Single-Driver Configurations

Often a single on-chip clock buffer is the simplest and best approach to clock distribution. A single adequately sized clock buffer is typically located at the perimeter to drive the entire clock net of the chip,

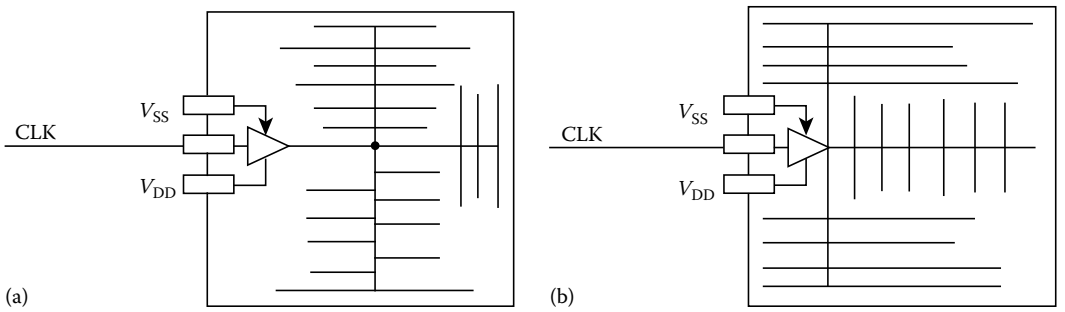


FIGURE 9.43 Single clock buffer placed in the I/O perimeter with dedicated power, ground pins (a) branching from a medial point on die (current density on line to medial point may be high) and (b) branching immediately (skew may be high).

as shown in Figure 9.43. This approach can also perform well in large systems if the clock is distributed in a low R_0C_0 routing layer, such as third-layer metal. The main advantage regarding skew is that no matter how the single-clock driver delays or otherwise responds to the external clock input, the output waveform is electrically common to all loads. No intervening buffers and no separate passive net segments develop skew. Moreover, if the global clock net comprises an electrically isochronic region ($NC_{gate} \gg l^2\,RC_{wire}$) in which clock loads are reasonably uniformly distributed, the clock net voltage rises virtually simultaneously at all points on the charging clock net, resulting in extremely low skew. Often this leads to lower skew than in a buffered clock fanout tree. There is also only one global clock wiring net, simplifying documentation.

On the other hand, skew can be larger and more routing dependent with a single buffer (than with some following schemes) when clock routing lengths vary significantly and wiring capacitance and resistance are significant. In such cases an isochronic net is not an accurate model, and performance depends on the way clocked loads are distributed on arms branching from the medial clock node. It is possible for neighboring flip-flops to be connected to the central driver via quite different path lengths, making prelayout simulation relatively uncertain and requiring considerable postlayout tuning. Another caution is that even if no actual skew is present because a single waveform is common to all loads, the rise time of the clock waveform may show considerable loading effects, so that threshold-dependent skew arises in the clocked loads. Finally, the potential for conducted switching noise problems is high with the single-buffer configuration because the entire clock net capacitance is charged in typically under 1 ns. Power supply decoupling and ground bound problems and even current density (electromigration limits) considerations may need to be given special attention if using this configuration. It is usually recommended that the single clock buffer be physically adjacent to the clock input pin, and the clock pin should be flanked by dedicated power and ground pins that feed only the clock driver. This principle, which applies in general to clock input buffers in all clocking schemes, keeps the clock switching noise out of the core power and ground bus lines. Also, the delay through a large clock buffer may be considerable, so a lightly loaded ''early clock'' can be picked off from the buffer input and used to retime chip output registers, or a PLL may advance the phase to the internal clock buffer to align internal and external timing regardless of the buffer delay.

An interesting central clock buffer design, which also has attributes of the clock trunk scheme which follows is reported in [9]. Here, the area for central clock driver fabrication is a strip across the center along one axis of the die. External clock is fed from one pin on the same axis as the buffer, and internal clock lines radiate systematically away from the central linear distributed buffer. Data flow is also highly structured and arranged to progress away from the central driver strip, further minimizing clock-to-data skew.

9.2.3.2 Four-Quadrant Clocking Approaches

In the quadrant-oriented approach, we may use up to four clock pads and four smaller clock drivers placed at the centers of the die edge, preferably also with dedicated power and ground from flanking pins. There may be one, two, or four external clock pins. Figure 9.44a shows a quadrant scheme tested in [24]. In Figure 9.44b, a single-pin quadrant-oriented scheme in 1 μm, two-layer metal CMOS achieved 0.6 ns skew among 400 registers. A four-pin quadrant approach was successfully used in [29] to develop a 90 MHz CMOS CPU. If more than one pin is used for clocking, pin count goes up but absolute delay through the clock buffers can be reduced. The maximum wiring RC delay on the internal clock net and the peak current and $L\,dI/dt$ effects through any one pin and bonding wire inductance may all be reduced in this case. Total loads on each of the four internal clock nets should be well balanced and/or the drivers sized for the actual number of loads in each quadrant. In many cases, reduction of each clocked area to one fourth of the die area can result in isochronic regions for which no further design attention other than total load balancing is required for clock routing within each region. The quadrant approach can also reduce the clock routing problem when the clock shares only two metallization layers available for all signal routing. Two other considerations apply to the quadrant schemes: (1) external tracking to multiple clock pins should be laid out with delay balancing in mind; (2) skew should be particularly considered on data paths which must cross from quadrant to quadrant, bridging timing across clock subnetworks.

9.2.3.3 Symmetric and Generalized Clock Buffer Trees

A symmetric or regular clock buffer tree (Figure 9.45a) has equal fanouts from buffers at the same level, equivalent (or isochronic) passive interconnect paths at each stage, identical buffer types at each level, and equal groups of loads at each leaf of the tree. This ideal can be approximated if loads are regularly distributed and layout and routing can be controlled to ensure equal interconnect delays from each buffer to its subtending buffers. The remaining skew will primarily be due to threshold variation in buffers and terminal loads.

A more general view of the clock buffer tree that arises with irregular layouts and routing is that the buffer tree has unequal interconnect delays and differing numbers of loads at each buffer. Figure 9.45b illustrates the electrical model of such a clock tree. (R and C values are all different.) The basic approach to skew control in such a generalized buffer tree is to size each buffer specifically for the individual loads it drives and the delay of the interconnect path to its input from the preceding level buffer. In practice, this means that generalized clock buffer trees may be the most handcrafted (and/or custom-tuned) of all designs, especially in a large system-level clock tree that extends down from a master clock source through multiple shelves, backplanes, connectors, circuit packs, and individual ICs. The system-level hierarchical clock tree design for the VAX 8800 is a good example described by Samaras [25]. Here, a two-phase clock was distributed to 20 large circuit packs over a 21 in. backplane with a global skew of 7.5 ns, for operation at about 50 MHz (37% skew).

Some basic methods and principles are, however, identifiable for designing generalized buffer trees so that it is not all just careful tuning and handcrafting:

1. Inverting buffers at each level will preserve clock duty cycle better than a tree of noninverting buffers.
2. Total delay (root to leaves) of the clock net is theoretically minimized when driving primarily capacitive loads, by a fanout ratio $e = 2.718\ldots$ at each level of the tree [23]. In practice, this implies a fanout ratio of about $n = 3$, with appropriately sized buffers, if delay is to be minimized. However, $n = 3$ can lead to relatively deep trees of many buffers in which skew can build up from threshold and time-constant variations as in Equation 9.4.
3. If identical buffers are used within each level, then the design aim is to make sure that the load of further buffers and/or interconnect RC load is delay-equivalent for each buffer. Dummy loads may be required to balance out portions of the clock tree whose total fanout is not needed. At the end-nodes of the tree equal numbers of standard loads should be grouped together on each leaf node, within a locally isochronic region.

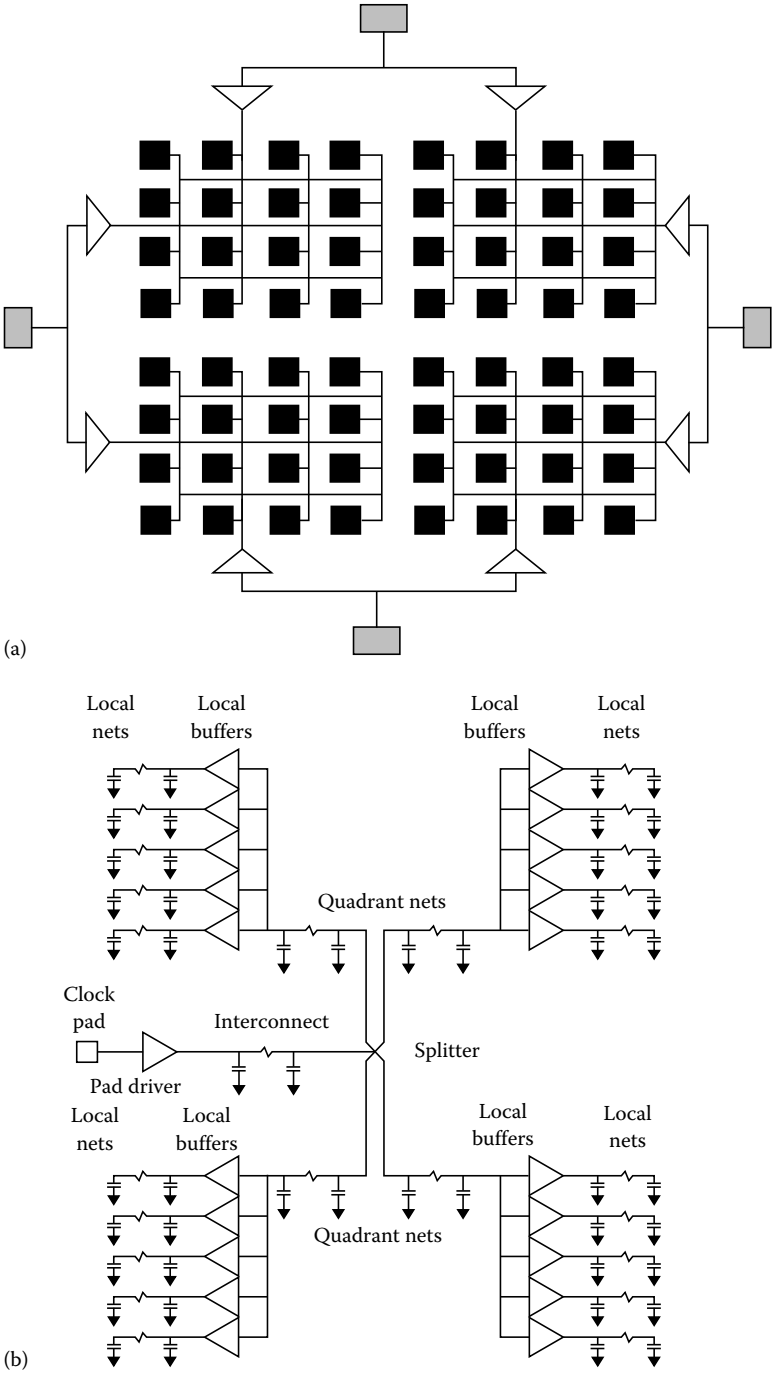


FIGURE 9.44 Quadrant-oriented clock distribution schemes: (a) 4 pins, 2 parallel buffers per quadrant. (From Nigam, N. and Keezer, D.C., A comparative study of clock distribution approaches for WSI, *Proc. IEEE 1993 Int. Conf. WSI*, 243–251, 1993.) (b) Single-pin quadrant scheme with 2 buffer levels. (From Boon, S., Butler, S., Byrne, R., Setering, B., Casalanda, M., and Scherf, A., High performance clock distribution for CMOS ASICS, *Proc. IEEE 1989 Custom Integrated Circuits Conf.*, 1989, 15.4.1–15.4.5.)

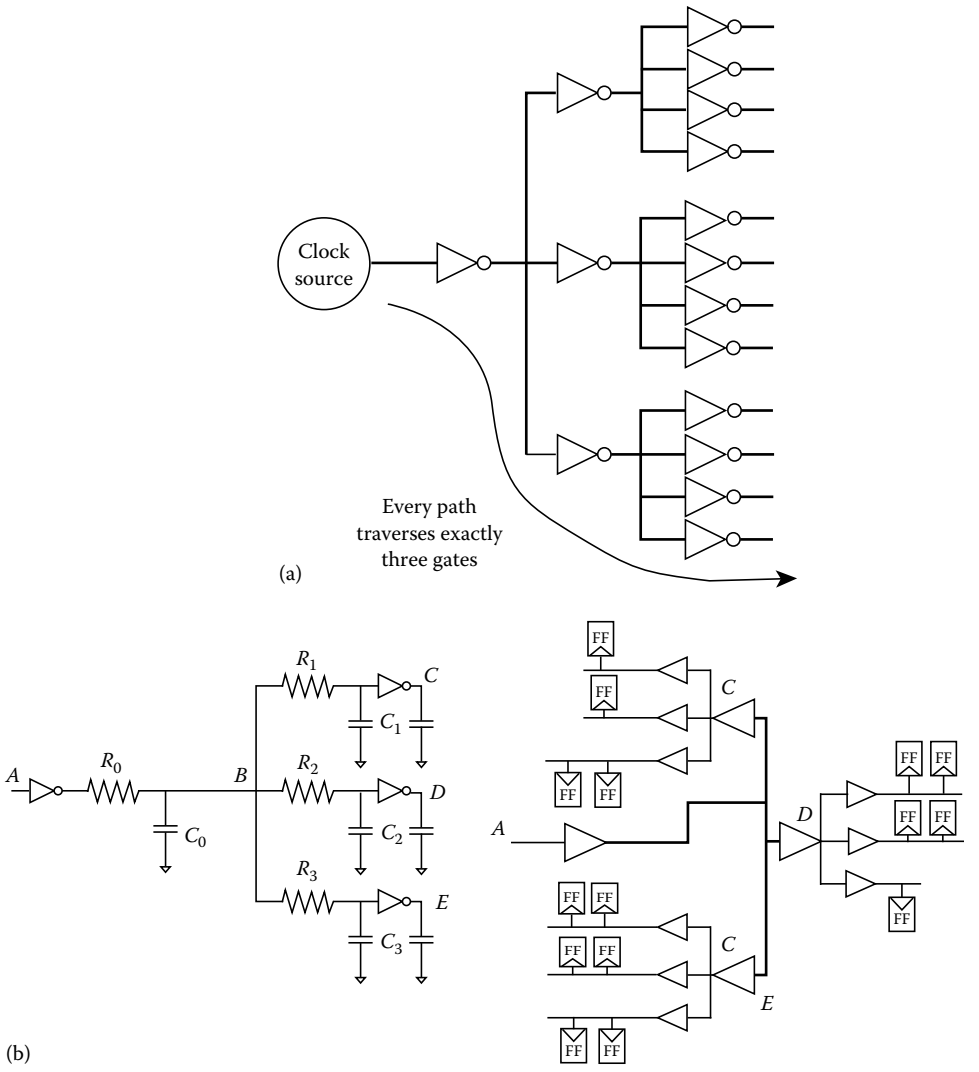


FIGURE 9.45 (a) Idealized symmetric buffer clock tree. (From Johnson, M.W. and Graham, M., *High-Speed Digital Design: A Handbook of Black Magic*, Prentice Hall, Englewood Cliffs, NJ, 1993, 348.) (b) Generalized clock buffer tree where interconnects and loads are not identical.

4. If the tree is deep, the skew among members of one local clock group at the bottom of the tree may be considerably smaller than the skew than the skew between groups of clocked loads at different end-nodes of the tree. If the inter- and intragroup skews are separately characterized, however, logic path delays can be designed to take advantage of the low intragroup skew and to allow greater timing margin on intergroup paths [25].
5. The choice of clock buffer type for use in any clock tree should take into account the effects of power supply sensitivity. For example, the "bootstrapped clock buffer" of Figure 9.46a can provide a very sharp rise time, although with relatively high delay through the buffer. Sharp rise times minimize skew due to switching threshold variations in the following buffers or clocked loads. The output switching time of the bootstrapped clock buffer is, however, relatively sensitive to supply

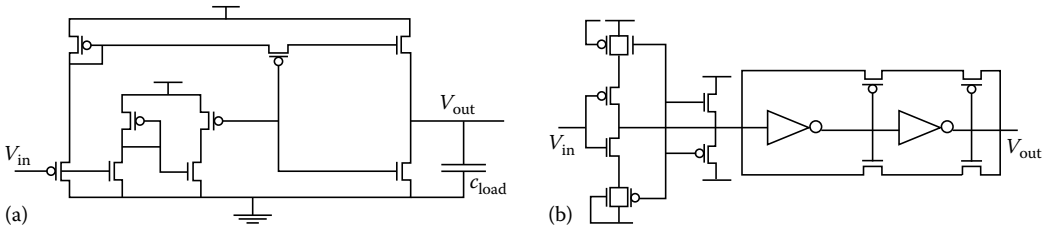


FIGURE 9.46 (a) Bootstrapped clock buffer for use in first level of a clock tree; and (b) phase-correcting clock buffer for use deeper in clock tree. (From Fried, J., *Proceeding of the IFIP Workshop on Wafer Scale Integration*, G. Saucier and J. Trilhe, Eds., North-Holland, Amsterdam, 1986, 127–141.)

voltage. On the other hand, the ''phase-correcting buffer'' of Figure 9.46b is very tolerant to supply variations, but is not as fast in its output rise time. This leads to a mixed buffer strategy in which the bootstrapped buffer is used in the relatively small population of buffers in the first few stages of a clock tree. Here, special attention can be paid to ensuring well-equalized power voltages. The phase-correcting buffer is more appropriate in later stages, nearer the more numerous individual loads, among which on-chip or wafer supply levels may exhibit more IR voltage drop variation.

Algorithms for the generalized clock tree design problem are also emerging. The algorithm in [28] can generate a buffer design based on custom transistor sizing to drive each heterogeneous load, all with the same nominal delay. A still more general optimization approach is under development by Cirit [5]. The sizes of all buffers in the tree are jointly optimized with respect to unequal interconnect RC totals and unequal loads that each drives, as in Figure 9.45b. Thus, the minimum ''total'' tree delay is found for which all paths from root to leaf of the tree also have ''equal'' delay. The procedure is intended to be iterated along with placement and routing alternatives until a set of feasible buffer sizes and acceptable total delay is found.

9.2.3.4 Clock Trunk Schemes

The clock trunk concept is gaining popularity and is now supported within several CAD systems for CMOS processes with two or more metallization layers. Three variants of clock trunk structures are shown in Figure 9.47. An input clock signal is buffered (its input pad is at the center of one side of the chip edge) and is routed either to the midpoint, or one or both ends of the internal clock ''trunk.'' The trunk itself is a metal line specially widened for low resistance, thereby making delay and the $R_0[C_{load} + C_0]$ rise time particularly small on the trunk portion of the overall clock net. As long as the lumped capacitive loads (C_{load}) dominate the trunk's C_0, the time constant of the trunk drops as it is widened. C_0 can be kept particularly low by forming the trunk in third-layer metal. The idea is to size and drive the clock trunk so that an isochronic region is created down the middle of the die, reducing all remaining clock path distances to not more than half the die diameter and setting up a situation in which the final distribution of clock from the isochronic trunk to the loads is in line with one routing axis. This means that branch routing from the trunk to loads can be exclusively contained within one metal layer and travel in a shortest direct line from trunk to load. This is highly desirable for processes in which all horizontal and vertical routing are dedicated to one metal layer or the other. Overall layout and routing is simplified and clock paths are predictable and uniform in routing and wiring type. If, however, the total fanout is very small, the wide metal trunk may add more to trunk capacitance than it decreases R_0 in the $R_0[C_{load} + C_0]$ product for the trunk. This is undesirable because for an isochronic trunk, we want C on the trunk to be dominated by its loads, not by the distributed capacitance of the trunk. In practice, therefore, the trunk scheme is typically recommended for fanouts of over 50. Below this, a single buffer scheme is recommended.

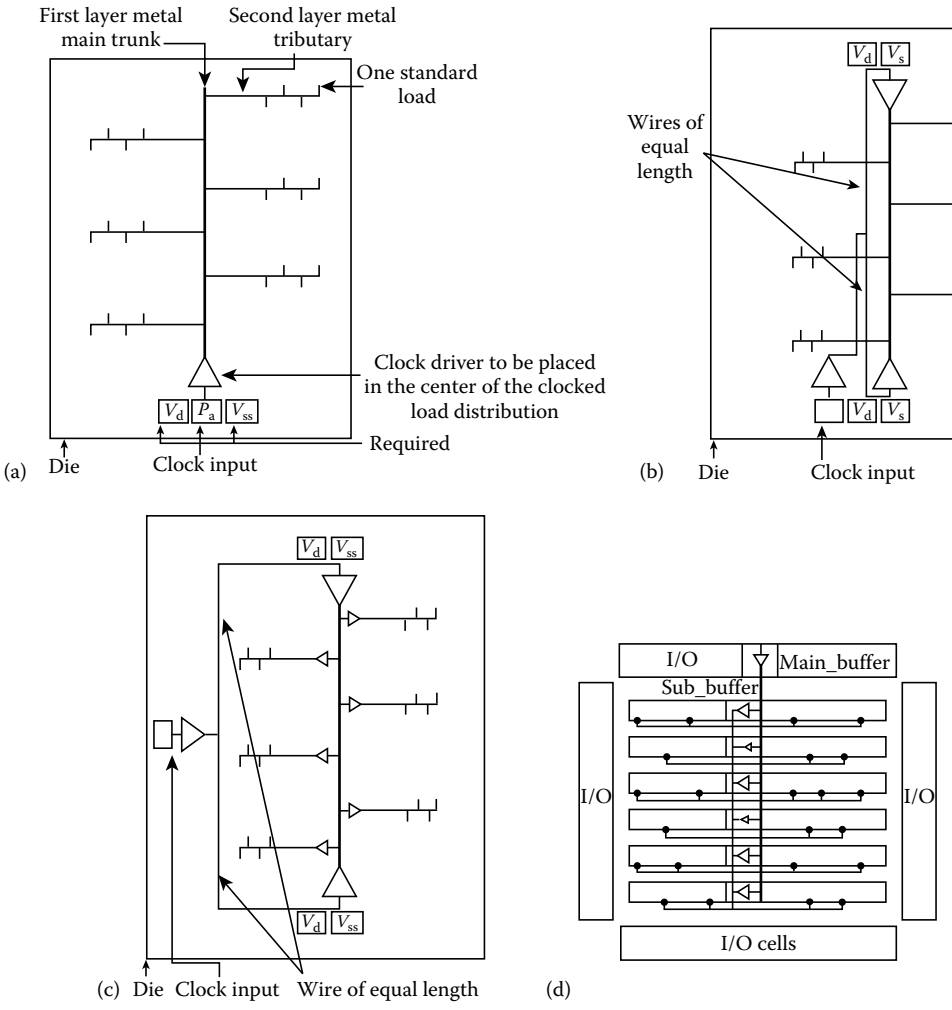


FIGURE 9.47 The clock trunk concept: (a) single-ended unbuffered clock trunk, (b) double-ended unbuffered, (c) buffered clock trunk, (From LSI Logic Corp., *Clock Scheme for One Micron Technologies*, Rev. 1.1, LSI Logic Application Note, Aug. 1992.) and (d) clock trunk with shorted branch buffer outputs. (From Saigo, T., Watanabe, S., Ichikawa, Y., Takayama, S., Umetsu, T., Mima, K., Yamamoto, T., Santos, J., and Buurma, J., *Proc. IEEE 1990 Custom Integrated Circuits Conf.*, 1990, 16.4.1–16.4.4.)

By designing with the following principles for single-ended, double-ended, and buffered clock trunks, clock nets of up to 2000 fanouts can achieve <1.5 ns skew in 1.0 or 0.7 μm CMOS gate array technology [21]. When the clock fanout is between 50 and 500 unit loads, a single-ended trunk scheme, as shown in Figure 9.47a provides a good trade-off among skew, area, and delay. A single clock driver input buffer is used to drive the trunk line, which is typically realized by six first-layer metal lines in parallel with metal filled in. Clock trunk sizing for a given fanout must set a minimum width to take current density limits into account, given that all of the clock net current flows through the trunk if the tributaries are not buffered. Tributaries of nominally constant fanout branch out in second-layer metal. To the extent possible, macrocells and hard-coded megacells should be laid out with the clock trunk in mind, ideally permitting the clock trunk to be located in the middle of the logic to which it fans out.

The tributary branches may or may not be buffered with their own clock drivers, depending on the total number of loads to be driven. Buffering primarily has the effect of reducing overall delay, rise time, and total current density in the trunk, by allowing a smaller trunk driver. Form a purely skew-oriented view, however, it is better not to have the secondary drivers as long as the trunk is isochronic and the size and delay of the main driver is acceptable. When using local buffering, it is important that the branch loads on all tributaries be balanced, more so than when using an unbuffered trunk. Local buffering is, therefore, primarily a way of distributing the total buffering load so that no one buffer needs to be extremely large.

For layout software simplicity, the main trunk may be constrained to use vertical (or horizontal) routing channels only. The main trunk (or each of possibly several main trunks) should be placed as close as possible to the centroid of area of all the loads which it drives. Layout or floorplanning software for commercial ASIC design can typically assist the designer in clock trunk placement by visualizing the spatial distribution of clock loads.

When a design has between 500 and 2000 clock fanouts, a double-ended clock trunk, as in Figure 9.47b or c is recommended. The double-ended clock drivers are internal buffers which use the I/O slots of two pins that will not be used externally thereafter. Both single-ended and double-ended clock trunk schemes use only one external clock input pin, with adjacent pins providing an AC ground and switching noise isolation by powering the drivers with dedicated V_{ss} and V_{DD} pins for the clock buffers. In general, clock input pins should always be surrounded by nondriven pins to minimize the possibility of cross talk coupling into the clock waveform. Clock pins also should be chosen so that minimal internal routing is required between the predriver associated with the input pad and the clock trunk drivers. As a clock trunk design is laid out, the spike current draw from V_{ss} due to simultaneous switching of large fanouts on the clock net should be assessed and considered in determining how many V_{ss} pins are needed in a particular design.

In the double-ended trunk scheme, some care must be taken to ensure that an equal-length path can be routed from the midpoint of the trunk, where the clock input branches to the trunk drivers at both ends of the clock trunk, and that a direct routing from the side of the die to the branch point at the center of the die is feasible. Particularly when preconfigured macrocell function have been placed, the metallization layer needed to bring the clock predriver into the middle of the die may be blocked. This leads to the recommendation that the clock input pin be placed on the side of the die that is in line with the clock trunk (see Figure. 9.47c). The line to the branch point and the two branch lines can then use the same metallization layer as the clock trunk and can be automatically provided for as part of the routing channel width reserved by the clock trunk layout software.

The two double-ended arrangements will have a basic skew given by the (nontrunk) R_0C_0 delay across one half of the chip's dimension (typically under 300 ps), plus skew due to any imbalance in buffer loads and thresholds. An advantage of the buffered clock trunk is that the capacitive load of the clock tree is distributed somewhat in both time and position across multiple buffer stages, reducing the current spikes occurring during a clock edge and their impact on ground bounce and injected power supply noise. On the other hand, a total of three or four buffer stages associated with this structure (for low skew in large applications) may cause high delay between the clock edge used to latch incoming and outgoing data on chip and the external system clock. Clock net fanouts achievable using the clock trunk scheme in commercial gate arrays are summarized in Table 9.10.

In even larger dies with famous of over 3 K flip-flops, multiple symmetrically driven double-ended clock trunks can be established to control the maximum distance of any point from a clock trunk. For example, with two trunks placed one fourth of the die width in from the sides, no load is over one fourth the die diameter from a trunk, and the loading of branch buffers is half that required with one trunk. Branch lines from different trunks should not be connected together where they meet in the middle of the die. These points are far enough away in terms of delay from their common driving points that joining them could cause power-wasting buffer output fights. In a further variant on the buffered clock trunk, buffers have their outputs ganged (i.e., shorted) by an additional vertical metal line parallel to the trunk,

TABLE 9.10 Fanouts of Clock Trunk Schemes in 1 μm CMOS Gate

Clock Frequency (MHz)	Single-Ended Trunk	Double-Ended Trunk	Double-Ended Trunk with Local Buffering
50	500	2000	>2000
60	450	1500	>1500
70	400	1200	>1200

Source: LSI Logic Corp., *Clock Scheme for One Micron Technologies*, Rev. 1.1, LSI Logic Application Note, Aug. 1992.

Note: Arrays (maximum number of clock loads driven with <1.5 ns skew).

close to the buffer outputs. The effect of shorting the branch buffers is to equalize the propagation delay through the trunk and distribute the capacitance per buffer more uniformly. This has been found to reduce skew considerably if the branch buffers were not equally loaded and also reduced skew (although less so) in the balanced buffer case [26].

9.2.3.5 Clock Ring Configuration

The clock ring approach shown in Figure 9.48 combines aspects of the clock trunk, quadrant, and the single large buffer approaches to achieve a combination of moderately low skew, and moderately low delay without the possibly high routing-dependent skew of the pure single-buffer scheme. The ring approach also simplifies overall clock and signal routing conflicts in a two-layer metal process. The external clock is buffered at entry with a moderate- to large-scale buffer, which drives a clock ring that follows the die perimeter. The ring is not a widened trunk because typically less than 50 other buffers are driven off the ring, not the entire clock net. Therefore, with relatively low C_{load} on the ring, it is not widened. The extra capacitance of a widened ring would be relatively high (on a square die, the total ring length will be four times the length of a corresponding central trunk) and only increase skew and delay. The ring drives secondary buffers sized to drive balanced groups of flip-flops in the core of the chip. The aim of the ring on a large die is not to create a wholly isochronic perimeter, although this could be approached by driving the ring at multiple symmetric locations. Rather, the ring establishes a relatively low-skew reference perimeter from which any interior clock load can be reached either by a purely

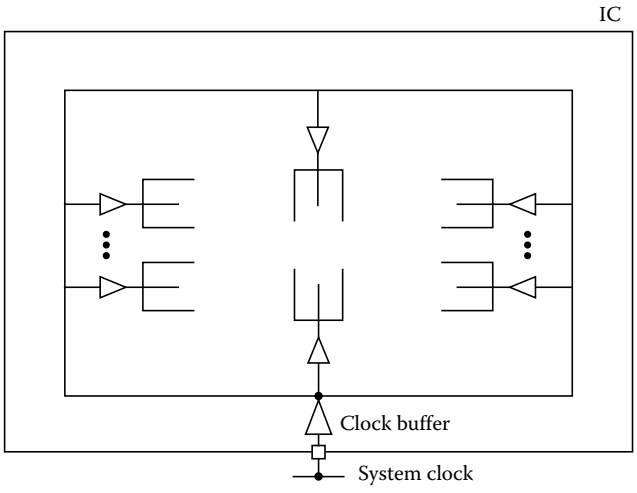


FIGURE 9.48 The clock ring concept.

vertical path or purely horizontal path (i.e., in a direct line of a single metal layer) no longer than one half the die size. The worst-case routing distance from the ring is the same as in the clock trunk scheme, but two thirds of all locations are within half that distance in the ring, whereas only one-half of all uniformly distributed loads are within half of the maximum distance of a central trunk. In practice, with good load balance on the secondary drivers, clock skew of 0.8–1.0 ns has been obtained in a designs of up to 30 K gates. If the secondary drivers are well balanced, skew in this architecture will depend primarily on the R_0C_0 delay from the ring driving point to its far side, around the periphery, typically 0.8 ns for a die of 350–400 mil. Relatively low chip delay is obtained by using the clock signal on the ring as an ''early clock'' with which to time I/O latches. The ring is electrically closed as this helps distribute the subbuffer capacitance and equalize delays, especially if driven at two opposing points.

9.2.3.6 H-Trees

The H-Tree is an area-efficient regular structure most suited to clock distribution in systems in which the synchronized modules are identical in size and placed in regular array. Figure 9.49 illustrates a 256 module H-tree tested on a 4 in. wafer by Keezer and Jain [17]. The scheme balances the R_0C_0 delay through the clock network by geometric symmetry so that the delay is nominally constant from the root to any leaf node. Loads are clocked only at leaf nodes of the tree. The minimum feature size of the process can be assumed to set the line width of the H-tree at its leaf nodes and each preceding level has progressively wider lines to maintain constant current density and to minimize impedance mismatch effects (at wafer scale and above) when no branching buffers exist. The H-tree is driven by a buffer at its root and may or may not have additional buffers at branching points. ASIC manufacturers have been able to achieve skew below 500 ps at fanouts of >5000 with experimental H-tree layouts in third-layer metal [22].

The H-tree approach is most practical only if an entire layer in a multilayer PCB or a third or fourth metallization layer in CMOS can be dedicated for H-tree clock distribution. Otherwise, the H-tree may encounter (or cause) a large number of routing blockages or require poly links which will disrupt

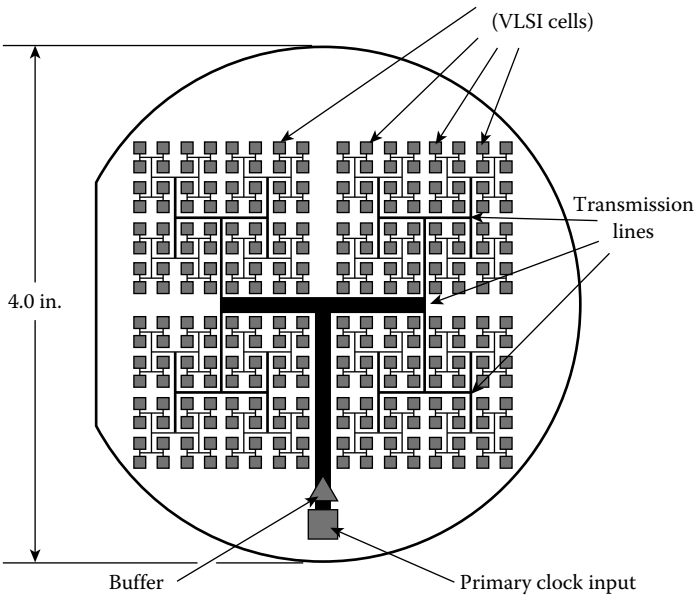


FIGURE 9.49 The H-tree concept illustrated in the form of a 256-cell passive H-tree for wafer-scale integration. (From Keezer, D.C. and Jain, V.K., *IEEE Int. Conf. WSI*, 1992, 168–175.)

the H-tree performance. In addition, many VLSI designs include memory cells or other hardcoded cells that are incompatible with the ideal symmetry of the H-tree. However, if a suitable layer is available for H-tree layout, it may be applied to random-logic designs by considering each leaf of the node as a clock supply point for all clocked loads within an isochronic region around each leaf node. The whole die is then tiled with overlapping isochronic regions, each fed from out of the plane by a leaf of the overlying H-tree. Each leaf of the H-tree might also use a variably sized buffer to drive the particular number of clocked loads of the random logic layout that fall in its particular leaf node zone.

Kung and Gal-Ezer [18] have given an expression for the time constant of an H-tree, which characterizes the total delay from root node to leaf:

$$\tau_H = 1.43\ N^3\left(3 - \frac{2}{N}\right)R_0C_0 \tag{9.6}$$

where an $N \times N$ array of leaf nodes is drives. Absolute delay rises as N^3 for large N. This in itself does not limit the clocking speed because, at least theoretically, more than one clock pulse could be propagating toward the leaf nodes within the H-tree. As a practical matter, however, the delay of the H-tree expressed in Equation 9.6 is essentially a rise time effect on the clock waveform. A slow rising edge out of the H-tree can lead to significant skew due to threshold variations in the leaf node buffers. These considerations apply to the on-chip context in which the H-tree clock network is dominated by RC diffusion delay effects. Equation 9.6 also describes an unbuffered H-tree. By placing buffers at selected levels of the overall tree, total propagation delay through the tree will increase, but the bandwidth of the tree may be preserved by effectively reducing N to the portions of the overall tree between buffers, in Equation 9.6. In contrast to the rapid bandwidth fall-off on-chip, at the multi-chip module level of system integration an H-tree may be designed from an impedance-controlled transmission line standpoint to obtain very high clock bandwidth. In experiments of this type, Bakoglu [3] has achieved 30 ps of skew at 2 GHz with a 16 leaf H-tree covering a 15×15 cm wafer area.

In an H-tree the total clock path length doubles each time one moves up two levels from the leaf nodes toward the root. Based on this, Kugelmass and Steiglitz [19] have shown that given σ_b and σ_w as the standard deviation of buffer delay (if present) and wire delay, respectively, the total delay of an H-tree considering buffers and wires has variance:

$$\sigma^2 = \sigma_b^2 \log_2(N) = \sigma_w^2 2\left(\sqrt{N} - 1\right) \tag{9.7}$$

and that the average case skew between any two leaf nodes is bounded by

$$E[\text{skew}] = \sigma_w 4\sqrt{\sqrt{(N-1)}\ln(N)} \tag{9.8}$$

where N is large and wire length effects dominate. Average case (or expected) skew is the maximum difference between clock times, averaged over many design trials, not the average clock time difference in any one design. Kugelmass and Steiglitz [19] also give results for the probability that a sample value of the skew exceeds the mean skew by a given factor in either an H-tree or a binary tree, based on assumptions that all wire length and buffer delay variables are independent and identically distributed:

$$P(\text{skew} > E[\text{skew}] + a) \leq \left[1 + \left(\frac{a}{E[\text{skew}]}\right)^2 48\ (\ln N)^2/\pi^2\right]^{-1} \tag{9.9}$$

where a is the amount of time by which the mean skew is exceeded. These expressions may be used to estimate skew-limited production yield at a given target clock speed.

TABLE 9.11 Comparative Performance of Clock Distribution Networks (8 × 8 Array of Loads Clocked at 31 MHz and Constant Total Power, 650 mW)

Scheme	Delay (ns)	Skew (ns)	Rise/Fall Time (ns)
3-level symmetric buffer tree	7	3	12.5
Single buffer H-tree	15	∼0.0	38
Clock trunk with branch buffers	13	4	14.2
Clock trunk with ganged branch buffers	14.2	2	16
4-pin-quadrant scheme, 2 buffers per quadrant	4.3	1.3	9

Source: Nigam, N. and Keezer, D.C., A comparative study of clock distribution approaches for WSI, *Proc. IEEE* 1993 *Int. Conf. WSI*, 1993, 243–251.

9.2.3.7 Delay, Skew, and Rise Time Comparison

The five clock distribution schemes described thus far were studied in a unified, experimental way by Nigam and Keezer [24] using HSPICE simulations. They compared each scheme on a 5 in. wafer holding an 8 × 8 grid of modules to be clocked. Each module presented at a total load of 2 pF and the interconnect R and C values were taken for a typical 2 μm double-metal CMOS process. Clock distribution lines were 10 μm wide for the buffer tree and the H-tree, except for its trunk, which was 40 μm. The clock trunk schemes used a 20 μm trunk width. All interconnect was modeled as distributed RC, with transmission line delay effects included. The results are tabulated in Table 9.11 and give an excellent overview of the relative characteristics of each method. The H-tree has essentially no skew, but has the highest delay and slowest clock edge, which can translate into skew due to threshold variations in the loads. The clock trunk has good skew and moderate delay. The best overall performance is achieved by the four-quadrant scheme, essentially by virtue of reducing the clocking area to one fourth of the overall size of the other clock networks.

9.2.3.8 Balanced Binary Trees

A balanced binary tree (BBT) is an unbuffered clock distribution tree in which each branch node has exactly two subtending nodes, and the delay from the root to all leaf nodes is made constant by placing branch points at the "balance point" of the two subtending trees from any node. BBTs are not simply clock buffer trees with fanouts of two. The significance of the BBT is that constant delay is achieved through multiple levels, without any buffers, and the BBT can be constructed by a fairly simple algorithm. Passive BBTs also may be used in practice to implement delay-equalized fanout networks between the active buffer levels of a larger buffered clock tree. The BBT concept should not be confused with the buffered clock tree concept in general, however. The key is that the generalized buffer clock tree does not have path delay equivalence if its buffers are removed, whereas the BBT has this property.

The basic ideas and methods of generalized balanced tree synthesis are explained in [6]. The clock tree that results has two branches at every node. Clocked loads appear only at the leaves of the tree. The line lengths at each level can be different than those at other levels of the tree, and the two line segments that are children of any node also can be of unequal lengths. The key, however, is that at each branch the total distance from the branch point to any subtending leaf via one outgoing path is equal to that in the other outgoing direction.

Figure 9.50 illustrates the basic procedure for BBT synthesis. The process works from the bottom up, by considering all leaf node positions, i.e., clock entry points for modules or isochronic local regions. Leaf nodes are subjected to a generalized matching or pairing process in which each node is paired with the other node closest to it in the Manhattan street length sense within the available routing channels. A first-level branch point is then defined at the midpoint on each line joining the paired leaf nodes. A similar pairing of the first-level branch points then defines a new set of line segments, each of which has two leaf nodes symmetrically balanced at its ends. Second-level branch are then defined at the RC balance point

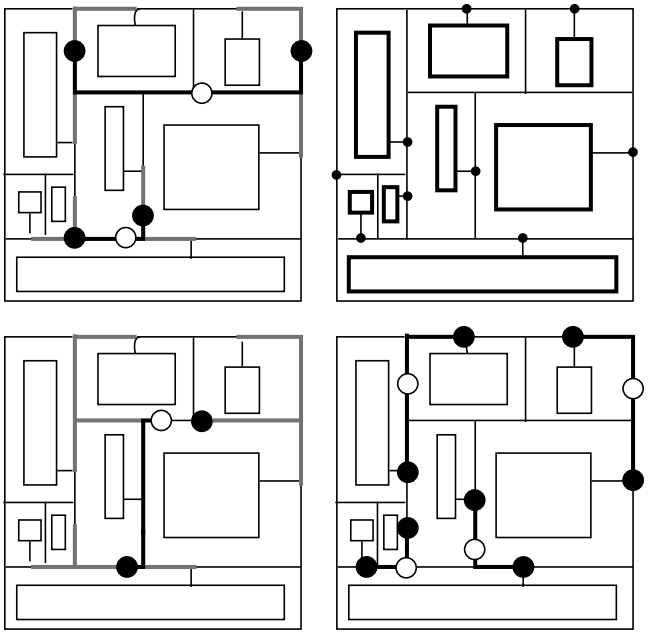


FIGURE 9.50 BBT synthesis in an eight-terminal net. Solid dots are roots of subtrees in the previous level; hollow dots are roots of new subtrees computed at the current level. (From Cong, J., Kahng, A., and Robins, G., *Proc. 4th IEEE Int. ASIC Conf.*, Sep. 1991, 14.5.1–14.5.4.)

on each line joining first-level branch points, and so on. Each iteration consists of finding the minimum path length matching of pairs of branch points from the previous iteration and defining the next set of branch points at the time-constant balance points on the line between just-matched, lower-level branch points. Assuming leaf nodes present equal loads, the first-level branch points are the midpoints of the pair-matching line segments. After that, the balance point for level i is the point on the line segment joining matched level $(i-1)$ branch points at which the total RC wiring time constants (simply total length if all wires and layers are identical) to the leaf nodes in the right and left directions are equal. This is repeated until only two branch points remain to be matched. A line is routed between them and driven by the clock signal at the final balance point, which defines the root of the BBT.

Clock trees developed in this way are in one sense highly structured and symmetric in that the total delay from root to any leaf is nominally constant, like the H-tree. Unlike the H-tree, however, the actual layout is irregular, allowing the BBT to accommodate the actual placement of cells and modules and to cope with the limited and irregular routing channels available in designs that do not use a completely regular layout.

Skew in BBTs has been considered theoretically and experimentally. Kugelmass and Steiglitz [19] showed that in a BBT with independent variations in the delay at each stage of the tree, with σ_0^2 variance, the expected skew is fairly tightly bounded by

$$E[\text{skew}] \leq \frac{4\sigma_0}{\sqrt{2\ln 2}} \ln N \tag{9.10}$$

where N is the number of leaf nodes of the tree.

Using the previous expressions we can compare the H-tree to a BBT. The comparison shows that when the regular structure of an H-tree is feasible, it is of relative merit for large fanouts because the expected skew grows more slowly ($O(N^{1/4}(\ln N)^{1/2})$) than the BBT tree in which expected skew grows as $O(\ln N)$.

For comparison, assuming 10,000 leaf nodes and the same σ_w per unit wiring length, the expected skew of the H-tree is about one half that expected of the BBT. This outcome is primarily because the BBT must be deeper (have more levels) than the H-tree for the same fanout.

Experimentally, Cong et al. [6] produced a large sample trials of 16- and 32-node BBT clock trees, synthesized on a 1000×1000 grid. It was shown that the BBT resulted in less than 2% of the skew from a corresponding minimum spanning tree (MST) for clock distribution to the same loads, even though the BBT had 24% to 77% more total routing length than the MST. The MST benchmark characterizes the skew that would typically result if the clock was routed as an ordinary signal net, with no special concern about skew.

An example of balanced clock tree synthesis supported by a gate array provider is [22], in combination with a three-level clock buffer tree hierarchy. Skew of $<$500 ps is achieved in 0.5 μm designs of up to 13,440 clocked loads. By using appropriately sized buffers and wire width at each level of the balanced tree, clock rise time is typically 0.8–0.9 ns at the terminal nodes. The clock tree compiler is invoked after floorplanning. The compiler takes into account the resistance and capacitance of different wire types, the length and width of wires, and the input capacitance of clock pins and buffers. Up to three active buffering levels can be used, with fanouts of up to 64, 14, and 15, respectively, from buffers at each level. The fanout subnet driven by each buffer is laid out as a (passive) BBT, so that the leaves of one balanced tree are the buffers that act as the roots of further passive binary balanced subtrees driven by the next buffering level. At the lowest level, local buffers each drive up to 15 loads via a final four-stage passive BBT.

9.2.3.9 Clocking Schemes Involving Phase-Locked Loops

A PLL is a negative-feedback control system in which the phase (and, implicitly, frequency) of a voltage-controlled oscillator (VCO) or phase shifter is brought into alignment, or to a predefined static phase offset, with respect to the phase of a periodic reference signal. The application of PLLs is most often to control skew and clock delay problems primarily at the multichip and interboard levels of system design.

Figure 9.51a shows how a PLL can be used to lock the on-chip clock phase at a selected point on-chip, to an external phase reference. Figure 9.51b shows the mid-trunk phase on a single-ended clock trunk being made to match the external phase reference. Here, the feedback line from the middle of the trunk to the PLL input is assumed to have negligible delay in itself because it is a metal line with only one standard load. Similarly, in the double-ended clock trunk scheme, the sense line can be connected one fourth of the way along the clock trunk. This will lock the internal system clock to the reference timing at two points on the clock trunk, as shown in Figure 9.51c, reducing overall skew to one fourth of that in the single-ended clock trunk scheme. The phase-sense line needs to be connected to the clock trunk at only one point because, by symmetry, the corresponding point from the other driver is similarly phase locked.

In general, when the phase-sense line has negligible delay, the clock phase at the sense point is driven into lock with the reference phase. Thus, in generating clock signals we can null out the delays of large buffers or drivers in the output circuits as well as their process and temperature-dependent variations, and, in general, coordinate clock and data phases at the inputs to another chip at any remote point by bringing the phase-sense line back from the actual point where the phase-controlled relationship is desired. In this way even the delay of an off-chip driver can be canceled out by including it within the PLL feedback loop. If delay in the phase-sensing feedback path is not negligible, then its effect is to advance the phase at the desired control point. The feedback delay can be compensated by a matching delay in the forward path from the VCO to the phase-sensing point, or at the PLL input. An inverter in the feedback signal path is also a convenient way to cause a 180° phase shift between referenced and VCO without requiring any loop delay.

With the addition of frequency divider (divide by N) in the feedback path, as in Figure 9.51d, the VCO operates at N times the frequency of the reference clock input. For $N = 2^n$ frequency multiplication, the feedback divider can be a simple ripple counter of n toggle flip-flop stages. For other multipliers, a synchronous counter is usually used. The delay-matching element at the PLL phase detector input compensates for delay from the feedback path divider. On-chip frequency multiplication can ease a

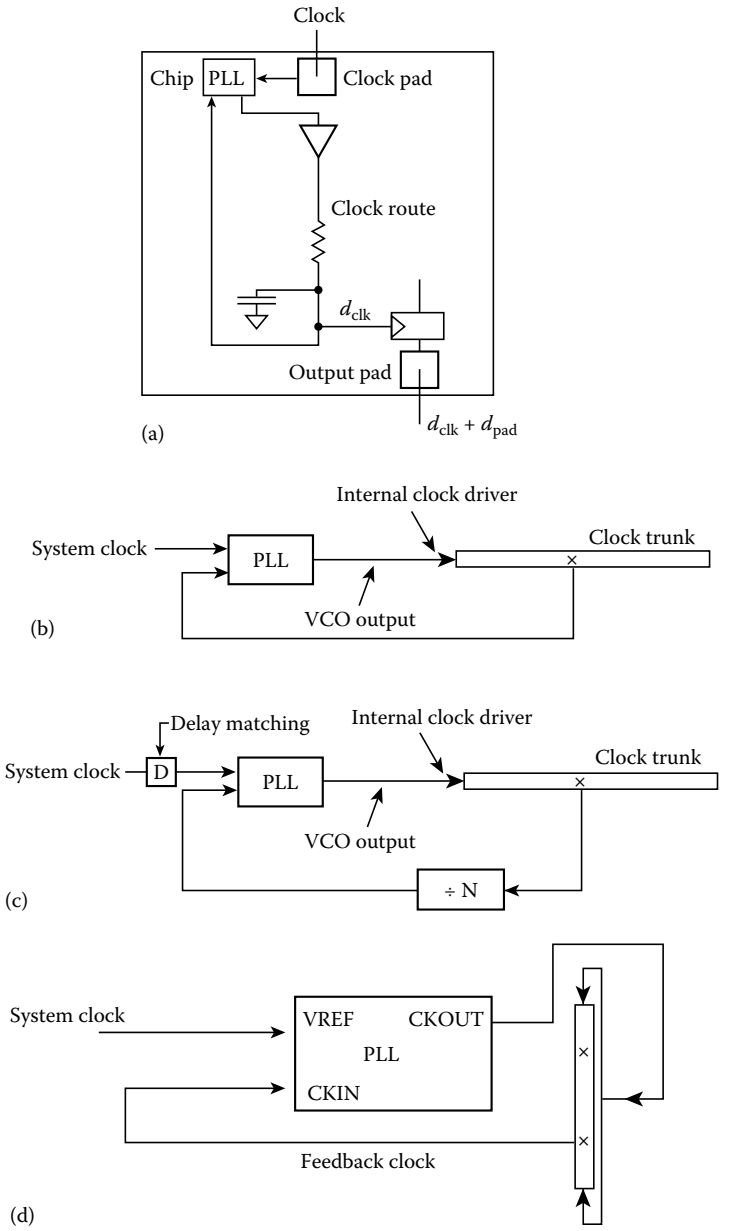


FIGURE 9.51 PLLs for skew and delay control: (a) canceling internal and clock net delay. (From Weste, N. and Eshraghian, K., *Principles of CMOS VLSI Design*, Addison-Wesley, Reading, MA, 1993, 335.) (b) Halving the skew in a single-ended clock trunk, (c) reducing double-ended clock trunk skew to one-fourth of the single-ended trunk, (d) one-chip frequency multiplication. (From LSI Logic Corp., *Phase-Locked Loop Application Note*, LSI Logic Application Note, Nov. 1991.)

number of system-level design problems. The overall system clock rate need not be equal to that of the fastest chip in the system. Transmission line effects across the relatively long distances of PCBs or backplanes can be reduced by operating at a lower clock frequency outside of the system ICs. The lower frequency of system reference distribution may also reduce power, and usually assists in meeting radiated

emission specifications for electronic equipment. Because a PLL regenerates the clock in each IC, a considerable amount of clock edge slew rate control can be used on the external system clock, further easing EMI and power supply switching noise problems. The difficulty of retaining clock waveform integrity getting on- and off-chip at high frequencies through inductive packaging and bonding leads is also eased for the same reason.

9.2.3.10 PLLs for CMOS

A block diagram of a PLL is shown in Figure 9.52a. The VCO exhibits a positive monotonic frequency of oscillation in response to a control voltage, characterized by the slope of its frequency vs. voltage curve. The loop filter, $H(f)$, is of a general low-pass characteristic, often of an all-poles design to avoid any jitter peaking (or AC gain) in the closed loop transfer function of the PLL. The loop filter must provide a DC coupled path between the phase detector and VCO. The phase (and/or frequency) detector compares the VCO output phase to the input reference phase and generates an output signal that is either of a DC nature or has a DC component that is proportional to the phase difference between the reference and feedback signal. The phase detector is characterized by the rate of change in the DC component of its output vs. phase input difference in V/rad.

A phase detector that is commonly used because of its all-digital nature and suitability for CMOS integration is Gardner's phase-frequency detector (PFD) [11] with charge pump outputs. The PFD produces an output that goes toward V_{DD} or V_{SS} in the presence of a negative or positive frequency offset, respectively, thereby slewing the VCO toward the lock frequency. Once in frequency lock, the PFD

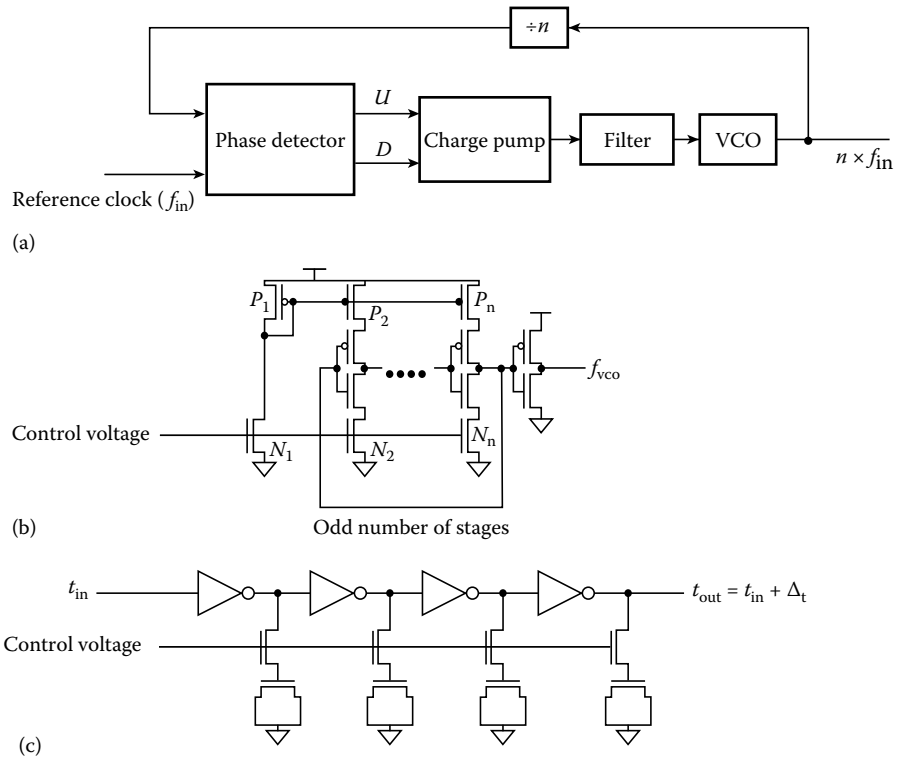


FIGURE 9.52 CMOS PLL circuits: (a) basic PLL block diagram, (b) CMOS VCO based on current-starved inverters, and (c) VCDL. (From Weste, N. and Eshraghian, K., *Principles of CMOS VLSI Design*, Addison-Wesley, Reading, MA, 1993, 336.)

produces pump up/pump down signals that vary in proportion to the time difference between reference and feedback clock edges at the PFD input. These pulse-width modulated signals drive a charge pump with a tristate buffer arrangement to either hold, bleed off, or supply charge to a capacitive storage element (i.e., the loop filter), thereby adjusting (and filtering) the voltage on the VCO control node to minimize the phase difference at the phase detector inputs.

For CMOS clocking system applications, the VCO is usually a form of a stable multivibrator in which the switching speed dependence of a CMOS inverter on its *n*-transistor pulldown current is exploited, as shown in Figure 9.52b. The VCO control voltage regulates the current flow in, and hence the speed of, each inverter stage through the extra *n*-transistor stage added to each inverter. Any odd number of such stages connected in a loop will oscillate, but now the relaxation period is voltage controlled. Weste and Eshragian [31] describe a 13-stage ''current-starved inverter'' VCO based on this approach. A related VCO design is based on varying the load capacitance seen by each inverter (in a chain of inverters) by applying the VCO control voltage to an *n*-MOSFET in series with the gate of another transistor configured as a capacitive load [16]. An on-chip RC loop filter can also be constructed from a CMOS transmission gate biased as a resistor and MOS gates used as capacitors (source and drain both connected to V_{SS}) [31].

If frequency multiplication is desired, a PLL with a true VCO is required. Otherwise, many PLL applications can use a voltage-controlled delay line (VCDL) in conjunction with a ''raw-clock'' input signal, as in Figure 9.52c. All the phase-lock feedback principles are the same, except we phase-shift the raw-clock input as required, rather than controlling the VCO oscillation phase. This eliminates the risk of the PLL ever failing to lock-in and may be simpler to fabricate. On the other hand, system design must take into account a more limited range of phase-shifting ability (a full half-cycle of phase control range may require a lot of delay stages) and to make sure initial delays are nominally centered within the positive-only delay control range of the VCDL. Another more subtle point is that while a VCO introduces a perfect integrator ($1/s$ term) in the PLL closed loop response, a VCDL does not. A VCDL, therefore, should not be simply substituted for a VCO without revisiting the closed loop response characteristics for noise bandwidth and possible jitter peaking.

Special power, grounding, and testing considerations apply when a PLL is used. A PLL is basically a linear circuit, so noise is especially important. Particularly when a frequency multiplying PLL is used, the VCO power supply should be well decoupled from system noise, and the input phase reference should be highly stable, as the PLL output clock will have N times the reference's phase noise. Noise voltages coupled into the analog PFD output and LPF signal path are similarly converted into phase noise that is N times worse than in a $\times 1$ PLL. Leadless on-chip decoupling capacitors are recommended as are dedicated power and ground pins for the PLL. The R and C components for the loop filter are often off-chip. In this case it is important that they are connected (depending on the $H(s)$ configuration) to the same analog ground reference as the VCO and PFD. The VCO output, or a divided down version of it, should be brought to an external pin for lock-in validation and as an aid in possible global system clock tuning. For testability, several other separate pins are typically required for independent access to I/O of the PFD, LPF, and VCO each. An on-chip PLL can require up to six or more pinouts.

9.2.3.11 Anceau's PLL Scheme

Anceau [1] developed a PLL-based approach for large systems in which modules are well-defined, relatively independent, and could be entirely self-timed if not for the need to avoid metastability in communication with other modules. Anceau recognized two natural system scales which are isochronic below different maximum frequencies. One is a global region encompassing the entire system, with a clock period determined by propagation distance delays, or, on-chip, by RC diffusion delays. The isochronic rate for this scale defines a slower clock rate for a system-wide communication bus. The second type of clocking region is smaller local regions which can run at full speed and are characterized by critical logic path delays and lumped capacitive loads within modules, not distance-dependent delays. Each smaller region will be free to operate in an almost self-timed mode.

The clocking style within each module (e.g., logic type and number of clock phases) can be as appropriate for the individual modules. Skew at the highest clock speeds in the system need be considered only within each module, except that timing must be controlled when reading the common data bus to avoid metastability. This is done by reference to the active edge of the slower-rate communications clock (comm_clk), formed by dividing down the master module clock frequency. The rising edge of comm_clk strobes the enabled driver data onto the bus. All other nondriving modules in the comm_clk cycle read the bus on an internal clock edge that is kept away in time form this transition in comm_clk, for metastability avoidance. Figure 9.53 illustrates the overall scheme. A PLL phase locks the module clock at a predefined angle relative to the comm_clk, thus keeping the raw module clock away from the transition times in the lower rate communication clock. The read timing is then safe because it is always preceded by the comm_clk transition on which new date were strobed to the bus. A monostable triggered by the comm_clk edge can be used inside each module as a delay generator to prohibit any bus read in the metastable region. This way, as long as modules write to the bus only on the comm_clk edge, other modules that read the bus will never do so at a moment when the bus data are still in transition.

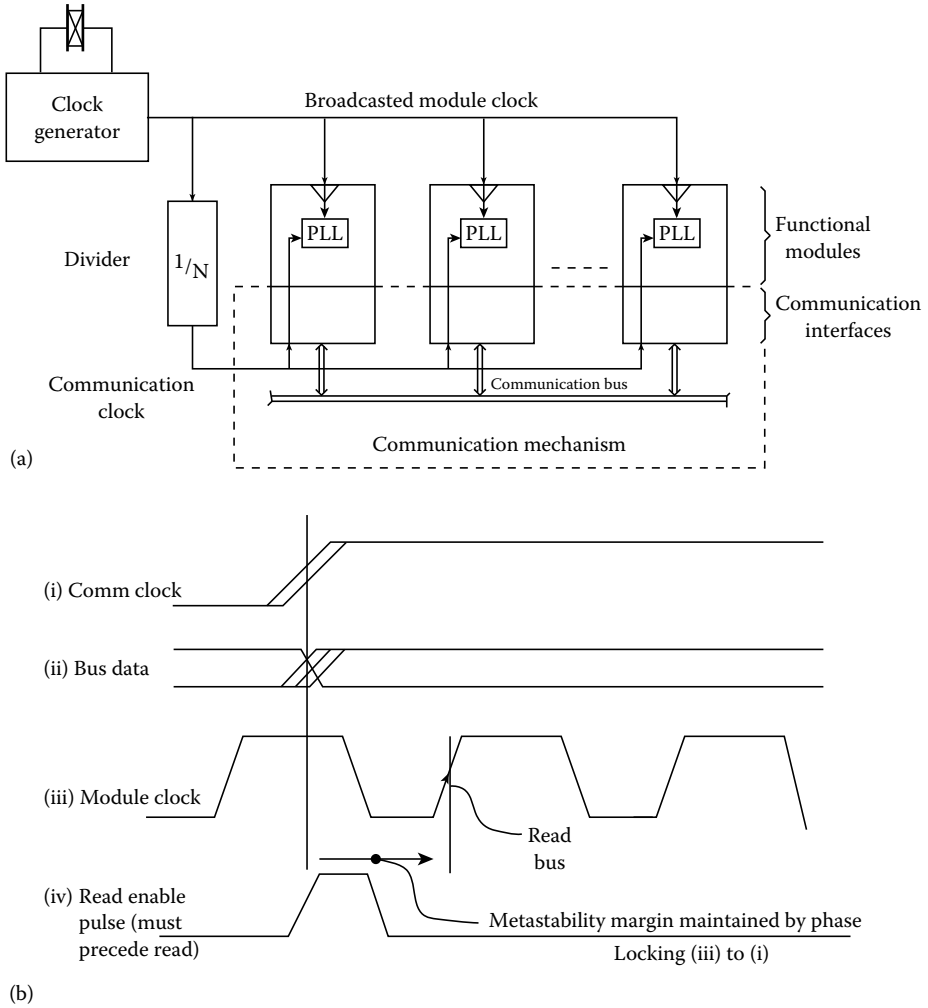


FIGURE 9.53 Anceau's scheme for metastability avoidance: (a) system architecture and (b) interface timing. (Adapted from Anceau, F., *IEEE J. Solid-State Circuits*, SC-17(1), 51, 1982.)

9.2.3.12 Grover's Interval-Halving PLL Scheme

A novel PLL-based approach to clock distribution in "electrically" large systems can synchronize all clocks in a large system on a single clock line [13,14]. In this scheme any number of nonisochronic points arbitrarily located on a single- or double-conductor reference line independently derive clock that is in absolute phase-lock to a common system-wide reference time. The central principle is that the time between appearances of an isolated pulse traveling down and back on a reference line is the same regardless of the point of observation, as shown in Figure 9.54a. This figure plots the trajectory in space-time of an isolated pulse that travels from a site at one end of a line and is returned at the end of the line to its origin (where it is electrically terminated). Figure 9.54a is drawn for the most general case of physically separate go and return conductors, looped at the right-hand end ($x = D$), but the space-time trajectory of the isolated pulse is identical if the line is a single conductor open-circuited at the end and driven by an impedance-matched source. Equivalently, a tristate buffer can terminate and regenerate the returning pulse at the end of the line for on-chip use. In either case, it is evident that the instant in time that is halfway between the outgoing and returning pulse edges is the same for all points of observation on the line, regardless of the propagation velocity of the line, i.e.,

$$\frac{t_1[x] + t_2[x]}{2} = t_1[D] = t_2[D] \equiv t_{\text{ref}} \tag{9.11}$$

where

$t_1[x]$, $t_2[x]$, are the times when the traveling pulse edge passes position x in Figure 9.54a
$t[D]$ is the time the reference pulse edge reaches (and departs) the reflection point

This time, called t_{ref}, is the midpoint between the two pluses as seen at every point of observation on the line.

This principle is adapted for single-line, skew-compensated clock distribution by periodic injection of a reference pulse onto a single conductor, reflection of this pulse at the end of the reference line, and generation of a local clock at all stations, such clock being phase-locked to the interval mid-time by a special interval halving PLL (IHPLL) circuit, as outlined in Figure 9.54b. The phase detector in the IHPLL is considerably simpler than the conventional PFD used in many CMOS PLL designs. This method can be adapted easily to a two-line operation, in which a full duty cycle waveform, rather than a narrow pulse, can be used to drive the looped reference line path. The reference line can be looped at one end and driven at the other, or split into two terminated lines, routed together as a pair, and driven together as shown in Figure 9.54c. In the latter case, all modules lock to the reference edge arrival at mid-path, rather than at its end.

In either of the two-conductor configurations, an edge-triggered, set–reset flip-flop function (Figure 9.54d) is the required phase detector. The two-conductor IHPLL approach avoids the need for an end-reflection or a tristate returning line driver, but requires layout of the reference line so that distances from the end are the same on both directions of the path at every tapping point. This is not hard to achieve on-chip, as the two halves of the looped path could be laid out identically. At the system level, however, the single-line variant has the advantage that uncontrolled cable and tracking lengths can be used without concern about delay equivalence in the return path, and an absolute minimum of cabling, connectors, and tracking is required for clock distribution. The interconnect is the same as that for a system in which clock is directly wired to all modules with a single line. However, this would normally be possible only if the whole system were one isochronic region.

Grover [13] reports experimental skew under 1 ns over 30 m on a coaxial cable which has an uncompensated delay of 147 ns. It was also shown that in the presence of the effects of the transmission line on the traveling reference pulse, the linear component of switching time error on the traveling reference pulse contributes no skew to the resultant clock phase. A phase-shifter variant of this scheme

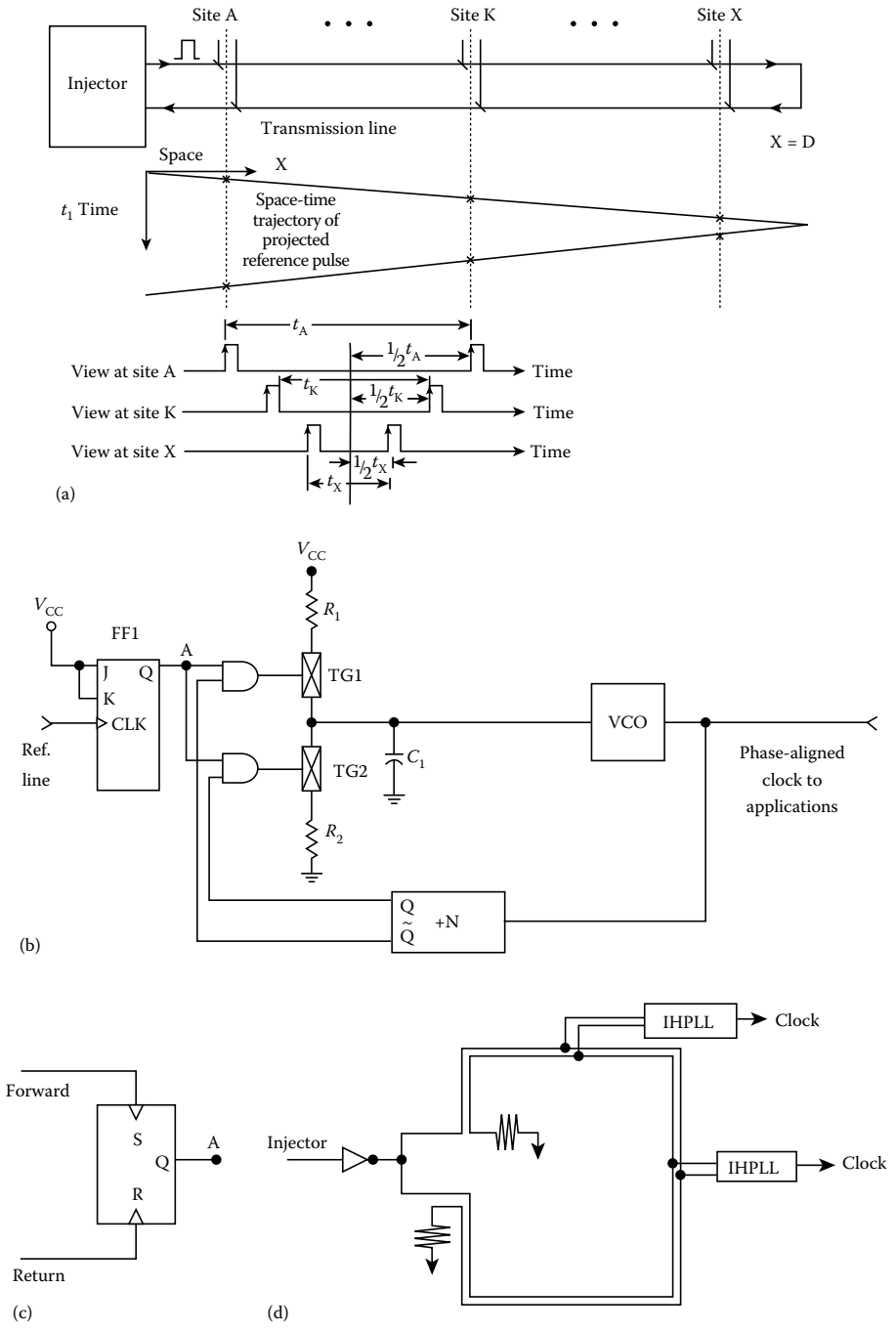


FIGURE 9.54 Grover's interval-halving clock distribution scheme: (a) interval-halving reference-time principle, (b) IHPLL circuit for single-line skew-compensated clock distribution, (From Grover, W.D., Method and apparatus for clock distribution and distributed clock synchronization, US Patent #5,361,277, Issued Nov. 1, 1994.) (c) phase detector for 2-line operation, and (d) driving two lines rather than one looped line.

uses a separate line to distribute raw-clock, which is then adaptively phase shifted at each point into the low-skew global phase by a voltage-controlled delay under the same feedback control sensing arrangement. Two-line operation, phase shifter, and other variations are described further in [14].

With this scheme, hierarchical clock distribution networks with delay-controlled cabling, delay-tuning, and numerous temperature- and load-dependent intermediate buffers may be replaced by one conductor with arbitrary routing. Both EMI and conducted noise are reduced by buffer driver elimination and because of the reduced average power of the reference pulse compared to the full clock signal. It may also be possible to add new clock-deriving taps in service, offering a growth path that is not limited by a predesigned clock–tree fanout limit. In many applications hybrids of reduced-depth clock trees, fanning out from skew-compensated roots on a single-line clock system of this type, may give the best combination of techniques.

Anceau's and Grover's schemes are similar in that a reference line is distributed to all modules and a PLL generates a local clock at each module. However, in the Anceau scheme modules do not run phase synchronously. Actual skew between modules remains arbitrarily high at the module clock frequency because the comm_clk line is set slow enough to be isochronic over the whole system. Actual delay in comm_clk, which is significant at the higher module rate, is not compensated at modules. Each module derives only enough information to coordinate its bus accesses with other modules, at the slower comm_clk frequency. In Grover's scheme, however, truly synchronous full-speed global clocking of all modules is achieved by returning the signal on the reference line (by reflection or looping) and exploiting the interval-halving time-reference principle and IH-PLL to cancel global skew. Gate-to-gate interaction on any clock cycle is feasible between modules in this case, as opposed to interaction only through the metastable-avoidance bus interface protocol.

9.2.3.13 Clock Tuning in Large Systems

In large systems, clock tuning at the chip, circuit pack, shelf, and rack levels of physical equipment may be required for the highest performance. Circuits to permit clock tuning can be a tapped delay-line circuit with a programmable selector, or (on a circuits card) a printed-in set of loops to be shunted out as needed by a suitcase jack, or a voltage-controlled varactor clock delay buffer. All of these are described in [15]. In general, to aid in the tuning process, one pin on each IC should be devoted to give external observability of the worst-case (if known) clock phase form inside the IC. This way the tuning process can compensate for the delays through I/O pads and clock buffers in large ICs.

Tuning begins by designing cable lengths, tracking, and connectors so that clock paths have nominally equivalent delays. The active tuning process then measures and adjusts relative delay starting from the master clock source to predefined levels of tuning points (TPs) electrically farther from the master clock source, denoted by TP_0. The delay measurement and adjustment repeats through lower level tuning points until the clock on in every IC is tuned. In going from the first to successive tuning points, it is preferable to refer delay measurements directly back to TP_0 each time. This may, however, be physically unmanageable, in which case delay tuning to level TP_n can be relative to TP_{n-1} although overall error relative to TP_0 will be higher in the relative tuning scheme. For systems that must grow in service, operational (i.e., in-service) signals should be the basis of the delay measurement, not requiring off-line signals or patterns.

One convenient way of indirectly measuring delay between points that are not easily accessed simultaneously for oscilloscope measurements is to make an oscillator out of the signal path to be measured. If the number of inversions in the path between TPs is odd, then an MUX can be switched to loop the tuning point signal at TP_n back to the TP_{n-1} driving point. A frequency counter can then measure the oscillation frequency, providing data to support automatic clock delay adjustments at the subordinate tuning point.

One mainframe computer used an automatic tuning scheme in which a clock phase-shifter chip produced multiple, slightly time-shifted copies of the clock on each system PCB. Individually selected delayed clock instances were then supplied to each IC on the board through a programmable crosspoint matrix IC. Each clock-receiving IC also provided several internal clock observation outputs to support

delay measurements down to the gate level. After automatically measuring the delays of the observable internal clocks in up to 30 ICs per board, the on-board clock selection matrix was programmed, giving each IC its best clock phase for overall system timing margins and minimal skew [30].

9.2.4 Future Directions

9.2.4.1 Current-Steered Logic

One way of reducing power supply noise injection from clocking is with current-steered logic. Experiments on differential current mode flip-flops in CMOS predict very short setup times (300–500 ps). Such devices would be very quiet electrically and much less susceptible to varying load effects on delay than on conventional CMOS. On the other hand, such devices might be about twice as large as conventional CMOS flip-flops and require more power. Using the bi-CMOS ability to integrate ECL type structures with CMOS may, however, be part of the solution for clocking very high speed medium-scale integrated devices.

9.2.4.2 Reduced Voltage Swing

Another potential method for reducing clocked load power consumption is to reduce voltage swings. Some experiments indicate a significant reduction in clock-related power and ground noise, but skew and delay objectives are more difficult to meet as the devices slow down in response to lower switching voltage swings.

9.2.4.3 Mixed Technology

Here, clock speed increases are envisaged by using current mode logic circuits selectively to implement critical timing paths in otherwise all-CMOS systems. New power reductions may also be obtained with ECL-based, high-speed, serial-multiplexed interfaces to replace wide buses which have many parallel CMOS drivers.

9.2.4.4 $\overline{Q}$ Elimination

Most logic families provide flip-flops with both Q and $\overline{Q}$ outputs as standard cells. An approach that could potentially have clock-related switching current and power is to eliminate the $\overline{Q}$ output buffer and develop corresponding logic synthesis tools to utilize inverted inputs and other logic means to assemble logic function without the $\overline{Q}$ outputs form flops. In one experiment of which the author is aware, $\overline{Q}$ buffers were removed, halving overall flip-flop power consumption, at the cost of only a 5% degradation in clock-to-Q delay.

9.2.4.5 Dedicated Layer for Clock Distribution

A number of workers are advocating or are already using dedicated third-layer metal for clock distribution. This affects process cost, but the advantages can be significant in high-performance applications. Third-layer metal is lower than other layers in resistance and capacitance. By moving the clock net, which is the largest single net in many designs, out of the other layers, routability of all other signals is improved and floorplanning simplified. Moreover, the clock tree can avoid uncertain delays due to unpredictable routing or due to polysilicon links in series when routing in fewer shared signal layers. In addition, noise due to clocking can be more easily isolated in the third-layer metal approach.

9.2.4.6 Optoelectronic Clock Distribution

Optical clock distribution takes advantage of the three-dimensional nature of imaging optics to remove all but the last-stage buffering levels of the clock distribution tree from the plane of the circuit, thereby eliminating multiple stages of buffering and metallization for clock routing. Figure 9.55 is an overview of the basic idea proposed by Clymer and Goodman [7]. The optical clock signal is generated off-chip and drives a laser diode at the top of the figure. The optical beam is expanded onto a transmission hologram,

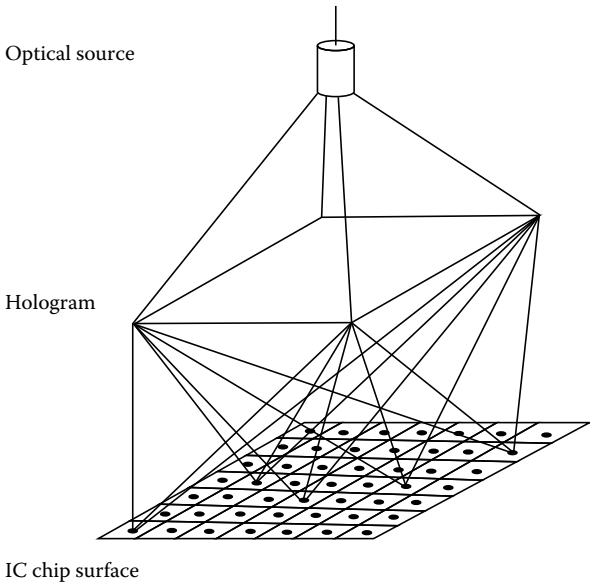


FIGURE 9.55 Clymer and Goodman's concept for wafer-scale optical clock distribution. (From Clymer, B.D. and Goodman, J.W., *Opt. Engin.*, 25(10), 1103, 1986.)

which focuses the light intensity onto predefined locations where optical detectors are fabricated into the wafer or die. The optical signal is detected, amplified, and used to drive a local clock generator-buffer which supplies a local isochronous regions. The optical path length differences are not equalized in this scheme because the optical path velocity is so high as to make the all optical path delays negligible as compared with the diffusion and lumped capacitive delays that determine the clock rare of the electronic system. When sources, detector, and packaging for this type of approach are developed, the potential exists for very low skew—high-speed clock distribution, with greater on-chip densities by eliminating most clock routing. One of the main challenges is in attaining uniform response times from the optical detector–amplifier combination (which tend to be sensitive to feature size variations) and the development of sources in the optical wavelength range for photodetectors that can be fabricated within the conventional CMOS circuit environment.

9.2.4.7 Reconfigurable Clock Nets

In WSI systems, where a single short in very large clock net may disable an entire wafer-level system, Fried [10] advocates methods of restructuring a clock net to enhance yield, primarily through the addition of a controllable tristate output stage to clock buffers within the clock distribution network. This way failed portions of the clock net can be isolated, or, with redundant interconnect and buffers, they may clocked by an alternate path. In particular, tristate buffers may be programmed on or off to select clock for each module from redundant connections to the central clock net, or to simply isolate failed clock net subregions from the drivers of unfailed portions.

References

1. F. Anceau, A synchronous approach for clocking VLSI systems, *IEEE J. Solid-State Circuits*, SC-17(1), 51–56, Feb. 1982.
2. M. Afghahi and C. Svensson, A unified single-phase clocking scheme for VLSI systems, *IEEE J. Solid-State Circuits*, 25(1), 225–233, Feb. 1990.

3. H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990, chap. 8.
4. S. Boon, S. Butler, R. Byrne, B. Setering, M. Casalanda, and A. Scherf, High performance clock distribution for CMOS ASICS, *Proc. IEEE 1989 Custom Integrated Circuits Conf*., San Diego, CA, pp. 15.4.1–15.4.5, 1989.
5. M. A. Cirit, Clock skew elimination in CMOS VLSI, *Proc. IEEE Int. Symp. Circuits Syst*., New Orleans, LA, pp. 861–864, 1990.
6. J. Cong, A. Kahng, and G. Robins, On clock routing for general cell layouts, *Proc. 4th IEEE Int. ASIC Conf*., Rochester, NY, pp. 14.5.1–14.5.4, Sep. 1991.
7. B. D. Clymer and J. W. Goodman, Optical clock distribution to silicon chips, *Opt. Engin*., 25(10), 1103–1108, Oct. 1986.
8. R. H. Cramback, C. M. Lee, and H. S. Law, High-speed compact circuits with CMOS, *IEEE J. Solid-State Circuits*, SC-17, 614–619, June 1982.
9. D. Dobberpuhl et al., A 200 MHz 64-b CMOS microprocessor, *IEEE JSSC*, 27(11), 1555–1567, Nov. 1992.
10. J. Fried, Power and clock distribution for WSI systems, *Proc. IFIP Workshop on Wafer Scale Integration*, G. Saucier and J. Trilhe, Eds. Amsterdam: North-Holland, pp. 127–141, 1986.
11. F. M. Gardner, Charge-pump phase-locked loops, *IEEE Trans. Commun*., COM-28, 1849–1858, Nov. 1980.
12. L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*. Reading, MA: Addison-Wesley, 1985, chap. 6.
13. W. D. Grover, A new method for clock distribution, *IEEE Trans. Circuits Syst. Part I*, 41(2), 149–160, Feb. 1994.
14. W. D. Grover, Method and apparatus for clock distribution and distributed clock synchronization, US Patent #5,361,277, Issued Nov. 1, 1994.
15. M. W. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*. Englewood Cliffs, NJ: Prentice Hall, 1993, chap. 11.
16. M. G. Johnson and E. L. Hudson, A variable delay line PLL of CPU-coprocessor synchronization, *IEEE J. Solid-State Circuits*, 23(5), 1218–1223, Oct. 1998.
17. D. C. Keezer and V. K. Jain, Design and evaluation of wafer scale clock distribution. *IEEE Int. Conf. WSI*, San Francisco, CA, pp. 168–175, 1992.
18. S.-Y. Kung and R. J. Gal-Ezer, Synchronous vs. asynchronous computation in very large scale integrated (VLSI) array processors, *Proc. SPIE*, 341, 53–64, 1982.
19. S. D. Kugelmass and K. Steiglitz, An upper bound on expected clock skew in synchronous systems, *IEEE Trans. Comput*., 39(12), 1475–1477, Dec. 1990.
20. LSI Logic Corp., *Phase-Locked Loop Application Note*, LSI Logic Application Note, Nov. 1991.
21. LSI Logic Corp., *Clock Scheme for One Micron Technologies*, Rev. 1.1, LSI Logic Application Note, Aug. 1992.
22. LSI Logic Corp, *Clock Distribution Schemes for 300K Technologies*, Rel. 2.0, LSI Logic Application Note, May 1993.
23. A. M. Moshen and C. A. Mead, Delay-time optimization for driving and sensing of signals on high-capacitance paths of VLSI systems, *IEEE Trans. Electron Devices*, ED-26, 540–548, 1979.
24. N. Nigam and D. C. Keezer, A comparative study of clock distribution approaches for WSI, *Proc. IEEE 1993 Int. Conf. WSI*, San Francisco, CA, pp. 243–251, 1993.
25. W. A. Samaras, The CPU clock system in the VAX 8800 family, *Digital Tech. J*., 4, 34–40, Feb. 1987.
26. T. Saigo, S. Watanabe, Y. Ichikawa, S. Takayama, T. Umetsu, K. Mima, T. Yamamoto, J. Santos, and J. Buurma, Clock skew reduction approach for standard cell, *Proc. IEEE* 1990 *Custom Integrated Circuits Conf*., Boston, MA, pp. 16.4.1–16.4.4, 1990.
27. C. L. Seitz, System timing, in *Introduction to VLSI Systems*, C. Mead and L. Conway, Eds. Reading, MA: Addition-Wesley, 1980, chap. 7.

28. J. Shyu, A. Sangiovanni-Vincentelli, J. Fishburn, and A. Dunlop, Optimization-based transistor sizing, *IEEE J. Solid-State Circuits*, 23(2), 400–409, Apr. 1988.
29. D. Tanksalvala et al., A 90 MHz RISC CPU designed for sustained performance, *IEEE Solid-State Circuits Conf.*, San Francisco, CA, pp. 52–53, Feb. 1982.
30. K. D. Wagner, A survey of clock distribution techniques in high speed computer systems, Report CRC 86-20. Stanford, CA: Stanford University Center for Reliable Computing, Dec. 1986.
31. N. Weste and K. Eshragian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd edn. Reading, MA: Addison-Wesley, 1993, pp. 317–335 (clocking strategies), pp. 334–336 (PLL methods), pp. 685–689.
32. D. F. Wann and M. A. Franklin, Asynchronous and clocked control structures for VLSI based interconnection networks, *IEEE Trans. Comput.*, C-32(3), 284–293, Mar. 1983.

9.3 MOS Storage Circuits

Josephine C. Chang and Bing J. Shev

In a large digital system, a sequence of operations must be performed for a particular function. The results of each operation depend on the results of previous operations. Therefore, the outputs of a logic circuit block typically depend not only on present input signals, but also on the history of the inputs. A CL circuit becomes more useful if it is combined with memory elements. To construct a sequential system, the most common and straightforward way is to employ a central clock to synchronize the sequence of operations.

Instead of using memory elements in a sequential system, we can use dynamic logic circuits to store temporary data. With the building blocks of inverters and transmission gates, the MOS transistors can be used as dynamic storage components to store data temporarily on the device capacitances. Dynamic storage is widely used in MOS technologies because of the simplicity of the required circuitry. Because a memory element such as a static circuit latch occupies a large area and consumes power, elimination of latches has a positive effect on circuit density and power consumption. However, the disadvantages of dynamic logic gates include high transient power disturbances and less noise margins in some applications [1].

Dynamic logic circuits design is based on the synchronized movement of charge through the MOS circuit. A typical capacitance value associated with a logic gate is on the order of a few femtofarads, with means the amount of charge $Q = CV$ dynamically stored on the capacitance is on the order of femtocoulombs. Therefore, perturbations from ideal behavior can become critical to the operation of a circuit.

9.3.1 Dynamic Charge Storage

The MOS technologies have two attractive features that lead to an efficient way to store data momentarily. These two features are the extremely high input impedance of MOS transistor and the ability of a MOS transistor to function as a nearly ideal electrical switch. In order to store the charge on a capacitive node, the node must be isolated from both the power supply and ground. Various types of storage nodes can be realized in CMOS technologies. For example, charge can be stored at a node between sources (or drains) of two MOS transistors such as nMOS–nMOS, pMOS–nMOS, and nMOS–pMOS [2]; or the source (or the drain) terminal of one MOS transistor connected to the gate terminal of a second MOS transistor. Because the stored charge will leak away over time, this circuit is termed ''dynamic storage circuit.''

Figure 9.56 shows the schematic diagrams of three combinations of source–drain connection. The distinction among the three connection types comes from the difference in voltage transmission levels for nMOS and pMOS gates.

Dynamic charge storage requires clocking the data at a sufficiently high rate so that the charge on the various nodes does not lead away significantly. Typically, this requires a minimum refresh rate of 500 Hz to 1 kHz, corresponding to a charge storage time of about to 2 ms.

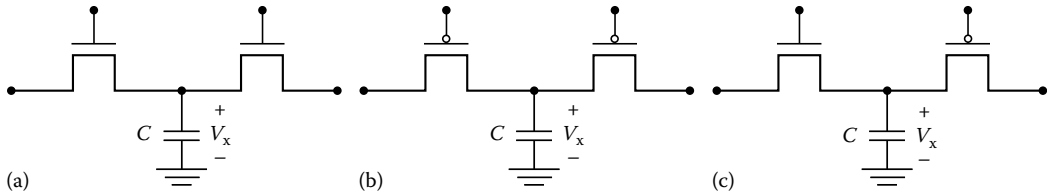


FIGURE 9.56 MOSFET source–drain connection storage nodes. (a) nMOS–nAMOS; (b) pMOS–pMOS; and (c) nMOS–pMOS.

9.3.1.1 nMOS–nMOS

An nMOS transistor is perfect for transmitting logic 0 signals, but imperfect for transmitting logic 1 signals due to the threshold voltage loss through the transistor. The voltage level of V_x which can be stored on the capacitor C is therefore limited by

$$0 \leq V_x \leq (V_{DD} - V_{th,n}) = V_{max} \tag{9.12}$$

where

$$V_{th,n} = V_{th\ 0,n} + \gamma_n\left(\sqrt{2\phi_{Fn} + V_{max}} - \sqrt{2\phi_{Fn}}\right) \tag{9.13}$$

Charge storage on an nMOS–nMOS node is affected by the leakage paths through the p-type bulk to the ground. This affects the long-term storage of a logic 1 value.

9.3.1.2 pMOS–pMOS

A pMOS–pMOS node is the complement storage component of an nMOS–nMOS node. The voltage level of V_x is limited by

$$V_{min} = \left|V_{th,p}\right| \leq V_x \leq V_{DD} \tag{9.14}$$

where

$$V_{th,p} = V_{th\ 0,p} + \left|\gamma_p\right|\left(\sqrt{\left|2\phi_{Fp}\right| + (V_{DD} - V_{min})} - \sqrt{2\left|\phi_{Fp}\right|}\right) \tag{9.15}$$

Because both p-channel MOS transistors have n-type bulks which are connected to V_{DD}, this type of storage node receives leakage current from the power supply. The logic 1 values can be held indefinitely, but the logic 0 values can only exist for a limited period of time.

9.3.1.3 nMOS–pMOS

A complementary nMOS–pMOS storage node can benefit from both nMOS and pMOS in transmitting logic 0 and logic 1, respectively. The voltage level which is stored on the capacitance is in the range of

$$0 \leq V_x \leq V_{DD} \tag{9.16}$$

If the maximum input value of V_{DD} is transmitted through the pMOS transistor and the minimum input value of 0 V is transmitted through the nMOS transistor. On the other hand, if the case is reversed, the

maximum input value is entered through the nMOS transistor and the minimum input value is entered through the pMOS transistor, then the voltage range is reduced to

$$\left|V_{\text{th,p}}\right| \leq V_{\text{x}} \leq V_{\text{DD}} - V_{\text{th,n}} \tag{9.17}$$

This type of operation should be avoided because it greatly reduces the noise margins. In a standard nMOS–pMOS storage node both leakage paths to the power supply and ground exist. The ability to retain logic 0 and logic 1 values depends on which leakage path dominates.

9.3.1.4 Source–Gate Connection

This type of storage node is the connecting point between the source terminal of a pass transistor and a gate terminal of another MOS transistor [3]. Electrical charge can be temporarily stored on or removed from the gate terminal of the second transistor. When the gate terminal of the pass transistor is at a logic low value, the pass transistor is turned off, and the charge on the gate terminal is isolated. This charge determines the stored logic value. If the stored charge is perfectly isolated, the logic value would be stored indefinitely. In a practical situation, the isolation is less than perfect, primarily because of leakage through the reverse-biased diode operation between the source diffusion region of the pass transistor and the substrate. In addition, leakage also can occur through the pass transistor. With the continuous advances in VLSI technologies, subthreshold leakage through the channel of the pass transistor becomes more important due to scale-down in device sizes. Leakage currents alter the node voltage, which may lead to a logic error.

Two major problems arise in maintaining the integrity of a stored logic state. First is the parasitic conduction paths in the transistors that lead to charge leakage. Leakage currents alter the node voltage, which may cause a logic error. The second problem is charge sharing, which occurs when two isolated storage nodes become connected by a switching event and must equalize their voltages by redistributing charge. Charge sharing may result in a logic error, or may block logic propagation entirely.

9.3.1.5 Charge Sharing

Beside charge leakage, a problem called charge sharing may also damage the integrity of a stored logic state. Charge sharing occurs when a dynamic charge-storage node is used to drive another isolated node in a switching network [4]. Typically, when two capacitors with different voltages are connected by a pass transistors, as shown on Figure 9.57, charge sharing may occur. When the pass transistor is turned on, the voltages on the capacitors equilibrate to some intermediate value. In Figure 9.57, capacitors C_1 and C_2 are in parallel when the transmission gate is conducting. This forces the voltages across C_1 and C_2 to be equal. If the two capacitors are charged to different initial voltages, charge sharing will occur when the

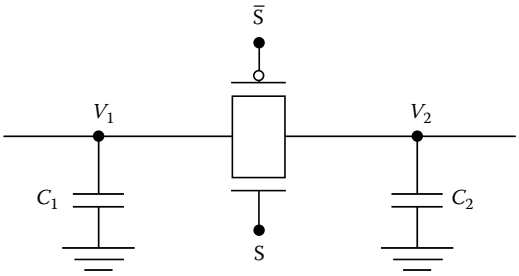


FIGURE 9.57 Charge-sharing-prone structure.

transmission gate turns on. Let the initial voltage charge on C_1 be V_1 and Q_1, and the initial voltage and charge on C_2 be V_2 and Q_2. The initial charge balance equation is

$$Q_1 + Q_2 = C_1V_1 + C_2V_2 \tag{9.18}$$

After the transmission gate turns on, the final charges on C_1 and C_2 become Q_1' and Q_2', respectively, and both capacitors are charged to the same value V'. The final charge balance equation is

$$Q_1' + Q_2' = (C_1 + C_2)V' \tag{9.19}$$

By applying the charge-conservation principle, we can obtain

$$V' = \frac{C_1V_1 + C_2V_2}{C_1 + C_2} \tag{9.20}$$

and

$$Q_1' = \frac{C_1}{C_1 + C_2}(C_1V_1 + C_2V_2) \tag{9.21}$$

A precharged circuit might work incorrectly due to charge-sharing errors, which could occur inside the pulldown network or at the output circuit. To control a precharged circuit, a gated clock can be present only at the input of the bottom transistor, while all other inputs to the gates of transistors in series in the pulldown chain must have a stable signal over the same clock phase to prevent charge-sharing problems. A ''sneak path'' is created when two pass transistors in series are both turned on at the same time and one is connected to V_{DD} while the other is connected to the ground. Charge can leak through this sneak path.

9.3.2 Shift Register

A frequent use of dynamic storage circuits is the shift register. Shift registers are most often used to provide temporary storage of digital signals. The shift register storage can be used as a simple way to delay the arrival of a signal for a specific number of clock cycles. Shift register storage is also frequently used as the temporary memory for a sequential logic circuit. In general, shift registers provide dense, limited access memory for many applications in digital integrated circuits.

9.3.2.1 Simple Shift Register

Figure 9.58 is the schematic diagram of a multistage MOS shift register, with each stage composed of a pass transistor and an inverter [5]. The nonoverlapping clock waveforms Φ_1 and Φ_2 are used. Assume that a logic signal is placed at the input of the first shift register stage while the Φ_1 clock is low and the transmission gate of the first stage is turned off. Next, when the Φ_1 clock goes high, if the signal at the input to the first stage is held constant, it will be propagated to the input of the inverter in the first stage. After a short delay, the output of the first inverter will provide the inverted logic signal to the input of the second shift register stage. At this time, the Φ_2 clock is low and the transmission gate in the second stage will not pass this signal. When the clock values change so that Φ_2 becomes high, the transmission gate of the second stage will propagate the output signal of the first stage to the second inverter, and then the output of the second stage is produced. This signal will be stopped by the transmission gate of the third stage because Φ_1 is low while Φ_2 is high. This sequence continues through the shift register chain as the clock signals alternate, causing the input signal to propagate through the shift register stages. The data are

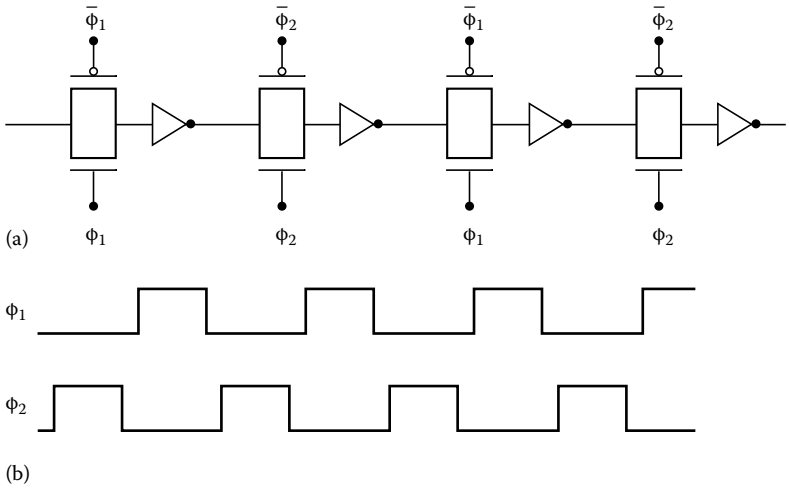


FIGURE 9.58 (a) A four-stage MOS register and (b) nonoverlapping waveforms of ϕ_1 and ϕ_2.

stored on the capacitances associated with the gate terminals of the inverter. The transmission gate acts as the switch that lets charge flow into and out of the capacitors when they are turned on. The charge is trapped at the capacitor when the transmission gate is turned off.

Each time the Φ_1 clock changes to a high value, the shift register input signal will propagate to the gate of the first inverter and the output signal of the first stage will be produced. A sequence of alternating Φ_1 and Φ_2 clock signals will cause an input signal to propagate through the whole structure at the rate of two stages of the shift register for each complete cycle of the clock signals. After N clock cycles, a logic input value will have shifted through $2N$ stages of the shift register chain. When a two-phase clock is used to control a shift register, it is important that the two clock phases do not overlap. If both phases of the clock were high simultaneously, a data value could propagate through multiple stages during the clock overlap time. This would cause erroneous operation of the shift register.

9.3.2.2 Parallel Shift Register

Several copies of the multistage shift registers can be combined in parallel with the same clock lines to form a parallel shift register to transmit a group of signals in lock-step fashion. Such a parallel shift of 8, 16, or 32 data bits is often used in microprocessor circuits. The basic structure of this set of shift registers demonstrates two principles which are important for the efficient geometrical layout of digital circuits. The data for the shift register flow from left to right while the control signals (Φ_1 and Φ_2 clocks) flow from top to bottom. Such an orthogonal structure of data paths and control signals within a circuit module is widely used to provide a regular organization of logic circuits within a VLSI chip. Physical layout of the shift register stages can be mirrored with respect to the ground and V_{DD} lines. This mirroring technique allows shared power and ground connections and reduces required circuit layout area. It is important to minimize the size of the basic shift register stage because this stage is repeated many times in a large shift register.

9.3.2.3 Clocked Barrel Shifter

A "barrel shifter" is a wraparound shifter that forms a very useful switch array [6]. The basic layout is shown in Figure 9.59. The inputs are labeled I_i; the shift controls $\Phi_2 \cdot \mathrm{SH}_i$, and the outputs O_i. The input lines run horizontally while the output lines run vertically. The operation of the first shift register

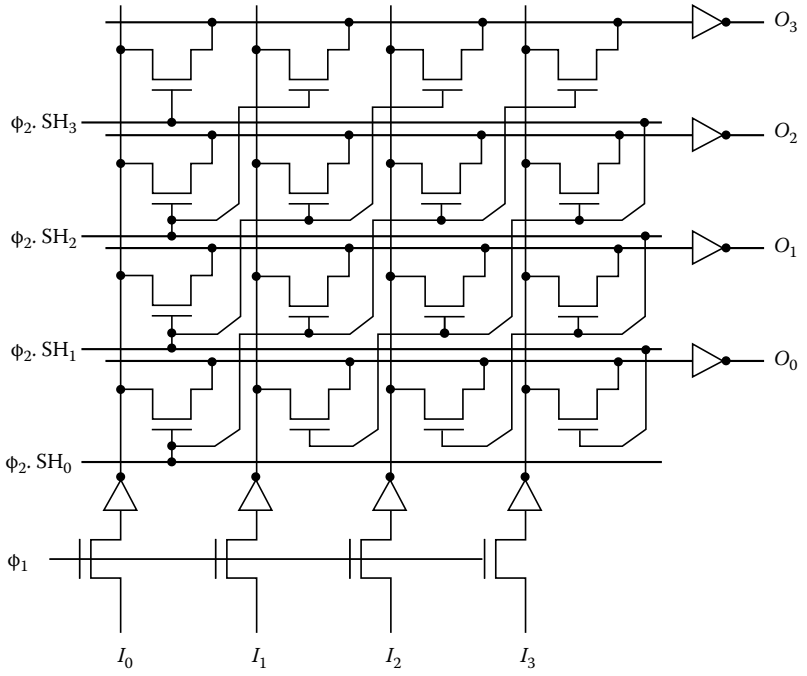


FIGURE 9.59 A four-bit clocked barrel shifter.

stage is the same as explained earlier. In the second stage for four output signals from the four inverters in the first stage can be shifted without changing the order or each signal can move up one, two, or three locations.

9.3.3 Dynamic CMOS Logic

The dynamic CMOS logic design consists of dynamic circuits based on precharging the output node to a particular level when the clock is at the logic 0 level. During the precharge phase, the inputs to the circuits change. When the clock is at the logic 1 value, the output of the logic gate may be pulled to a complementary value, depending on the input conditions.

The choice of using static or dynamic logic is dependent on many criteria. When low-power performance is desired, it appears that dynamic logic has some inherent advantages in a number of areas including reduced switching activity due to hazards, elimination of short-circuit dissipation, and reduced parasitic node capacitances. Static logic circuits have advantages on charge sharing and precharge operation.

Static circuits design can exhibit spurious transitions due to races. These spurious transitions dissipate extra power over that required to perform the computation. The number of these extra transitions is a function of input patterns, internal state assignment in the logic design, delay skew, and logic depth. Although it is possible with careful logic design to eliminate these transitions, dynamic logic intrinsically does not have this problem because any node can undergo at most one power-consuming transition per clock cycle [7].

Short-circuit currents are found in static CMOS circuits. However, by sizing transistors for equal rise and fall times, the short-circuit component of the total power dissipated can be kept to $<$20% of the dynamic switching component. Dynamic logic does not exhibit this problem, except for those cases in which static pullup devices are used to control charge sharing or when clock skew is significant. Dynamic

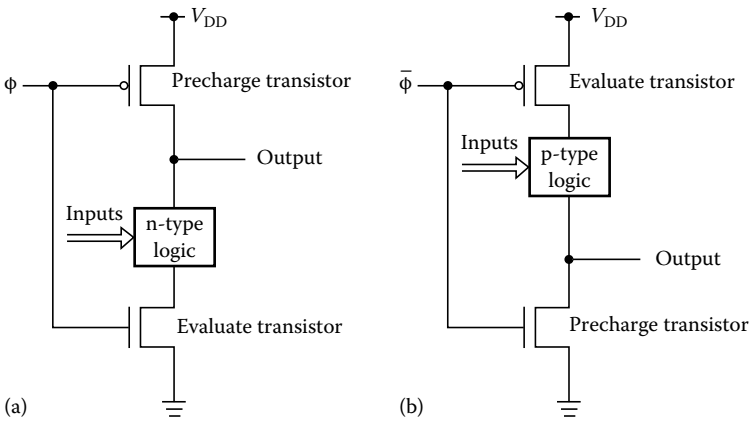


FIGURE 9.60 A precharge–evaluate logic gate.

logic typically used fewer transistors to implement a given logic function, which directly reduces the amount of capacitance being switched and thus has a direct impact in the power-delay product.

9.3.3.1 Precharge–Evaluate Logic

The schematic diagram of a basic precharge–evaluate logic is shown in Figure 9.60. It consists of an *n*MOS logic structure whose output node is precharged to V_{DD} by a pMOS precharge transistor; and conditionally discharged by the *n*-transistor network connected to the ground. Alternatively, an *n*-transistor precharge to the ground and a pMOS logic structure to conditionally discharge to V_{DD} may be used. A single-phase clock Φ is used for high-speed operation. For the former case, the precharge phase occurs when the clock Φ is low. The path to the ground is activated via the *n*-transistor network when the clock Φ is high. The input capacitance of this logic gate is the same as a pseudo-nMOS gate which has a single *p*-transistor, with the gate connected to the ground, as a load device. The pullup time is better than a pseudo-nMOS gate by virtue of the active switch but the pulldown time is increased due to the ground switch.

9.3.3.2 Clocked CMOS Logic

Clocked CMOS logic (C^2MOS) gates were originally used to build low-power dissipation logic gates. The reasons for the reduced dynamic power dissipation stem mainly from metal–gate CMOS layout considerations. The main use of such logic structures is to form clocked structures that incorporate latches or interface with other dynamic forms of logic structure. The gates have the same input capacitance as regular complementary gates, but larger rise and fall times due to the serially connected clocking transistors.

The schematic diagram of a clocked CMOS logic gate is shown in Figure 9.61. In this circuit, the clocked transistors are placed in series with the transistors in the *p*- and *n*-type logic blocks. The primary use of C^2MOS is in dynamic shift registers. In a C^2MOS dynamic shift register, the *p*-type logic black is a *p*-transistor network and the *n*-type logic is an *n*-transistor network. All transistors can normally be chosen as minimum-size devices because each stage is only required to drive the capacitance of an identical shift register stage.

Although the C^2MOS circuit requires the same number of transistors, external connections, and clock phases as the standard CMOS dynamic shift register, the layout is greatly simplified because the source/drain regions of the two *p*-channel transistors can be merged, and the corresponding regions of the two *n*-channel transistors can be merged. This feature helps to reduce circuit capacitance, number of contacts, and layout area.

Operation of the C^2MOS circuit is quite simple. The gates of the *p*MOS pullup transistor and the nMOS pulldown transistor of the inverter are both connected to the input signals. For a valid logic input,

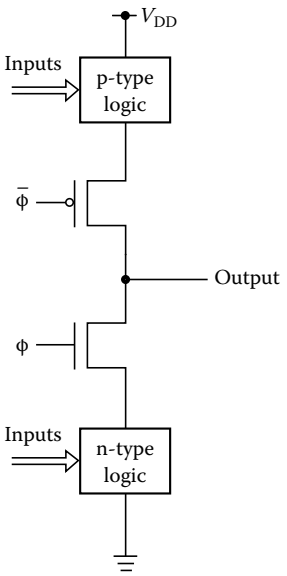


FIGURE 9.61 A clocked CMOS logic gate.

one of these transistors is turned off while the other is turned on. Clocked transistors placed in series with the pullup and pulldown transistors serve to connect these transistors to the output node when the clock Φ is high. If the input signals match the *n*-type logic portion, the output storage node will be discharged. Otherwise, the output storage node will be charged. When the clock Φ is low, the output node will remain in its present state. In contrast to other clocked logic circuits, the output of C^2MOS is available during the entire clock cycle, although it is actively driven only when the clock is high. A C^2MOS circuit is more susceptible to interference from the load circuit attached to the stage because the load capacitance is the storage node for the dynamic charge.

9.3.3.3 Domino CMOS Logic

The domino logic gate design can provide glitch-free cascades of nMOS logic structures. It is a modification to the clocked CMOS logic gate to allow a single clock to precharge and evaluate a cascaded set of dynamic logic blocks. A domino logic gate consists of two elements: a precharge–evaluate logic stage followed by a static inverter buffer at the output, as shown in Figure 9.62. The logic gate can be built in two forms: mostly *n*-transistors and mostly *p*-transistors. During the precharge phase when clock Φ is low, the output node of the dynamic gate is precharged high, and inverted by the static buffer to provide a logic 0 output for the domino CMOS gate. As subsequent logic stages are driven by this buffer, transistors in subsequent logic blocks will be turned off during the precharge phase. When the gate is evaluated, the node voltage of the logic stage is conditionally pulled down according to the input signal values. If the logic condition of the gate is satisfied, the node voltage is pulled down. It is inverted by the static buffer to provide a logic 1 output. Each gate in sequence can make at most one transition ($1 \rightarrow 0$). Hence, the buffer can only make a transition from ($0 \rightarrow 1$). In a cascaded set of logic blocks, each state evaluates and causes the next stage to evaluate in the same manner as a stack of dominos fall. Any number of logic stages may be cascaded, provided that the sequence can evaluate within the given clock phase. A single clock can be used to precharge and evaluate all logic gates within a block.

The structure has some limitations. First, only noninverting structures can be constructed. Second, each logic gate must be buffered. Finally, in common with a clocked CMOS gate, charge redistribution can be a problem. The effects of these problems can be minimized. For example, in complex logic circuits, such as ALUs, the necessary XOR gates may be implemented conventionally as complementary gates and driven by the last domino circuit. The buffer is often needed to drive large capacitive load and does not contribute to any extra cost.

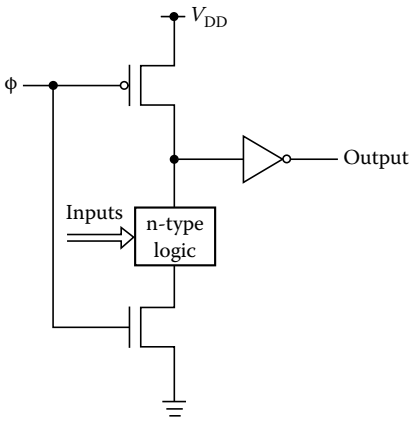


FIGURE 9.62 A domino CMOS logic gate.

Static storage of charge can be realized by a domino logic gate by including a weak *p*-transistor. A weak *p*-transistor is one that has low gain, which is realized with a small W/L ratio. It should have a small gain in order not to fight with the pulldown transistors, yet to balance the effects of leakage. This will allow low frequency or static operation when the clock is held high. In this case, the pullup speed could be an order of magnitude slower than the pulldown speed. Notice that the precharge transistor may be eliminated if the time between evaluation phases is long enough to allow the weak pullup to charge the output node.

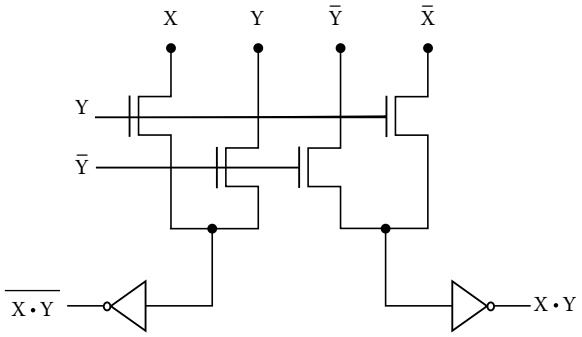


FIGURE 9.63 A CPL AND/NAND cell.

A domino logic gate has advantages over a simple precharge–evaluate logic structure. For example, the static buffer provides output-driving capability to either V_{DD} or the ground. In the precharge–evaluate logic gate the output can be favorably driven only to the ground in response to logical conditions, not to V_{DD}. When the logic condition of the precharge–evaluate gate is not satisfied, dynamic charge storage at the output must maintain the logic 1 value. The dynamic logic portion of a domino CMOS gate always has a fanout of 1, thereby simplifying device sizing within the gate structure.

9.3.3.4 Complementary Pass-Transistor Logic

The complementary pass-transistor logic (CPL) gate is constructed by using an nMOS pass-transistor network for logic function and eliminating the pMOS latch [8]. It consists of complementary inputs and outputs, nMOS pass transistor logic network, and CMOS output inverters. Figure 9.63 is the schematic diagram of a CPL AND/NAND cell. The pass-transistors function as pulldown and pullup devices. Thus, the pMOS latch can be eliminated, allowing the advantage of differential circuits to be fully utilized. One attractive feature of the CPL gate is that complementary outputs are produced by the simple four-transistor circuits. Because the logic 1 value level of the pass-transistor outputs is lower than the supply voltage V_{DD} by the threshold voltage of the pass-transistors, the signals must be amplified by the output inverters. In addition, the CMOS output inverters shift the logic threshold voltage and drive the capacitive load. The logic threshold shift is necessary because that of the output inverter is lower than half of the supply voltage, due to the lowering of the logic 1 value.

The CPL gate is attractive because fewer transistors are required to implement important functions. However, a CPL gate has two basis problems. First, the threshold drop across the single-channel pass-transistor results in reduced current drive and hence slower operation at a reduced supply voltage. This is important for low-power design because it is desirable to operate at the lowest possible voltage levels.

Second, because the logic 1 input value at the regenerative inverters is not V_{DD}, the pMOS device in the inverter is not fully turned off, and hence direct-path static power dissipation could be significant. To solve these problems, reduction of the threshold voltage has proven effective, although if taken too far it will incur a cost in dissipation due to subthreshold leakage and reduced noise margins.

9.3.3.5 Cascade Voltage Switch Logic

The cascade voltage switch logic (CVSL) gate is a differential style of logic circuit design requiring both true and complement signals to be routed to the gates [9]. Two complementary nMOS switch structures are constructed and then connected to a pair of cross-coupled p pullup transistors as shown in Figure 9.64a. When the inputs switch, node voltages Q and $\overline{Q}$ are either pulled up or down. Positive feedback applied to the p pullup transistors causes the gate to switch. The logic trees may be further minimized from the fully differential form using logic minimization algorithms. This version is slower than a

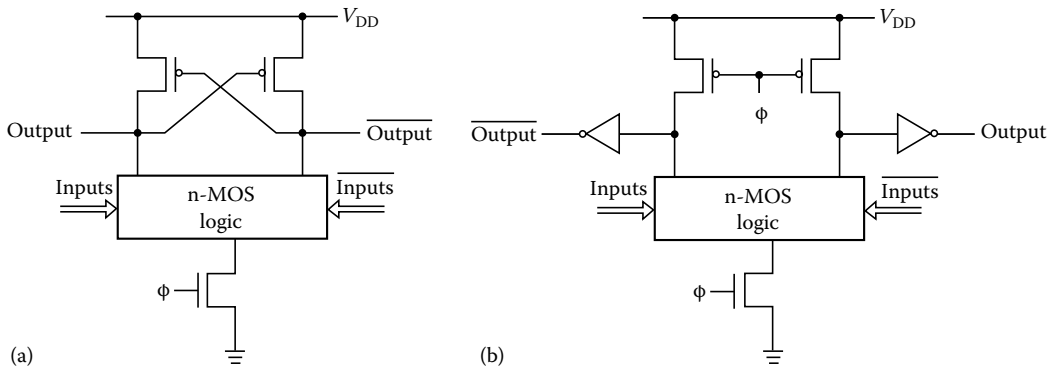


FIGURE 9.64 A CVSL gate: (a) static version and (b) dynamic version.

conventional complementary gate employing a *p*-tree and an *n*-tree because during the switching action, the *p* pullup transistors must compete with the *n* pulldown tree. The schematic diagram of a dynamic charge-storage version of the CVSL logic gate design is shown in Figure 9.64b. It consists of two domino logic gates with complementary input logic trees. The advantage of CVSL gate over a domino logic gate is the capability to generate a complete logic function rather than just the noninverting logic function. However, extra silicon area is needed.

9.3.3.6 NORA CMOS Dynamic Logic

NORA logic is capable of handling signal race problems in transmission pates [10]. It is based on dynamic CMOS logic, but uses latches instead of transmission gates to control signal flow. In a NORA logic dynamic nMOS and pMOS, logic circuits are cascaded into a C^2MOS latch. Figure 9.65 shows the schematic diagrams of both Φ stage and $\bar{\Phi}$ stage. Static inverters are provided at the outputs of dynamic circuits to realize logic inversion. This allows direct implementation of arbitrary functions without modification. In the Φ stage, the logic circuit used $\Phi = 0$ for precharge and $\bar{\Phi} = 1$ for evaluation. The latch accepts data when $\Phi = 1$ and holds the data when $\Phi = 0$. No new data can be accepted during the hold time. The operation in the $\bar{\Phi}$ stage is similar when reversing clock signals are used.

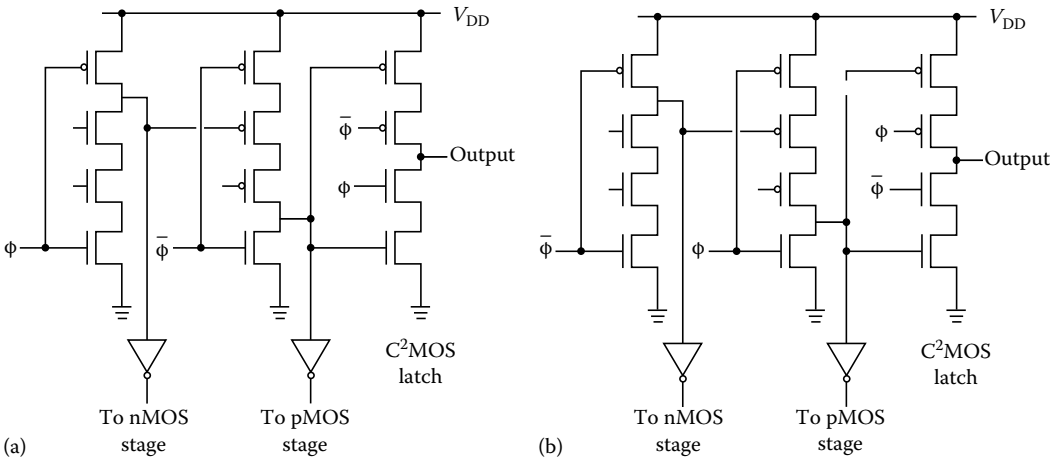


FIGURE 9.65 NORA clock stages: (a) Φ stage and (b) $\bar{\Phi}$ stage.

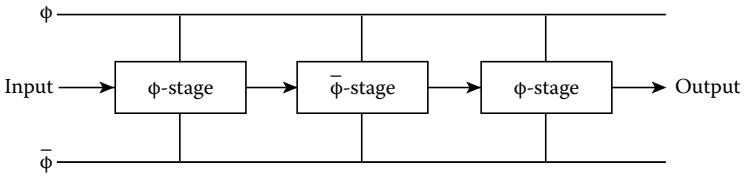


FIGURE 9.66 A NORA chain.

By alternating Φ and $\bar{\Phi}$ clock stages makes NORA chains well suited for pipelined logic. The schematic diagram of a generic structure of a NORA chain is shown in Figure 9.66 [2]. Logic flows through the chain at a rate set by the clock. The problem of logic races by using transmission gates as latches between logic circuits has been eliminated because of the dynamic C^2MOS latch circuit.

References

1. N. Wang, *Digital MOS Integrated Circuits*, Englewood Cliffs, NJ: Prentice Hall, 1989.
2. J. P. Uyemura, *Circuit Design for CMOS VLSI*, Boston, MA: Kluwer Academic, 1992.
3. R. L. Geiger, P. E. Allen, and N. R. Strader, *VLSI Design Techniques for Analog and Digital Circuits*, New York: McGraw-Hill, 1990.
4. A. Mukherjee, *Introduction to nMOS and CMOS VLSI Systems Design*, Englewood Cliffs, NJ: Prentice Hall, 1986.
5. L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Reading, MA: Addison-Wesley, 1985.
6. E. D. Fabricius, *Introduction to VLSI Design*, New York: McGraw-Hill, 1990.
7. A. Chandrakasan, S. Sheng, and R. W. Brodersen, Low-power CMOS digital design, 27(4), 473–484, April 1992.
8. K. Yano et al., A 3.8-ns CMOS 16 × 16-b multiplier using complementary pass-transistor logic, *IEEE J. Solid-State Circuits*, 25(2), 388–395, April 1990.
9. N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Reading, MA: Addison-Wesley, 1993.
10. N. F. Goncalves and H. J. De Man, NORA: A racefree dynamic CMOS technique for pipelined logic structures, *IEEE J. Solid-State Circuits*, 18(3), 261–266, June 1983.

9.4 Microprocessor-Based Design

Roland Priemer

9.4.1 Introduction

During the past three decades, microprocessors have become components that are routinely and widely used in machines and systems that engineers design. This is due to their flexibility and ability to perform tasks at a low cost. Because they are programmable, microprocessors are used to achieve operations of devices and systems with complexity that we have come to take for granted. Engineers are embedding microprocessors into systems that are being employed in virtually all fields of human endeavor.

The competition among the numerous manufacturers of microprocessors has brought about a great variety of microprocessors to increase their suitability in ever-widening fields, more products, and new markets. Moreover, turnaround times have become short enough and costs have come down enough so that system and application engineers have the option to specify the design of a customized microprocessor to meet their application-specific performance requirements.

This section is intended to introduce the reader to design with a microprocessor. It is assumed that the reader is acquainted with the material that is generally covered in an introductory course on digital systems. The goal is to help the designer who is not experienced with the design of microprocessor-based systems to come to a basic understanding of what is involved. Two representative and significantly different microprocessors will be used to do this. These are the Zilog Z80, an enduring general-purpose microprocessor, and the Motorola M68HC11, also called a microcontroller. These and similar microprocessors have been used in many diverse kinds of dedicated applications.

The design and development of the hardware and the software are two broad aspects to utilizing a microprocessor. Here, the emphasis is on digital hardware design. To understand the role that a microprocessor plays in the design process we must understand the operation of several major building blocks. These are memory, architecture of a microprocessor (programmer model), the system bus and timing of bus signals, and supportive devices to interface these building blocks together. At the level of design to be considered here, hardware and software are and should remain inseparable. From both points of view, we gain a better understanding of the other. However, due to space limitations, no significant presentation of programming in assembly language is included in the sequel. Instead, software is only utilized to see how it influences hardware requirements and how it makes hardware work.

9.4.2 Features of a Microprocessor-Based System

Conceptually, Figure 9.67 depicts a microprocessor-based system. The inputs to the microprocessor are all binary (can only take on one of two values) or digital signals. However, the origins and meanings of these signals can be very diverse. Inputs can be due to manually activated switch closures from buttons or keypads or they can be due to switches that are embedded within sensors that detect, for example, presence or absence of an object, pressure that is over or under a certain level, temperature that is above or below a particular temperature, presence or absence of an ultrasound or light beam, voltage that is above or below a desired value, etc. An input can come from a real-time clock. Inputs can also be information about the status or readiness of devices that receive outputs from the microprocessor-based system.

A single digital signal is called a bit, and it can only take on one of two logic values, i.e., logic one or logic zero, which in the circuits that we will use correspond to 5 or 0 V, respectively. Typically, these circuits are designed to accept as logic one any voltage in the range 3.5–5, and accept as logic zero any

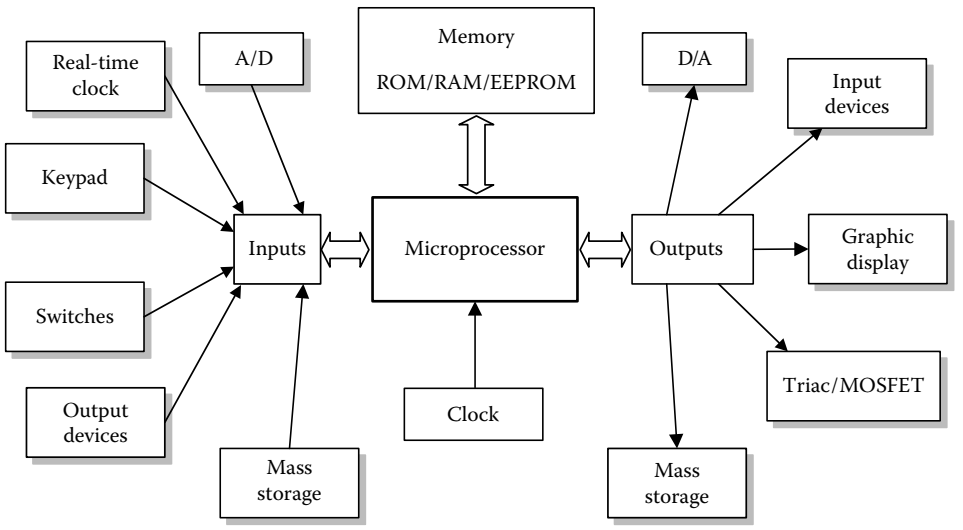


FIGURE 9.67 Conceptual diagram of a microprocessor-based system.

voltage in the range 0–1.5. Voltages in the range 1.5–3.5 will produce indeterminate results. Digital hardware using other voltage levels is also available. Of particular interest for portable applications are microprocessors using 3 V (or even 1.5 V) and 0 V. Unless explicitly noted, all signals are voltages as they vary with respect to a common reference.

Taken together, n-bits that are labeled, for example, with $d_{n-1}, d_{n-2}, \ldots, d_1, d_0$ can have any one of 2^n different binary combinations or binary assignments. A group of $n = 8$ bits is called a byte, which can take on any one of 256 different binary assignments. Another commonly used quantity is $2^{10} = 1024$, which is denoted by 1K. Thus, for example, we get $16\text{K} = 2^{14} = 16{,}384$.

Inputs can be analog in origin, and the microprocessor receives the result of sampling and analog-to-digital (A/D) conversion of an analog signal. Usually, analog signals are electrical signals generated by transducers that convert physical quantities of interest to voltages, such as, for example, speed of a motor, weight of an object, sound of someone speaking, stress of material under test, temperature in a building, electrocardiogram of a patient, acceleration of some structure, wind direction, oil pressure in an engine, etc. The analog signal can be the output of a receiver in a wireless communication application. The possibilities are endless. Thus, a microprocessor (the program it is executing) can receive information about time and the state of the world in which it is intended to operate.

The memory module of Figure 9.67 serves several purposes. First of all, it contains the binary instruction codes of the program that is executed by the microprocessor. The program instruction codes are another kind of input to the microprocessor. Memory may also be used for temporary storage and later retrieval of data produced by the program as it executes the algorithms that process the inputs. And, memory may contain various kinds of data tables that are referenced according to the instructions of the executing program. In contrast to the memory module, which must respond in time comparable to microprocessor speed of operation, slower mass storage devices also provide inputs that may be programs that will be transferred to memory for subsequent execution or that may be data that will be utilized by an executing program.

An important feature of using a microprocessor is that the functionality of an application is achieved mainly with software and the interaction through hardware between the software and the application environment. From a mass production viewpoint, this is almost a matter of trading hardware, a repeated expense, for software, a one-time expense, whenever possible. Thus, we are not concerned to design a general-purpose computer, but instead, we want to design the hardware that is sufficient to support the software that comprises the functionality of an application. Furthermore, using a microprocessor can achieve more than this, because with programmed hardware we can realize fundamental tasks such as conditional checking, storage, waiting, arithmetic, and timing, to mention a few, and higher level tasks such as complex algorithm calculations, decision making, control of machines, management of data, complex graphical interaction with the system user, and so on. When programmed properly, a microprocessor provides the ability to perform tasks that are difficult or impossible to achieve with any other kind of means or hardware.

The outputs from the microprocessor in Figure 9.67 are all digital signals. Like inputs, the meaning and destination of outputs can also be very diverse. Outputs can be one bit quantities that are intended to turn on and off, for example, light and sound indicators and devices such as, for example, TRIACs that can control AC power supply to equipment like AC motors, lights, heaters, and transformers, and, for example, power MOSFETs that can control DC power supply to equipment like solenoids, relays, DC motors, and lights. Outputs can be complex data structures that support graphical interaction between the user and the microprocessor-based system. As for inputs, here too the possibilities are endless. Outputs may be necessary to control the input devices or inform them of the readiness to receive inputs.

Outputs can also be 8, 16, or other multi-bit quantities that are, for example, printer control and text character codes, data to be placed in mass storage, a binary value to be transferred to a digital-to-analog (D/A) converter to synthesize an analog signal, data to be placed in memory, codes of characters to be displayed, data to be wirelessly communicated, protocols to establish communication links, and many other possibilities.

The clock module produces a voltage pulse train, called the system clock. Since microprocessor activity occurs due to level changes of the system clock, the speed at which the microprocessor executes instructions depends on the frequency of this pulse train. For example, if some particular microprocessor can fetch and execute an instruction in 5 clock cycles, and it can be clocked with a frequency of 250 MHz, then the microprocessor can execute the instruction in 20 ns. For most microprocessors, the number of clock cycles required to fetch and execute an instruction varies with the kind of instruction.

With all of the above modules interconnected to achieve a particular task, it will be the activity of the microprocessor, which is determined by the software that it executes, that controls the acquisition of inputs, processes the inputs, implements the intended functionality of the system, and generates and controls the placement of outputs.

Figure 9.68 shows a more functional structure than Figure 9.67 of a microprocessor-based system. An important feature of this structure is the use of a set of common communication paths, called the system bus, between all major modules. To identify any particular device, the microprocessor places a binary number, called an address, on the m-bit address bus. Actually, the electronic circuitry that comprises the microprocessor sets the voltages on the m address bus conductors to some combination of 0 and 5 V. There are 2^m different addresses that can be placed on the address bus. Every other module receives the address, and only the particular device that recognizes its own address, which is assigned and incorporated by the system designer, is supposed to respond. The data bus simultaneously transfers n-bits of data from the microprocessor to any device or from any device to the microprocessor. There are microprocessors with the number of address bits ranging from $m = 10$ to $m = 64$ and the number of data bits ranging from $n = 4$ to $n = 64$. We will work with microprocessors that produce an $m = 16$ bits wide address, which permits $2^{16} = 65{,}536 = 64\text{K}$ different addresses, and that use an $n = 8$ bits wide data bus. The control bus informs all modules about the kind of activity that is presently occurring or is about to take place on the system bus.

Usually, the different modules in the system have different time and electrical and operating characteristics. It is the purpose of the interfaces to make compatible the modalities of these different modules. Often, it is interface design that is the focus of an overall hardware design effort. Depending on the kind of control signals provided by a microprocessor and the kind of control signals that are preferred or required on the control bus, a microprocessor may also require an interface circuit to the system bus.

There are numerous different standard system bus definitions. For example, the public availability of the definition of the IBM PC bus has permitted the connection of innumerable different accessories, produced by many different manufacturers, to this bus. Such accessories and associated software were designed to be compatible with the definition of this bus standard. Over the past three decades, such standardization has contributed to the wide success of the PC in the home and in the workplace.

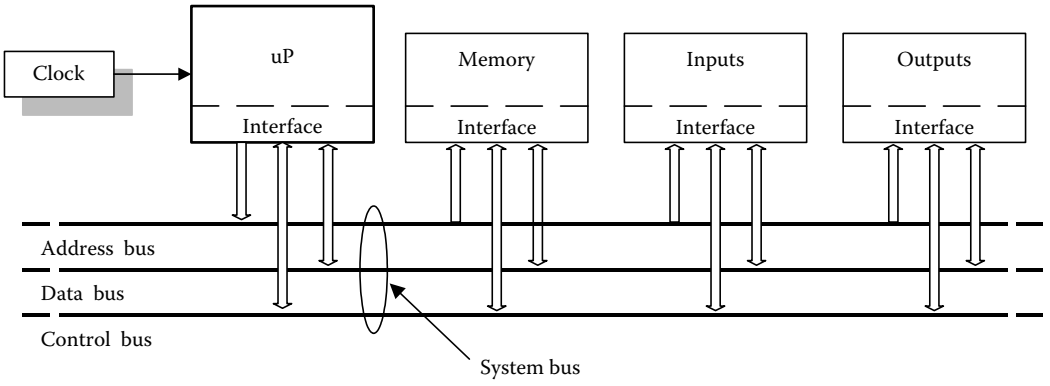


FIGURE 9.68 System block diagram.

However, in a dedicated application, adherence to a bus standard may not be necessary, particularly when there is no need to interconnect standardized modules. Here, we use the particular control signals of the microprocessor to form the system control bus as needed.

9.4.3 Memory

Almost all of the internal and external activity of a microprocessor depends on the ability to store, transfer, and retrieve data. A designer must understand how these operations take place. By way of presenting some memory devices, some terminology and supportive hardware are also introduced.

A circuit for a memory cell is shown in Figure 9.69. It is also called a binary cell (BC). Here, the activity of the flip-flop is determined by the logic signals at the points labeled *S*, for select, and R/$\overline{\text{W}}$, for read or write. The small circles attached to the OR gates and the buffers are called bubbles, and they mean logical complement. Thus, the buffer followed by a bubble is a NOT gate, and the flip-flop is made with two NOR gates. Another commonly used notation for logical complement is a small right triangle, as shown in the figure.

Note that we are using positive logic so that logic 1 means that a signal is high (5 V) and logic 0 means that a signal is low (0 V). When an operation or activity is enabled by taking a control signal high(low), then we say the control is active high(low). More briefly and without regard to the required level, an operation is enabled by asserting (or activating) its control signal(s).

The read and select control inputs of the BC are active high, while the write control input is active low, which is indicated by the bar over the *W*. To store a bit in the BC, the sequence of operations is (1) set the input, (2) set the R/$\overline{\text{W}}$ control signal low, and (3) momentarily select the cell. And, the write operation is complete.

A set of BCs that can be read or written in parallel is called a register, and the set of bits that a register can hold is called a word. A four words by 3 bits/word memory module is shown in Figure 9.70. The $m = 2$ bits input a_1a_0 is decoded with the 2×4 decoder, which has its own active low enable control. When enabled, the decoder output selects, according to its inputs, just one register (row of BCs) for a read or write operation. The binary assignment to a_1a_0 is called the address of the word to be referenced. For a read operation, the OR gates receive the bits of the selected word, while all other OR gate inputs are logic 0. For a write operation all BCs in a column receive the same input bit, while only the BC in the selected row accepts it. With additional columns, we can increase the register length to *n*-bits/word, and with an $m \times 2^m$ decoder we can have a memory size of 2^m registers, while only requiring *m* address bits to specify any particular memory register that is the object of a read or write operation (memory reference).

Figure 9.70 implies that the system data bus must consist of an *n*-bit unidirectional input bus to transfer data from memory to the microprocessor and an *n*-bit unidirectional output bus to transfer data from the microprocessor to memory. Practically, to reduce the number of communication paths, an *n*-bit bidirectional system data bus is preferred. This can be accommodated by connecting to each memory I/O bit pair, say the *i*th pair, an arrangement of tristate buffers as shown in Figure 9.71, where d_i

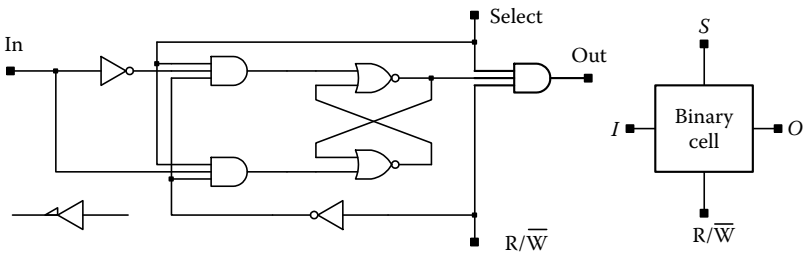


FIGURE 9.69 Circuit of a BC with select and R/$\overline{\text{W}}$ control.

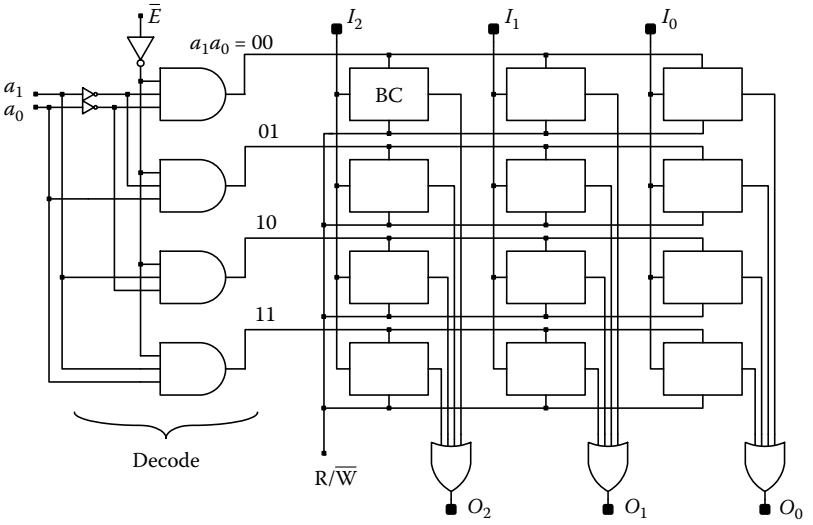


FIGURE 9.70 A multiword read/write memory circuit.

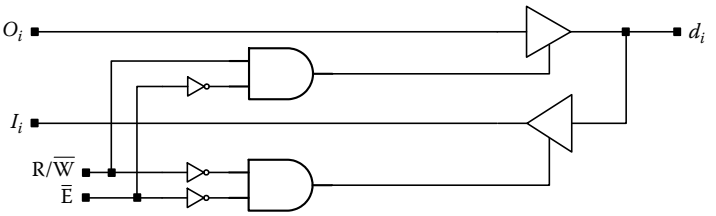


FIGURE 9.71 Unidirectional to bidirectional bus conversion.

is connected to the *i*th bit of the system data bus. When both buffers are in tristate (high impedance state), which occurs when $\overline{E}$ is high, then d_i is independent of O_i and I_i. Thus, the memory module can be electrically disconnected from the system bidirectional data bus. When $\overline{E}$ is asserted, then R/$\overline{W}$ controls the direction of data transfer. Figure 9.72 shows a more concise notation for the entire memory module. Here, the two address lines are grouped into one bus, and the three bidirectional data lines are grouped into another bus.

Since there is no particular required address sequence for reading or writing data to this memory, it is called random access memory (RAM). Since stored data remain intact as long as power is supplied to the circuit, this memory is called SRAM, contrary to another memory type, called dynamic RAM (DRAM),

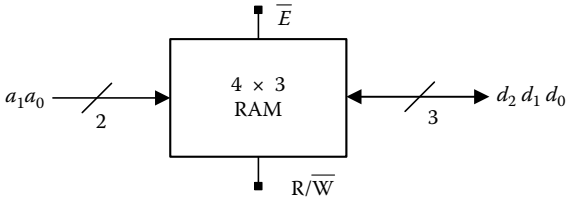


FIGURE 9.72 A memory module.

where the logic content of a BC is not based on the state of a bistable circuit, but is instead based on the presence or absence of charge on a cell capacitor. Since charge on capacitors dissipates, DRAM must be periodically refreshed, which requires additional control circuitry. However, DRAM can be fabricated to achieve much higher bit densities than of SRAM, resulting in the substantially lower cost of DRAM.

RAMs in various dimensions and package types are available. Commonly available SRAM chips range in size from 256×1 to $1024\text{K} \times 8$ (a 1 megabyte RAM), and the function and number of the control signals also vary a little. Commonly available DRAM chips range in size from $16\text{K} \times 1$ to $512\text{M} \times 1$ (a 512 megabit RAM). Important parameters for each memory type are the read and write cycle times, which must be less than the width of read and write time windows allowed by the devices that will reference this memory.

Larger RAM modules are constructed by the cascade and parallel connection of smaller RAMs. This is illustrated in Figure 9.73, which shows a $32\text{K} \times 8$ RAM made from $16\text{K} \times 4$ RAMs. If $\overline{E}$ is asserted, which takes the entire module out of tristate, then address bit a_{14} enables either RAMs 1 and 2 or RAMs 3 and 4 to access the data bus. The remaining address bits select a particular 4 bit register within each $16\text{K} \times 4$ RAM. Here, RAMs 1 and 3 hold the upper half of each data byte, while RAMs 2 and 4 hold the lower half of each data byte.

Whenever power is removed from the circuits of SRAM or DRAM, the data content is lost, and such memory devices are said to be volatile. To provide software for execution immediately after power is applied to a microprocessor, ROM is used. Figure 9.74 shows eight words by 2 bits/word ROM that has been, for example, programmed with the data given in the table. By opening, which is indicated with an *x*, the connections called links the ROM is programmed. These links remain open through power-down and power-up cycles, and thus, a ROM is said to be nonvolatile. The tristate buffers are controlled by the OE signal so that the ROM output can be electrically disconnected from a system data bus, which may then be used for transfer of data among other devices. An important parameter of a ROM is its read cycle time.

By providing a manufacturer a table of contents, a ROM can be programmed at the time of fabrication. Blank (unprogrammed) ROMs are available that can be one time programmed (OTP) in the field. These are called PROMs for PROM. There are also PROMs that can be erased by exposing the IC to a certain level of ultraviolet light for several minutes. Erasing is made possible by using an IC package with a clear

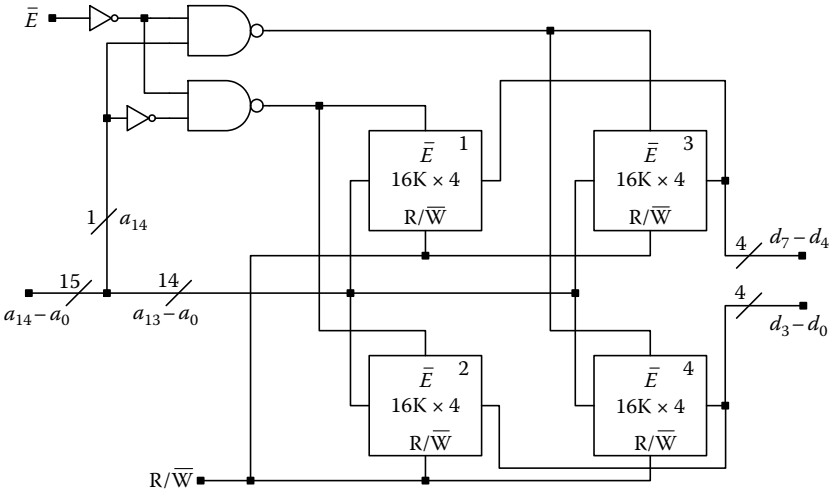


FIGURE 9.73 Cascade and parallel construction of a RAM module.

Address			Data	
a_2	a_1	a_0	O_1	O_0
0	0	0	0	1
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

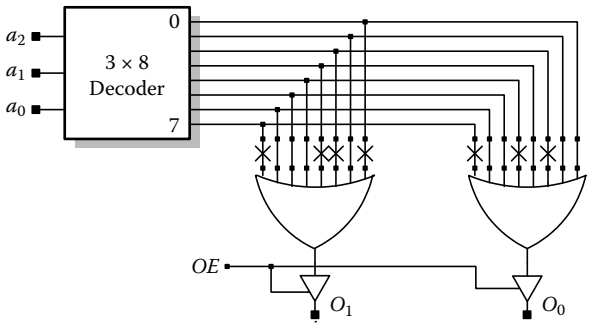


FIGURE 9.74 An eight words by 2 bits/word programmable ROM.

window just above the IC. These are called EPROMs for erasable PROM. After an EPROM has been erased, it can be programmed again.

Another EPROM type is the EEPROM that can be electrically erased by essentially overwriting previously stored data. However, the write cycle time is significantly longer than the write cycle time of conventional RAM. Commonly available PROMs and EPROMs range in size from 2K × 8 to 1024K × 8 (a 1 megabyte ROM). Commonly available EEPROMs are not as large as other ROMs, and they cost more than other ROM types. Like RAM modules, larger ROM modules can be made by combining smaller ROMs in a manner illustrated in Figure 9.73.

A recently developed memory type is the flash EEPROM, or flash memory. Its fabrication methodology is relatively inexpensive, and yields high-density memory modules having large data storage capacities. To rewrite to it, it must first be erased in bulk. These memories are used in, for example, digital cameras and portable flash mass storage devices.

Figure 9.75 shows an 8K × 8 EPROM and the package pin assignment. To place a byte into a particular ROM register, or program the ROM, the register address and intended content must first be supplied at

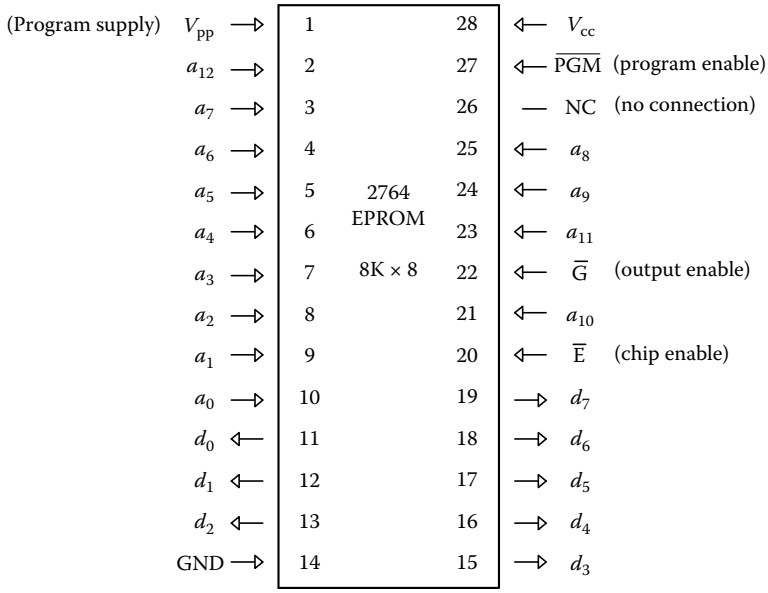


FIGURE 9.75 Pin assignment of a typical (the 2764) EPROM.

the address and data pins, respectively. Then, in addition to V_{CC}, the V_{PP} supply must be provided, the chip enable is asserted, and the $\overline{PGM}$ input is activated for a particular time duration. An instrument called a PROM programmer is used to do this. Some PROM programmers work by connection to a computer, and, in conjunction with software running on the computer, it can program a great variety of PROMs, which are selected from a menu of those PROMs supported by the software, with code in a file that it receives from the computer.

A programmed ROM can serve several purposes. Most computers have a ROM that contains program code that is always executed at power-up. Typically, this software, or firmware, is a utility (called a boot) that transfers software from a mass storage device to RAM. Then, after this transfer is complete, or perhaps after some additional ROM resident initializing code has executed, the software in RAM starts to execute. In a dedicated application, such as, for example, the microprocessor-based systems that control a car engine, it is more likely that ROM will contain all of the application software ready for use whenever the system is powered-up.

ROM is also used for other purposes. For example, by connecting three variables to the address inputs of the ROM given in Figure 9.74, this ROM can, through programming, be used to realize any two Boolean functions of three variables. ROMs are used to hold tables of data. For example, multiplication can be performed through table look-up. If the 4 bit binary codes of two digits that we want to multiply are used to form an 8 bit address of, for example, a 256×8 ROM, then the upper and lower halves of the retrieved data could each be the 4 bit binary codes of the product digits. The idea is that the address is formed with the input(s) of an operation (or argument of a function), and the precomputed and stored data are the output of the operation (or value of the function). EEPROMs are often used to hold tables of data that are generated during program execution, which must be available after a power-down and power-up cycle. A battery-backed RAM can also be used for such purposes.

Another way to use a ROM is for signal generation. By using a counter to supply an address sequence to a ROM, each bit of the data as it is clocked out of the ROM could be used as a control signal of some process. The frequency of the clock that drives the counter determines real time, and the variation between 0 and 1 of any particular data bit determines the resulting control signal shape. On the other hand, if the data word sequence coming from a ROM is inputted to a D/A converter, then a ROM can be used to generate an arbitrarily shaped analog signal that is repeated each time the counter repeats its count sequence.

The kind of memory that is used, its size, and the addresses to which any particular memory will be responsive must meet the requirements of an application. This information is often given in the form of a system memory map. The size of this map is the range of addresses that a microprocessor can specify.

With $m = 16$, an address in binary is denoted by $a_{15}a_{14}\ldots a_1a_0$. The most significant address bit a_{15} splits the entire 64K word memory space into two 32K word blocks, and 0xxxxxxxxxxxxxxx, where x means do not care, is any address in the lower 32K word block, while 1xxxxxxxxxxxxxxx is any address in the upper 32K word block. Similarly, address bits $a_{15}a_{14}$ together split the entire memory space into four 16K word blocks. If $a_0 = 0$, then the address is an even number, while if $a_0 = 1$, then it is an odd number.

Sometimes, it is more convenient to use hexadecimal notation. The symbol \$ will be placed in front of all numbers written with hexadecimal notation. Thus, \$XXXX is any 16 bit number in hexadecimal notation, while \$XX is any 8 bit number. As the most significant hexadecimal address digit changes by \$1, the address changes by 4K.

Suppose a design must include 8K bytes of EPROM starting at address \$0000, 1K bytes of EEPROM starting at address \$4000, and 32K bytes of RAM starting at address \$8000. The ROM, EEPROM, and RAM must be responsive to addresses in the ranges \$0000–\$1FFF, \$4000–\$43FF, and \$8000–\$FFFF, respectively.

Since the three most significant address bits split the memory space into eight 8K word blocks, the ROM should be enabled when these address bits are $a_{15}a_{14}a_{13} = 000$, and then the remaining address bits $a_{12}a_{11}\ldots a_1a_0$ covering an 8K word space are the address input to the ROM. Similarly, the EEPROM

should be enabled when $a_{15} \ldots a_{10} = 010000$, and the remaining address bits $a_9 \ldots a_1 a_0$ covering a 1K word space are the address input to the EEPROM. The RAM should be enabled when $a_{15} = 1$.

Thus, the size and desired position of a memory device determine how to place it in the memory space. The other important aspect of interfacing memory to the system bus and eventually to a microprocessor is that a memory module must be responsive to control signals issued by the microprocessor to present and accept data in certain particular time windows. This will be considered further when we look at the timing of microprocessor control signals. An alternative mode of operation would be for the microprocessor to automatically wait for a response signal from memory whenever it references memory before completing the memory reference cycle.

A memory map for this design is shown in Figure 9.76. Here we see the size, kind, and position in the memory space of the microprocessor of actual memory devices. When possible, it is also useful to describe the location and purpose of particular program code modules.

Assuming that the EPROM contains program code, some example instructions have been placed at the beginning of this memory. These are binary numbers that are machine instruction codes for the Z80 microprocessor. The Z80 microprocessor is a type of processor that after being reset, fetches instruction codes starting at address $0000. The first instruction, i.e., $ED $56, is the Z80 machine code that selects its method 1 for responding to interrupts. Another commonly used notation for hexadecimal is to attach the suffix H. The next instruction, i.e., 31H, is the machine code that loads the stack pointer (SP) register with an address given by the next 2 bytes, i.e., $F000. These instructions could be the part of a program that initializes the processor. To understand how an instruction is processed, it is useful to study the architecture of a microprocessor.

For programming convenience, programs are usually written using mnemonics of instruction codes that are indicative of instruction activity. The set of mnemonics and associated notational convention for all of the instruction codes that a particular microprocessor can execute form the assembly language of the microprocessor. Different microprocessors, with different instruction sets have different assembly languages. Furthermore, different manufacturers adopt different mnemonics

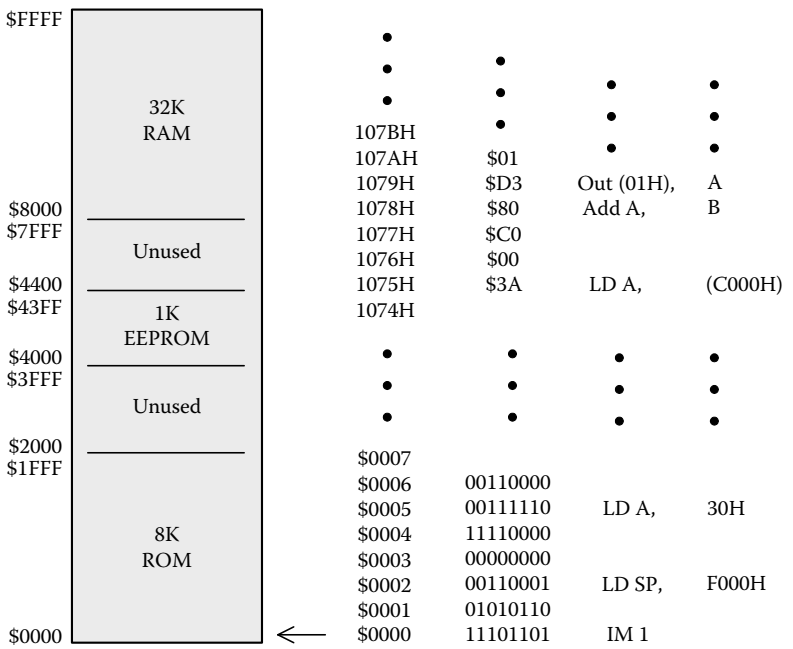


FIGURE 9.76 Memory map, sample ROM content, and assembly language source.

for their microprocessor instruction sets, causing another variation among assembly languages. However, in principle, there remain many common attributes from one assembly language to another. A program written in assembly language must be converted into machine code for storage and eventual execution. A program that performs this conversion is called an assembler. From the viewpoint of the assembler, an assembly language program is an input character string, and the machine code output is another character string.

9.4.4 Microprocessor Architecture

Employing a microprocessor in a dedicated application does not require detailed knowledge about its internal behavior. However, it is useful to have some insight about how instructions, i.e., their codes, are processed (executed) by the hardware. Understanding the relationship between software and hardware can affect the selection of a particular microprocessor, and the design process. Moreover, the hardware designer should also understand the programming model of a microprocessor.

From a programmer's viewpoint, a microprocessor is defined by its programming model and its instruction set, which together comprise the architecture of the microprocessor. The programming model consists of the set of internal registers that are involved in the execution of operations as specified by the instruction set. These registers do different specialized tasks. The purpose, capability, and number of these registers can vary greatly from one microprocessor to another.

Basically, within the programming model, microprocessors have four kinds of registers. There are address registers that are used to form and hold addresses to be used for referencing memory and other devices to obtain program instruction codes and their operands and to specify source and destination memory locations for data read and write operations. There are data registers that can be the source of data for an operation or the destination for the result of an operation. There are operational registers that have associated hardware to perform, for example, logical and arithmetic operations. And, there are status/control registers that configure the operation of the microprocessor and support different kinds of conditional instructions.

An operational register possessed by most microprocessors is the accumulator. It is commonly denoted by register A. Figure 9.77 illustrates how an A register functions within a microprocessor. Associated with reg. A are the ALU and the condition code register (CCR) or status register (SR). Inputs to the ALU can come from reg. A, reg. B, and the CCR, and its activity is determined by the function select word $f_{k-1} \ldots f_1 f_0$. Reg. A receives its input from the ALU, and reg. B receives its input from the n-bit internal data bus. The activity of this circuit is determined by the control signals that are applied at all of the points labeled with triangles. For example, by asserting the E, for enable, input of reg. A, the n-bit word coming from the ALU is latched (loaded) into reg. A. The content of reg. A can be placed on the internal data bus by asserting the control signal of the n tristate buffers connected to its output. The content of the internal n-bit data bus can be latched into reg. B by asserting its E control input.

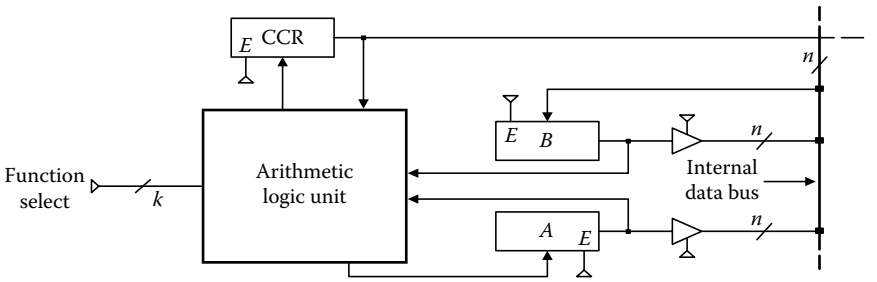


FIGURE 9.77 Register-to-register transfer activity of an accumulator.

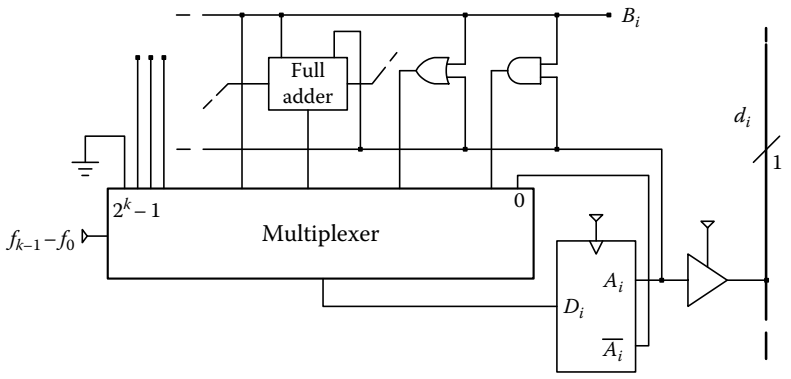


FIGURE 9.78 Bit level activity of an ALU.

Figure 9.78 illustrates how the ALU performs its task at the gate (bit) level. Notice how the function select lines determine, as they do for the MUXs of all the other bits, which MUX input is latched into the *i*th flip-flop of the accumulator when its *E* control input is asserted. For example, if $f_{k-1} \ldots f_0 = 0 \ldots 0$, then the accumulator will be complemented, as if the microprocessor has just read in the machine code for the complement accumulator instruction, which has the Z80 assembler mnemonic CPL. Thus, the machine codes of instructions such as ADD A, B; RLA; INC A; LD A, 80 H; etc., eventually determine the binary assignments of these function select lines to accomplish the instruction tasks. There can be as many as 2^k different operations involving the accumulator that can be achieved by this ALU structure.

All of the control bits required in Figures 9.77 and 9.78 come from memory called control ROM or microstore. Figure 9.79 illustrates the data paths of a microprogrammable microprocessor. Each word, consisting of perhaps 16–128 bits, in control ROM is called a microinstruction. The busing of microinstructions throughout the microprocessor is not shown in Figure 9.79. Instead, these interconnections are indicated by labeling with the small triangles.

Figure 9.79 also illustrates a level of design called the register transfer level. There is still another level of greater detail before we get to a level of IC design detail that shows the interconnection of transistors, resistors, diodes, and conductors. This is the logic gate level. However, at that level, the detail of design would probably be too much and detract from an understanding of microprocessor operation. Here the intent is to exemplify how hardware processes an instruction code.

Each microinstruction is partitioned into a set of fields, and one of these fields holds the accumulator function select line assignment. Other fields hold data that signify: (1) which registers are the source of data for the internal and external buses; (2) which registers are the destination of data on the internal and external buses; (3) register activity control such as clear, increment, decrement, load, and tristate; (4) a *j*-bit address of the next microinstruction; and (5) the external control word that informs external devices of the present microprocessor activity. Each microinstruction coming out of control ROM can cause many activities to take place at the same time within the microprocessor. The other registers in the diagram of Figure 9.79 perform the following tasks.

MAR—Memory Address Register. It holds the address that the microprocessor can place on the external address bus, and it is loaded from the internal address bus by asserting its enable input.

PC—Program Counter. This register holds the address of the next instruction or instruction operand. It can be cleared, loaded, and incremented by asserting appropriate enable inputs. To fetch program code (either an instruction code or instruction operand), its content is transferred to the MAR. Usually, after each time its content has been used to fetch a byte of program code, its content is incremented.

SP—Stack Pointer. It holds an address that points to RAM that can be used for temporary storage. It can be loaded, incremented, and decremented by asserting appropriate enable inputs. Certain

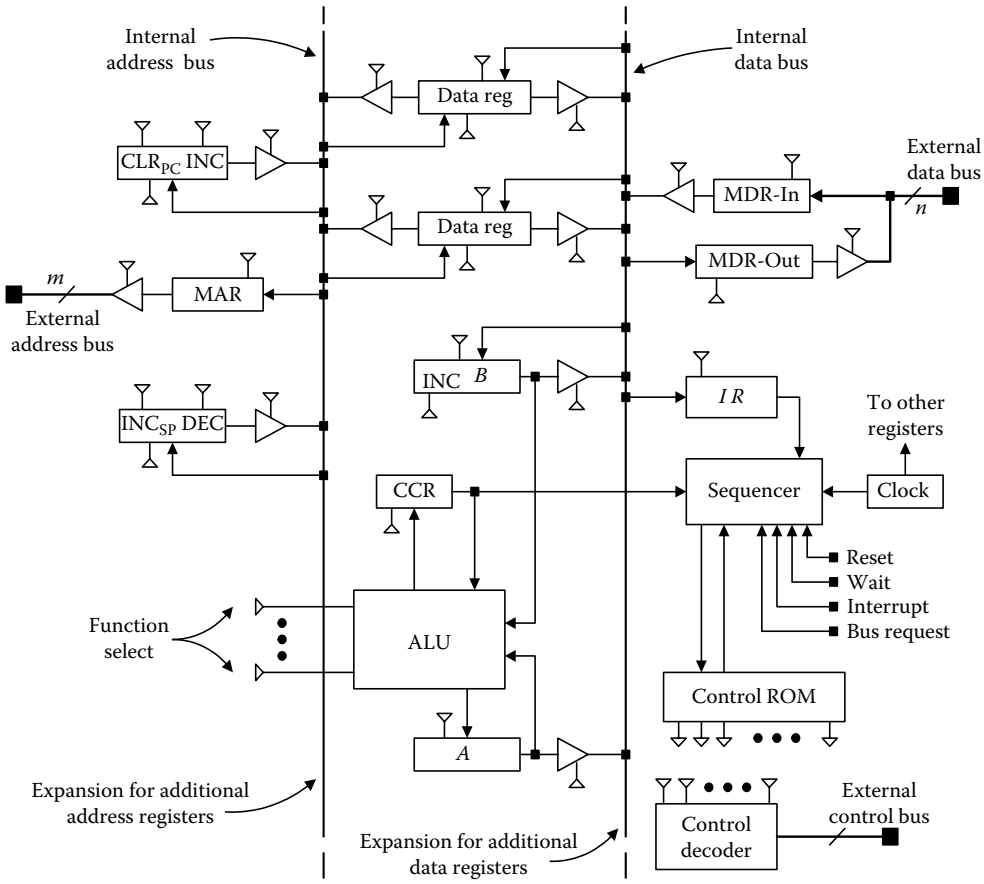


FIGURE 9.79 Data paths of a microprogrammable microprocessor.

instructions can cause its content to be transferred to the MAR for a memory read or write operation. The RAM that is referenced with the SP is called the stack. Typically, the SP is implicitly decremented before (after) writing to the stack, and it is implicitly incremented after (before) reading from the stack. Thereby, the stack is a last in and first out memory area that is used to support, among other things, subroutine calls.

MDR-IN—Memory Data Register IN. It receives its input from the external data bus, and it can drive the internal data bus.

MDR-OUT—Memory Data Register OUT. It receives its input from the internal data bus, and it drives the external data bus.

IR—Instruction Register. When the microprocessor is executing an instruction fetch from external memory, this register receives the content of the MDR-IN register, which is then assumed to be an instruction code.

CCR.—Each bit in this SR is indicative of the result of some previous microprocessor operation. The meanings of the flags (bits) in the CCR vary from one microprocessor to another. Typically, this register contains: (1) a carry flag that is set or reset depending on whether or not the previous add or subtract instruction produced a carry or required a borrow out of the most significant bit of the arithmetic operation, (2) a zero flag that is set if the previous instruction produced a zero result and reset if the instruction result is not zero, (3) an interrupt enable flag that can be set or reset by an instruction to

allow for software control over whether or not the microprocessor can respond to maskable interrupts, and (4) other flags. Usually, a microprocessor's instruction set includes program flow control instructions that are conditioned on these status flags such that if the flag is set (reset), then the instruction is executed, and if the flag is reset (set), then the instruction is not executed (skipped).

Sequencer. The sequencer consists mainly of a ROM and some sequential logic. It uses the instruction code and SR flags to form an address to its own ROM from which is obtained a *j*-bit microinstruction address that is applied to the control ROM. Furthermore, in response to asynchronous external control signals, such as reset, interrupt, bus request, wait, and others, it generates addresses of microinstructions that cause activity appropriate for these inputs.

Control Decoder. Depending on the input coming from control ROM, it generates external control signals that are intended to be used to inform external devices about the present activities of the microprocessor and to synchronize activities on the external address and data buses.

Data Register. This register is used for temporary storage of data. Usually, there are several registers like it. Some can also drive the lower byte of the internal address bus, while others can also drive the upper byte of the internal address bus. Still others can receive data from the upper or lower byte of the internal address bus. Some may have low-level arithmetic/logic capabilities.

The execution of each microprocessor instruction involves the execution of a set of microinstructions, called a microprogram. Thus, the functionality of a microprocessor's instruction set is determined by the microprograms stored in control ROM. After system reset or the completion of each instruction, unless an external asynchronous input is active, the microprogram that performs an instruction fetch is executed. It uses the content of the PC to point to the instruction, and it increments the PC so that the PC points to an instruction operand or the next instruction. Then, depending on the instruction code and SR content presented to the sequencer, a particular microprogram is executed to complete execution of an instruction. If an instruction code requires that operands be fetched, then the PC is further incremented so that, after an instruction has executed, the PC is pointing to the next instruction.

Microprograms for program flow control instructions such as JUMP or BRANCH elsewhere cause the PC to be loaded with the address operand of the instruction. Moreover, microprograms for instructions such as CALL or BRANCH subroutine first cause the SP to be used for storing in the stack the content of the PC before loading the PC with the address operand. Then, the microprogram for an instruction such as RETURN, which is used to terminate a subroutine, causes the SP to be used for retrieving from the stack the address of the instruction following the CALL subroutine instruction, which is then loaded into the PC.

This architecture can be expanded to include additional address registers and other special purpose address registers such as index registers, another PC and SP, additional accumulators and data registers, and so on. And, the widths of the address and data bus can be increased, contingent upon fabrication issues. Moreover, algorithmic instruction types can be supported since executing the desired activity of an instruction is a matter of writing a microprogram to accomplish all of the required register-to-register transfer activities.

9.4.5 Design with a General Purpose Microprocessor

The Z80 is a general purpose microprocessor, and it is available in a 40-pin dual in-line package (DIP). It is an 8 bit machine, i.e., its data bus is 8 bits wide. It has also evolved into a great variety of similar microprocessors. From a software development viewpoint, this machine has numerous features, as an examination of its extensive instruction set and programming model, which is given in Figure 9.80, will show. It has an accumulator (A), status or flag (F) register, six data registers (B, C, D, E, H, and L) that can be paired for use as address registers (BC, DE, and HL), an SP, a PC, two index registers (IX and IY), a register (I) that is used by its indirect vectored interrupt processing method, and a 7-bit counter register (R) that can be utilized for DRAM refresh. Furthermore, it can quickly change programming context by swapping with the alternate (primed) register set. Among its over 600 instructions, there are data transfer, arithmetic, logical and rotate, branch (or jump), stacking, I/O, program control, exchange,

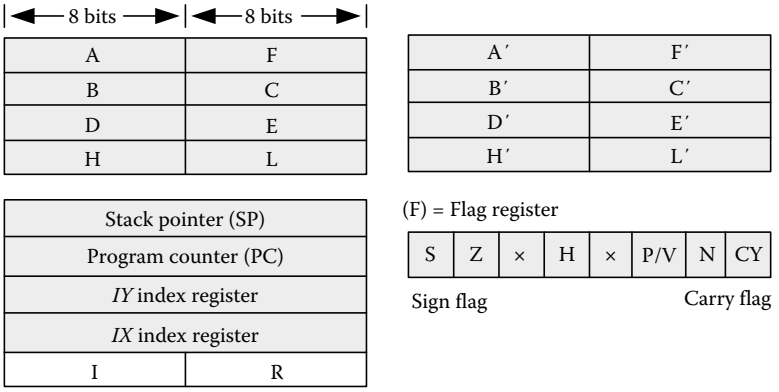


FIGURE 9.80 Z80 programming model.

block transfer, search, and bit manipulation instructions. Due to space limitations, the information given here is necessarily limited. Complete information can be found in textbooks or the data book from the manufacturer.

The pin assignment of the Z80 is shown in Figure 9.81. Power is supplied at pins 11 and 29, and a conventional crystal-controlled oscillator is used to provide the system clock ϕ at pin 6. If the load of each system address bus bit is only one or two TTL loads, then the outputs $a_{15}a_{14}\ldots a_1a_0$ can drive the system address bus. However, to accommodate a greater load, two unidirectional tristate octal buffers, as shown in Figure 9.81, can be used. These buffers also have hysteresis, which helps to produce sharper bus signals. Similarly, it is likely that the Z80 data signals $d_7d_6\ldots d_1d_0$ must be buffered to accommodate

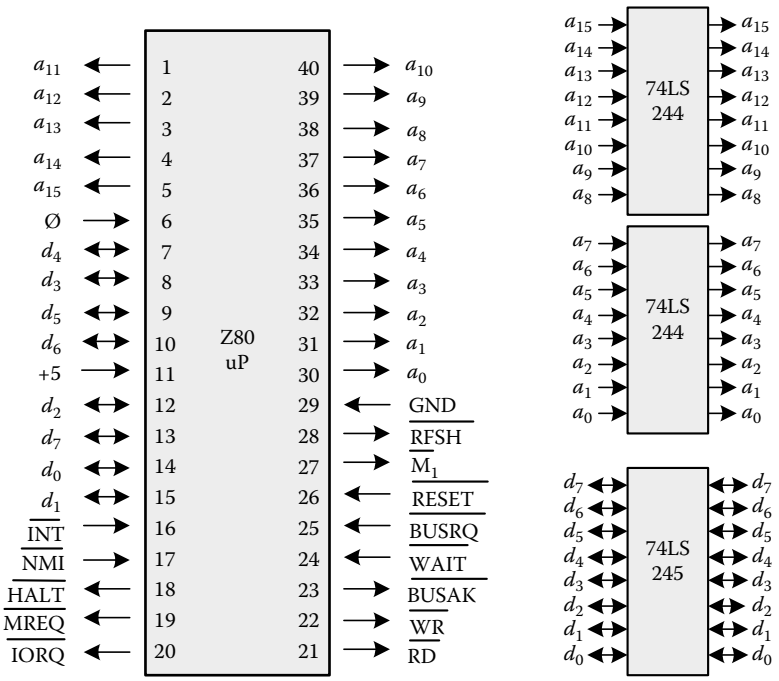


FIGURE 9.81 Pin assignment of the Z80 microprocessor.

system data bus loading. A bidirectional buffer, as shown in Figure 9.81, can be used for this. Z80 control signals must be used to control buffer direction. Also, define the system data bus by terminating it with pull-up resistors. The Z80 control signals perform the following tasks.

$\overline{\text{RESET}}$—This active low input resets the interrupt enable flag, clears registers I and R, causes all control signals to become inactive, sets the interrupt processing method to method 0, and clears the PC. When this input is released, the Z80 starts to fetch the first instruction code (op-code) from the memory register with address held by the PC.

$\overline{\text{MREQ}}$—This output becomes active whenever the Z80 is performing an operation, such as op-code fetch or instruction operand fetch that references an external device with a 16 bit address. It indicates that the address bus holds a valid address. The devices that respond are said to be positioned in the memory space.

$\overline{\text{IORQ}}$—This output becomes active whenever the Z80 is executing either an IN, for input, or an OUT, for output, instruction, both of which reference external devices with an 8 bit address that is placed on $a_7 \ldots a_0$. It also indicates that the address bus holds a valid 8 bit address. The devices that respond are said to be positioned in the I/O space, which is separate from the memory space.

$\overline{\text{RD}}$—This indicates that an external device must place valid data on the data bus, which the Z80 will soon accept.

$\overline{\text{WR}}$—This indicates that the Z80 is driving the data bus with valid data.

$\overline{\text{M}}_1$—This output is active while the Z80 is fetching an op-code. The only other time it becomes active is in response to a maskable interrupt.

$\overline{\text{HALT}}$—This output becomes active when the Z80 has stopped fetching additional instructions due to having executed a HALT instruction. The microprocessor can only continue instruction execution upon activation of an interrupt input.

$\overline{\text{WAIT}}$—This input, when active, causes the Z80 to hold constant its address and control signal outputs until this input is released (no longer active). When referenced, a slow memory or I/O device can cause the Z80 to wait by asserting the $\overline{\text{WAIT}}$ input until sufficient time has elapsed to allow the device to respond to the reference.

$\overline{\text{RFSH}}$—When active, this indicates that $a_6 \ldots a_0$ holds the content of counter register R, which, along with an active $\overline{\text{MREQ}}$ can be used to refresh dynamic memory.

$\overline{\text{NMI}}$—Whenever this nonmaskable interrupt input goes through an active low edge, an internal flip-flop is set, and other interrupts are disabled. The interrupt is said to be latched. At the completion of every instruction, this flip-flop is checked, and if it is set, then the Z80 first stacks the PC and then it loads the PC with \$0066, where the first op-code of an interrupt service routine must be located. This way of loading the PC in response to an interrupt is called a direct interrupt, and it is intended for an unconditional and fast response to, for example, battery low detected, temperature too high detected, or some other urgent situation, since it requires no further action from the interrupting device.

$\overline{\text{INT}}$—At the completion of every instruction the Z80 checks this maskable interrupt input. If it is active and interrupts have been enabled with the EI instruction, then interrupt processing commences according to the interrupt method stipulated by the IM i, $i = 0$, 1, or 2, instruction. If $i = 1$, for the direct method, the PC is first stacked, and then it is loaded with 0038H, the address used to obtain the first op-code of an interrupt service routine. If $i = 0$, for the vectored method, the Z80 acknowledges the interrupt by asserting the $\overline{\text{IORQ}}$ line while the $\overline{\text{M}}_1$ signal is active. This combined activity is then interpreted externally to be an interrupt acknowledge signal, denoted by $\overline{\text{INTA}}$. In response to the $\overline{\text{INTA}}$ signal, the interrupting device then has the opportunity to place the op-code for the one byte Z80 instruction RST, N on the data bus, where $N = 0, 1, \ldots, 7$. Then, the Z80 stacks the PC and loads it with an address given by 8 N. Thus, depending on N, a 3-bit code embedded within the RST instruction, the first op-code of

the interrupt service routine can be located in one of eight memory locations. If $i = 2$, for the indirect vectored method, the PC is first stacked, and then the Z80 acknowledges the interrupt by asserting the $\overline{\text{INTA}}$ signal. In response to the $\overline{\text{INTA}}$ signal, the interrupting device must then place a byte given by xxxxxxx0 on the data bus. The Z80 then uses this byte as the low address byte and the content of the I register as the high address byte to fetch two consecutive bytes from memory that are then loaded into the PC. Thus, the I register points to a 256 byte block of memory, where there can be 128 interrupt vectors one of which is selected by the byte provided by the interrupting device.

$\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$—The first signal is an input that indicates to the Z80 that an external device wants to take control of the system bus. The Z80 completes execution of the present machine cycle, takes its address, data, and tristate control signals into tristate, and acknowledges the request with an active $\overline{\text{BUSAK}}$ signal.

Most of these signals are representative of the kinds of control signals that microprocessors have. Their usefulness becomes more apparent as we look at the timing of microprocessor activity and the impact this has on the design of hardware so that the microprocessor can accomplish read and write data transfers. Figure 9.82 shows the Z80 timing diagram for an op-code fetch, and Figure 9.83 shows timing diagrams for the other memory and I/O references. For the purpose of studying the timing of events, we do not have to know actual binary assignments on the address and data bus lines. Therefore, the

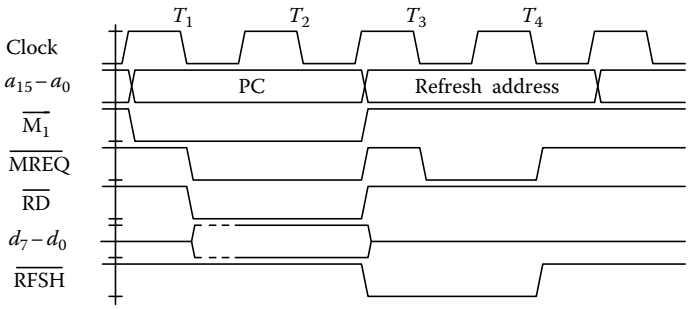


FIGURE 9.82 Z80 timing diagram for an op-code fetch.

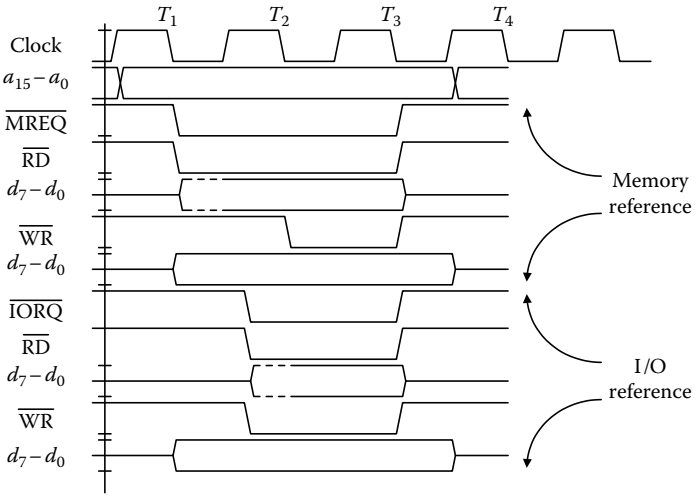


FIGURE 9.83 Z80 timing diagrams.

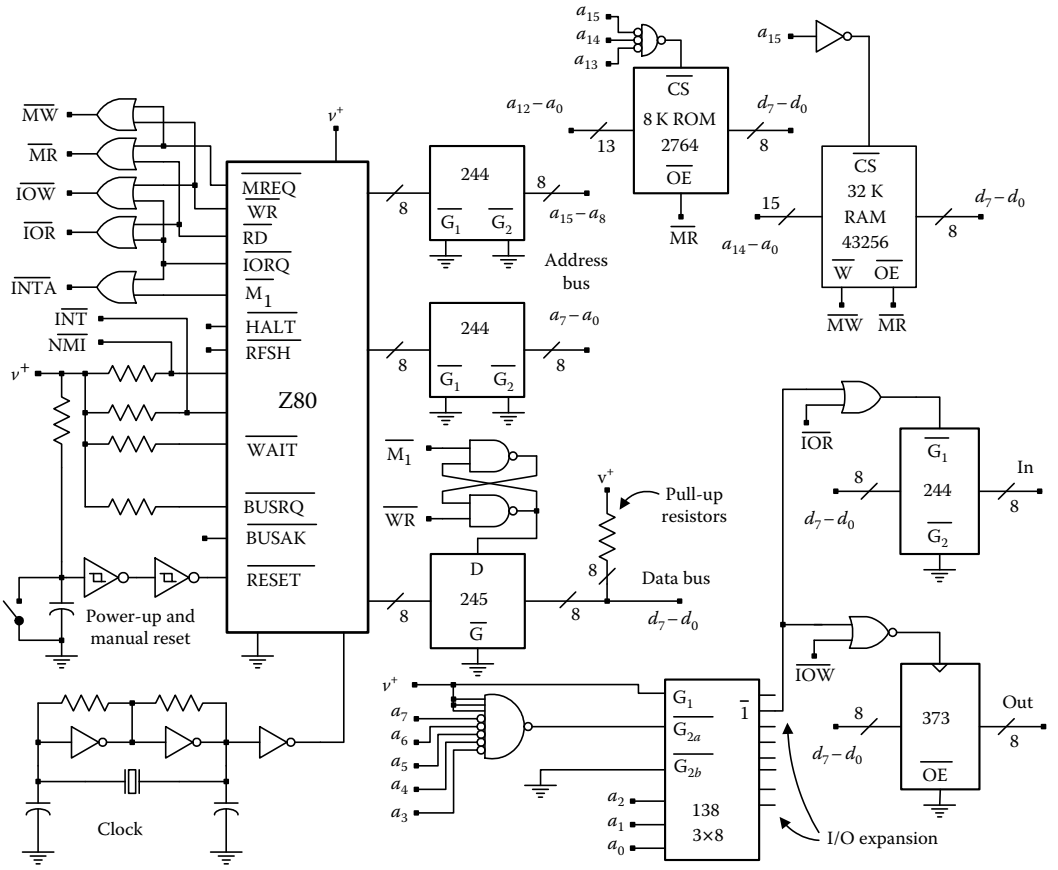


FIGURE 9.84 Z80-based microcomputer.

convention in these diagrams is intended to indicate when these signals either become relevant for the operation at hand or switch to the present binary assignment, whatever it may be, from some previous binary assignment.

As we look at the timing diagrams, it will also be useful to see how these signals are used. Figure 9.84 shows a schematic of a microcomputer designed according to the memory map of Figure 9.76. Whenever possible, labeling is used to indicate connections.

Depending on the instruction, the Z80 uses from 4 to 23 clock cycles to execute an instruction. A group of clock cycles within the execution time of an instruction that accomplishes a major activity is called a machine cycle. Each clock cycle within a machine cycle is labeled with T_i, $i = 1, 2, \ldots$, and all activities occur at either leading or trailing clock edges. To better understand such groupings, suppose that after executing many instructions that were retrieved from ROM, the Z80 is about to execute the next few instructions located in memory shown in Figure 9.76.

The PC is set to PC = 1075H, pointing to the op-code at memory location 1075H. This is the op-code for load accumulator with data using the two bytes that follow the op-code to form an address to point to the data, i.e., LD A (C000H). This is a 3 byte instruction, and, according to the Z80 manual, it requires 13 clock cycles, which are grouped into four machine cycles, to execute.

Referring to Figure 9.82, at the leading edge of the first clock cycle T_1 of the first machine cycle of this instruction the Z80 asserts the $\overline{M_1}$ signal, and it drives the external address bus with the address 1075H. At the trailing edge of clock cycle T_1, it asserts the $\overline{MREQ}$ signal and the $\overline{RD}$ signal. At the time that $\overline{RD}$

becomes active, the Z80's data bus input register is enabled to follow the content of the external data bus. At the leading edge of clock cycle T_3, the Z80's data bus input register is disabled from following the external data bus, and the content of this register is accepted as an op-code.

During the time interval from the trailing edge of T_1 until just before the leading edge of T_3, which is slightly less than 1.5 clock cycles, external hardware has the opportunity to place the addressed data, which will be processed as an instruction op-code, on the data bus. A designer must ensure that external hardware is fast enough to do this in a timely manner. If it is necessary to use an external device with an access time that is greater than this allotted time, then a counter, called a wait state generator, can be used. Once the slow external device detects (by address and control signal decode) that it is supposed to put valid data on the data bus, then it enables the wait state generator, which should be driven by the system clock, to activate the $\overline{\text{WAIT}}$ input of the microprocessor for a number of clock cycles that will give the device an opportunity to place data on the data bus. Each such clock cycle is called a wait state (labeled with T_w), and the number of wait states needed will depend on the access time of the device being referenced. If several external devices need to generate different numbers of wait states, then the outputs of the wait state generators of all such devices can be OR'd to present just one wait input to the microprocessor. Thereby, the machine can run as fast as each different external device will allow.

During the next 2 clock cycles, the received op-code is interpreted so that the Z80 becomes set to perform two more memory references, called machine cycles M_2 and M_3, to fetch each byte of the address operand. Machine cycle M_1 consists of 4 clock cycles, and while the Z80 is processing a retrieved op-code during clock cycles T_3 and T_4, the address bus is available for DRAM refresh. During machine cycle M_4, the address C000H, which was obtained during M_2 and M_3, is placed on the external address bus, and execution of this instruction is completed by transferring the content of memory location C000H to the accumulator. Machine cycles M_2 and M_3 each require 3 clock cycles, and machine cycle M_4 also requires 3 clock cycles. If whenever this RAM module is referenced, it issues wait states, then the actual number of clock cycles required to execute this instruction will be more than 13 clock cycles.

Notice that the op-code determines the kind and number of additional machine cycles that are necessary to complete execution of an instruction. It also determines how much the PC must be incremented. Thus, whenever execution of an instruction has been completed, the PC is pointing to an op-code of the next instruction. Furthermore, virtually all activity involves some kind of synchronized register-to-register transfer.

The op-code at location 1078H is the code for the 1 byte instruction, ADD A, B. Since this addition requires no additional memory reference, it requires 4 clock cycles (1 machine cycle) to execute. The next instruction, i.e., OUT (01H), A, is a 2 byte instruction, and it requires 3 machine cycles to execute. Since this is an OUT instruction, the Z80, during machine cycle M_3 does the following: (1) transfers the address 01 H, which it obtained during M_2, to the lower byte of the MAR to drive the lower byte of the system address bus, (2) transfers the content of the accumulator to the data bus out register at the trailing edge of T_1 to drive the external data bus, which, as shown in Figure 9.83, incurs a set-up delay, (3) activates the $\overline{\text{IORQ}}$ control signal at the leading edge of T_2, which incurs a set-up delay, and (4) activates the $\overline{\text{WR}}$ control signal, which also incurs a set-up delay, until the trailing edge of T_3. Thus, the I/O device that is supposed to receive the content of the accumulator has slightly less than 1.5 clock cycles during which it must capture the content of the system data bus. By decoding the address, which for an I/O device is called an I/O port address, and the control signals, an I/O interface can enable an external register to start to follow the data bus shortly after $\overline{\text{IORQ}}$ and $\overline{\text{WR}}$ have become active. Then, by the time that the $\overline{\text{WR}}$ signal becomes inactive, the external register should hold the content of the data bus.

9.4.6 Interfacing

The circuitry, or more broadly, the method that makes compatible the operations of devices so that these devices can exchange signals (data or codes) is loosely called an interface. The devices on opposite sides of an interface can be different in many ways. Interfacing two devices can encompass a variety of

requirements, such as, for example, (1) impedance matching, (2) voltage or current level conversion, (3) translation of control signal meanings, (4) protocol conversion, (5) signal timing alignment, (6) electrical isolation, (7) exchange of status information, (8) data format conversion, and (9) resolution of other incompatibility issues.

Figure 9.84 gives some examples of the goals of interface design. The unidirectional and bidirectional buffers make compatible the drive (current source) capability of the microprocessor and the drive requirements of the address and data bus. The particular control signals of the Z80 are interfaced to the system control bus with combinatorial logic to provide drive as well as more explicit control signals. The interfaces between the memory devices and the system bus utilize the address bus to position memory in the desired locations in the memory space. We must be certain that only one device can be a source to the data bus at a time. The interfaces also utilize control bus signals to produce the particular kinds of control signals required by the memory devices and to activate memory at times compatible with microprocessor timing.

Figure 9.84 also shows an input and an output port. The interface for this parallel I/O port also utilizes the address bus, and, instead of using control bus signals that go active due to memory reference instructions, it uses control bus signals that go active due to the IN and OUT instructions of the Z80. For output, it decodes the control bus and the lower half of the address bus to enable (clock) the octal latch to capture the content of the data bus at the right time and in response to the intended (its own) address $01. Thus, by executing an output instruction, like the one located at address $1079 in Figure 9.76, the octal latch receives the content of the accumulator. For input, the interface decodes the address and control bus to enable (take out of tristate) the octal buffer, which permits it to drive the data bus with its input at the right time and in response to the intended address. Thus, by executing an input instruction such as IN A, (01H), the accumulator will receive the byte at the buffer input. Depending on the address decode logic, the I/O port can be positioned anywhere in the I/O space of the microprocessor.

By using the $\overline{\text{MREQ}}$ control signal, instead of the $\overline{\text{IORQ}}$ control signal, and the entire address bus, I/O ports can also be positioned in the memory space of the microprocessor. The variety of conventional memory move instructions can then be used for I/O. Some microprocessors, like the M68HC11, do not have I/O space separate from memory space and instructions for I/O in addition to memory move instructions. Instead, no distinction is made between memory and I/O references. This eliminates the need for explicit control signals that interface hardware requires to distinguish memory and I/O references.

One or more parallel I/O ports can serve many different applications. With circuits like those given in Figure 9.85, a bit of an output port can turn on and off a light or sound indicator, or DC or AC power

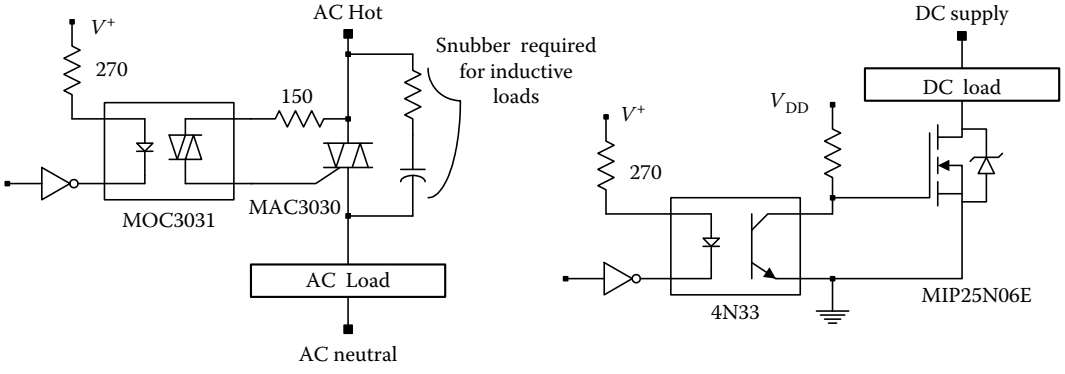


FIGURE 9.85 Circuits for opto-isolated power control.

supply to a load. With a set of bits, software can produce codes that control a device such as a printer. Since the octal latch in Figure 9.84 has a tristate output, O/E could be controlled by address and control signal decode from another microprocessor to place the octal latch output on another data bus. Thereby we accomplish data transfer between two machines. Or, software can produce binary time functions to control some process. An analog signal can be generated by outputting to a D/A converter, as shown in Figure 9.86.

Through a parallel input port a program can receive information. This may be the position of a set of switches, as shown in Figure 9.87, or the output of some kind of an encoder, such as a keypad encoder, rotating shaft position encoder, A/D converter, as shown in Figure 9.88, or even data from another microprocessor.

For data transfer from one system to another, the sender needs to know if the intended recipient is ready to receive data. And, if the data has been sent, then has it been received? To facilitate asynchronous

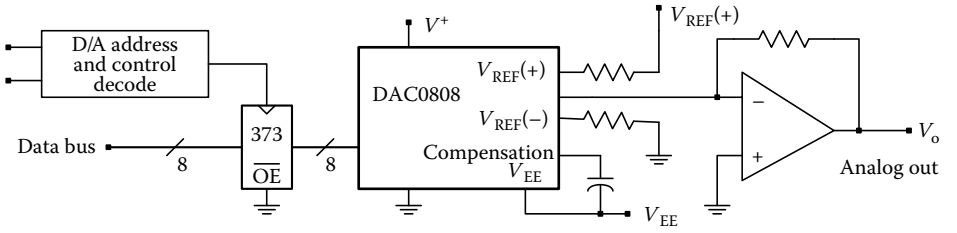


FIGURE 9.86 Output to a D/A converter.

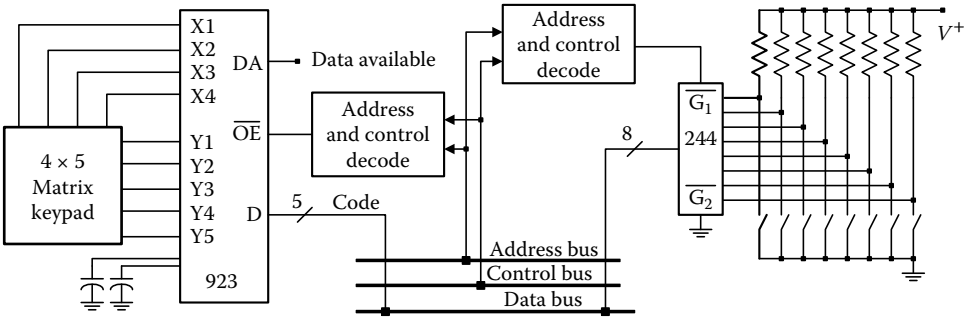


FIGURE 9.87 Circuits to input switch closures.

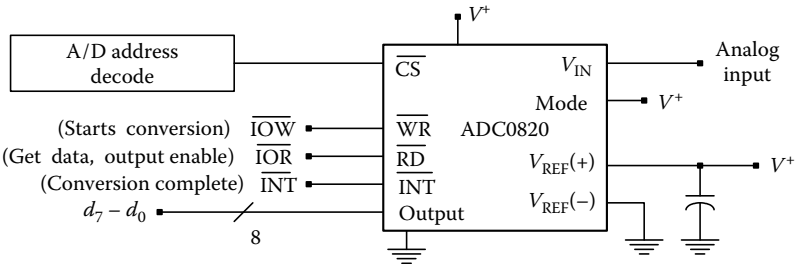


FIGURE 9.88 Write controlled A/D that includes sample and hold.

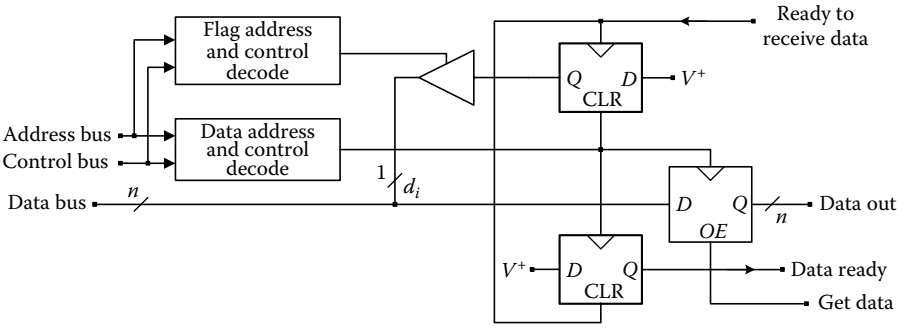


FIGURE 9.89 Output port with handshaking.

transfer of data, consider the output interface shown in Figure 9.89. This interface is the output port given in Figure 9.84 and additional circuitry to support ready to receive data and data ready flags. The sender can poll (obtain and check) the ready to receive data flag by inputting the data at the address assigned to flags. If data bit d_i is reset (logic 0), then continue polling the ready to receive data flag. If d_i is set (logic 1), then write data to the address assigned to be the data port, indivisibly reset the ready to receive data flag, and set the data ready flag. Thereby the sender cannot find that the ready to receive data flag is set until the recipient has found that the data ready flag is set, obtained the data, and set the ready to receive data flag, however little or much time this takes. Notice that setting the ready to receive data flag indivisibly resets the data ready flag. Thereby the recipient cannot find that data is available until it is new data, however little or much time this takes. There must be provision to reset the ready to receive data and data ready flags at system reset. The exchange of status information and the action among the flags is called handshaking. Figure 9.90 shows an input interface that includes handshaking. Here too, there must be provision to clear the data ready and ready to receive data flags at system reset.

To accomplish data transfer, the I/O methods of Figures 9.89 and 9.90 require that a program must continually poll the flags concerned with I/O. This works as long as the microprocessor is not required to do anything else. If the microprocessor must be used to do other tasks, then I/O can be serviced by either periodically polling I/O status flags or by using I/O status flags to interrupt the microprocessor while it is executing some program. Periodic polling of status flags may or may not be satisfactory. This depends on how much and how often I/O occurs.

Interrupt processing of the Z80, as well as the M68HC11, starts by asserting an interrupt input of the microprocessor. If more than one device must interrupt the microprocessor then using a microprocessor

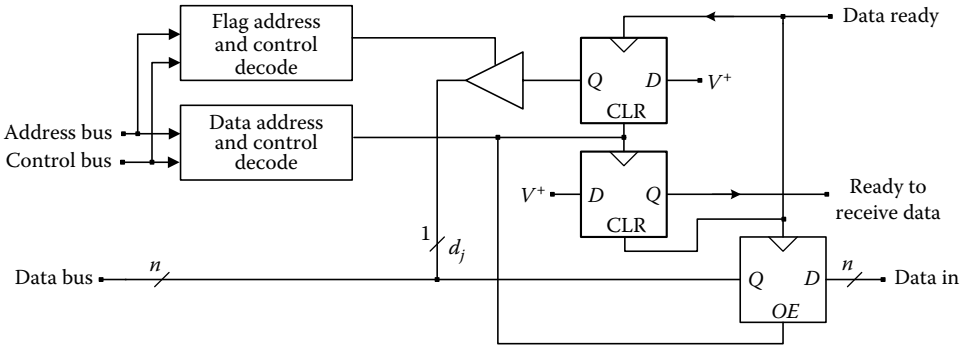


FIGURE 9.90 Input port with handshaking.

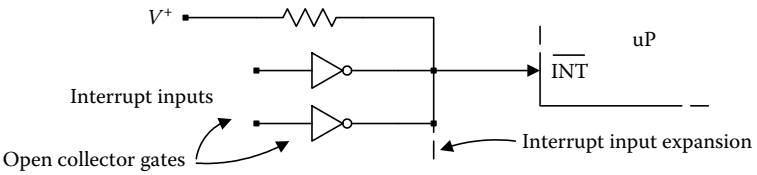


FIGURE 9.91 A wire OR'd circuit.

with many interrupt inputs might be useful. Or, all interrupt sources can be wire OR'd with open collector gates, as shown in Figure 9.91, to produce one interrupt signal. For example, the flip-flop output *Q* for the ready to receive flag in Figure 9.89 and the flip-flop output *Q* for the data ready flag in Figure 9.90 could each be connected to an inverter input in Figure 9.91. Either or both flags can then interrupt the processor. Any number of additional open collector gates could be added to include additional interrupt sources. By this method, the microprocessor discovers the occurrence of an interrupt, but must find out which device caused the interrupt.

If the Z80 is operating in direct interrupt mode, then the interrupt service routine, which starts at address 0038H, can input all of the status flags, check which flags are set, and respond with service priority determined by the software. Thus, any number of interrupts can be accommodated.

If the Z80 is operating in indirect vectored mode, then an interrupt causes it to respond with an active $\overline{\text{INTA}}$. The processor then accepts a byte from the data bus that must be provided by the interrupting device, and this byte, along with the content of the I register, is used as an address to get the interrupt service routine address from memory that is then loaded into the PC. Thus, the I register points to an interrupt vector table, and each interrupting device can point to the address of its own service routine.

If there is more than one interrupt source, then, since only one device is allowed to drive the data bus, a priority must be established, where the interrupting device with highest priority is serviced first. A daisy chain can be used so that hardware determines priority. This is shown in Figure 9.92. The device connected to the top of the chain has the highest priority. The $\overline{\text{INTA}}$ signal propagates down the chain until it is blocked by a set status flag. This condition is then used to place a response to $\overline{\text{INTA}}$ on the data bus.

In applications, such as control, it is often necessary that a program keep track of elapsed time. This can be accomplished with a circuit like the one shown in Figure 9.93. The 16 bit counter is clocked by the system clock or a clock derived from the system clock. The state of the counter is compared to the output of two registers, and a match, which clears the counter, is then used to interrupt the microprocessor. Thus, through I/O ports, software can set the time interval between interrupts, which can be disabled

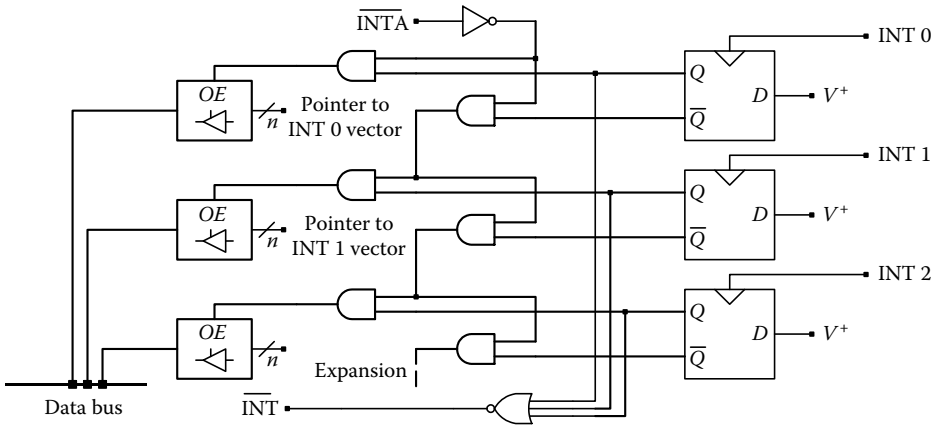


FIGURE 9.92 Priority with a daisy chain.

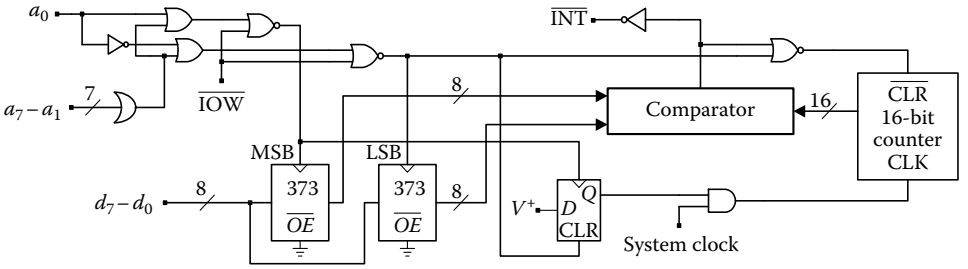


FIGURE 9.93 Timer-counter circuit.

(a write to the LSB) or enabled (a write to the MSB), and thereafter the microprocessor will be periodically interrupted to perform tasks in real time.

Solutions to interfacing problems are not unique. Variables such as the number of I/O devices, I/O device characteristics, amount of I/O, required response times, handshaking, software and hardware trade-offs, cost of parts and eventual manufacture, etc. all influence interface design. Also, there usually are numerous options to solving a particular problem. All of this requires an understanding of system (microprocessor) bus activity to properly time the occurrence of data transfer events. Then, there are issues of response to asynchronous events and status information about data transfer, polling and interrupt processing, and hardware and software trade-offs.

Solutions to common I/O problems such as parallel I/O, serial I/O, timer functions, etc. are often available within single IC packages. Typically, manufacturers of microprocessors provide a family of compatible ICs for each microprocessor. This can simplify the design task and reduce package count. Or, to reduce package count, consideration should also be given to combining gate level hardware into a single package of a PLD. Standardizing I/O can also reduce I/O coats. For example, the universal serial bus (USB) has been widely accepted to accommodate a great variety of I/O requirements.

9.4.7 Design with a Microcontroller

The M68HC11 is a microcomputer system within a single IC package. It is available in a 48 pin DIP and a 52 pin leaded chip carrier (LCC). In addition to the microprocessor, it can be configured for a variety of resources within the same package. These additional internal resources can be ROM (PROM or EPROM), RAM, parallel I/O ports, serial I/O ports, EEPROM, timer, and an A/D converter. Through a reserved bank of 64 status/control and I/O registers, software can select modes of operation of this additional hardware from a multitude of options. With these additional resources, especially the timer and A/D converter, the device is also called a microcontroller.

The programming model of the M68HC11 is shown in Figure 9.94. Registers A and B are both accumulators, and they can be used together as accumulator D for 16 bit add, subtract, multiply, and

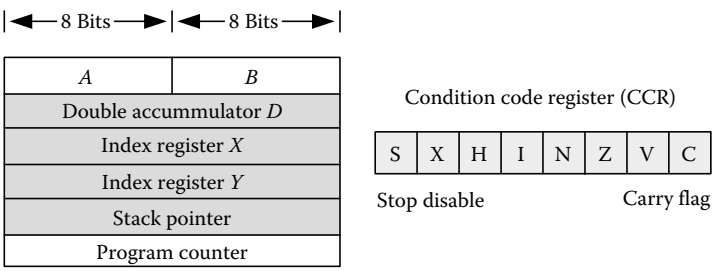


FIGURE 9.94 Programming model of the M68HC11.

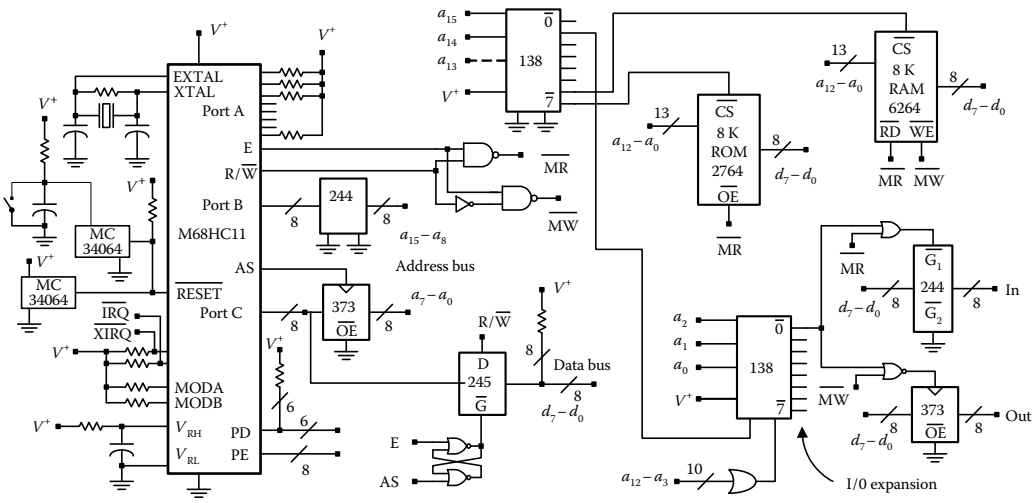


FIGURE 9.95 M68HC11A8-based microcomputer.

divide instructions. The machine language (op-codes), instruction set, and mnemonics are not the same as for the Z80, and so, the M68HC11 has its own assembly language.

The microcontroller also has different modes of operation. One of four modes is determined by the inputs at the two pins labeled MODA and MODB. These modes are MODB = 0, MODA = 0, special bootstrap, (01) special test, (10) normal single chip, and (11) normal expanded. We will consider the single chip and expanded modes of operation.

The M68HC11A8 is a version that has eight analog input channels, and in normal expanded mode of operation its pin functions are shown in Figure 9.95, which gives a diagram of a conventional microcomputer. Notice the selection of the normal expanded mode. For convenient reference, 38 of the 52 pins are grouped and labeled as follows: port A = pins 27–34, port B = pins 35–42, port C = pins 9–16, port D = pins 20–25, and port E = pins 43–50. The remaining pins are used for power supply (V_{SS} = pin 1 and V_{DD} = pin 26), the crystal for the internal clock circuit (pins 7 and 8), $\overline{\text{RESET}}$ (pin 17), nonmaskable interrupt ($\overline{\text{XIRQ}}$ = pin 18), maskable interrupt ($\overline{\text{IRQ}}$ = pin 19), control bus (AS = pin 4, E clock = pin 5, and R/$\overline{\text{W}}$ = pin 6), and analog input range (VRL = pin 51 to VRH = pin 52).

Port B provides the upper address byte. Port C is the data bus, and, to economize on the package pin count, the lower address byte is time-multiplexed with the data bus. The control signal AS, for address strobe, signifies when the data bus contains the lower address byte. The 74HC373 captures the lower address byte when AS is active. Thus, the external system address bus consists of port B and the output of the octal latch. The remainder of the control bus consists of the R/$\overline{\text{W}}$ signal and the E clock signal, which has a frequency equal to one-fourth the system clock frequency.

Since all I/O is memory mapped, the timing of system bus activity is especially straightforward. Figure 9.96 shows M68HC11 timing for all external memory references. After the lower address byte has been captured, the E clock goes high, the data bus is available for data transfer, and the R/$\overline{\text{W}}$ signal determines the direction. The E clock and the R/$\overline{\text{W}}$ signal together control read and write operations. During a read operation, the M68HC11 begins to follow the data bus at the leading edge of the E clock, and it accepts the content of the data bus at the trailing edge of the E clock. Thus, external hardware has slightly less than 2 clock cycles to provide valid data. During a write operation, the microprocessor begins to drive the data bus with valid data at the leading edge of the E clock. Thus, external hardware has slightly less than 2 clock cycles to follow the data bus, and at the trailing edge of the E clock it must accept the content of the data bus. All external memory references require one E clock cycle.

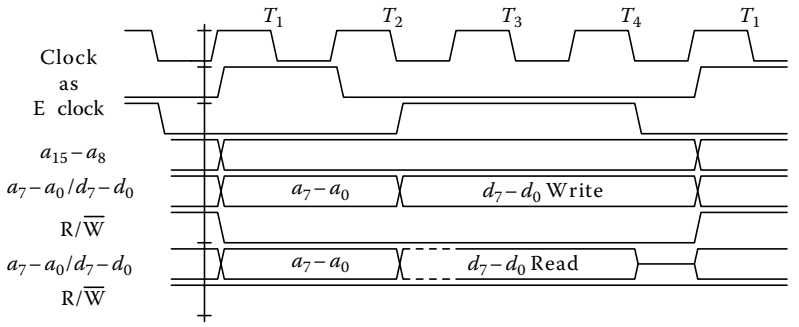


FIGURE 9.96 M68HC11 timing diagram for external memory references.

At power-up, an active low reset is accomplished with the MC34064 device in Figure 9.95. The output of this device switches low whenever its input supply is below a particular limit, and the output goes into tristate whenever its input exceeds a particular limit. Upon reset, the microprocessor initializes internal registers to default conditions, and then it reads memory locations $FFFE and $FFFF to obtain the reset vector that it loads into the PC. Thus, a designer can locate code that is to be executed at reset almost anywhere in the memory space.

Interrupt processing occurs in a similar way. Just prior to fetching each op-code the microprocessor checks the $\overline{\text{IRQ}}$ input. If it is active, and if the *I*-bit (interrupt mask) in the CCR is reset, then the microprocessor stacks 9 bytes (PC, D, IX, IY, and CCR) from the registers in its programming model, disables further interrupts by setting the *I*-bit, and then it reads memory locations $FFF2 and $FFF3 to obtain a vector that is loaded into the PC. The vector is the address of (points to) the first instruction of the $\overline{\text{IRQ}}$ interrupt service routine, which means that a designer can locate this service routine almost anywhere in the memory space.

The response to many interrupting sources can be handled in the same way as is illustrated by Figures 9.89 through 9.91, where the flags of interrupting devices are wire OR'd into one interrupt input to the microprocessor. Then, upon an interrupt, the service routine must first find out which device caused the interrupt. The control bus does not provide any special means to accomplish this. Moreover, as the service routine responds to each interrupt, there must be provision to clear each associated interrupting flag.

With the same vectored method, the M68HC11 is designed to accommodate numerous other external and internal sources of interrupts such as internal timer overflow, illegal op-code, software interrupt, nonmaskable interrupt, to name a few. Each interrupt is associated with two particular memory locations within the vector table, where its vector must be placed. This provides flexibility, and it does not require additional hardware for a reaction by an interrupting device to further control signals.

With the resources and method of interrupt processing of this microprocessor, the memory map of an M68HC11-based system is somewhat fixed. Figure 9.97 shows the memory map of the system given in Figure 9.95.

Through access to the system bus, additional memory and I/O ports can be added to this system. There are some restrictions. Internal address and control signal decoding positions the internal EEPROM block to start at memory location $B600. Because of its long write cycle time, we cannot store data into this memory with just a regular memory write instruction. There must be nonvolatile memory at the top end of the memory space to hold the vector table. The positions of the internal RAM and the control register block are programmable. The default locations are given in Figure 9.97. They can be positioned to start at any 4K boundary according to the data stored in the INIT register, which has address $103D after reset.

Many pins of the M68HC11 serve more than one purpose. For example, the mode select inputs are read during reset to determine the mode of operation. Thereafter, MODA is an active low output that

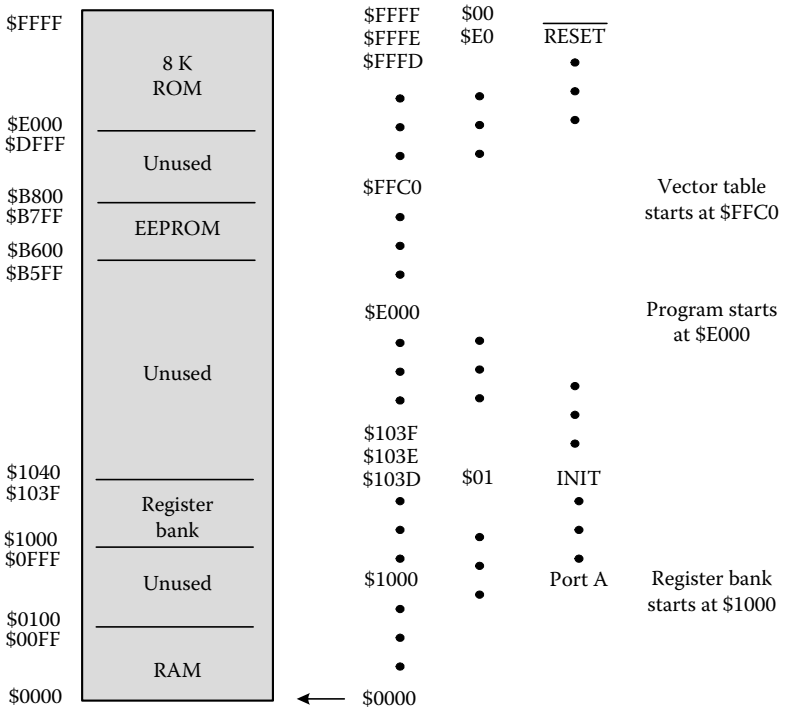


FIGURE 9.97 Memory map and sample ROM content.

becomes active during the first E cycle of each instruction. This is useful for debugging to know when the data bus must hold an op-code. The MODB pin can be used to provide standby power to maintain the content of the internal RAM when V_{DD} is not present.

Port A can serve as three input pins (PA0–PA2), four output pins (PA3–PA6), and one pin (PA7) that can be configured for input or output, depending on the data direction control bit labeled DDRA7 in the PACTL register at memory location $1026. From a programmer's viewpoint, port A is a memory register with address $1000, and the register that controls its I/O functionality is another memory register.

The M68HC11 has a free-running 16 bit counter that is clocked by the output of a programmable prescaler, which is clocked by the E clock. The 2 bytes of the counter can be read through buffers at locations $100E (high byte) and $100F (low byte). All M68HC11 timer functions are based on this counter. The 29 registers at locations $100B–$1027 are all concerned with using and configuring various kinds of timer functions. Thus, software can initiate a periodic real-time interrupt (RTI), which has an associated RTI vector located at $FFF0 and $FFF1.

Another purpose of port A is to provide access to some of these timer functions. For example, pins PA0–PA2 can be used to measure the edge-to-edge time durations of incoming pulses. The M68HC11 contains circuitry for edge detection, and for each input the edge polarity that is to be detected is programmable. Also, for each input there is a unique interrupt vector location associated with edge detection. Thus, to measure the period of a pulse train input at PA0, for example, the counter state must be captured upon an edge detection. For PA0, counter capture is controlled by the IC3F flag in register TFLG1 at location $1023. Registers at locations $101A and $101B receive the counter state. For PA0, edge detection interrupt is enabled with the IC3I flag in register TMSK1 at location $1022. The vector for this interrupt is located at $FFEA and $FFEB. If the *I*-bit in the CCR is reset, then an edge at PA0 will receive interrupt service, which should read the captured counter. Comparing counter states from successive edges of the same polarity can then be used to find the period.

Port D can serve as a general purpose 6-bit I/O port with address $1008. The direction of each bit is programmable through the DDRD register at location $1009. When serial communication is enabled, this port provides asynchronous serial input (RxD) at PD0 and serial output (TxD) at PD1. The baud rate is controlled by the contents of the BAUD register at location $102B. The five registers at locations $102B–$102F are all concerned with using and configuring asynchronous serial communication. From a programmer's viewpoint serial I/O are accomplished by a parallel read from and parallel write to memory location $102F, respectively. There are actually two responsive registers at this address that are distinguished by the R/$\overline{W}$ signal.

The other four pins of port D provide synchronous high speed serial communication. Pins PD3 and PD2 are used for transmitting and receiving serial data, respectively. Pin PD4 carries a clock signal to synchronize data transfer, and pin PD5 can be used to indicate the start of a data transfer. The three registers at locations $1028–$102A are all concerned with using and configuring synchronous serial communication. To reduce package size, pin count and communication paths, there are numerous devices that use serial I/O. Through port D, the M68HC11 can communicate with, for example, serial in LED/LCD display drivers, serial data out A/D converters, serial in/out EEPROMs, or even another microprocessor.

Port E can serve as an 8-bit digital input port with address $100A. These inputs are also each connected to a sample and hold circuit and then to an eight channel analog MUX, the output of which goes to a successive approximation A/D converter. The A/D converter produces an unsigned 8-bit number that is proportional to a DC voltage in the range VRL to VRH. An analog input equal to VRL (VRH) yields an A/D conversion result of $00 ($FF). Each A/D conversion requires 32 E clock cycles. Flags in register ADTCL at location $1030 control A/D conversion. Four consecutive conversions of a single channel, or one conversion of each of the lower four or the upper four channels can be obtained, depending flags in the ADCTL register. A/D conversion results are available from registers at locations $1031–$1034.

Needless to say, the M68HC11 is a complex machine with many options among its numerous features under software control. Features such as software security, failure detection and recovery, power-down/standby, and others have not been discussed.

In the single-chip mode as shown in Figure 9.98, ports B and C and the control signals serve other purposes. Here, port B is an 8-bit output port with address $1004. Pin 6, which is the R/$\overline{W}$ signal in the

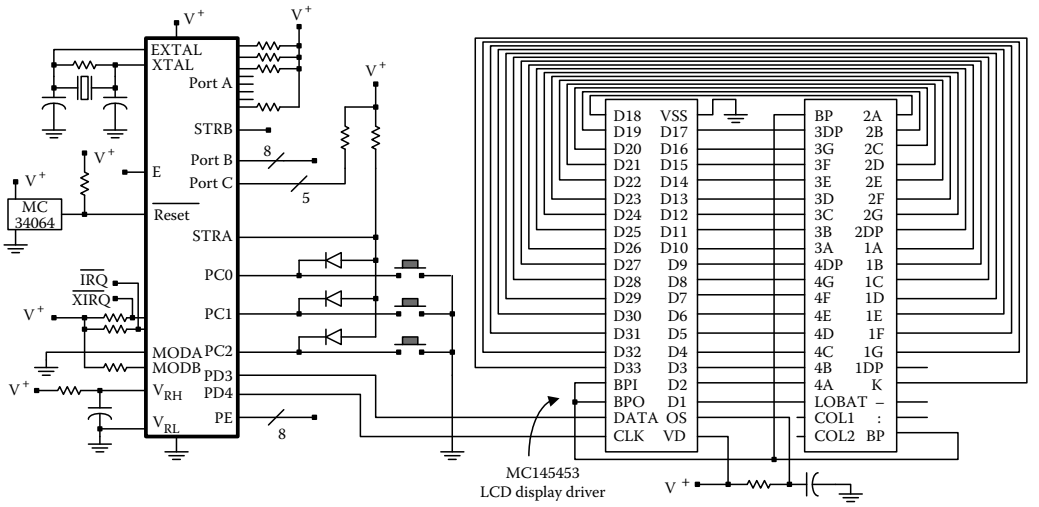


FIGURE 9.98 M68HC11 microcontroller used for a three button and four-digit LCD display device.

expanded mode, can now be configured as an output strobe (STRB) that produces a pulse whenever a write to $1004 (port B) occurs. Port C is a general purpose I/O port with address $1003, and its data direction is controlled by the DDRC register at location $1007. The eight pins of port C also go to a register with address $1005. Pin 4, which is the AS signal in the expanded mode, is now an input strobe (STRA). An edge, the polarity of which is programmable, at this input will cause the data at the port C pins to be latched into the register PORTCL with address $1005. Full handshaking is implemented, because when data are latched into the PORTCL register, the flag STAF in SR PIOC at location $1002 becomes set, and after both the flag and PORTCL have been read, the flag becomes reset.

In single-chip mode, there can be as many as 27(12) output bits and 11(26) input bits, depending on data direction control, or fewer I/O bits if some of these pins are used for A/D, serial communication, or timer functions. Moreover, software must reside in the internal 8K byte ROM of the M68HC11A8 or the internal 12K byte ROM of the M68HC11E9. Like external ROM, this ROM is positioned at the top end of the memory space. It must receive its content at the time of manufacture. With some restrictions, it can be enabled or disabled by the ROMON flag of the CONFIG register at location $103F. This is a special one byte EEPROM so that it will retain its content through power-down and power-up cycles. The M68HC711E9 is an EPROM version of the M68HC11E9. The EPROM can be programmed/erased in the field for development in the single-chip mode. There is also an OTP version of the E9. There are several other members of the M68HC11 family of microcontrollers with varying amounts of hardware resources.

Recently, the M68HC11 has evolved into the M68HC12. In addition to more of the resources available in the M68HC11, the M68HC12 has two expanded modes of operation using either an 8 bit data bus or a 16 bit data bus. Most noteworthy of this processor is its extensive and complex instruction set. In particular, there are instructions that make it convenient to implement fuzzy logic controllers, including fuzzification, inference engine, and defuzzification computing.

In contrast to the evolution of the M68HC11, the MSP430 microcontroller family from Texas Instruments has a very small instruction set, like an RISC machine, with instructions executing in single clock cycles. With its 14 bit A/D converter and fast multiply and accumulate instructions, it can be applied to do digital signal processing. Operating on 3 V, this processor is particularly well suited for low voltage, low power consumption, and portable applications.

The 8051-based family of microcontrollers from Intel has been widely utilized over the past three decades in embedded systems. It is manufactured by many companies. The Harvard architecture of its CPU is a distinctive feature of the processor.

The PIC microcontroller family from Microchip, Inc. is particularly easy to employ. Like most microcontrollers, it can be programmed in C, assembly language, and BASIC. Very low-cost development tools are available for it.

All of these microcontroller families are available with a variety of resources including ROM, RAM, EEPROM, parallel and serial I/O, A/D conversion, timers, LCD drivers, flash memory, and more within a package. When single package hardware resources can match application requirements, economical and compact hardware designs can be achieved.

9.4.8 Design Guidelines

Looking back at Figure 9.67, we see that its simplicity is deceptive. Nonetheless, in view of Figure 9.98 or even Figure 9.84, Figure 9.67 represents the typical embedded microprocessor application. To use a microprocessor requires an awareness of what may seem like an untold amount of information (facts). This should not and cannot be avoided. Time spent in the beginning to know the details will likely save time and expense that may have to be spent later to make revisions. Here, we have only raised a few issues to see the possibilities.

As we look at different processors we find that while the programming model, instruction set, and hardware resources change, hardware design is concerned with interfacing to achieve electrical, timing, and functional compatibility. There are some important principles that carry over from one

microprocessor to another. It is a component that is fundamentally intended to provide trade-offs between hardware and software. This is a matter of degree, and will vary from one application to another. Much care must be taken to suitably allocate the hardware/software trade-offs.

Software and hardware development tools such as software simulators, hardware in circuit emulators, logic analyzers, cross-assemblers and cross-compilers, real-time kernels, and others are necessary to be competitive in product development. Software will likely carry the greater burden to achieve products that function according to desired modes of operation, and it will also incur the greater development cost.

All of the manufacturers of microcontrollers mentioned above provide low-cost evaluation modules and software development tools. Once an application has been well defined, methods of implementation are sufficiently understood and required resources have been specified, it is worthwhile to exercise evaluation modules of several different microcontrollers to compare and find that microcontroller best suited for incorporation into the application.

References

1. Ayala, K. J., 2000, *The 80251 Microcontroller*, Upper Saddle River, NJ: Prentice-Hall.
2. Bierl, L., 2000, *MSP430 Family: Mixed Signal Microcontroller Application Reports*, Dallas, TX: Texas Instruments, Inc.
3. Cavenor, M. and Arnold, J., 1989, *Microcomputer Interfacing: An Experimental Approach Using the Z80*, Upper Saddle River, NJ: Prentice-Hall.
4. Cady, F. M., 1997, *Software and Hardware Engineering, Motorola M68HC12*, New York: Oxford University Press.
5. Driscoll, F. F., Coughlin, R. F., and Villanucci, R. S., 1994, *Data Acquisition and Process Control with the M68HC11 Microcontroller*, Columbus, OH: Merrill.
6. Haznedar, H., 1991, *Digital Microelectronics*, Redwood City, CA: Benjamin/Cummings.
7. Morton, T. D., 2001, *Embedded Microcontrollers*, Upper Saddle River, NJ: Prentice-Hall.
8. Noergaard, T., 2005, *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*, Amsterdam/Boston: Elsevier.
9. Peatman, J. B., 1998, *Design with PIC Microcontrollers*, New York: Prentice-Hall.
10. Rafiquzzaman, M., 2005, *Fundamentals of Digital Logic and Microcomputer Design*, Hoboken, NJ: Wiley.
11. Short, K. L., 1998, *Embedded Microprocessor Systems Design, An Introduction Using the Intel 80C188EB*, New York: Prentice-Hall.
12. Spasov, P., 2002, *Microcontroller Technology, The 68HC11*, Englewood Cliffs, NJ: Prentice-Hall.
13. Uffenbeck, J., 1985, *Microcomputers and Microprocessors, The 8080, 8085, and Z80, Programming, Interfacing, and Troubleshooting*, Englewood Cliffs, NJ: Prentice-Hall.
14. Wolf, W., 2005, *Computers as Components, Principles of Embedded Computing System Design*, San Fransico, CA: Elsevier.

9.5 Systolic Arrays

Kung Yao and Flavio Lorenzelli

9.5.1 Concurrency, Parallelism, Pipelining, and Systolic Array

9.5.1.1 Motivations and Definitions

Real-time high throughput rate processing constitutes one of the most demanding aspects of modern digital signal processing. In order to achieve the desired throughput rate, various forms of concurrent operations are needed. ''Concurrency'' denotes the ability of a processing system to perform more

than one operation at a given time. Concurrency can be achieved through either parallelism or pipelining, or both. ''Parallelism'' addresses concurrency by replicating some desired processing functions many times. High throughput rate is achieved by having simultaneous operations performed by these functions on different parts of the program. On the other hand, ''pipelining'' tackles concurrency by breaking some demanding part of the task into many smaller simpler pieces, with many corresponding processing elements (PEs), so that processing can be performed in a pipeline manner. This digital pipe is arranged so that it is capable of processing the instructions and data independent of the number of PEs in the pipe. Then, high throughput rate can be achieved by having fast PEs in the pipe. As we shall see, a ''systolic array'' can exploit both the parallelism and pipelining capability of some algorithms.

The term systolic array was coined by Kung and Leiserson [2] to denote one simple class of concurrent processors, in which processed data move in a regular and periodic manner similar to that of the systolic pumping action of the blood by the heart. The earlier definition of a systolic array by Kung [3] requires (1) only a small class of PEs is in the array, with each element in a class performing identical operation; (2) all operations are performed in a synchronous manner independent of the processed data—the only control data broadcast to the PEs is the synchronous clock signal; and (3) the PEs have only nearest-neighbor communications. These regular structure and local communication properties of a systolic array are consistent with efficient modern VLSI designs. Later, various extensions of these assumptions were made (1) some of the PEs can perform a limited number of different functions, depending on the presence of some control data; (2) wavefront array allows PEs to start/end/control their own processing tasks, depending on the data; and (3) PEs can have communications to few nearby neighbors; wraparound communications among PEs located at the edge of the array are allowed.

Systolic arrays can be designed as linear arrays or two-dimensional rectangular or triangular arrays. In Figure 9.99a, consider a uniprocessor system requiring μ time unit to complete a basic operation. If some task requires N such repeated identical operations, the effective throughput rate of this system is given by $r_a = 1/N\mu$. In Figure 9.99b, consider a linear array consisting of a single pipe with N such PEs. Then the rate of this linear systolic array is given by $r_b = 1/\mu$. This demonstrates the pipelining aspects of the array. In Figure 9.99c, consider a rectangular array consisting of M pipes, with each pipe having N PEs. The rate of this rectangular systolic array is given by $r_c = M/\mu$. This demonstrates both the pipelining and parallelism of the array. While the three models in Figure 9.99 are overly simple, nevertheless they demonstrate the fact that if a given task can be designed for systolic processing, different systolic arrays can yield significantly higher throughput rates as compared to a uniprocessor of a given capability. This is the most basic aspect of systolic processing in which a higher hardware complexity is traded for a higher throughput rate.

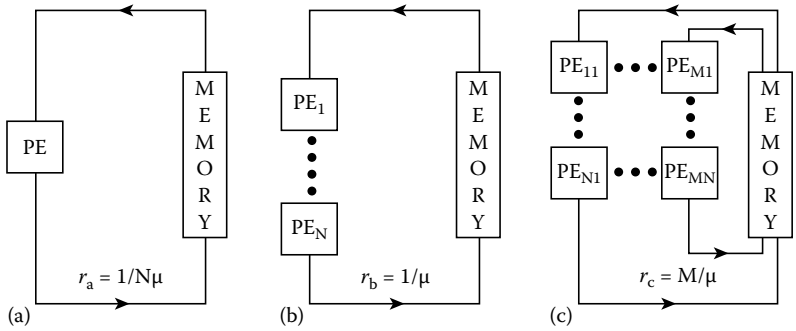


FIGURE 9.99 (a) A uniprocessor system, (b) a linear systolic array, and (c) a rectangular systolic array.

9.5.1.2 Systolic Arrays for Correlation

Consider the linear correlation of a data sequence $\{x_1, x_2, \ldots, x_M\}$ with a weight sequence $\{a_1, a_2, \ldots, a_N\}$ to yield an output sequence $\{y_1, y_2, \ldots, y_{M-N+1}\}$ given by $y_i = a_1 x_i + a_2 x_{i+1} + \cdots + a_N x_{i+N-1} = \Sigma_{J=1}^{N} a_j x_{i+j-1}$, $i = 1, 2, \ldots, M-N+1$. For the case of $N=3$ and $M > N$, we have

$$\begin{aligned} y_1 &= a_1 x_1 + a_2 x_2 + a_3 x_3 \\ y_2 &= a_1 x_2 + a_2 x_3 + a_3 x_4 \\ y_3 &= a_1 x_3 + a_2 x_4 + a_3 x_5 \\ &\vdots \end{aligned}$$

Here, we show two of many possible systolic arrays that can implement the above correlation operations. Design B1 in Figure 9.100a uses three identical PEs to perform the accumulation (multiply and add) operation. Here, the weights a_i are preloaded to the cells and stay throughout the computation. Partial results y_i move systolically from cell to cell. Starting at the third iteration, $y_1, y_2, \ldots$, are outputted from the rightmost cell at the rate of one output per iteration. For each iteration, an x_i is broadcast to all the cells, and a y_i, initialized to zero, enters the leftmost cell. The broadcasted data x_i is marked with an arrow $\downarrow$ in Table 9.12. Indeed, by comparison we see y_1, y_2, and y_3, outputted at iteration $T=3$, 4, and 5, agree with those given from the correlation equations.

In design B2, shown in Figure 9.100b, each input x_i is again broadcasted to each cell, each y_i stays at each cell to accumulate terms, while the weights a_i circulate around the cells in the array. A tag bit is associated with a_1 to reset the contents of the accumulator, while a tag bit is associated with a_3 to output the contents of the accumulator after the first two iterations. Data movements in design B2 are shown in

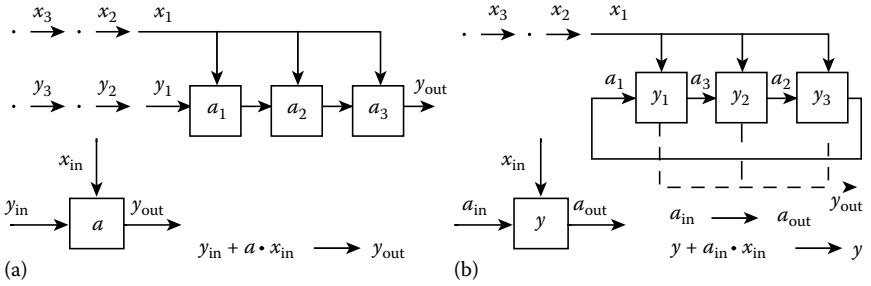


FIGURE 9.100 (a) Systolic array design B1 for correlation; and (b) systolic array B2 for correlation.

TABLE 9.12 Data Movement in Design B1

Iteration	Cell 1	Cell 2	Cell 3
$T=1$	$\downarrow x_1\ a_1$	$\downarrow x_1\ a_2$	$\downarrow x_1\ a_3$
$T=2$	$\downarrow x_2\ a_1$	$a_1 a_1 + \downarrow x_2\ a_2$	$x_1 a_2 + \downarrow x_2\ a_3$
$T=3$	$\downarrow x_3\ a_1$	$x_2 a_1 + \downarrow x_3\ a_2$	$x_1 a_1 + x_2 a_2 + \downarrow x_3\ a_3 \rightarrow y_1$
$T=4$	$\downarrow x_4\ a_1$	$x_3 a_1 + \downarrow x_4\ a_2$	$x_2 a_1 + x_3 a_2 + \downarrow x_4\ a_3 \rightarrow y_2$
$T=5$	$\downarrow x_5\ a_1$	$x_4 a_1 + \downarrow x_5\ a_2$	$x_3 a_1 + x_4 a_2 + \downarrow x_5\ a_3 \rightarrow y_3$

TABLE 9.13 Data Movement in Design B2

Iteration	Cell 1	Cell 2	Cell 3
$T=1$	$0 + a_1 \overset{\downarrow}{x_1}$	$a_3 \overset{\downarrow}{x_1}$	$a_2 \overset{\downarrow}{x_1}$
$T=2$	$a_1x_1 + a_2 \overset{\downarrow}{x_2}$	$0 + a_1 \overset{\downarrow}{x_2}$	$a_2x_1 + a_3 \overset{\downarrow}{x_2}$
$T=3$	$a_1x_1 + a_2x_2 + a_3 \overset{\downarrow}{x_3} \rightarrow y_1$	$a_1x_2 + a_2 \overset{\downarrow}{x_3}$	$0 + a_1 \overset{\downarrow}{x_3}$
$T=4$	$0 + a_1 \overset{\downarrow}{x_4}$	$a_1x_2 + a_2x_3 + a_3 \overset{\downarrow}{x_4} \rightarrow y_2$	$a_1x_3 + a_2 \overset{\downarrow}{x_4}$
$T=5$	$a_1x_4 + a_2 \overset{\downarrow}{x_5}$	$0 + a_1 \overset{\downarrow}{x_5}$	$a_1x_3 + a_2x_4 + a_3 \overset{\downarrow}{x_5} \rightarrow y_3$

Table 9.13. Note that resets occur at cell 1 at iteration 1, cell 2 at iteration 2, cell 3 at iteration 3, cell 4 at iteration 4, etc. Similarly, output y_1 occurs from cell 1 at iteration 3, y_2 from cell 2 at iteration 4, y_3 from cell 3 at iteration 5, etc.

9.5.1.3 Systolic Array Design Techniques

Systolic array designs, as shown above for the correlation case, can be obtained by ad hoc approaches. More formal procedures for the systematic design of systolic arrays have been proposed by Moldovan [7], Quinton [9], Kung [5], Rao [10], Darte and Delosme [1], and others. All those more formal procedures are collectively referred to as dependence graph mapping techniques for systolic array design.

In this approach, an algorithm must be formulated in the ''single assignment algorithm'' form. Each variable has a unique value during the evaluation of the algorithm. Those variables with multiple values can be converted to single values by vectorizing the variables through the introduction of new indices. As an example, consider the matrix multiplication of $C = AB$, where $A = [a_{ik}]$ is $N_1 \times N_3$, $B = [b_{kj}]$ is $N_3 \times N_2$, and $C = [c_{ij}]$ is $N_1 \times N_2$. A conventional formulation of this algorithm contains the expression, $c_{ij} = c_{ij} + a_{ik}\, b_{kj}$ for $i = 1$ to N_1, $j = 1$ to N_2, and $k = 1$ to N_3. We note, c_{ij} has multiple values for $k = 1, \ldots, N_3$. We can modify it to have single values by replacing it by the variable c_{ijk}. The previous equation for c_{ij} then becomes $c_{ijk} = c_{ij(k-1)} + a_{ik}\, b_{kj}$, $c_{ij0} = 0$, c_{ijN_3}, $i = 1, \ldots, N_1$, $j = 1, \ldots, N_2$, $k = 1, \ldots, N_3$.

All algorithm variables are assumed to be indexed variables with V variable names, denoted by the generic names of V_m, $1 \le m \le V$. In the above matrix–matrix multiplication problem $V = 3$, and we can take $X_1, = c$, $X_2 = a$, and $X_3 = b$. For each variable name, the domain of the index vectors is a subset in an S-dimensional space. This subset is called the algorithm's ''index space'' and S is its dimension. For most iterative signal processing problems, time is usually one of the index space coordinates. For the preceding matrix–multiplication problem, we need to propagate a_{ik} across the j variables as well as b_{kj} over the i variables in order to perform the basic multiplication operation. These and a_{ik} and b_{kj} are propagating variables because they involve no computations, but need to be made available at various stages of the computation. In the matrix–matrix problem, clearly $S = 3$ and the index space is $\mathbf{S}_0 = \{(i, j, k)\colon 1 \le i \le N_1, 1 \le j \le N_2, 1 \le k \le N_3,\}$. Furthermore, the initializations of the new variables are given by $a(i, 0, k) = a_{ik}$, $b(0, j, k) = b_{jk}$, $c(i, j, 0) = 0$, $c(i, j, N_3) = c_{ij}$, and the algorithm is finally given by $a(i, j, k) = a(i, j, -1, k)$, $b(i, j, k) = a(i - 1, j, k)$, $c(i, j, k) = c(i, j, k - 1) + a(i, j, j)b(i, j, k)$, for $(i, j, k) \in \mathbf{S}_0$.

In general, a point (or node) in the index space is called an ''index point.'' Thus, $X_m(I)$ is the variable X_m defined at the index point I. A dependence graph mapping is a representation of a single assignment algorithm, where the dependencies among the variables are represented by directed arcs among the nodes. A basic property of the class of algorithms of interest is that of ''shift-invariance.''

An algorithm is shift-invariant if the dependence graph is regular. That is, $X(I)$ depends on $Y(J)$, then $X(I + K)$ depends on $Y(J + K)$ for all I, J, and K in the index space. Three well-known shift-invariance algorithms include

1. Uniform recurrence equations (URE): $X_1(I) = F_1(X_1(I - D_1), \ldots, X_V(I - D_V))$, $X_i(I) = X_i(I - D_i)$, $2 \le i \le V$. Computation occurs only in $F_1(\cdot)$ and propagations in all the other variables, Clearly, the final form of the above matrix–matrix multiplication algorithm is a URE algorithm with

$V = 3$, $F_1(\cdot) = c(i, j, k)$, with $X_1(\cdot) = c(\cdot)$, $X_2(\cdot) = a(\cdot)$, $X_3(\cdot) = b(\cdot)$, $I = (i, \mathrm{i}, k)$, $D_1 = [0, 0, 1]^{\mathrm{T}}$, $D_2 = [0, 1, 0]^{\mathrm{T}}$, and $D_3 = [1, 0, 0]^{\mathrm{T}}$.

2. Generalized uniform recurrence equations (GURE): $X_m(I) = F_m(X_{m1}(I - D_{m1}), \ldots, X_{mk(m)}(I - D_{mk(m)}))$, $1 \leq m \leq V$, where $m_1, \ldots, m_{k(m)}$, belongs to $\{1, \ldots, m\}$. In GURE, we can have computations in all V functions of $F_m(\cdot)$. The number of independent variables $m_{k(m)}$ depends on each m. The shift index dependence, $I - D_{mi}$, is fixed for each X_{mi}.
3. Regular iterative algorithm (RIA): $X_m(I) = F_m(X_{m1}(I - D_{m1,m}), \ldots, X_{mk(m)}(I - D_{mk(m)m}))$, $1 \leq m \leq V$. Here, the shift index dependency, $I - D_{m_i m}$, is not fixed but is a function of m_i and m.

Each processor of the systolic array is assumed to have all the necessary computational modules to compute $F_m(\cdot)$. For URE, we need only one such module, but for GURE and RIA, we need V modules. The time required for the computation of $F_m(\cdot)$ is denoted by τ_m, and the minimum time between such computations is denoted by h_m. In most cases, we can set τ_m and h_m to unity. The design of a processor array to perform the algorithm requires spatial and temporal assignments. Each $X_m(I)$ must be assigned to a processor at each integral time slot. The processor "allocation function," $A(I)$ assigns all variables with the index I to the processors in the array. The "scheduling function," $S_m(I)$, assigns the start of the computation for the variable $X_m(I)$. The simplest form of scheduling and processor allocations are based on the projection of the high multidimensional dependence graph onto the lower dimensional processor array. Variables represented by nodes in the dependence graph are mapped to processors which perform the computations. The directed arcs of the dependence graphs are transformed to physical communication links in the processor array.

The essence of the allocation function $A(\cdot)$ is thus to return for every index value $I \in \mathbf{S}_0$ a vector which indicates the processor in charge of the computation represented by a point in a lower dimensional space. Analogously, the scheduling function $S_m(\cdot)$ provides the relative start of the execution for the computation indexed by I. These two functions cannot be chosen independently because two computations assigned to the same processor cannot be scheduled for the same time ("compatibility constraint"). Additional details on this constraint are given later. While in principle $A(\cdot)$ and $S_m(\cdot)$ can be any function, we shall consider only "affine functions," in the sense that $A(I) = A^{\mathrm{T}}I$, $S_m(I) = \lambda^{\mathrm{T}} i + \gamma_m$, where A is a suitable matrix, λ a vector, and γ_m an integral constant.

The dependence graph of an algorithm can be interpreted as a "lattice" embedded in a multidimensional integral space (i.e., a proper bounded subset of $\mathbf{Z}^s$, where $\mathbf{Z}$ is the set of relative integers), enclosed in a convex polyhedron. We assume the lattice to be "dense" in the sense that all the integral points in it correspond to actual computations. The whole procedure of mapping an algorithm onto a systolic-type processor consists of two conceptually different but interdependent operations of using a space transformation and a time transformation. The former actually "projects" the dependence graph onto a lower dimensional structure which then can be mapped one-to-one onto the physical array, while the latter gives the start of the execution of each computation.

For simplicity, consider the projection of the S-dimensional space onto an $(S-1)$ dimensional processor space. The more general problem of projecting the dependence graph onto an $(S-p)$-dimensional space ($p \geq 1$) can be expressed using a similar but more involved notation and is omitted here. Instead of considering allocation functions, we refer to the "projection vector u," which is orthogonal to the processor space onto which we project. Assume that we have chosen both the projection and the scheduling vectors (u and λ, respectively). For normalization purposes, they are chosen to be coprime vectors, such that the greatest common divisor of their components is 1, and their first nonzero element is positive.

Two sets of constraints must be satisfied by u and λ. Assume nodes I and J are located along a direction parallel to the projection vector u such that $J = I + \alpha u$, $\alpha \in \mathbf{Z}$. Then, the computations associated with the two nodes will be projected onto the same processor. Consequently, compatibility constraint requires that they be performed at different times. Analytically, this is equivalent to $|\lambda^{\mathrm{T}} u| \; \lambda \max_{m=1,\ldots,V} h_m$, which for $h_m = 1$ simplifies to $|\lambda^{\mathrm{T}} u| > 0$. Thus, for this case λ and u cannot be perpendicular. Furthermore, the

quantity $c = \Delta|\lambda^T u|$ represents the number of time slots between successive calculations scheduled on the same processor. $1/c$ is sometimes called the "efficiency" of the processors because the larger the c, the more time the processors can idle. One common approach is to select the projection vector and the scheduling vector to achieve the highest efficiency, with c being as close to 1 as possible.

Consider the case in which the variable $X_m(I)$ depends on $X_n(I - D_{nm})$. The "precedence constraint" implies the calculation of $X_n(I - D_{nm})$ must be scheduled to be completed before the start of the calculation of $X_m(I)$. Analytically, the precedence constraint is equivalent to $\lambda^T D_{nm} + \gamma_m - \gamma_n \geq \tau_n = 1$, for all $1 \leq m \leq v$ and for all dependences D_{nm}. If the γ constants are chosen to be all equal, the precedence constraint becomes $\lambda^T D_{nm} \geq 1 \; \forall \, m = 1, \ldots, V$.

Assume the precedence and compatibility constraints are satisfied and λ and u are coprime vectors. Then it is possible to extend both vectors to two unimodular matrices. A matrix with integral entries is called "unimodular" when its determinant is equal to ± 1. This implies that they admit integral inverses. The unimodular extension of coprime vectors is not unique. We will choose U and Λ to be the unimodular extended matrices that have u and λ, respectively, as their first columns. It is possible to show that the columns of any S-dimensional unimodular matrix can constitute a basis for the space Z^S. Moreover, if we denote $\sigma_1, \ldots, \sigma_s$ to be the columns of $\Sigma = U^{-T}$, then we have $\sigma_1^T u = 1$ and $\sigma_1^T u = 0$ for all $i = 2, \ldots, S$. Therefore, $\{\sigma_2, \ldots, \sigma_s\}$ will be a basis of the processor space of the resulting logic array. Similarly, the first column of T (the inverse of Λ^T) t_1, represents the direction in which time increases by one step; i.e., it is the vector defining the hyperplane of the points computed at the same time. The other columns of T (denoted by $t_2, \ldots, t_s$) are a basis of such a hyperplane.

If we denote by $\Sigma_+ = [\sigma_2, \ldots, \sigma_s]$ the matrix basis of the processor space, the allocation function and the scheduling function have the form $A(I) = \Sigma_+^T I, S_m(I) = \lambda^T I + \gamma_m$, $m = 1, \ldots, V$. With these elements we can have the complete description of the final array. The processors are labeled by $A(I) = \Sigma_+^T I$ as I ranges over the index space. The dependences D_{nm} are mapped onto communication links $\Sigma_+^T D_{nm}$ and the delay registers on such links must be in number equal to $\lambda^T D_{nm} + \gamma_m - \gamma_n - \tau_n = \lambda^T D_{nm} - 1$.

Reconsider the systolic correlation problem using the weights $\{a_1, \ldots, a_k\}$ and the data $\{x_1, \ldots, x_n\}$, as discussed earlier. Recall the correlation is given by $y_i = a_1 x_i + a_2 x_{i+1} + \cdots + a_k x_{i+k-1}$, $1 \leq i \leq n + 1 - k$. A recurrence equation formulation of this equation is given by $y(i,j) = y(i,j-1) + w(i,j)x(i,j), y(i,0) = 0, y_i = y(i,k)$; $w(i,j) = w(i-1,j)$, $w(1,j) = a_j$; and $x(i,j) = x(i+1,j-1)$, $x(i,0) = x_{i-1}$, all with $1 \leq i \leq n + 1 - k$, $1 \leq j \leq k$. A dependence graphical representation of these equations is shown in Figure 9.101a.

A URE reformulation of the recurrence equations yields $X_1(I) = y(i,j) = F_1(X_1(I - D_1), X_2(U - D_2), X_3(I - D_3))$, $X_2(I) = w(i,j) = X_2(I - D_2)$, $X_3(I) = x(i,j) = X_3(I - D_3)$, with the index point $I = [i,j]^T$ and

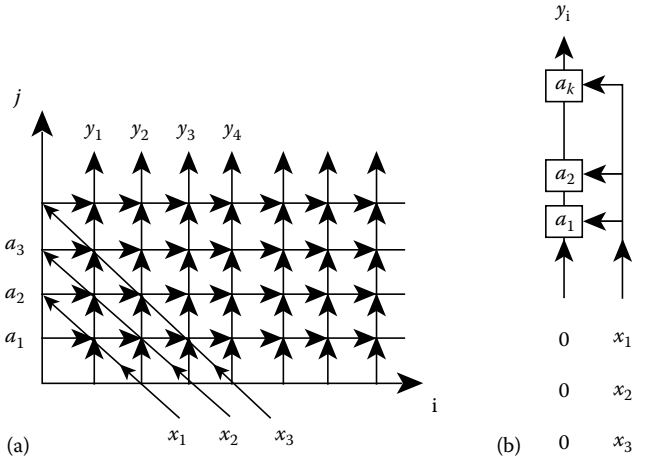


FIGURE 9.101 (a) Two-dimensional dependence graph and (b) one-dimensional dependence graph.

displacement vectors $D_1 = [0, 1]^T$, $D_2 = [1, 0]^T$, and $D_3 = [-1, 1]^T$. In particular, consider the URE representation of the B1 design based on the choice of $u = [1, 0]^T$, $\Sigma_+ = [0, 1]^T$, $\lambda = [1, 1]^T$, $\gamma_m = 0$, $m = 1$, 2, and 3, and $\tau_n = h_n = 1$, $n = 1$, 2, and 3. Then, the two-dimensional graph of Figure 9.101a is projected onto the one-dimensional graph of Figure 9.101b. Specifically, for any index $I = [i, j]^T$, the processor allocation function yields $A(I) = \Sigma_+^T I = j$, $i \le j \le k$, which is a valid projection from two-dimensions to one. On the other hand, the index point for each input data x_1 (with variable name of X_3) is given by $I = [i, l - i + 1]^T$, $l = 1, \ldots, n$. Then the scheduling function $S_3(I)$ is given by $S_3(I) = \lambda^T I = [1, 1]\ [i, 1 - i + 1]^T = i + l - i + 1 = l + 1$, $l = 1, \ldots, n$. This indicates each x_1 for the previously given I must be available at all the processors at time $l + 1$. Thus, there is no propagation and this means all the x_1 must be broadcasted to all the processors. However, the simplistic definition of a systolic design does not allow broadcasting. Indeed, the precedence constraint is *not* satisfied with D_3. That is, $\lambda^T D_3 = [1, 1]\ [-1, 1]^T = 0 \not\ge \tau_n = 1$. Of course, other choices of λ and u generate other forms of systolic array architecture for correlation. For more complicated signal processing tasks such as QR decomposition (QRD), recursive least-squares (LS) estimation, singular value decomposition (SVD), Kalman filtering (KF), etc., the design of efficient systolic array architectures are generally difficult. The dependence graph mapping technique provides a systematic approach to such designs by providing the proper selections of these λ and u vectors.

9.5.2 Digital Filters

The application of digital filtering has spread tremendously in recent years to numerous fields, such as signal processing, digital communications, image processing, and radar processing. It is well known that the sampling rate, which is closely related to the system clock, must be higher than the Nyquist frequency of the signals of interest. It follows that in order to perform real-time filtering operations when high frequency signals are involved, high-speed computing hardware is necessary.

Pipelining techniques have been widely used to increase the throughput of synchronous hardware implementations of a transfer function of a particular algorithm. Most algorithms can be described in a number of different ways, and each of these descriptions can be mapped onto a set of different concurrent architectures. Different descriptions may lead to realizations with entirely different properties, and can have a dramatic impact on the ultimate performance of the hardware implementation. Pipelining can also be used for other than throughput increase. For a fixed sample rate, a pipelined circuit is characterized by a lower power consumption. This is due to the fact that in a pipelined system capacitances can be charged and discharged with a lower power supply. Because the dissipated power depends quadratically on the voltage supply, the power consumption can be reduced accordingly.

An increase in the speed of the algorithm also can be achieved by using parallelism. By replicating a portion of the hardware architecture, similar or identical operations can be performed by two or more concurrent circuits, and an intelligent use of this hardware redundancy can result in a net throughput increase, at the expense of area. Note that VLSI technologies favor the design in which individual sections of the layout are replicated numerous times. A regular and modular design can be achieved at relatively low costs. For a fixed sample rate, parallelism can be exploited for a low power design due to the reduced speed requirements on each separate portion of the circuit.

Much work has been done in the field of systolic synthesis of finite and infinite impulse response (FIR/IIR) filters, as can be seen from the literature references. In the following subsection, we consider possible strategies that can be used to increase the throughput of the concurrent architectures of the FIR and IIR filters.

9.5.2.1 FIR Filters

FIR filters have been largely employed because of certain desirable properties. In particular, they are always stable, and causal FIR filters can posses linear phase. A large number of algorithms have been devised for the efficient implementation of FIR filters, which minimize the number of multipliers, the

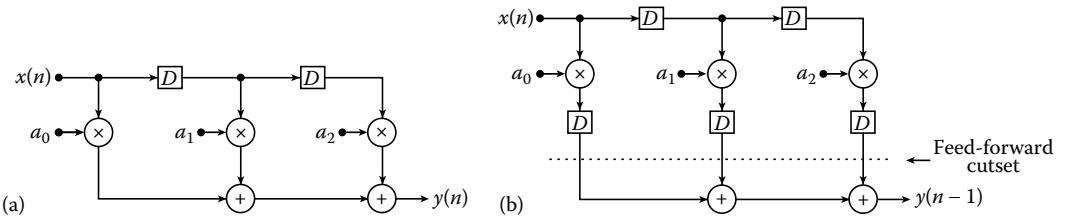


FIGURE 9.102 A three-tap FIR filter: (a) with no pipelining, the throughput is limited by the rate of one multiplication and two additions and (b) with pipelined circuit, the throughput is increased to the rate of one multiplication or two additions.

round-off noise, or the coefficient sensitivity. The generic expression that relates the output $y(n)$ at time n to the inputs $x(n-i)$ at times $(n-i)$ $i=0,1,\ldots,q$ is given by

$$y(n) = \sum_{i=0}^{q} a_i x(n-i)$$

where the $\{a_i\}_{i=0}^{q}$ are the FIR filter coefficients. Here, we consider only issues of pipelining and parallelism.

The pipeline rate, or throughput, of implemented nonrecursive algorithms such as FIR filters can be increased without changing the overall transfer function of the algorithms by means of a relatively simple modification of the internal structure of the algorithm. In particular, one set of latches and storage buffers can be inserted across any feed-forward cutset of the data flow graph. Figure 9.102b, illustrates the increase of throughout achieved by pipelining in a second-order three-tap FIR filter. The sample rate of the circuit of Figure 9.102a is limited by the throughput of one multiplication ''and'' two additions. After placing the latches at the locations shown in Figure 9.102b, the throughput can be increased to the rate of one multiplication ''or'' two additions. Pipelining can be used to increase the sample rate in all the cases in which no feedback loops are present. The drawbacks of pipelining are an increased latency and a larger number of latches and buffers.

Parallelism can be used to increase the speed of an FIR filter. Consider Figure 9.103, in which the three-tap FIR filter of Figure 9.102 was duplicated. Because at each time instant two input samples are processed and two samples are output, the effective throughput rate is exactly doubled.

As can be seen from Figure 9.103. parallelism leads to speed increase at a considerable hardware cost. For many practical implementations, parallelism and pipelining can be used concomitantly, when either method alone would be insufficient or limited by technology such as I–O, clock rate, etc.

9.5.2.2 IIR Filters

These are recursive filters in the sense that their output is function of current inputs as well as past outputs. The general I–O relationship is expressed by

$$y(n) = \sum_{j=1}^{p} a_i y(n-j) + \sum_{i=o}^{q} b_i x(n-i) \tag{9.22}$$

where the $\{a_j\}_{j=1}^{p}$ are the coefficients associated to the recursive part, and the $\{b_i\}_{i=0}^{q}$ are the coefficients associated to the nonrecursive portion of the filter. The associated transfer function is written as the following z-transform

$$H(z) = \frac{\sum_{i=0}^{q} b_i z^{-i}}{1 - \sum_{j=1}^{p} a_j z^{z-j}}$$

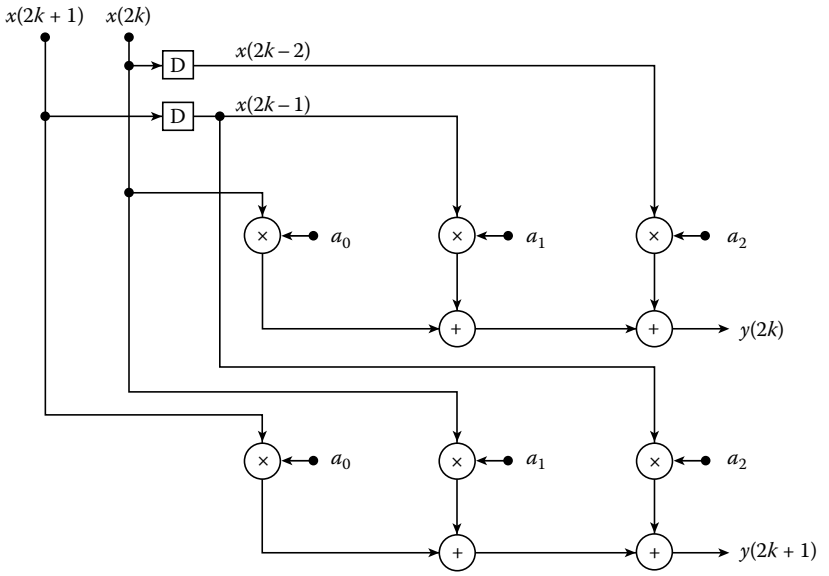


FIGURE 9.103 Three-tap FIR filter whose hardware has been duplicated to achieve double throughput rate.

For stability reasons, it is required that all the poles (i.e., the zeroes of the denominator of $H(z)$) be inside the unit circle in the z-plane.

Consider a circuit in which L loops are present each with latency τ_k, $k = 1, \ldots, L$. The number of latches present in each loop is equal to ν_k, $k = 1, \ldots, L$. Then the throughput period cannot be shorter than

$$T_{\max} \equiv \max_{k=l,\ldots,L} \left\lceil \frac{\tau_k}{\nu_k} \right\rceil$$

The pipeline can be increased by increasing the number of latches internal to the feedback loops. The computational latency associated with the internal feedback prevents one from introducing pipeline simply by inserting latches on feedforward cutsets. In fact, inserting latches in the loop would change the overall transfer function. This difficulty can be overcome by recasting the algorithm into an equivalent formulation from an I–O point of view. The transformations applied to the algorithm, prior to the mapping, have the purpose of creating additional concurrency, thereby increasing the achievable throughput rate. Without ever changing the algorithm's transfer function, additional delays are introduced inside the recursive loop. These delays are subsequently used for pipelining. In the sequel, we briefly describe two types of look-ahead techniques that generate the desired algorithmic transformations, namely the clustered and the scattered look-ahead techniques proposed by Loomis and Sinha [18] and Parhi and Messerschmitt [19], respectively. Look-ahead techniques are based on successive iterations of the basic recursion, in order to generate the desired level of concurrency. The implementation is then based on the iterated version of the algorithm.

Clustered look-ahead: In a pth order recursive system, the output at time n is a function of the past output samples $y(n-1)$, $y(n-2), \ldots, y(n-p)$. In the clustered look-ahead technique, the recursion is iterated m times so that the current output is a function of the cluster of p consecutive samples $y(n-m)$, $y(n-m-1), \ldots, y(n-m-p)$. The original order-$p$ recursive filter is emulated by a $(p+m)$th filter, where m canceling poles and zeroes have been added. In this way the m delays generated inside the feedback loop can be used to pipeline by m stages.

By iterating 9.22 m times, we can derive the following I–O relationship

$$y(n) = \sum_{j=1}^{p-1} \left[\sum_{k=j+1}^{p} a_k r_{j+m-k} \right] y(n-j-m) + \sum_{j=0}^{m-1} \sum_{k=0}^{q} b_k x(n-k-j)$$

where the coefficients $\{r_i\}$ can be precomputed off-line, and are such are such that $r_i = \Sigma_{k=1}^{p} a_k r_{i-k}$, $i > 0$, $r_0 = 1$, and $r_i = 0$, $i = -(p-1), \ldots, -1$. This implementation requires $(p+m)$ multiplications for the nonrecursive part, and p for the recursive part, for a total of $(2\,p+m)$, which grows linearly with m. The transfer function is equal to

$$H(z) = \frac{\sum_{j=0}^{m-1} \sum_{k=0}^{q} b_k z^{-k-j}}{1 - \sum_{j=1}^{p-1} \left[\sum_{k=j+1}^{p} a_k r_{j+m-k} \right] z^{-j-m}}$$

The clustered look-ahead technique does not guarantee that the resulting filter is stable because it may introduce poles outside the unit circle.

Consider the following simple example with a stable transfer function:

$$H(z) = \frac{1}{1 - 1.3z^{-1} + 0.35z^{-2}}$$

with poles at $z = 0.7$ and $z = 0.5$. The two-stage equivalent filter can be obtained by introducing the canceling pole-zero pair at $z = -1.3$, as follows:

$$H(z) = \frac{1 + 1.3z^{-1}}{(1 - 1.3z^{-1} + 0.35z^{-2})(1 + 1.3z^{-1})} = \frac{1 + 0.9z^{-1}}{1 - 1.34z^{-2} + 0.455z^{-3}}$$

Because a pole is found at $z = -1.3$, this transfer function is clearly unstable.

Scattered look-ahead: In the scattered look-ahead technique, the current output sample $y(n)$, is expressed in terms of the (scattered) p past outputs $y(n-m)$, $y(n-2m), \ldots, y(n-mp)$. The original order-p filter is now emulated by an order-mp filter. For each pole of the original filter, $(m-1)$ canceling pole-zero pairs are introduced at the same distance from the origin as the original pole. Thus, stability is always assured. The price we must pay is higher complexity, on the order of mp. To best describe the technique, it is convenient to write the transfer function $H(z)$ as a ratio of polynomials, i.e., $H(z) = N(z)/D(z)$. The transformation can be written as follows:

$$H(z) = \frac{N(z)}{D(z)} = \frac{N(z) \prod_{k=1}^{m-1} D\left(z e^{j(2\pi k/m)}\right)}{\prod_{k=0}^{m-1} D(z e^{j(2\pi k/m)})}$$

Note that the transformed denominator is now a function of z^{-m}.

Consider the example of the previous section. For the scattered look-ahead technique, it is necessary to introduce pole-zero pairs at $z = 0.7\ e^{\pm j(2\pi/3)}$ and $z = 0.5\ e^{\pm j(2\pi/3)}$. The transformed denominator equals $1 - 0.125z^{-3}$.

The complexity of the nonrecursive part of the transformed filter is $(pm+1)$ multiplications, while the recursive part requires p multiplications, for a total of $(pm+p+1)$ pipelined multiplications. Although the complexity is still linear in m, it is much higher than in the clustered look-ahead technique for a large value of p.

Parhi and Messerschmitt [19] presented a technique to reduce the complexity of the nonrecursive portion down to $O(p \log_2 m)$, applicable when m is a power of 2. This technique can be described as follows. Assume that the original recursive portion of the given IIR filter is given by

$$H(z) = \frac{1}{1 - \sum_{j=1}^{p} a_j^{(1)} z^{-j}}$$

An equivalent two-stage implementation of the same filter can be obtained by multiplying numerator and denominator by the polynomial $(1 - \Sigma_{j=1}^{p}(-1)^j a_j^{(1)} z^{-j})$, which is given by

$$H(z) = \frac{1 - \sum_{j=1}^{p} (-1)^j a_j^{(1)} z^{-j}}{1 - \sum_{j=1}^{p} a_j^{(2)} z^{-j}}$$

where the set of coefficient $\{a_j^{(2)}\}_{j=1}^{p}$ is obtained from the original set $\{a_j^{(1)}\}_{j=1}^{p}$ by algebraic manipulation. By repeating this process $\log_2 m$ times one can obtain an m-stage pipelined implementation, equivalent to the original filter. In this way the hardware complexity only grows logarithmically with the number of pipelining stages.

Bidirectional systolic arrays for IIR filtering: Lei and Yao [16] showed that many IIR filter structures can be considered as special cases of a general class of systolizable filters, as shown in Figure 9.104. These filters can be pipelined by rescaling the time so that $z' = z^{1/2}$, and by applying a cutset transformation. This time rescaling causes the hardware factorization to reduce to merely 50%, which is quite inefficient. Lei and Yao [17] later proposed two techniques to improve the efficiency of these bidirectional IIR filters.

In the first method (''overlapped subfilter scheme''), one makes use of the possibility to factor the numerator and the denominator of the given transfer function. For instance, if

$$H(z) = \frac{N(z)}{D(z)} = \underbrace{\frac{N_a(z)}{D_a(z)}}_{H_a(z)} \cdot \underbrace{\frac{N_b(z)}{D_b(z)}}_{H_b(z)}$$

where $a + b = p$, $a - b = 0$, 1, or 2, and p is the number of modules of the original transfer function. Then the two subfilters, $H_a(z)$ and $H_b(z)$, can be realized on the same systolic array of $a + 1$ modules, as in Figure 9.105. A multiplexer at the input of the array chooses the incoming data at even time instants, and the data from the output of the first module at odd time instants. The modules alternately perform operations associated to $H_a(z)$ and $H_b(z)$ in such away as to interleave the operations and have an overall 100% efficiency.

In the second technique (''systolic ring scheme'') the number of modules is about half of the order of the original transfer function. The modules of the new structure are arranged as a systolic ring, as in Figure 9.106. For example, a five-module ring can be used to implement a ten-module IIR filter: module i performs the operations associated to modules i and $(5 + i)$ of the original array, for $i = 1, \ldots, 5$. Note that in the original structure every other module is idle. The resulting ring is therefore 100% efficient.

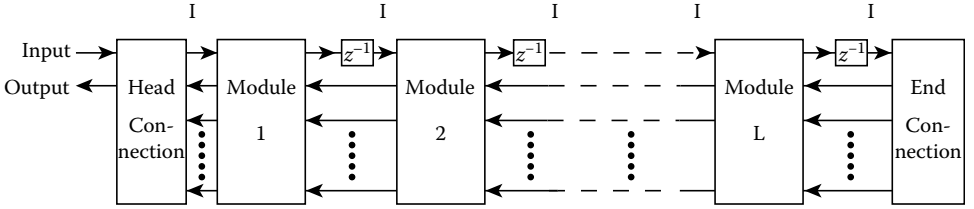


FIGURE 9.104 A general structure of bidirectional IIR filters.

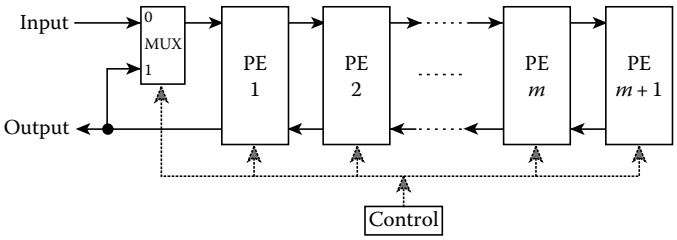


FIGURE 9.105 The overlapped subfilter scheme for IIR filtering.

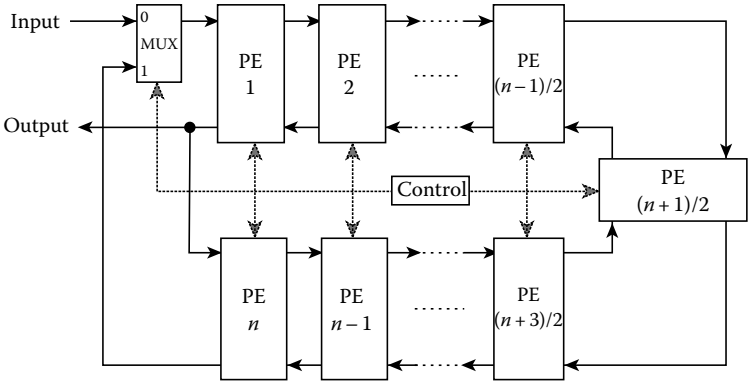


FIGURE 9.106 The systolic ring scheme for IIR filtering.

9.5.3 Systolic Word and Bit-Level Designs

Previous discussions on systolic array designs have taken place at the word level. This is to say that the smallest data or control item exchanged between pairs of processors is constituted by a ''word'' representable by B bits. Each processor in a word-level system has the capability of performing word-level operations. Some may be as complex as floating point multiplications or others as simple as square root operations, etc. The systolic array approach can be applied at various different levels beyond the word level, according to what is sometimes referred to as ''granularity'' of the algorithm description. Systolic arrays and associated dependence graphs can, in fact, be defined at high levels of description, in which each individual processor can, in principle, be a whole mainframe computer or even a separate parallel processor array. The communication between processors thus takes the form of complex protocols and entire data sequences. According to the same principle, the algorithm description can also be done at the lowest level of operation, namely, at the bit level, at which each processor is a simple latched logic gate, capable of performing a logic binary operation. The exchanged data and control also take the form of binary digits. The different approaches due to the different granularity of description have different merits and can be advantageously used in various circumstances or at different steps of the design.

These considerations bring to one possible design strategy, namely, the ''hierarchical systolic design.'' In this approach the complete systolic design is broken down to a sequence of hierarchical steps, each of which define the algorithm at different levels of granularity. At first, the higher level of description is adopted, the relative dependence graph is drawn, and, after suitable projection and scheduling, a high-level systolic architecture is defined. Subsequently, each high-level processor is described in terms of finer scale operations. Dependence graph and systolic architecture corresponding to these operations are produced and embedded in the higher level structure previously obtained. The process can continue down to the desired level of granularity. The simplest form of hierarchical design implies two steps. The first step involves the design of the word-level architecture. Second, the operations performed by each

work-level processor are described at bit level. The corresponding bit-level arrays are then nested into the word-level array, after ensuring that data flows at both levels are fully compatible.

The hierarchical approach has the merits of reducing the complexity of each step of the design. The dependence graphs involved usually have reduced dimensionality (thus, are more manageable), and the procedure is essentially recursive. The drawback of a hierarchical design is that it implicitly introduces somewhat arbitrary boundaries between operations, thereby reducing the set of resulting architectures. An approach that leaves all options open is to consider the algorithm at bit level from the outset. This approach has led to new insights and novel architectures. The price to pay is that the designer must deal with dependence graphs of higher dimensionality. As an example, the dependence graph of the inner product between two N-vectors, $c = \Sigma_{i=0}^{N-1} a_i b_i$ is two dimensional. If the same inner product is written at bit level, i.e., $c_k = \Sigma_{i=0}^{N-1} \Sigma_{j=0}^{B-1} a_{i,j} b_{i,k-j} +$ carries, $k = 0, \ldots, B-1$, then it produces a three-dimensional dependence graph.

Examples of the two design procedures applied to the convolution problem are considered below. First, consider the factors that can make bit-level design advantageous:

- *Regularity*. Most bit-level arrays are highly regular. Only relatively simple cells need to be designed and tested. The communication pattern is simple and regular. Neighbor-to-neighbor connections allow high packing density and low transmission delays.
- *High pipeline rate*. Because the individual cells have reduced computation time (on the order of the propagation delay through a few gates), the overall throughput can be made very high.
- *Inexpensive fault tolerance*. The use of bypass circuitry can be made without wasting too much of the silicon area.

It must be borne in mind that bit-level arrays realistically cannot be operated in wavefront array mode because the interprocessor hand-shaking protocols would be too expensive as compared to the data exchange. A good clock signal distribution is therefore needed to synchronize the array operations. In systolic arrays, unlike synchronous architectures of a different sort, only the incremental clock skew must be minimized by suitably designing the clock signal distribution lines. This problem may become particularly delicate in bit-level arrays, where the number of processors involved is very high.

9.5.3.1 Bit-Level Design of a Serial Convolver

Bit-level systolic design was first proposed by McCanny and McWhirter [29]. Subsequently, they and others have applied this technique to various algorithms. As a simple example, consider the bit-level design of a serial convolver. The word-level output of an N point convolver can be written as $y_k = \Sigma_{i=0}^{N-1} a_i x_{k-i}$, $k = 0, 1, \ldots$, where $\{a_i\}_{i=0}^{N-1}$ is a given set of coefficients and x_i, $i = 0, 1, \ldots$, is a sequence of input data. Coefficients and data values are assumed to be B-bit words. The word-level dependence graph is shown in Figure 9.107, together with one possible systolic realization. In this case, the coefficients are permanently stored in each individual cell. I/O values are propagated in opposite directions. In each PE, the corresponding coefficient is multiplied by the incoming data value. This product is added to the partial output value and the accumulated result is propagated forward. Each cell performs the simple multiply and add operation expressed by $y_{k,i+1} \leftarrow y_{k,i} + a_i x_{k-i}$, $y_k = y_{k,N}$.

According to the hierarchical approach, one must now proceed to determine the dependence graph corresponding to the bit-level description of the multiply-and-add operation. The complete dependence graph can be subsequently obtained by embedding the finer scale graph into the higher level graph. If both a_i and x_i are B-bit binary numbers, then the jth bit of y_{ki}, can be computed according to $y_{k,i,j} = y_{k,i,j} + s_{i,k,j}, s_{i,k,j} + \Sigma_{l=0}^{B-1} a_{i,l} x_{k-i,j-l} +$ carries, where $a_{i,l}$ and $x_{i,l}$, $l = 0, \ldots, B-1$, represent the lth bit of a_i and x_i. The dependence graph corresponding to this operation is given in Figure 9.108, where subscripts only indicate the bit position, and $B = 3$. Note that this graph is quite similar to the graph corresponding to a convolver, apart from the carry bits, which are taken care of by the insertion of an additional row of cells.

The combined dependence graph, obtained from the word dependence graph of Figure 9.107, in which each cell is replaced by the bit-level dependence graph of Figure 9.108, is given in Figure 9.109.

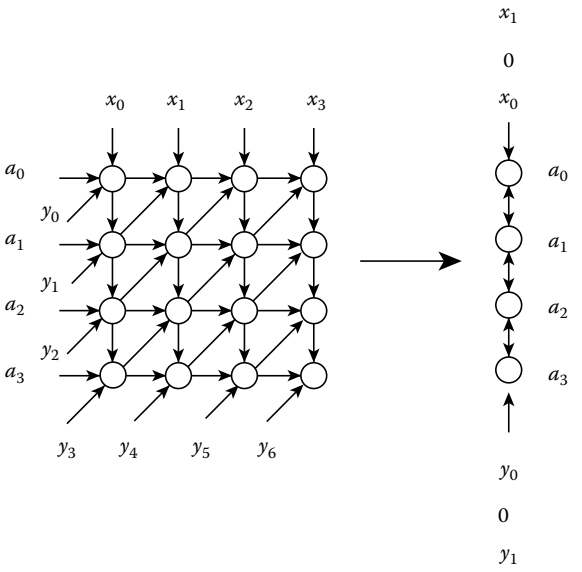


FIGURE 9.107 World-level dependence graph of an N point convolution operation with one possible systolic realization.

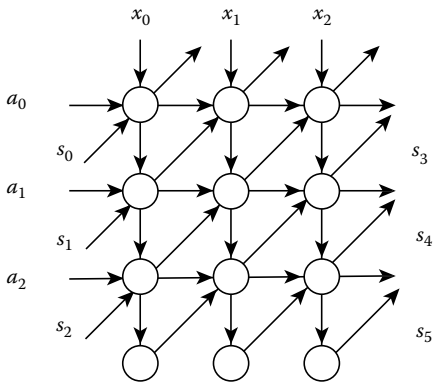


FIGURE 9.108 Bit-level dependence graph corresponding to the multiply-and-add operation.

The data flows are fully compatible at both word and bit levels. At this point, a full two-dimensional bit-level systolic array can be obtained from the final dependence graph by simply replacing each node with latched full adder cells. Different linear systolic implementations can be obtained by projecting the combined dependence graph along various directions. One possibility is again to keep the coefficients residents in individual cells, and have input data bits and accumulated results propagate in opposite directions. The schematic representation of the systolic array with these features is drawn in Figure 9.109. Judgment about the merits of different projections involves desired data movement, I–O considerations, throughput rate, latency time, efficiency factor (ratio of idle time to busy time per cell), etc.

As discussed previously, the convolution operation can be described at bit level from the very beginning. In this case the expression for the jth bit of the kth output can be expressed as follows:

$$y_{k,j} = \sum_{i=0}^{N-1} \sum_{l=0}^{B-1} a_{i,l} x_{k-i,j-1} + \text{carries} \tag{9.23}$$

By using this expression as a starting point, one is capable of generating a number of feasible systolic realizations potentially much larger than what is attainable from the two-step hierarchical approach. The reason for this can be simply understood by nothing that in this formulation no arbitrary precedence relationship is imposed between the two summations on i and l, whereas earlier we required that the summation on l would always ''precede'' the summation on i. The result is a fairly complicated three-dimensional dependence graph of size $N \times B \times$ number of inputs, as shown in Figure 9.110. Observe that

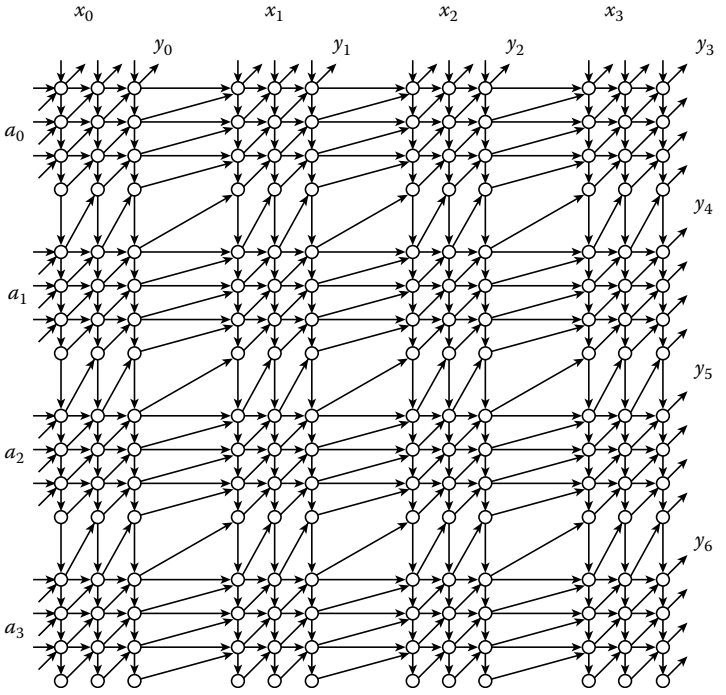


FIGURE 9.109 Bit-level dependence graph for convolution obtained by embedding the bit-level graph into the word-level graph.

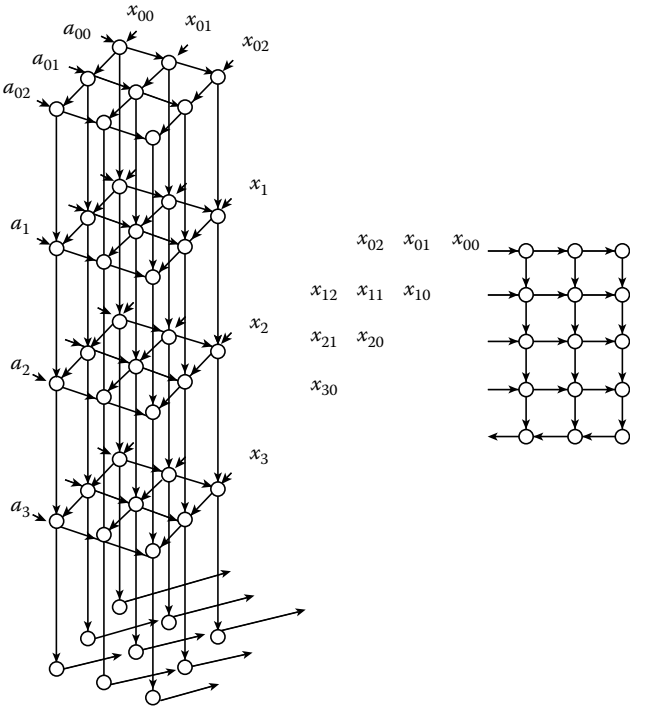


FIGURE 9.110 General three-dimensional bit-level dependence graph for convolution.

the bottom level of the dependence graph corresponds to the summation over l in Equation 9.23. In the same figure a schematic two-dimensional bit-level systolic realization of the algorithm is given, in which the coefficient bits are held in place. Projections along different directions have different characteristics and may be considered preferable in different situations. The choice ultimately must be made according to given design constraints or to efficiency requirements.

The concept of bit-level design, as considered here, can be applied to a large variety of algorithms. Indeed, it has generated a number of architectures, including FIR/IIR filters, arrays for inner product computation, median filtering, image processing, eigenvalue problems, Viterbi decoding, etc.

9.5.4 Recursive LSs Estimation

9.5.4.1 LSs Estimation

The LS technique constitutes one of the most basic components of all modern signal processing algorithms dealing with linear algebraic and optimization of deterministic and random signals and systems. Specifically, some of the most computationally intensive parts of modern spectral analysis, beam formation, direction finding, adaptive array, image restoration, robotics, data compression, parameter estimation, and KF all depend crucially on LS processing.

Regardless of specific application, an LS estimation problem can be formulated as $Ax \approx y$, where the $m \times n$ data matrix A and the $m \times 1$ data vector y are known, and we seek the $n \times 1$ desired solution x. In certain signal processing problems, rows of A are composed of sequential blocks of lengths n taken from a one-dimensional sequence of observed data. In other n-sensor multichannel estimation problems, each column of A denotes the sequential outputs of a given sensor. In all cases, the desired solution x provides the weights on the linear combinations of the columns of A to optimally approximate the observed vector y in the LS sense. When $m = n$ and A is nonsingular, then an exact solution for x exists. The Gaussian elimination method provides an efficient approach for determining this exact solution. However, for most signal processing problems, such as when there are more observations than sensors, and thus $m > n$, then no exact solution exists. The optimum LS solution $\hat{x}$ is defined by $\|A\hat{x} - y\| = \min_x \|Ax - y\|$. The classical approach in LS solution is given by $\hat{x} = A^+ y$, where A^+ is the pseudo-inverse of A defined by $A^+ = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}}$. The classical LS approach is not desirable from the complexity, finite precision sensitivity, and processing architecture points of views. This is due to the need for a matrix inversion, the increase of numerical instability from "squaring of the condition number" in performing the $A^{\mathrm{T}}A$ operation, and the block nature of the operation in preventing a systolic update processing and architecture for real-time applications.

The QRD approach provides a numerically stable technique for LS solution that avoids the objections associated with the classical approach. Consider a real-valued $m \times n$ matrix A with $m \geq n$ and all the columns are linearly independent (i.e., rank $A = n$). Then, from the QRD, we can find a $m \times m$ orthogonal matrix Q such that $QA = \overline{R}$. The $m \times n$ matrix $\overline{R} = [R^{\mathrm{T}}, 0^{\mathrm{T}}]^{\mathrm{T}}$ is such that R is an $n \times n$ upper triangular matrix (with nonzero diagonal elements) and 0 is an all-zero $(m - n) \times n$ matrix. This upper triangularity of R is used crucially in the following LS solution problem.

Because the l_2 norm of any vector is invariant with respect to an orthogonal transformation, an application of the QRD to the LS problem yields $\|Ax - y\|^2 = \|Q(Ax - y)\|^2 = \|\overline{R}x - f\|^2$, where f is an $m \times 1$ matrix given by $f = Qy = [u^{\mathrm{T}}, v^{\mathrm{T}}]^{\mathrm{T}}$. Denote $e = Ax - y$ as the "residual" of the LS problem. Then, the previous LS problem is equivalent to $\|e\|^2 = \|Ax - y\|^2 = \|[Rx, 0x]^{\mathrm{T}} - [u^{\mathrm{T}} - v^{\mathrm{T}}]^{\mathrm{T}}\|^2 = \|Rx - u\|^2 + \|v\|^2$. Because R is a nonsingular upper triangular square matrix, the back substitution procedure of the Gaussian elimination method can be used to solve for the exact solution $\hat{x}$ of $R\hat{x} = u$. Finally, the LS problem reduces to $\min_x \|Ax - y\|^2 = \|A\hat{x} - y\|^2 = \|-R\hat{x} - f\|^2 = \|R\hat{x} - u\|^2, + \|v\|^2 = \|v\|^2$. For the LS problem, any QRD technique such as the Gram–Schmidt method, the modified-Gram–Schmidt (MGS) method, the Givens transformation, and the Householder transformation is equally valid for

finding the matrix R and the vector v. For a systolic implementation, the Givens transformation yields the simplest architecture, but the MGS and Householder transformation techniques are also possible with slight advantages under certain finite precision conditions.

9.5.4.2 Recursive LSs Estimation

The complexity involved in the computation of the optimum residual $\hat{e}$ and the optimum LS solution vector $\hat{x}$ can become arbitrarily large as the number of samples in the column vectors of A and y increases. In practice, we must limit m to some finite number greater than the number of columns n. Two general approaches in addressing this problem are available. In the ''sliding window'' approach, we periodically incorporate the latest observed set of data (i.e., ''updating'') and possibly remove an older set of data (i.e., ''downdating''). In the ''forgetting factor'' approach, a fixed scaling constant with a magnitude between 0 and 1 is multiplied against the R matrix and thus exponentially forget older data. In either approach, we find the optimum LS solution weight vector $\hat{x}$ in a recursive LSs manner. As the statistics of the signal change over each window, these $\hat{x}$ vectors change ''adaptively'' with time. This observation motivates the development of a recursive LSs solution implemented via the QRD approach. For simplicity, we consider only the updating aspects of the sliding window recursive LSs problem.

Let m denote the present time of the sliding window of size m. Consider the $m \times n$ matrix $A(m)$, the $m \times 1$ column vector $y(m)$, the $n \times 1$ solution weight column vector $x(m)$, and the $m \times 1$ residual column vector $e(m)$ expressed in terms of their values at time $m-1$ as $A(m) = [\alpha\ (1), \ldots, \alpha\ (m)]^{\mathrm{T}} = [A(m-1)^{\mathrm{T}}, \alpha\ (m)]^{\mathrm{T}}$, $y(m) = [y_1, \ldots, y_m]^{\mathrm{T}}\ [y(m-1)^{\mathrm{T}}, y^{\mathrm{T}}m]^{\mathrm{T}}$, $x(m) = [x_1(m), \ldots, x_n(m)]^{\mathrm{T}}$, and $e(m)\ A(m) \times (m) - y(m) = [e_1(m), \ldots, e_n(m)]^{\mathrm{T}}$. By applying the orthogonal matrix $Q(m) = [Q_1(m)^{\mathrm{T}}, Q_2(m)^{\mathrm{T}}]^{\mathrm{T}}$ of the QRD of the $m \times n$ matrix $A(m)$, we obtain $Q(m)\ A(m) = [R(m)^{\mathrm{T}},\ 0^{\mathrm{T}}]^{\mathrm{T}} = R_0(m)$ and $Q(m)y(m) = [Q_1(m)^{\mathrm{T}}, Q_2(m)^{\mathrm{T}}]^{\mathrm{T}}\ y(m) = [u(m)^{\mathrm{T}},\ v(m)^{\mathrm{T}}]^{\mathrm{T}}$. The square of the l_2 norm of the residual e is then given by $\in(m) = \|e(m)\|^2 = \|A(m)x(m) - y(m)\|^2 = \|Q(m)(A(m)x(m) - y(m))\|^2 = \|R(m)x(m) - u(m)\|^2 + \|v(m)\|^2$. The residual is minimized by using the back substitution method to find the optimum LS solution $\hat{x}(m)$ satisfying $R(m)\hat{x}(m) = u(m) = [u_1(m), \ldots, u_n(m)]^{\mathrm{T}}$. It is clear that the optimum residual $\hat{e}(m)$ is available after the optimum LS solution $\hat{x}(m)$ is available as seen from $\hat{e}(m) = A(m)\hat{x}(m) - y(m)$. It is interesting to note that it is not necessary to first obtain $\hat{x}(m)$ explicitly and then solve for $\hat{e}(m)$ as shown earlier. It is possible to use a property of the orthogonal matrix $Q(m)$ in the QRD of A and the vector $y(m)$, to obtain $\hat{e}(m)$ explicitly. Specifically, note $\hat{e}(m) = A(m)\hat{x}(m) - y(m) = Q_1(m)^{\mathrm{T}}\ R(m)\hat{x}(m) - y(m) = [Q_1(m)^{\mathrm{T}}\ Q_1(m) - I_m]y(m) = -Q_2(m)^{\mathrm{T}}\ Q_2(m)y(m) = -Q_2(m)^{\mathrm{T}}v(m)$. This property is used explicitly in the following systolic solution of the last component of the optimum residual.

9.5.4.3 Recursive QRD

Consider the recursive solution of the QRD. First, assume the decomposition at step $m-1$ has been completed as given by $Q(m-1)A(m-1) = [R(m-1)^{\mathrm{T}}, 0^{\mathrm{T}}]^{\mathrm{T}}$ by using a $(m-1) \times (m-1)$ orthogonal matrix. Next, define a new $m \times m$ orthogonal transformation $T(m) = [Q(m-1), 0; 0, 1]$. By applying $T(m)$ on the new $m \times n$ data $A(m)$, which consists of the previously available $A(m-1)$ and the newly available row vector $\alpha(m)^{\mathrm{T}}$ we have

$$T(m)A(m) = \begin{bmatrix} Q(m-1) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A(m-1) \\ \alpha(m)^T \end{bmatrix} = \begin{bmatrix} Q(m-1)A(m-1) \\ \alpha(m)^T \end{bmatrix}$$
$$= \begin{bmatrix} R(m-1) \\ 0 \\ \alpha(m)^T \end{bmatrix} = R_1(m)$$

While $R(m-1)$ is an $n \times n$ upper triangular matrix, $R_1(m)$ does not have the same form as the desired $R_0(m) = [R(m)^{\mathrm{T}}, 0^{\mathrm{T}}]^{\mathrm{T}}$ where $R(m)$ is upper triangular.

9.5.4.4 Givens Orthogonal Transformation

Next, we want to transform $R_1(m)$ to the correct $R_0(m)$ form by an orthogonal transformation $G(m)$. While any orthogonal transformation is possible, we will use the Givens transformation approach due to its simplistic systolic array implementation. Specifically, denote $G(m) = G_n(m)G_{n-1}(m)\ldots G_1(m)$, where $G(m)$ as well as each $G_i(m)$, $i = 1,\ldots n$, are all $m \times m$ orthogonal matrices. Define

$$G_i(m) = \begin{array}{c} \\ 1 \\ \\ i \\ \\ m \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & & i & & m \end{array} \\ \begin{bmatrix} 1 & & 0 & & 0 \\ & 1 & & & \\ 0 & & c_i(m) & & s_i(m) \\ & & & 1 & \\ 0 & & -s_i(m) & & c_i(m) \end{bmatrix} \end{array}, \quad i = 1,\ldots,n$$

as a $m \times m$ identity matrix, except that the (i, i) and (m, m) elements are specified as $c_i(m) = \cos\theta_i(m)$, where $\theta_i(m)$ represents the rotation angle at the ith iteration, the (i, m) element as $s_i(m) = \sin\theta_i(m)$, and the (m, i) element as $-S_i(m)$. By cascading all the $G_i(m)$, $G(m)$ can be reexpressed as

$$G(m) = \begin{bmatrix} k(m) & 0 & d(m) \\ 0 & I_{m-n-1} & 0 \\ h^T(m) & 0 & \gamma(m) \end{bmatrix}$$

where $k(m)$ is $n \times n$, $d(m)$ and $h(m)$ are $n \times 1$, and $\gamma(m)$ is 1×1. In general $k(m)$, $d(m)$, and $h(m)$ are quite involved functions of $c_i(m)$ and $s_i(m)$, but $\gamma(m)$ is given simply as $\gamma(m) = \prod_{i=1}^{n} c_i(m)$ and will be used in the evaluation of the optimum residual.

Use $G(m)$ to obtain $G(m)T(m)A(m) = G(m)R_1(m)$. In order to show the desired property of the n orthogonal transformation operations of $G(m)$, first consider

$$G_1(m)R_1(m) = \begin{bmatrix} c_1(m) & & & & s_1(m) \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ s_1(m) & & & & c_1(m) \end{bmatrix} \begin{bmatrix} x & x & \cdots & x \\ 0 & x & \cdots & x \\ & & \cdots & x \\ 0 & 0 & \cdots & x \\ x & x & \cdots & x \end{bmatrix}$$

$$= \begin{bmatrix} x & x & \cdots & x \\ 0 & x & \cdots & x \\ & & \cdots & x \\ 0 & 0 & \cdots & x \\ 0 & x & \cdots & x \end{bmatrix}$$

In the preceding expression, an x denotes some nonzero valued element. The purpose of $G_1(m)$ operating on $R_1(m)$ is to obtain a zero at the $(m, 1)$ position without changing the $(m-2) \times n$ submatrix from the second to the $(m-1)$st rows of the r.h.s. of the expression. In general, at the ith iteration, we have

$$G_i(m) \begin{bmatrix} x & x & . & . & . & x \\ & x & . & . & . & x \\ & & & & & x \\ & & & \vdots & & \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & x & x & x \\ & & & \underset{i-1}{} & & \end{bmatrix} = \begin{bmatrix} x & x & . & . & . & x \\ & x & . & . & . & x \\ & & & & & x \\ & & & \vdots & & \\ 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 0 & x & x \\ & & & \underset{i}{} & & \end{bmatrix}$$

The preceding zeroing operation can be explained by noting that the Givens matrix $G_i(m)$ operates as a $(m-2)\times(m-2)$ identity matrix on all the rows on the right of it except the ith and the mth rows. The crucial operations at the ith iteration on these two rows can be represented as

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \quad \begin{bmatrix} 0 & \cdots & 0 & r_i & r_{i+1} & \cdots & r_n \\ 0 & \cdots & 0 & \underset{i}{a_i} & a_{i+1} & \cdots & a_n \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \cdots & 0 & r_i^T & r_{i+1}^T & \cdots & r_n^T \\ 0 & \cdots & 0 & \underset{i}{0} & a_{i+1}^T & \cdots & a_n^T \end{bmatrix}$$

For simplicity of notation, we suppress the dependencies of i and m on c and s. Specifically, we want to force $a_i^{\mathrm{T}}=0$ as given by $0=a_i^{\mathrm{T}}=-sr_i+ca_i$. In conjunction with $c^2+s^2=1$, this requires $c^2=r_i^2/(a_i^2+r_i^2)$ and $s^2=a_i^2/(a_i^2/r_i^2)$. Then $r_i^{\mathrm{T}}=cr_i+sa_i=\sqrt{(a_i^2+r_i^2)}$, $c=r_i/r_i^{\mathrm{T}}$, and $s=a_i/r_i^{\mathrm{T}}$. This shows from the individual results of $G_1(m)$, $G_2(m),\ldots,$ $G_n(m)$, the overall results yield $Q(m)$ $A(m)=G(m)\overline{R}(m)=[R(m)^{\mathrm{T}}$, $0^{\mathrm{T}}]^{\mathrm{T}}=R_0(m)$, with $Q(m)=G(m)T(m)$.

9.5.4.5 Recursive Optimal Residual and LS Solutions

Consider the recursive solution of the last component of the optimum residual $\hat{e}(m)=[\hat{e}_1(m),\ldots,$ $\hat{e}_m(m)]^{\mathrm{T}}=-Q_2(m)^{\mathrm{T}}\,v(m)=-Q_2(m)^{\mathrm{T}}\,[v_1(m),\ldots,v_m(m)^{\mathrm{T}}]$. Because $Q_2(m)=[Q_2(m-1),\ 0;\ h(m)^{\mathrm{T}}$ $Q_1(m-1),\ \gamma(m)]$, then $\hat{e}(m)=[\hat{e}_1(m),\ldots,\hat{e}_m(m)]^{\mathrm{T}}=Q_2(m)=-[Q_2^{\mathrm{T}}(m-1),\ Q_1^{\mathrm{T}}(m-1)h(m);\ 0,\ \gamma(m)]$ $[v_1(m),\ldots,v_m(m)]$. Thus, the last component of the optimum residual is given by $\hat{e}(m)=-\gamma(m)$ $v_m(m)=-\prod_{i=1}^{n}c_i(m)v_m(m)$, which depends on all the products of the cosine parameters $c_i(m)$ in the Givens QR transformation, and $v_m(m)$ is just the last component of $v(m)$, which is the result of $Q(m)$ operating on $y(m)$.

As considered earlier, the LS solution $\hat{x}$ satisfies the triangular system of equations. After the QR operation on the extended matrix $[A(m),\ Y(m)]$, all the $r_{ij}, j\geq i=1,\ldots,n$ and $u_i, i=1,\ldots,n$ are available. Thus, $\{\hat{x}_1,\ldots,\hat{x}_n\}$ can be obtained by using the back substitution method of $\hat{x}_i=(u_i-\Sigma_{j-i+1}^{n}r_{ij}\hat{x}_j/r_{ij})$, $i=n,\ n-1,\ldots,\ 1$. Specifically, if $n=1$, then $\hat{x}_1=u_1/r_{11}$. If $n=2$, then $\hat{x}_2=u_2/r_{22}$ and $\hat{x}_1=u_1-r_{12}\hat{x}_2/$ $r_{11}=u_1/r_{11}-u_2r_{12}/r_{11}r_{22}$. If $n=3$, then $\hat{x}_3=u_3/r_{33},\hat{x}_2=u_2-r_{23}x_3/r_{22}=u_2/r_{22}-r_{23}u_3/r_{22}r_{23}$, and $\hat{x}_1=u_1-r_{12}\hat{x}_2-r_{13}\hat{x}_3/r_{11}=u_1/r_{11}-r_{12}u_2/r_{11}r_{22}+u_3[-r_{13}/r_{11}r_{33}+r_{12}r_{23}/r_{11}r_{22}r_{33}]$.

9.5.4.6 Systolic Array Implementation for QRD and LS Solution

The recursive QRD considered above can be implemented on a two-dimensional triangular systolic array based on the usage of four kinds of processing cells. Figure 9.111a shows the boundary cell for the generation of the sine and cosine parameters, s and c, needed in the Givens rotations. Figure 9.111b shows the internal cell for the proper updating of the QRD transformations. Figure 9.111c shows the single output cell needed in the generation of the last component of the optimal residual $\hat{e}_m(m)$ as well as the optimal LS solution $\hat{x}(m)$. Figure 9.111d shows the delay cell which performs a unit time delay for proper time skewing in the systolic processing of the data.

Figure 9.112 shows a triangular systolic array capable of performing the recursive QRD for the optimal recursive residual estimation and the recursive LSs solution by utilizing the basic processing cells in Figure 9.111. In particular, the associated LS problem uses an augmented matrix $[A,y]$ consisting of the $m\times n$ observed data matrix A and the $m\times 1$ observed vector y. The number or processing cells in the triangular array consists of n boundary cells, $n(n+1)/2$ internal cells, one output cell, and n delay cells.

The input to the array in Figure 9.112 uses the augmented matrix

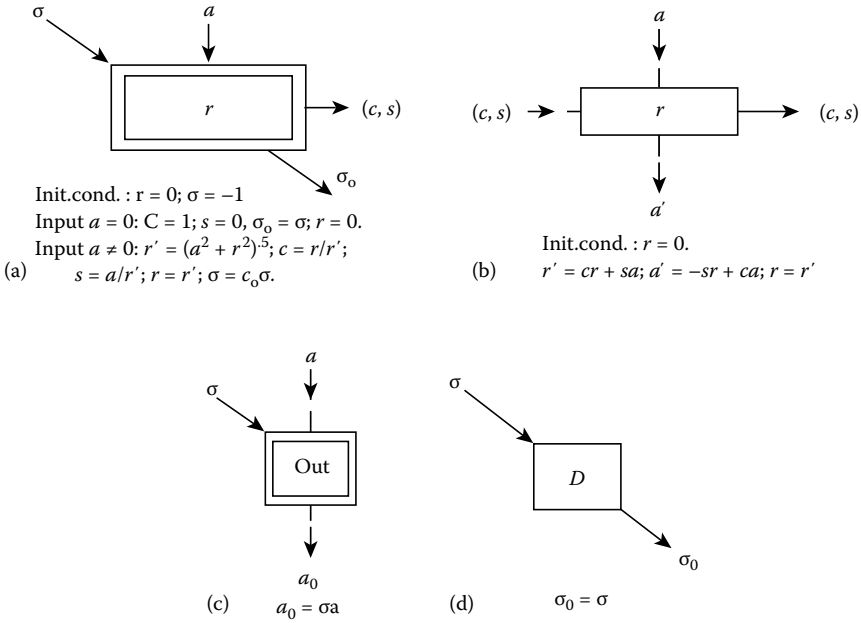


FIGURE 9.111 (a) Boundary cell; (b) internal cell; (c) output cell; and (d) delay cell.

$$[A,Y] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & y_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & y_2 \\ & & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} & y_m \end{bmatrix}$$

skewed in a manner such that each successive column from left to right is delayed by a unit time as given by

$$\begin{bmatrix} a_{11} & 0 & & \cdots & 0 & 0 \\ a_{21} & a_{12} & 0 & \cdots & 0 & 0 \\ a_{31} & a_{22} & a_{13} & \cdots & 0 & 0 \\ & & & \vdots & & \\ a_{n1} & a_{(n-1)2} & & & a_{1n} & 0 \\ a_{(n+1)1} & a_{n2} & & & a_{2n} & y_1 \\ & & & \vdots & & \\ a_{m1} & a_{(m-1)2} & & & a_{(m-n+1)n} & y_{m-n} \\ & & & \vdots & & \\ 0 & & & & a_{mn} & y_{m-1} \\ 0 & & & & 0 & y_m \end{bmatrix} \begin{matrix} k=1 \\ 2 \\ 3 \\ \vdots \\ n \\ n+1 \\ \vdots \\ m \\ \vdots \\ m+n-1 \\ m+n \end{matrix}$$

We see that at time k, input data consists of the kth row of the matrix, and moves down with increasing time. However, in Figure 9.112, purely for drawing purpose in relation to the position of the array, the relevant rows of data are drawn as moving up with increasing k.

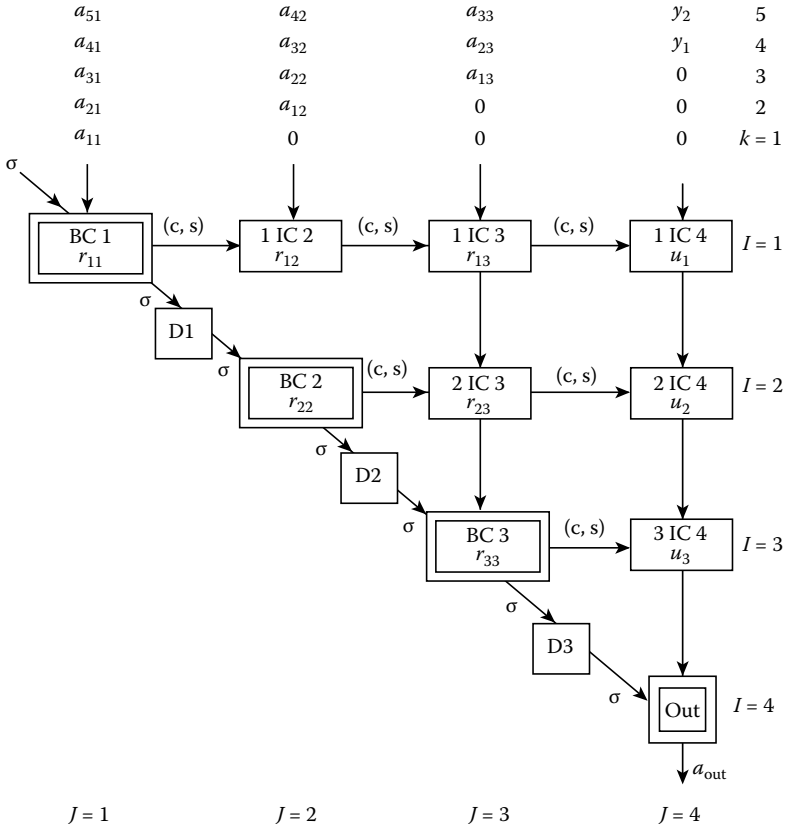


FIGURE 9.112 Triangular systolic array implementation of an $n = 3$, QRD-recursive, LSs solver.

Consider some of the iterative operations of the QRD for the augmented matrix $[A, y]$ for the systolic array in Figure 9.112. At time $k = 1$, a_{11} enters BC 1 and results in $c = 0$, $s = 1$, and $r_{11} = a_{11}$. All other cells are inactive. At $k = 2$, a_{21} enters BC 1, with the results $c = a_{11}\sqrt{(a_{11}^2 + a_{21}^2)}$, $s = a_{21}\sqrt{(a_{11}^2 + a_{21}^2)}$,$r_{11} = \sqrt{a_{11}^2 + a_{21}^2}$. This r_{11} corresponds to that of r_i^{T}, while the preceding c and s correspond to the c and s in the Givens transformation. Indeed, the new a_i^{T} is zero and does not need to be saved in the array. Still, at $k = 2$, a_{12} enters 1 IC 2 and outputs $a^{\mathrm{T}} = 0$ and $r_{12} = a_{12}$. At $k = 3$, a_{13} enters BC 1, and the Givens rotation operation continues where the new r_i is given by the previously processed r_i^T and a_i is now given by a_{13}. Meanwhile, a_{22} enters at 1 IC 2. It outputs $a^{\mathrm{T}} = -a_{21}a_{12}/\sqrt{(a_{11}^2 + a_{21}^2)} + -a_{22}a_{21}/\sqrt{(a_{11}^2 + a_{21}^2)}$, which corresponds to that of a_{i+1}^T, and $r_{12} = a_{11}a_{12}/\sqrt{(a_{11}^2 + a_{21}^2)} + -a_{21}a_{22}/\sqrt{(a_{11}^2 + a_{21}^2)}$, which corresponds to that of r_{i+1}^T. In general, the top (i.e., $I = 1$) row of the processing cells performs Givens rotation by using the first row to operate on the second, third,..., mth rows (each row with $n + 1$ elements), such that the $\{21, 31, \ldots, m1\}$ locations in the augmented matrix are all zeroed. The next row ($I = 2$) of cells uses the second row to operate on the third,..., mth rows (each row with n elements), such that locations at $\{32, 42, \ldots, m2\}$ are zeroed. Finally, at row $I = n$, by using the nth row to operate on the $(n + 1)$st,..., mth rows, elements at locations $\{(n + 1)n, (n + 2)n, \ldots, mn\}$ are zeroed. We also note that the desired cosine values in $\gamma(m)$ are being accumulated by c along the diagonal of the array. Delay cells $\{D_1, D_2, \ldots, D_n\}$ are used to provide the proper timing along the diagonal.

The cell BC 1 (at $I = J = 1$) terminates in the QR operation at time $k = m$, while the cell at $I = 1$ and $J = 2$ terminates at $k = m + 1$. In general, the processing cell at location (I, J) terminates at

$k = I + J + m - 2$. In particular, the last operation in the QRD on the augmented matrix is performed by the cell at $I = n$ and $J = n + 1$ at time $k = 2n + m - 1$. Then, the last component of the optimum residual $e_m(m)$ exits the output cell at time $k = 2n + m$.

After the completion of the QRD obtains the upper triangular system of equation, we can "freeze" the r_{IJ} values in the array to solve for the optimum LS solution $\hat{x}$ by the back substitution method. Specifically, we can append $[I_n, 0]$, where I_n is a $n \times n$ identity matrix and 0 is an $n \times 1$ vector of all zeroes, to the bottom of the augmented matrix $[A, y]$. Of course, this matrix is skewed as before when used as input to the array. In particular, immediately after the completion of the QR operation at BC 1, we can input the unit value at time $k = m + 1$. This is stage 1 of the back substitution method. Due to skewing, a unit value appears at the $I = 1$ and $J = 2$ cell at stage 3. Finally, at stage $(2n - 1)$, which is time $k = m + 2n - 1$, the last unit value appears at the $I = n$ and $J = 1$ cell. For our example of $n = 3$, this happens at stage 5. The desired LS solution $\hat{x}_1$ appears at stage $(2n + 1)$ (i.e., stage 7 for $n = 3$), which is time $k = 2n + m + 1$, while the last solution $\hat{x}_n$ appears at stage $3n$ (i.e., stage 9 for $n = 3$), which is time $k = 3n + m$. The values of$\{\hat{x}_1, \hat{x}_2, \hat{x}_3\}$ at the output of the systolic array are identical to those given by the back substitution method solution of the LS problem.

9.5.5 Kalman Filtering

KF was developed in the late 1950s as a natural extension of the classical Wiener filtering. It has profound influence on the theoretical and practical aspects of estimation and filtering. It is used almost universally for tracking and guidance of aircraft, satellites, GPS, and missiles as well as many system estimation and identification problems. KF is not one unique method, but is a generic name for a class of state estimators based on noisy measurements. KF can be implemented as a specific algorithm on a general-purpose mainframe/mini/microcomputer operating in a batch mode, or it can be implemented on dedicated system using either DSP, ASIC, or custom VLSI processors in a real-time operating mode.

Classically, an analog or a digital filter is often viewed in the frequency domain having some low-pass, bandpass, high-pass, etc. properties. A KF is different from the classical filter in that it may have multiple inputs and multiple outputs with possibly nonstationary and time-varying characteristics performing optimum states estimation based on the unbiased minimum variance estimation criterion.

In the following discussions, we first introduce the basic concepts of KF, followed by various algorithmic variations of KF. Each version has different algorithmic and hardware complexity and implementational implications. Because there are myriad of KF variations, we then consider two simple systolic versions of KF.

9.5.5.1 Basic KF

The KF model consists of a discrete-time linear dynamical system equation and a measurement equation. A linear discrete-time dynamical system with $n \times 1$ state vector $x(k+1)$, at time $k+1$, is given by $x(k+1) = \mathrm{A}(k)x(k) + B(k)u(k) + w(k)$, where $x(k)$ is the $n \times 1$ state vector at time k, $A(k)$ is an $n \times n$ system coefficient matrix, $B(k)$ is an $n \times p$ control matrix, $u(k)$ is a $p \times 1$ deterministic vector, which for some problems may be zero for all k, and $w(k)$ is an $n \times 1$ zero-mean system noise vector with a covariance matrix $W(k)$. The input to the KF is the $m \times 1$ measurement (also called observation) vector $y(k)$, modeled by $y(k) = C(k)x(k) + v(k)$, where $C(k)$ is an $m \times n$ measurement coefficient matrix, and $v(k)$ is a $m \times 1$ zero-mean measurement noise vector with an $m \times m$ positive-definite covariance matrix $V(k)$. The requirement of the positive-definite condition on $V(k)$ is to guarantee the Cholesky (square root) factorization of $V(k)$ for certain KF algorithms. In general, we will have $m \leq n$ (i.e., the measurement vector dimension is less than or equal to that of the state vector dimension). It is also assumed that $w(k)$ is uncorrelated to $v(k)$. That is, $E\{w(i)v(j)^{\mathrm{T}}\} = 0$. We also assume each noise sequence is white in the sense $E\{w(i)w(j)^{\mathrm{T}}\} = E\{v(i)v(j)^{\mathrm{T}}\} = 0$, for all $i \neq j$.

The KF provides a recursive linear estimation of $x(k)$ under the minimum variance criterion based on the observation of the measurement $y(k)$. Let $\hat{x}(k)$ denote the optimum filter state estimate of $x(k)$ given

measurements up to and including $y(k)$, while $\hat{x}_+(k)$ denotes the optimum predicted state estimate of $x(k)$ given measurement up to and including $y(k-1)$. Then the $n \times n$ "optimum estimation error covariance matrix" is given by $P(k) = E\{(x(k) - \hat{x}(k))(x(k) - \hat{x}(k))^{\mathrm{T}}\}$, while the "minimum estimation error variance" is given by $J(k) = \text{Trace } P(k) = E\{(x(k) - \hat{x}(k))^{\mathrm{T}} (x(k) - \hat{x}(k))\}$. The $n \times n$ "optimum prediction error covariance matrix" is given by $P_+(k) = E\{(x(k) - x_+(k))(x(k) - x_+(k))^{\mathrm{T}}\}$.

The original KF recursively updates the optimum error covariance and the optimum state estimate vector by using two sets of update equations. Thus, it is often called the "covariance KF." The "time update equations" for $k = 1, 2, \ldots$, are given by $x_+(k) = A(k-1)\ \hat{x}(k-1) + B(k-1)u(k-1)$ and $P_+(k) = A(k-1)\ P(k-1)A^{\mathrm{T}}(k-1) + W(k-1)$. The $n \times n$ "Kalman gain matrix" $K(k)$ is given by $K(k) = P_+(k)C^{\mathrm{T}}(k)[C(k)P_+(k)C^{\mathrm{T}}(k) + V(k)]^{-1}$. The "measurement update equations" are given by $\hat{x}(k) = x_+(k) + K(k)(y(k) - C(k)x_+(k))$ and $P(k) = P_+(k) - K(k)P_+(k)$. The first equation shows the update relationship of $\hat{x}(k)$ to the predicted state estimate $x_+(k)$, for $x(k)$ based on $\{\ldots, y(k-2), y(k-2), y(k-1)\}$, when the latest observed value $y(k)$ is available. The second equation shows the update relationship between $P(k)$ and $P_+(k)$. Both equations depend on the $K(k)$, which depends on the measurement coefficient matrix $C(k)$and the statistical property of the measurement noise, covariance matrix $V(k)$. Furthermore, $K(k)$ involves an $m \times m$ matrix inversion.

9.5.5.2 Other Forms of KF

The basic KF algorithm considered above is called the covariance form of KF because the algorithm propagates the prediction and estimation error covariance matrices $P_+(k)$ and $P(k)$. Many versions of the KF are possible, characterized partially by the nature of the propagation of these matrices. Ideally, under infinite precision computations, no difference in results is observed among different versions of the KF. However, the computational complexity and the systolic implementation of different versions of the KF are certainly different. Under finite precision computations, especially for small numbers of bits under fixed point arithmetics, the differences among different versions can be significant. In the following discussions we may omit the deterministic control vector $u(k)$ because it is usually not needed in many problems. In the following chol (.) qr (.), and triu (.) stand for Cholesky factor, QRD, and triangular factor, respectively.

1. **Information filter**. The inverse of the estimation error covariance matrix $P(k)$ is called the information matrix and is denoted by $PI(k)$. A KF can be obtained by propagating the information matrix and other relevant terms. Specifically, the information filter algorithm is given by time updates for $k = 1, 2, \ldots,$ of $L(k) = A^{-\mathrm{T}}(k-1)PI(k-1)A^{-1}(k-1) \times [W^{-1}(k-1) + A^{-\mathrm{T}}(k-1)\ PI(k-1)A^{-\mathrm{T}}(k-1)]^{-1}$, $d_+(k) = (I - L(k))A^{-\mathrm{T}}(k-1)PI(k-1)A^{-1}(k-1)$. The measurements updates are given by $d(k) = d_+(k) + C^{\mathrm{T}}(k)V^{-1}(k)y(k)PI(k) = PI_+(k) + C^{\mathrm{T}}(k)V^{-1}(k)C(k)$.
2. **Square-root covariance filter (SRCF)**. In this form of the KF, we propagate the square root of $P(k)$. In this manner, we need to use a lower dynamic range in the computations and obtain a more stable solution under finite precision computations. We assume all three relevant covariance matrices are positive-definite and have the factorized form of $P(k) = S^{\mathrm{T}}(k)S(k)$, $W(k) = S_W^T(k)S_W(k)$, $V(k) = S^{\mathrm{T}}(k)\ S_v(k)$. In particular, $S(k) = \text{chol}(P(k))$, $S_W(k) = \text{chol}(W(k))$, $S_V(k) = \text{chol}(V(k))$, are the upper triangular Cholesky factorizations of $P(k)$, $W(k)$, and $V(k)$, respectively. The time updates for $k = 1, 2, \ldots,$ are given by $x_+(k) = A(k-1)\hat{x}(k-1)$, $U(k) = \text{triu}(\text{qr}([S(k-1)A^{\mathrm{T}}(k-1); S_W(k-1)]))$, $P_{+s}(k) = U(k)(1{:}n;\ 1{:}n)$. The measurement updates are given by $P_+(k) = P_{+s}^{\mathrm{T}}(k)P_{+s}(k)$,/,$K(k) = P_+(k)C^+(k)[C(k)P_+(k)C^+(k) + V(k)]^{-1}$, $\hat{x}(k) + K(k)(y(k) - C(k)x_+(k))$, $Z(k) = \text{triu}(qr([S_v(k),\ 0_{mn}; P_{+s}(k)C^+(k), P_{+s}(k)]))$ and $S(k) = \text{Z}(k)(m+1{:}\ m+n, m+1{:}\ m+n)$.
3. **Square-root information filter (SRIF)**. In the SRIF form of the KF, we propagate the square root of the information matrix. Just as in the SRCF approach, as compared to the conventional covariance form of the KF, the SRIF approach, as compared to the SRIF approach, needs to use a lower dynamic range in the computations and obtain a more stable solution under finite precision computations. First, we denote $SI(k) = (\text{chol}(P(k)))^{-1}$, $SI_W(k) = (\text{chol}(W(k)))^{-1}$, and

$SI_V(k) = (\text{chol}(V(k)))^{-1}$. The time updates for $k = 1, 2, \ldots$, are given by $U(k) = \text{triu}(\text{qr}([SI_W(k-1), 0_{n\times n}, 0_{n\times 1}; SI(k-1)A^{-1}(k-1), SI(k-1)A^{-1}(k-1), b(k-1)]))$, $P_{+S}(K) = U(k)(n+1{:}2n, n+1{:}2n)$ and $b_+(k) = U(k)(n+1{:}\ 2n,\ 2n+1)$. The measurement updates are given by $Z(k) = \text{triu}(\text{qr}([P_{+S}(k), b_+(k); SI_V(k)C(k), SI_V(k)y(k)]))SI(k) = Z(k)(1{:}n, 1{:}n)$, and $b(k) = Z(k)(1{:}\ n, n+1)$. At any iteration, $\hat{x}(k)$ and $P(k)$ are related to $b(k)$ and $SI(k)$ by $\hat{x}(k) = SI(k)b(k)$ and $P(k) = (SI^{\text{T}}(k)SI(k))^{-1}$.

9.5.5.3 Systolic Matrix Implementation of the KF Predictor

The covariance KF for the optimum state estimate $\hat{x}(k)$ includes the KF predictor $x_+(k)$ In particular, if we are only interested in $x_+(k)$ a relatively simple algorithm for $k = 1, 2, \ldots$, is given by $K(k) = P_+(k)C^{\text{T}}(k)[C(k)P_+(k)C^{\text{T}}(k) + V(k)]^{-1}$, $x_+(k+1) = A(k)\ x_+(k) + A(k)\ K(k)\ [y(k) - C(k)x_+(k)]$ and $P_+(k+1) = A(k)P_+(k)A^{\text{T}}(k) - A(k)K(k)C(k)P_+(k)A^{\text{T}}(k) + W(k)$. To start this KF prediction algorithm, we use $\hat{x}(0)$ and $P(0)$ to obtain $x_+(1) = A(0)\hat{x}(0)$ and $P_+(1) = A(0)P(0)A^{\text{T}}(0) + W(0)$. The above operations involve matrix inverse; matrix–matrix and matrix–vector multiplications; and matrix and vector additions. Fortunately, the matrix inversion of $\alpha = C(k)P_+(k)C^{\text{T}}(k) + V(k)$ can be approximated by the iteration of $\beta(i+1) = \beta(i)[2I - \alpha\ \beta(i)]$, $i = 1, \ldots, I$. Here, $\beta(i)$ is the ith iteration estimate of the inverse of the matrix α. While the preceding equation is not valid for arbitrary α and $\beta(i)$, for KF applications, we can use $I = 4$ because a good initial estimate $\beta(1)$ of the desired inverse is available from the previous step in the KF. Clearly, with the use of the above equation for the matrix inversion, all the operations needed in the KF predictor can be implemented on an orthogonal array using systolic matrix operations of the form $D = B \times A + C$, as shown in Figure 9.113.

The recursive algorithm of the KF predictor is decomposed as a sequence of matrix multiplications, as shown in Table 9.14. In step 1 the $n \times n$ matrix $P_+(k)$ and the $m \times n$ matrix $C^{\text{T}}(k)$ are denoted as B and A, respectively. The rows of B (starting from the n, $n-1, \ldots, 1$ row) are skewed and inputted to the $n \times n$ array starting at time 1. By time n (as shown in Figure 9.113), all the elements of the first column of B (i.e., $b_{n1}, \ldots, b_{11}$) are in the first column of the array. At time $n+1, \ldots, 2n-1$, elements of the second to nth columns of B are inputted to the array and remain there until the completion of the BA matrix multiplication. At time $n+1$, a_{11} enters $(1,1)$ cell and starts the BA process. At time $n+m$, a_{1m} enters the $(1,1)$ cell. Of course, additional times are needed for other elements in the second to the nth rows of A to enter the array. Further processing and propagation times are needed before all the elements of $D = BA = P_+(k)C^{\text{T}}(k)$ are outputted. However, in step 2, because B remains as $P_+(k)$, we do not need to input it again, but only append $A(k)$ (denote as $\tilde{A}$ in Figure 9.113) in the usual skewed manner after the previous $A = C^{\text{T}}(k)$. Thus, at time $n+m+1$, $\tilde{a}_{11}$ enters the $(1, 1)$ cell. By time $n+m+n$, $\tilde{a}_{1n}$ enters the $(1, 1)$ cell. Thus, step 1 takes $n+m$ time units, while step 2 takes only n time units. In step 3 m time units are needed to load $C(k)$ and m time units are needed to input $P_+(k)C^{\text{T}}(k)$, resulting in $2m$ time units. Steps 4 and 5 perform one iteration of the inverse approximation. In general, $I = 4$ iterations is adequate, and $16m$ time units are needed. Thus far, all the matrices and vectors are fed continuously into the array with no delay. However, in order to initiate step 13, the $(n, 1)$ component of $A(k) - A(k)K(k)C(k)$ is needed, but not available. Thus, at the end of step 11, an additional $(n-3)$ time units of delay must be provided to access this component. From Table 9.14, a total of $9n + 22m$ time units is needed to perform one complete KF prediction iteration.

9.5.5.4 Systolic KF Based on the Faddeev Algorithm

A form of KF based on mixed prediction error covariance $P_+(k)$ and information matrix $PI\ (k) = P^{-1}(k)$ updates can be obtained from the covariance KF algorithm. For $k = 1, 2, \ldots$, we have $x_+(k) = A(k-1)\hat{x}(k-1) + B(k-1)$, $P_+(k) = A(k-1)PI^{-1}(k-1)A^{\text{T}}(k-1) + W(k-1)$, $PI(k) = P_+^{-1}(k) + C^{\text{T}}(k)V^{-1}(k)C(k)K(k) = PI^{-1}(k)C^{\text{T}}(k)V^{-1}(k)$ and $\hat{x}(k) = x_+(k) + K(k)(y(k) - C(k)x_+(k))$. The algorithm starts with the given $\hat{x}(0)$ and $P(0)$, as usual. Because this algorithm requires the repeated use of matrix inversions for $(PI(k-1))$, $(P_+(k))^{-1}$, $(V(k))^{-1}$ as well as $P(k) = (PI(k))^{-1}$, the following "Faddeev algorithm" is suited for this approach.

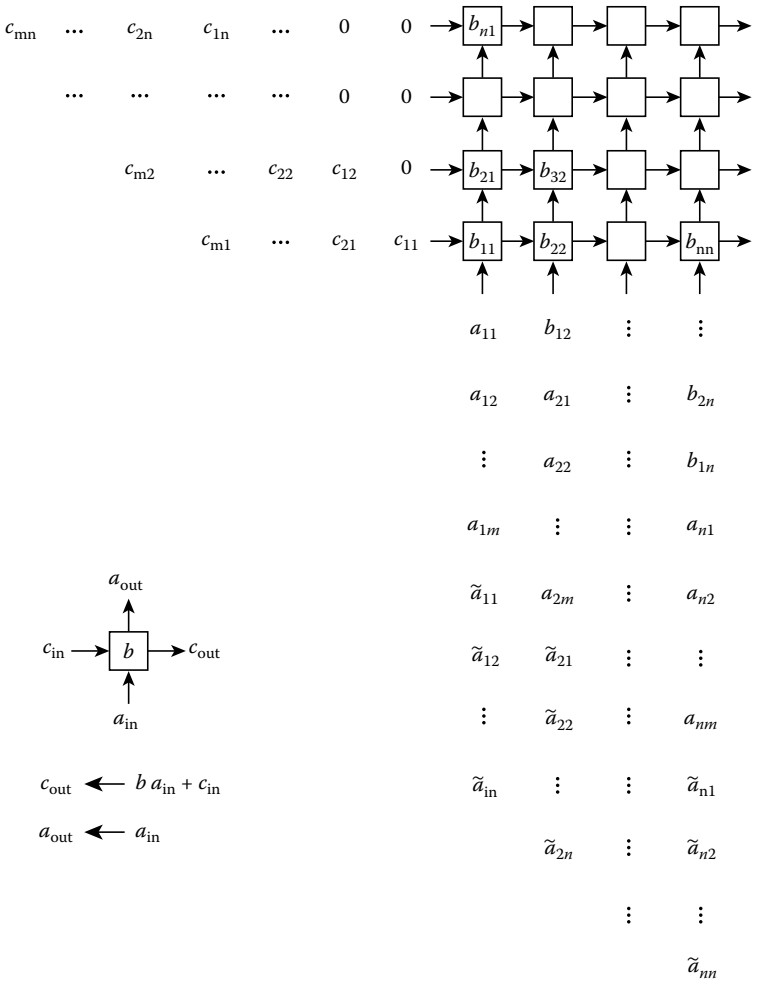


FIGURE 9.113 Systolic matrix multiplication and addition of $B \times A + C$.

TABLE 9.14 Systolic Matrix Operations of a KF Predictor

Step	B	A	C	D	Time
1	$P_+(k)$	$C^T(k)$	0	$P_+(k)C^T(k)$	$n+m$
2	$P_+(k)$	$A^T(k)$	0	$P_+(k)A^T(k)$	N
3	$C(k)$	$P_+(k)\ C^T(k)$	$V(k)$	$C(k)P_+(k)\ C^T(k)+V(k)=\alpha$	$2m$
4	α	$-\beta(i)$	$2I$	$2I-\alpha\beta(i)$	$2Im$
5	$\beta(i)$	$2I-\alpha\beta(i)$	0	$\beta(i+1)$	$2Im$
6	$P_+(k)\ C^{\mathrm{T}}(k)$	β	0	$K(k)$	$n+m$
7	$A(k)$	$K(k)$	0	$A(k)K(k)$	$n+m$
8	$A(k)$	$x_+(k)$	0	$A(k)x_+(k)$	1
9	$-C(k)$	$x_+(k)$	$y(k)$	$y(k)-C(k)x_+(k)$	$m+1$
10	$A(k)K)(k)$	$-C(k)$	$A(k)$	$A(k)-A(k)K(k)C(k)$	$2n$
11	$A(k)K(k)$	$y(k)-C(k)x_+(k)$	$A(k)x_+(k)$	$x_+(k+1)$	1
12	$n-3$				
13	$A(k)-A(k)K(k)C(k)$	$P_+(k)A^T(k)$	$W(k)$	$P_+(k)$	$2n$

Consider an $n \times n$ matrix A, an $n \times m$ matrix B, a $p \times n$ matrix C, and a $p \times m$ matrix D arranged in the form of a compound matrix $[A\ B; -C\ D]$. Consider a $p \times n$ matrix W multiplying $[A\ B]$ and added to $[-C\ D]$, resulting in $[A\ B; -C + WAD + WB]$. Assume W is chosen such that $-C + WA = 0$, or $W = CA^{-1}$. Then, we set $D + WB = D + CA^{-1}B$.

In particular, by picking $\{A, B, C, D\}$ appropriately, the basic matrix operations needed above can be obtained using the Faddeev algorithm. Some examples are given by

$$\begin{matrix} A & I \\ -I & 0 \end{matrix} \Rightarrow D + W\ B\ = A^{-1}$$

$$\begin{matrix} I & B \\ -C & 0 \end{matrix} \Rightarrow D + W\ B\ = C\ B$$

$$\begin{matrix} I & B \\ -C & D \end{matrix} \Rightarrow D + W\ B\ = D + C\ B$$

$$\begin{matrix} A & B \\ -I & 0 \end{matrix} \Rightarrow D + W\ B\ = A^{-1}B$$

A modified form of the previous Faddeev algorithm first triangularizes A with an orthogonal transformation Q, which is more desirable from the finite precision point of view. Then, the nullification of the lower left portion can be performed easily using the Gaussian elimination procedure. Specifically, applying a QRD, $Q[A\ B] = [R\ QB]$. Then, applying the appropriate W yields

$$\begin{bmatrix} R & QB \\ -C + W\ Q\ A & D + W\ Q\ B \end{bmatrix} = \begin{bmatrix} R & Q\ B \\ 0 & D + C\ A^{-1}B \end{bmatrix} \tag{9.24}$$

The preceding mixed prediction error covariance and information matrix KF algorithm can be reformulated as a sequence of Faddeev algorithm operations, as given in Table 9.15. The times needed to

TABLE 9.15 Faddeev Algorithm Solution to KF

Step	Compound Matrix		$D + WB$	Time
1	I $-A(k-1)$	$\hat{x}(k-1)$ $B(k-1)u(k-1)$	$x_+(k)$	$n+1$
2	$P^{-1}(k-1)$ $-A(k-1)$	$A^{\mathrm{T}}(k-1)$ $W(k-1)$	$P_+(k)$	$2n$
3	$V(k-1)$ $-C^{\mathrm{T}}(k)$	I 0	$C^{\mathrm{T}}(k)V^{-1}(k-1)$	$m+n$
4	$P_+(k)$ $-I$	I 0	$p_+^{-1}(k)$	$2n$
5	I $-C^{\mathrm{T}}(k)V^{-1}(k)$	$C(k)$ $P_+^{-1}(k)$	$P^{-1}(k)$	N
6	$P^{-1}(k)$ $-I$	$C^{\mathrm{T}(k+1)V^{-1}(k)}$ 0	$K(k)$	$2n$
7	I $C(k)$	$x_+(k)$ $y(k)$	$y(k) - C(k)_{x_+}(k)$	$m+1$
8	I $-K(k)$	$y(k) - C(k)x_+(k)$ $x_+(k)$	$\hat{x}(k)$	$m+1$

perform steps 2, 3, 4, and 6 are clearly just the sum of the lengths of the two matrices in the corresponding steps. Step 1 requires only n times unit to input the second row of matrices because $\hat{x}(k-1)$ is already located in the array from the previous iteration (step 8 output) and one time unit to output $x_+(k)$. Due to the form of $[-I\ 0]$ in step 4, $C(k)$ of step 5 can be inputted before the completion of $P_+^{-1}(k)$ in step 4. Thus, only n time units are needed in step 5. Similarly, $x_+(k)$ of step 7 can be inputted in step 6. Thus, we need only $m+1$ time units to input $[C(k)\ y(k)]$ and complete its operations. In step 8 only $m+1$ time units are needed as in step 1. Thus, a total of $9n+3m+3$ time units are needed for the Faddeev algorithm approach to the KF.

9.5.5.5 Other Forms of Systolic KF and Conclusions

While the operations of a KF can be expressed in many ways, only some of these algorithms are suited for systolic array implementations. For a KF problem with a state vector of dimension n and a measurement vector of dimension m, we have shown the systolic matrix–matrix multiplication implementation of the predictor form of the KF needs $9n+22m$ time steps for each iteration. A form of KF based on mixed update of prediction error covariance and information matrices is developed based on the Faddeev algorithm using matrix–matrix systolic array implementation. It has a total of $9n+3m+3$ time steps per iteration. A modified form of the SRIF algorithm can be implemented as a systolic array consisting of an upper rectangular array of $n(n+1)/2$ internal cells, and a lower n-dimensional triangular array of n boundary cells, and $(n-1)^2/2$ internal cells, plus a row of n internal cells, and $(n-1)$ delay cells. It has a total of n-boundary cells, $((n-1)^2+2n^2+2n)/2$ internal cells, and $(n-1)$ delay cells. Its throughput rate is $3n$ time steps per iteration. A modified form of the SRCF algorithm utilizing the Faddeev algorithm results in a modified SRCF form of a KF consisting of a trapezodial section, a linear section, and a triangular section systolic array. The total of these three sections needs $(n+m)$ boundary cells, n linear cells, and $((m-1)^2+2nm+(n-1)^2)/2$ internal cells. Its throughput rate is $3n+m+1$ time steps per iteration. The operations of both of these systolic KF are quite involved and detailed discussions are omitted here. In practice, in order to compare different systolic KFs, one needs to concern oneself not only with the hardware complexity and the throughput rate, but other factors involving the number of bits needed finite precision computations, data movement in the array, and I–O requirements as well.

9.5.6 Eigenvalue and SVDs

Results from linear algebra and matrix analysis have led to many powerful techniques for the solution of wide range of practical engineering and signal processing problems. Although known for many years, these mathematical tools have been considered too computationally demanding to be of any practical use, especially when the speed of calculation is an issue. Due to the lack of computational power, engineers had to content themselves with suboptimal methodologies of simpler implementation. Only recently, due to the advent of parallel/systolic computing algorithms, architectures, and technologies, have engineers employed these more sophisticated mathematical techniques. Among these techniques are the so-called eigenvalue decomposition (EVD) and the SVD. As an application of these methods, we consider the important problem of spatial filtering.

9.5.6.1 Motivation–Spatial Filtering Problem

Consider a linear array consisting of L sensors uniformly spaced with an adjacent distance d. A number M, $M<L$, of narrowband signals of center frequency f_0, impinging on the array. These signals arrive from M different spatial direction angles $\theta_1,\ldots,\theta_M$, relative to some reference direction. Each sensor is provided with a variable weight. The weighted sensor outputs are then collected and summed. The goal is to compute the set of weights to enhance the estimation of the desired signals arriving from directions $\theta_1,\ldots,\theta_M$.

In one class of beamformation problems, one sensor (sometimes referred to as main sensor) receives the desired signal perturbed by interference and noise. The remaining $L-1$ sensors (auxiliary sensors)

are mounted and aimed in such a way as to collect only the (uncorrelated) interference and noise components. In this scenario the main sensor gain is to be kept at a fixed value, while the auxiliary weights are adjusted in such a way as to cancel out as much perturbation as possible. Obviously, the only difference of this latter cast is that one of the weights (the one corresponding to the main sensor) is kept at a constant value of unity.

Let the output of the ith sensor, $i = 1, \ldots, L$, at discrete time $n = 0, 1, \ldots$, be given by

$$\bar{x}_i(n) = \Re\left\{ [x_i(n) + v_i(n)] e^{j(2\pi f_0 n)} \right\} \quad x_i(n) = a_i(n) \sum_{k=1}^{M} S_k(n) e^{j2\pi(i-1)d \sin\theta_k/\lambda}$$

where

$a_i(n)$ is the antenna gain at time n
S_k is the complex amplitude of the kth signal, inclusive of the initial phase
λ is the signal wavelength

The vectors $x_i(n)$ and $v_i(n)$ are analytic signal representations. The noise $v_i(n)$ is assumed to be uncorrelated white and Gaussian, of power σ_N^2. In order to avoid the ill effects of spatial aliasing, let us also assume that $d \leq \lambda/2$. The outputs of the sensor array for times $n = 0, 1, \ldots, N$, can be collected in matrix form as follows:

$$\underbrace{X}_{N\times L} = \underbrace{S}_{N\times M} = \underbrace{A}_{M\times L} + \underbrace{V}_{N\times L}$$

The matrix A is referred to as the "steering matrix." In the case in which $a_i(n) = 1$ for all i, the matrix A is Vandermonde and full rank, and its kth row can be expressed as $A(\theta) = (1, e^{j2\pi d \sin\theta/\lambda}, \ldots, e^{j2\pi(L-I)d \sin\theta_k/\lambda})$.

The data correlation matrix, $R_X = E\{X^H X\}$, where $E(\cdot)$ is the ensemble average operator, is equal to $R_X = A^H R_s A + \sigma_N^2 I, R_s = \mathrm{E}\{S^H S\}$. We note:

1. The matrix R_s has rank M by definition as does the matrix $A^H R_s A$.
2. The rows of A are in the range space of R_X.
3. The value σ_N^2 is an eigenvalue of R_X with multiplicity $L - M$, given the det $(R_X - \sigma_N^2 I) = 0$, and the rank of $A^H R_s A$ is M.

The EVD of R_X can therefore be written as $R_X = V_S \Lambda_S V_S^H + \sigma_N^2 V_N V_N^H$, where V_S is $L \times M$, V_N is $L \times (L - M)$, $V_S^H V_S = I$ and $V_N^H V_N = 1$, and $V_S^H V_N = 0$. Moreover, we have that $A V_N = 0$. Let $A(\theta)$ be a generic steering vector, defined as $A(\theta) + (1, e^{j2\pi d \sin\theta/\lambda}, \ldots, e^{j2\pi(L-I)d \sin\theta/\lambda})$. Then, the function $\pm(\theta) = 1/|A(\theta) V_N|^2$ has M poles at the angles $\theta = \theta_k k = 1, \ldots, M$. Alternatively, any linear combination, w, of the columns of V_N is such that $E\{\|Xw\|_2\} = \min_z E\{\|Xz\|_2\} = \sigma_N$. In other words, the signals impinging from angular directions $\theta_1, \ldots, \theta_M$ are totally canceled out in the system output.

The desired weighting vector for our spatial filtering problem can consequently be expressed as $w = V_N p$, $p = [p_1, \ldots, p_{L-M}]^{\mathrm{T}}$, for any nonzero vector p. From the above discussion, we see that the solution to the original spatial filtering problem can be obtained from the EVD of the correlation matrix $R_X = E\{X^H X\}$. In practice the sample correlation matrix $\hat{R}_X$ is used instead, where the ensemble average is replaced by a suitable temporal average.

The computation of the covariance matrix implies the computation of the matrix product $X^H X$. Some small elements in X are then squared and the magnitude of the resulting element can become comparable or smaller than the machine precision. Rounding errors can often impair and severely degrade the computed solution. In these cases it is better to calculate the desired quantities (correlation eigenvalues and eigenvectors) directly from the data matrix X using the SVD technique as considered next.

9.5.6.2 EVD of a Symmetric Matrix

Consider an $L \times L$ real symmetric matrix $A = A^{\mathrm{T}}$. In the previous spatial filtering example, $A = R_X$. Let

$$G(i,j,\theta) = \begin{array}{c} \\ i \\ \\ \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} i & & j & & \end{array} \\ \begin{pmatrix} I_{i-1} & & & & \\ & c & & s & \\ & & I_{j-i-1} & & \\ & -s & & c & \\ & & & & I_{L-j} \end{pmatrix} \end{array}$$

be an "orthogonal Givens rotation matrix," where $c = \cos\theta$ and $s = \sin\theta$. Pre- or postmultiplication of A by G leaves A unchanged, except for rows (columns) i and j, which are replaced by a linear combination of old rows (columns) i and j. A "Jacobi rotation" is obtained by simultaneous pre- and postmultiplication of a matrix by a Givens rotation matrix, as given by $G(i, j, \theta)^{\mathrm{T}} A G(i, j, \theta)$, where θ is usually chosen in order to zero out the (i, j) and (j, i) entries of A.

The matrix A can be driven toward diagonal form by iteratively applying Jacobi rotations, as given by $A_0 \leftarrow A$, $A_{k+1} \leftarrow G_k^T A G_k$, where G_k is a Givens rotation matrix. A "sweep" is obtained by applying $L(L-1)/2$ Jacobi rotations, each nullifying a different pair of off-diagonal elements, according to a prespecified order. Given the matrix $A_k = (a_{pq}^{(k)})$ at the kth iteration, and a pair of indices (i, j), the value of $\tan\theta$ can be obtained from the following equations:

$$u = \frac{a_{jj}^{(k)} - a_{ii}^{(k)}}{2a_{ij}^{(k)}}, \quad \tan\theta = \frac{sign(u)}{|u| + \sqrt{1+u^2}} \tag{9.25}$$

It is possible to demonstrate that each Jacobi rotation reduces the matrix off-norm. The matrix A_k indeed tends to diagonal form and for all practical purposes it reaches it after $\mathbb{O}(\log L)$ sweeps.

The matrix V of eigenvectors is obtained by applying the same rotations to a matrix initialized to the identity, as follows: $V_0 \leftarrow I$, $V_{k+1} \leftarrow V_k G_k$. A two-dimensional systolic array implementation of the previous algorithm is shown in Figure 9.114, for the case $L = 8$. At the beginning of iteration k, processor P_{ij} contains elements

$$\begin{pmatrix} a_{2i-1,2j-1}^{(k)} & a_{2i-1,2j}^{(k)} \\ a_{2i,2j-1}^{(k)} & a_{2i,2j}^{(k)} \end{pmatrix} \quad i,j = 1,2,\ldots,L/2,$$

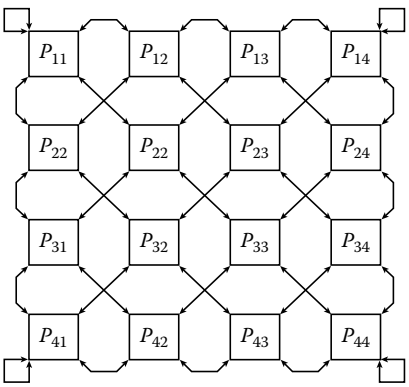


FIGURE 9.114 Systolic array for an EVD of a symmetric matrix based on Jacobi rotations.

The diagonal processors compute the rotation parameters and apply the rotation to the four entries they store. Subsequently, they propagate the rotation parameters horizontally and vertically to their neighbors, which, upon receiving them, apply the corresponding rotation to their stored entries. After the rotation is applied, each processor swaps its entries with its four neighbors along the diagonal connections. The correct data movement at the edges of the array is also shown in Figure 9.114. A correct scheduling of operations requires that each processor be idle for two out of three time steps, which translates into an "efficiency" of 33%. Each sweep takes 3 $(L-1)$ time steps, and the number of sweeps can be chosen on the order of $\log L$.

9.5.6.3 SVD of a Rectangular Matrix via the Hestenes Algorithm

Consider an $N \times L$ real matrix A, $N \geq L$. Its SVD can be written as follows:

$$\underbrace{A}_{N\times L} = \underbrace{U}_{N\times L}\,\underbrace{\Sigma}_{L\times L}\,\underbrace{V^{\mathrm{T}}}_{L\times L}$$

where U and V have orthonormal columns. The matrix $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_L)$ is the diagonal matrix of singular values, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_L \geq 0$. Consider the following recursion $A_0 \leftarrow A$, $A_{k+1} \leftarrow A_k\, G_k$, where the Givens rotations are chosen not to zero out entries of A_k, but to orthogonalize pairs of its columns. A sweep is now defined as a sequence of Givens rotations that orthogonalize all $\binom{L}{2}$ pairs of columns exactly once. Observe the similarity with the algorithm described previously for the calculation of eigenvalues and eigenvectors. If $G(i, j, \theta)$ is the Givens rotation which orthogonalizes columns i and j of A, then $G(i, j, \theta)^{\mathrm{T}} M G(i, j, \theta)$ is the Jacobi rotation that zeroes out the entries (i, j) and (j, i) of $M = A^{\mathrm{T}}A$. A sweep (as defined here) of rotations applied to the rectangular matrix A corresponds exactly to a sweep (as defined earlier) of rotations applied to the symmetric matrix M.

At any time step, the original matrix A can be expressed as follows:

$$A = A_k V_k^T, \quad V_k = \prod_{i=1}^{k} G_k$$

where V_k has orthonormal columns for any k (by definition of Givens rotations). After a number of sweeps (on the order of $\log L$) the matrix A_k approaches a matrix, W, of orthogonal columns, $A_k \rightarrow W$, $V_k \rightarrow V$. If σ_i is the norm of the ith column of W, $i = 1, \ldots, L$, then we have $W = U\ \mathrm{diag}\ (\sigma_1, \ldots, \sigma_L)$, $A = U\Sigma V^{\mathrm{T}}$.

This SVD approach based on the Hestenes algorithm can be realized on a Brent–Luk [58] linear systolic array, as shown in Figure 9.115, for the case $L = 8$. Each processor stores a pair of columns; in particular, the procedure starts by storing columns 2 $k-1$ and $2k$ in processor P_k. Each processor computes the rotation parameters which orthogonalize the pair of columns.

Let x and z be the two stored columns. Let ξ and ζ be their norms, and $\acute{\eta}$ be their inner product. Then the value of $\tan\theta$ from

$$u = \frac{\zeta - \xi}{2\eta}, \quad \tan\theta = \frac{\mathrm{sign}(u)}{|u| + \sqrt{1 + u^2}}$$

After applying the computed rotation, each processor swaps its pair of columns with the two neighboring processors along the connections shown in Figure 9.115. The column indices stored in each processor at the different steps of a single sweep are given in Table 9.16. Note that all the $\binom{L}{2}$ pairs of indices are generated by using the Brent–Luk scheme. The stopping criteria can be set in advance. A possible criterion is by inspecting the magnitude of the rotating angles. When they are all in absolute value below

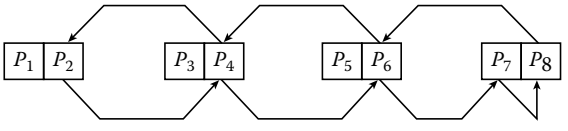


FIGURE 9.115 Linear systolic array for SVD of a rectangular matrix based on the Hestenes algorithm.

TABLE 9.16 Movement of Matrix Columns during One Sweep ($L = 8$)

P_1	P_2	P_3	P_4
(1, 2)	(3, 4)	(5, 6)	(7, 8)
(1, 4)	(2, 6)	(3, 8)	(5, 7)
(1, 6)	(4, 8)	(2, 7)	(3, 5)
(1, 8)	(6, 7)	(4, 5)	(2, 3)
(1, 7)	(8, 5)	(6, 3)	(4, 2)
(1, 5)	(7, 3)	(8, 2)	(6, 4)
(1, 3)	(5, 2)	(7, 4)	(8, 6)
(1, 2)	(3, 4)	(5, 6)	(7, 8)

a given threshold, then the algorithm can stop. More commonly, a predetermined number of sweeps is chosen ahead of time. Observation shows that a number of sweeps on the order of $\log L$ is sufficient for convergence.

9.5.6.4 SVD of a Rectangular Nonsymmetric Matrix

9.5.6.4.1 Via the Jacobi Algorithm

The SVD algorithm described in the previous section has the drawback of a somewhat complicated updating procedure. In many signal processing applications, continuous updating of the matrix decomposition as new samples are appended to the data matrix is required. Such problems occur in spectral analysis, direction-of-arrival estimation, beam forming, etc. An efficient updating procedure for the SVD of rectangular matrices of growing row size is given by the algorithm described in this section, based on the succession of two basic operations: a QR updating step, followed by a rediagonalization operation. This algorithm is otherwise known as a version of the Kogbetliantz algorithm for triangular matrices.

Given the $m \times L$ data matrix at time m, $A_m = [a_1, \ldots, a_m]^T$, where a_i, $i = 1, \ldots, m$ are the rows of A_m, one defines the exponentially weighted matrix $B_m(\beta)\, A_m$, where $B_m(\beta)$ is the diagonal forgetting matrix $B_m(\beta) + \text{diag}(\beta^{m-1}, \beta^{m-2}, \ldots, \beta, 1)$, and $0 < \beta \leq 1$ is the forgetting factor. The updating problem is to determine the SVD of the updated weighted matrix $B_{m+1}(\beta)\, A_{m+1}$, given the SVD at time m,

$$B_m(\beta) A_m = U_m \sum_m V_m$$

Often only the singular values and right singular vectors are of interest. This is fortunate because the left singular matrix grows in size as time increases, while the sizes of Σ_m and V_m remain unchanged.

The algorithm can be summarized as follows. Given the matrices V_m and Σ_m and the new data sample x_{m+1},

$$\sum\nolimits'_m \leftarrow \begin{pmatrix} \beta & \sum \\ & m \\ x_{m+1} & V_m \end{pmatrix} V'_m \leftarrow V_m,$$

the QR updating step

$$\begin{pmatrix} \sum'_m \\ 0 \end{pmatrix} \leftarrow Q_{m+1} \sum\nolimits'_m$$

the rediagonalization using permuted Jacobi rotations

for $k = 1, \ldots, 1$,
  for $i = 1, \ldots, n-1$, $j = i+1$,
    $\Sigma'_m \leftarrow \prod_{ij} G(i, j, \theta)\Sigma'_m G(i, j, \varphi)^{\mathrm{T}} \prod_{ij}$,
    $V'_m \leftarrow V'_m G(i, j, \varphi)^{T} \prod_{ij}$.
end
end
    $\Sigma_{m+1} \leftarrow \Sigma'_m$,
    $V_{m+1} \leftarrow V'_m$.

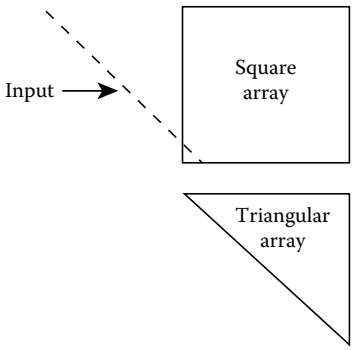


FIGURE 9.116 Two-dimensional systolic array for updating the Jacobi SVD algorithm for a nonsymmetric matrix.

In the preceding algorithm, the parameter l determines both the number of computations between subsequent updates and the estimation accuracy at the end of each update step. When l is chosen equal to the problem order L, then one complete sweep is performed. In practice, the l parameter can be chosen as high as $\sim 10L$ (usually for block computations or slow updates) and as small as 1 (for very high updating rates, with understandable degradation of estimation accuracy). The matrices Σ_m and Σ'_m are upper triangular at all times. This is ensured by the application of the permuted left and right Givens rotations in the rediagonalization step. After the application of any Jacobi rotation, the rotated rows and columns are subsequently permuted. This expedient not only preserves the upper triangularity of the Σ matrices, but also makes it possible for the rotations to be generated on the diagonal and propagated along physically adjacent pairs of rows and columns. All these features make this algorithm a very attractive candidate for a systolic implementation.

A schematic diagram of the systolic array proposed by Moonen et al. [63] is shown in Figure 9.116, where the triangular array stores the matrices Σ_m and Σ'_m, for all m, and the square array stores the V-matrix. The incoming data samples, x_{m+1}, are input into the V-array, where the vector-by-matrix multiplication $x_{m+1}\ V_m$ is performed. The output is subsequently fed into the triangular array. As it propagates through the array, the QR updating step is carried on: left rotations are generated in the diagonal elements of the array and propagated through the corresponding rows of the Σ matrix. One does not need to wait for the completion of the QR step to start performing the Jacobi rotations associated with the diagonalization step. It is known that the two operations (QR update and diagonalization) can be interleaved without compromising the final result. The parameters relative to the left rotations are, as before, propagated along the rows of the triangular matrix, while the right rotation parameters move along the columns of Σ_m, and are passed on to the V-array. Due to the continual modification of the matrix V_m caused by these right rotations, and because of the use of finite precision arithmetic, the computed right singular matrix may deviate from orthogonality. It is also known that in a realistic environment the norm of $V_m V_m^{\mathrm{T}} - I$ grows linearly with m. Reorthogonalization procedures must therefore be included in the overall scheme. A complicated reorthogonalization procedure based on left rotations, which interleaves with the other operations, was described in Moonen et al. [63]. An improved reorthogonalization algorithm was proposed by Vanpoucke and Moonen [66], where the matrix V_m is stored in parametrized form, thereby guaranteeing orthogonality at all times. The resulting triangular array and its modes of operation were also described.

References

1. A. Darte and J. M. Delosme, Partitioning for array processors, Tech. Rep. LIP-IMAG 90–23, Laboratoire de l'Informatique du Parallélisme, Ecole Supérieure de Lyon, Oct. 1990.
2. H. T. Kung and C. E. Leiserson, Systolic arrays (for VLSI), in *Introduction to VLSI Systems*, C. A. Meads and L. A. Conway, Eds. Reading, MA: Addison-Wesley, 1980, Chapter 8.3.

3. H. T. Kung, Why systolic architecture? *Computer*, 15, 37–46, 1982.
4. S. Y. Kung, K. S. Arun, R. J. Gal-Elzer, and D. V. B. Rao, Wavefront array processor: Language, architecture, and applications, *IEEE Trans. Comput*., 31, 1054–1066, 1982.
5. S. Y. Kung, *VLSI Array Processing*, Englewood Cliffs, NJ: Prentice Hall, 1988.
6. F. Lorenzelli, Systolic mapping with partitioning and computationally intensive algorithms for signal processing, PhD thesis, University of California, Los Angeles, 1993.
7. D. I. Moldovan, On the analysis and synthesis of VSLI algorithms, *IEEE Trans. Comput*., 31, 1121–1126, 1982.
8. M. Newman, *Integral Matrices*, New York: Academic Press, 1972.
9. P. Quinton, The systematic design of systolic arrays, IRISA Tech. Report 193, April 1983.
10. S. Rao, Regular iterative algorithms and their implementations on processor arrays, PhD thesis, Stanford, CA: Stanford University, 1985.
11. E. Angelidis and J. E. Diamessis, A novel method for designing FIR filters with nonuniform frequency samples, *IEEE Trans. Signal Process*., 42, 259–267, 1994.
12. S. Chanekar, S. Tantaratana, and L. E. Franks, Multiplier-free IIR filter realization using periodically time-variable state-space structures, I and II, *IEEE Trans*, *Signal Process*., 42, 1008–1027, 1994.
13. L. A. Ferrari and P. V Sankar, Minimum complexity FIR filters and sparse systolic arrays, *IEEE Trans. Comput*., 37, 760–764, 1988.
14. A. Jayadeva, A new systolic design for digital IIR filters, *IEEE Trans. Circuits Syst*., 37, 653–654, 1990.
15. S. C. Knowles, J. G. McWhirter, R. F. Woods, and J. V. McCanny, Bit-level systolic architectures for high performance IIR filtering, *J. VLSI Signal Process*., 1, 9–24, 1989.
16. S. M. Lei and K. Yao, A class of systolizable IIR digital filters and its design for proper scaling and minimum output roundoff noise, *IEEE Trans. Circuits Syst*., 37, 1217–1230, 1990.
17. S. M. Lei and K. Yao, Efficient systolic array implementation of IIR digital filtering, *IEEE Trans. Circuits Syst*., 39, 581–584, 1992.
18. H. H Loomis Jr. and B. Sinha, High-speed recursive digital filter, *Circuits*, *Syst. Signal Process*., 3, 267–294, 1984.
19. K. K. Parhi and D. G. Messerschmitt, Pipeline interleaving and parallelism in recursive digital filters. I. Pipelining using scattered look-ahead and decomposition, *IEEE Trans. Acoust. Speech Signal Process*, 37, 1099–1117, 1989.
20. N. R. Shanbhag and K. K. Parhi, *Pipelined Adaptive Digital Filters*, Boston, MA: Kluwer Academic, 1994.
21. R. F. Woods and J. V. McCanny, Design of high performance IIR digital filter chip, *IEEE Proc. E*, *Comput. Digital Tech*., 139, 195–202, 1992.
22. C. W. Wu and J.-C. Wang. Testable design of bit-level systolic block FIR filters, *Proc. IEEE Int. Symp. Circuits Syst*., San Diego, CA, pp. 1129–1132, 1992.
23. R. Wyrzykowski and S. Ovramenko, Flexible systolic architecture for VLSI FIR filters, *IEEE Proc. E*, *Comput. Digital Tech*., 139, 170–172, 1992.
24. L. W. Chang and J. H. Lin, A bit-level systolic array for median filter, *IEEE Trans. Signal Process*., 40, 2079–2083, 1992.
25. J. M. Delosme, Bit-level systolic array for real symmetric and Hermitian eigenvalue problems, *J. VLSI Signal Process*., 4, 69–88, 1992.
26. R. A. Evans, J. V. McCanny, J. G. McWhirter, A. Wood, and K. W. Wood, A CMOS implementation of a systolic multibit convolver chip, *Proc. VLSI*, 227–235, 1983.
27. G. Fettweis and H. Meyr, High-rate Viterbi processor: A systolic array solution, *IEEE J. Sel. Topics Commun*., 8, 1520–1534, 1990.
28. S. C. Knowles, J. G. McWhirter, R. F. Woods, and J. V. McCanny, Bit-level systolic architectures for high performance IIR filtering, *J. VLSI Signal Process*., 1, 9–24, 1989.
29. J. V. McCanny and J. G. McWhirter, On the implementation of signal processing function using one bit systolic arrays, *Elect. Lett*., 18, 241–243, 1982.

30. J. V. McCanny, J. G. McWhirter, and S. Y. Kung, The use of data dependence graphs in the design of bit-level systolic arrays, *IEEE Trans. Acoust., Speech Process.*, 38, 787–793, 1990.
31. J. V. McCanny, R. F. Woods, and M. Yan, Systolic arrays for high-performance digital signal processing, in *Digital Signal Processing: Principles, Device, and Applications*, N. B. Jones and J. D. M. Watson, Eds, New York: Peter Peregrinus, 1990, pp. 276–302.
32. C. L. Wang, An efficient and flexible bit-level systolic array for inner product computation, *J. Chin. Inst. Eng.*, 41, 567–576, 1991.
33. C. W. Wu, Bit-level pipelined 2-D digital filters image processing, *IEEE Trans. Circuits Syst. Video Technol.*, 1, 22–34, 1991.
34. M. G. Bellanger and P. A. Regalia, The FLS-QR algorithm for adaptive filtering: The case of multi-channel signal, *Signal Process*, 22, 115–126, 1991.
35. J. M. Cioffi, The fast adaptive ROTOR'S RLS algorithm, *IEEE Trans. Acoust. Speech Signal Process.*, 38, 631–653, 1990.
36. W. M. Gentleman and H. T. Kung, Matrix triangularization by systolic arrays, *Proc. SPIE, Real-Time Signal Process*, 298, 298–303, 1981.
37. S. Haykin, *Adaptive Filter Theory*, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1991.
38. F. Ling and J. G. Proakis, A recursive modified Gram-Schmidt algorithm with applications to least-squares estimation and adaptive filtering, *IEEE Trans. Acoust. Speech Signal Process*, 34, 829–836, 1986.
39. K. J. R. Liu, S. F. Hsieh, K. Yao, and C. T. Chiu, Dynamic range, stability, and fault tolerant capability of finite-precision RLS systolic array based on Givens rotations, *IEEE Trans. Circuits Sys.*, June, 625–636, 1991.
40. K. J. R. Liu, S. F. Hsieh, and K. Yao, Systolic block householder transformation for RLS alogrithm with two-level pipelined implementation, *IEEE Trans. Signal Process.*, 40, 946–958, 1992.
41. J. G. McWhirter, Recursive least-squares minimization using a systolic array, *Proc. SPIE, Real-Time Signal Process. VI*, 431, 105–112, 1983.
42. J. G. McWhirter, Algorithm engineering in adaptive signal processing, *IEEE Proc. F.*, 139, 226–232, 1992.
43. B. Yang and J. F. Böhme, Rotation-based RLS algorithms: Unified derivations, numerical properties, and parallel implementations, *IEEE Trans. Acoust. Speech Signal Process.*, 40, 1151–1167, 1992.
44. M. J. Chen and K. Yao, On realization of least-squares estimation and Kalman filtering by systolic arrays, in *Systolic Arrays*, W. Moore, A. McCabe, and R. Urquhart, Eds. Bristol, U.K.: Adam Hilger, 1986, pp. 161–170.
45. F. Gaston, G. Irwin, and J. McWhirter, Systolic square root covariance Kalman filtering, *J. VLSI Signal Process.*, 2, 37–49. 1990.
46. J. H. Graham and T. F. Kadela, Parallel algorithm architectures for optimal state estimation, *IEEE Trans. Comput.*, 34, 1061–1068, 1985.
47. R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.*, 82, 35–45, 1960.
48. P. G. Kaminiski, Discrete square root filtering: A survey of current techniques, *IEEE Trans. Autom. Control*, 16, 727–735, 1971.
49. S. Y. Kung and J. N. Hwang, Systolic array design for Kalman filtering, *IEEE Trans. Signal Process.*, 39, 171–182, 1991.
50. R. A. Lincoln and K. Yao, Efficient systolic Kalman filtering design by dependence graph mapping, in *VLSI Signal Processing III*, R. W. Brodersen and H. S. Moscovitz, Eds., New York: IEEE Press, 1988, pp. 396–407.
51. J. G. Nash and S. Hansen, Modified Faddeev algorithm for matrix manipulation, *Proc. SPIE*, 495, 39–46, 1984.
52. C. C. Paige and M. A. Saunders, Least squares estimation of discrete linear dynamic systems using orthogonal transformation, *SIAM J. Numer. Anal.*, 14, 180–193, 1977.

53. G. M. Papadourakis and F. J. Taylor, Implementation of Kalman filters using systolic arrays, *Proc. Int. Conf. Acoust*., *Speech and Signal Process*., Denver, CO, pp. 783–786, 1987.
54. P. Rao and M. A. Bayoumi, An algorithm specific VLSI parallel architecture for Kalman filter, in *VLSI Signal Processing IV*, H. S. Moscovitz, K. Yao, and R. Jain, Eds., New York: IEEE Press, 1991, pp. 264–273.
55. T. Y. Sung and Y. H. Hu, Parallel implementation of the Kalman filter, *IEEE Trans. Aero. Electr. Syst*., 23, 215–224, 1987.
56. H. Yeh, Systolic implementation of Kalman filters, *IEEE Trans. Acoust*., *Speech*, *Signal Process*., 36, 1514–1517, 1988.
57. E. Biglieri and K. Yao, Some properties of singular value decomposition and their application to signal processing, S*ignal Process*., 18, 277–289, 1989.
58. R. Brent and F. T. Luk, The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays, *SIAM J. Sci. Stat. Comput*., 6, 69–84, 1985.
59. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Baltimore, MD: Johns Hopkins University Press, 1989.
60. S. Haykin, *Adaptive Filter Theory*, 2nd ed., Englewood Cliffs, NJ: Prentice Hall, 1991.
61. M. R. Hestenes, Inversion of matrices by biorthogonalization and related results, *J. Soc. Ind. Appl. Math*., 6, 51–90, 1958.
62. F. T. Luk, A triangular processor array for computing singular values, *Linear Algebra Appl*., 77, 259–273, 1986.
63. M. Moonen, P. Van Dooren, and J. Vandewalle, A systolic array for SVD updating, *SIAM J. Matrix Anal. Appl*., 14, 353–371, 1993.
64. R. O. Schmidt, A signal subspace approach to multiple emitter location and spectral estimation,'' PhD thesis, Stanford, CA: Stanford University, 1981.
65. G. W. Stewart, A Jacobi-like algorithm for computing the Schur decomposition of a nonhermitian matrix, *SIAM J. Sci. Stat. Comput*., 6, 853–864, 1985.
66. F. Vanpoucke and M. Moonen, Numerically stable Jacobi array for parallel subspace tracking, *Proc. SPIE*, 2296, 403–412, 1994.
67. F. D. Van Veen and K. M. Buckley, Beamforming: A versatile approach to spatial filtering, *IEEE ASSP Mag*., 5, 4–24, 1988.

10

Data Converters

10.1 Digital-to-Analog Converters **10**-1
Introduction • Basic Converter Design Issues • Converter Architectures • Techniques for High-Resolution DACs • Sources of Conversion Errors • Low-Spurious DAC Design Examples
References **10**-28
10.2 Analog-to-Digital Converters **10**-29
Introduction • Nyquist Rate Converters • Oversampled Converters
Acknowledgment **10**-57
References **10**-57
Further Information **10**-58

Bang-Sup Song
University of California, San Diego

Ramesh Harjani
University of Minnesota

10.1 Digital-to-Analog Converters

Bang-Sup Song

10.1.1 Introduction

Digital-to-analog converters (DACs), referred to as decoders in communications terms, are devices by which digital processors communicate with the analog world. Although DACs are used as key elements in analog-to-digital converters (ADCs), they find numerous applications as stand-alone devices from CRT display systems and voice/music synthesizers to automatic test systems, waveform generators, digitally controlled attenuators, process control actuators, and digital transmitters in modern digital communications systems.

The basic function of the DAC is the conversion of input digital numbers into analog waveforms. An N-bit DAC provides a discrete analog output level, either voltage or current, for every level of 2^N digital words, $\{D_i;\ i = 0, 1, 2, \ldots, 2^N - 1\}$, that is applied to the input. Therefore, an ideal voltage DAC generates 2^N discrete analog output voltages for digital inputs varying from $000\ldots00$ to $111\ldots11$ as illustrated in Figure 10.1 for the 4 bit example. The output has a one-to-one correspondence with the input

$$V_{\text{out}}(D_i) = V_{\text{ref}}\left(\frac{b_N}{2} + \frac{b_{N-1}}{2^2} + \cdots + \frac{b_2}{2^{N-1}} + \frac{b_1}{2^N}\right) \tag{10.1}$$

where V_{ref} is a reference voltage setting the output range of the DAC and $b_N, b_{N-1}, \ldots, b_1$ is the binary representation of the input digital word D_i. In the unipolar case, as shown, the reference point is 0 when the digital input D_0 is $000\ldots00$, but in bipolar or differential DACs, the reference point is the midpoint of

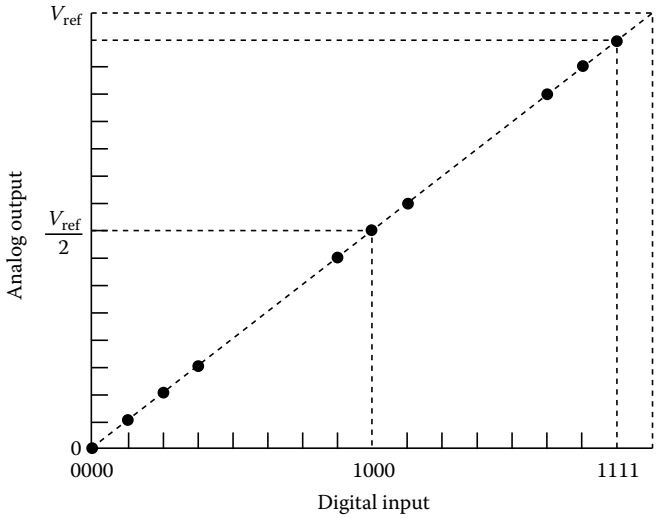


FIGURE 10.1 Transfer characteristics of a unipolar DAC.

the full scale when the digital input is 100 . . . 00, and the range is defined from $-V_{ref}/2$ to $V_{ref}/2$. Although purely current-output DACs are possible, voltage-output DACs are common in most applications.

10.1.1.1 Signal-to-Noise Ratio and Dynamic Range

The resolution is a term used to describe a minimum voltage or current that a DAC can resolve. The fundamental limit of a DAC is the quantization noise due to the finite resolution of the DAC. If the input digital word is N bits long, the minimum step that a DAC can resolve is $V_{ref}/2^N$. If output voltages are reproduced with this minimum step of uncertainty, an ideal DAC should have a maximum signal-to-noise ratio (SNR) of

$$\text{SNR} = \frac{3}{2}2^{2N} \approx 6N + 1.8 \text{ (dB)} \tag{10.2}$$

where SNR is defined as the power ratio of the maximum signal to the inband uncorrelated noise. For example, an ideal 16 bit DAC has an SNR of about 97.8 dB. The spectrum of the quantization noise is evenly distributed up to the Nyquist bandwidth (half the sampling frequency). Therefore, this inband quantization noise decreases by 3 dB when the oversampling ratio (OSR) is doubled. This implies that when oversampled, the SNR within the signal band can be made lower than the quantization noise limited by Equation 10.2.

The resolution of a DAC is usually characterized in terms of SNR, but the SNR accounts only for the uncorrelated noise. The real noise performance is better represented by TSNR, which is the ratio of the signal power to the total inband noise, including harmonic distortion (HD). Also, a slightly different term is often used in place of the SNR. The useful signal range or dynamic range is defined as the power ratio of the maximum signal to the minimum signal. The minimum signal is defined as the smallest input, for which the TSNR is 0 dB, while the maximum signal is the full-scale input. Therefore, the SNR of nonideal DACs can be lower than the ideal dynamic range because the noise floor can be higher with a large signal present. In practice, DACs are limited not only by the quantization noise, but also by nonideal factors such as noises from circuit components, power supply coupling, noisy substrate, timing jitter, insufficient settling, and nonlinearity.

10.1.2 Basic Converter Design Issues

The performance of a DAC can be specified in terms of its linearity, monotonicity, and conversion speed. In most conventional DACs, except for the oversampling DACs the linearity and monotonicity are limited by how accurately the reference voltage/current is divided using passive/active components.

10.1.2.1 Linearity

Differential nonlinearity (DNL). The output range of an N-bit DAC is equally divided into 2^N small units, as shown in Figure 10.1, and one least significant bit (LSB) change in the input digital word makes the analog output voltage change by $V_{\text{ref}}/2^N$. The DNL is a measure of the deviation of the actual DAC output voltage step from this ideal voltage step for 1 LSB. The DNL is defined as

$$\text{DNL} = \frac{V_{\text{out}}(D_{i+1}) - V_{\text{out}}(D_i) - V_{\text{ref}}/2^N}{V_{\text{ref}}/2^N} \quad \text{for } i = 0, 1, \ldots, 2^N - 2 \text{ (LSB)} \tag{10.3}$$

and the largest positive and negative numbers are usually quoted to specify the static performance of a DAC.

Integral nonlinearity (INL). The overall linearity of a DAC can be specified in terms of the INL, which is a measure of deviation of the actual DAC output voltage from the ideal straight line drawn between two endpoints, 0 and V_{ref}. Because the ideal output is $i \times V_{\text{ref}}/2^N$ for any digital input D_i, the INL is defined as

$$\text{INL} = \frac{V_{\text{out}}(D_i) - i \times V_{\text{ref}}/2^N}{V_{\text{ref}}/2^N} \quad \text{for } i = 0, 1, \ldots, 2^N - 1 \text{ (LSB)} \tag{10.4}$$

and the largest positive and negative numbers are usually quoted to specify the static performance of a DAC.

However, several definitions of INL may result depending on how two endpoints are defined. In some DAC architectures the two endpoints are not exactly 0 and V_{ref}. The nonideal reference point causes an offset error, while the nonideal full-scale range gives rise to a gain error. In most DAC applications, these offset and gain errors resulting from the nonideal endpoints do not matter, and the integral linearity can be better defined in a relative measure using a straight line linearity concept rather than the end point linearity in the absolute measure. The straight line can be defined as two endpoints of the actual DAC output voltages or as a theoretical straight line adjusted to best fit the actual DAC output characteristics. The former definition is sometimes called endpoint linearity, while the latter is called best straight line linearity.

10.1.2.2 Monotonicity

The DAC output should increase over its full range as the digital input word to the DAC increases. That is, the negative DNL should be <-1 LSB for a DAC to be monotonic. Monotonicity is critical in most applications, in particular, in digital control applications. The source of nonmonotonicity is an inaccuracy in binary weighting of a DAC. For example, the most significant bit (MSB) has a weight of one half of the full range. If the MSB weight is smaller than the ideal value, the analog output change can be smaller than the ideal step $V_{\text{ref}}/2^N$ when the input digital word changes from $0111\ldots11$ to $1000\ldots00$ at the midpoint of the DAC range. If this decrease in the output is >1 LSB, the DAC becomes nonmonotonic. The similar nonmonotonicity can take place when switching the second or lower MSB bits in binary-weighted multibit DACs.

Monotonicity is inherently guaranteed if an N-bit DAC is made of 2^N elements for thermometer decoding. However, it is impractical to implement high-resolution DACs using 2^N elements because the number of elements grows exponentially as N increases. Therefore, to guarantee monotonicity in practical applications, DACs have been implemented using either a segmented DAC or an

integrator-type DAC. Oversampling interpolative DACs also achieve monotonicity using a pulse-density modulated bitstream converted into analog voltages by a lossy integrator or by a low-pass filter.

Segmented DACs. Applying a two-step conversion concept, a DAC can be made in two levels using coarse and fine DACs. The fine DAC divides one coarse MSB segment into fine LSBs. If one fixed MSB segment is subdivided to generate LSBs, matching among MSB segments creates a nonmonotonicity problem. However, if the next MSB segment is subdivided instead of the fixed segment, the segmented DAC can maintain monotonicity regardless of the MSB matching. This is called ''next-segment approach.'' Unless the next segment approach is used to make a segmented DAC with a total $M+N$ bits, the MSB DAC should have a resolution of $M+N$ bits for monotonicity, while the LSB DAC requires an N-bit resolution. Using the next-segment approach, an MSB DAC made of 2^M identical elements guarantees monotonicity, although INL is still limited by the MSB matching.

To implement a segmented DAC using two resistor-string DACs, voltage buffers are needed to drive the LSB DAC without loading the MSB DAC. Although the resistor-string MSB DAC is monotonic, overall monotonicity is not guaranteed due to the offsets of the voltage buffers. The use of a capacitor-array LSB DAC eliminates a need for voltage buffers. The most widely used segmented DAC is a current-ratioed DAC, whose MSB DAC is made of identical elements for the next-segment approach, but the LSB DAC is a current divider. A binary-weighted current divider can be used as an LSB DAC, as shown in Figure 10.2. For monotonicity, the MSB M-bits are selected by a thermometer code, but one of the MSB current sources corresponding to the next segment of the thermometer code is divided by a current divider for fine LSBs.

Integrator-type DACs. As mentioned, monotonicity is guaranteed only in a thermometer-coded DAC. The thermometer coding of a DAC output can be implemented either by repeating identical DAC elements many times or by repeatedly using the same element. The former requires more hardware, but the latter more time. In the continuous time integrator-type DAC the integrator output is a linear ramp and the time to stop integrating can be controlled by the digital codes. Therefore, monotonicity can be maintained. Similarly, the discrete time integrator can integrate a constant amount of charge repeatedly and the number of integrations can be controlled by the digital codes. The integration approach can give high accuracy, but its disadvantage is its slow speed limiting its applications.

Although it is different in concept, oversampling interpolative DACs modulate the digital code in to a bitstream, and its pulse density represents the DAC output. Due to the incremental nature of the pulse density modulation, oversampling DACs are monotonic. A DAC for the pulse-density modulated

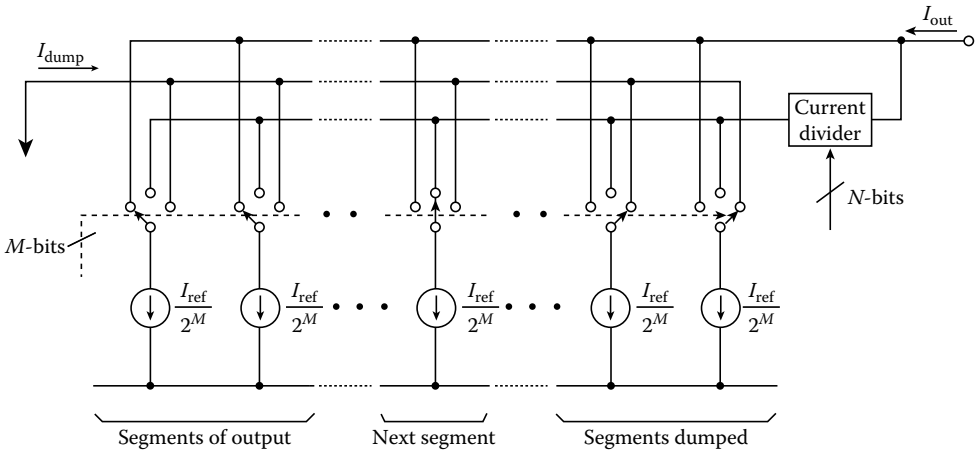


FIGURE 10.2 Segmented DAC for monotonicity.

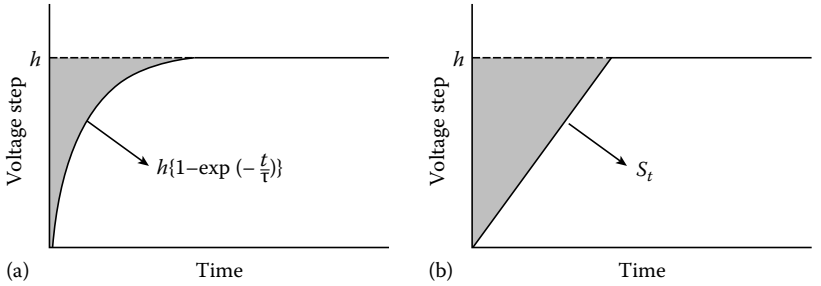


FIGURE 10.3 Errors in step response: (a) settling and (b) slewing.

bitstream is a lossy integrator. The integrator integrates a constant charge if the pulse is high, while it subtracts the same charge if the pulse is low. In principle, it is equivalent to the discrete time integrator DAC, but the output is represented by the average charge on the integrator.

10.1.2.3 Conversion Speed

The output of a DAC is a sampled-and-held step waveform held constant during a word clock period. Any deviation from the ideal step waveform causes an error in the DAC output. High-speed DACs usually have a current output, but even current-output DACs are either terminated with a 50 to 75 Ω low-impedance load or buffered by a wideband transresistance amplifier. Therefore, the speed of a DAC is limited either by the RC time constant of the output node or by the bandwidth of the output buffer amplifier. Figure 10.3 illustrates two step responses of a DAC when it settles with a time constant of τ and when it slews with a slew rate of S, respectively. The transient errors given by the shaded areas of Figure 10.3 are $h\tau$ and $h^2/2S$, respectively. This implies that a single time-constant settling of the former case only generates a linear error in the output, which does not effect the DAC linearity, but the slew-limited settling of the buffer generates a nonlinear error. Even in the single-time constant case (the former), the code-dependant time constant in settling can introduce a nonlinearity error because the settling error is a function of the time constant τ. This is true for a resistor-string DAC, which exhibits a code-dependent settling time because the output resistance of the DAC depends on the digital input.

The slew rate limit of the buffer is a significant source of nonlinearity since the error is proportional to the square of the signal, as shown in Figure 10.3b. The height and the width of the error term change with the input. The worst-case HD when generating a sinusoidal signal with a magnitude V_0 with a limited slew rate of S is [1]

$$\mathrm{HD}_k = 8\frac{\sin^2\frac{\omega T_c}{2}}{\pi k(k^2-4)} \times \frac{V_0}{ST_c}, \quad k = 1, 3, 5, 7, \ldots \tag{10.5}$$

where T_c is the clock period. For a given distortion level, the minimum slew rate is given. Any exponential system with a bandwidth of ω_0 gives rise to signals with the maximum slew rate of $2\omega_0 V_0$. Therefore, by making $2\omega_0 V_0 > S_{\min}$, no slew rate is limited and the DAC system will exhibit no distortion.

10.1.3 Converter Architectures

Many circuit techniques are used to implement DACs, but a few popular techniques used widely today are of the parallel type, in which all bits change simultaneously upon applying an input code word. Serial DACs, on the other hand, produce an analog output only after receiving all digital input data in a sequential form. When DACs are used as stand-alone devices, their output transient behavior limited by

glitch, slew rate, word clock jitter, settling, etc. are of paramount importance, but are used as subblocks of ADCs, DACs need only to settle within a given time interval. In stand-alone DAC applications, the digital input word made of *N*-bits should be synchronously applied to the DAC with a precise timing accuracy. Thus, the input data latches are used to hold the digital input during the conversion. The output analog sample-and-hold (S/H), usually called deglitcher, is often used for the better transient performance of a DAC. The three most popular architectures in integrated circuits are DACs using a resistor string, ratioed current sources, and a capacitor array. The current-ratioed DAC finds the greatest application as a stand-alone DAC, while the resistor-string and capacitor-array DACs are used mainly as ADC subblocks.

10.1.3.1 Resistor-String DACs

The simplest voltage divider is a resistor string. Reference levels can be generated by connecting 2^N identical resistors in series between V_{ref} and 0. Switches to connect the divided reference voltages to the output can be either a 1-out-of-2^N decoder or a binary tree decoder as shown in Figure 10.4 for the 3 bit example. Because it requires a good switch, the stand-alone resistor-string DAC is easier to implement using CMOS. However, the lack of switches does not limit the application of the resistor string as a voltage reference divider subblock for ADCs in other process technologies.

Resistor strings are used widely as reference dividers, an integral part of the flash ADC. All resistor-string DACs are inherently monotonic and exhibit good differential linearity. However, they suffer from a poor integral linearity and also have the drawback that the output resistance depends on the digital input code. This causes a code-dependent settling time when charging the capacitive load of the output bus. The code-dependent settling time has no effect on the reference divider performance as an ADC subblock, but the performance is severely degraded as a stand-alone DAC. This nonuniform settling time problem can be alleviated by adding low-resistance parallel resistors and by compensating the MOS switch overdrive voltages.

In bipolar technology, the most common resistors are thin-film resistors made of tantalum, Ni–Cr, or Cr–SiO, which exhibit very low voltage and temperature coefficients. In CMOS either diffusion or undoped poly resistors are common. Four of the most frequently used resistors are listed in Table 10.1. Conventional trimming or adjustment techniques are impractical to be applied to all 2^N resistor elements. The following four methods are often used to improve the integral linearity of resistor-string DACs.

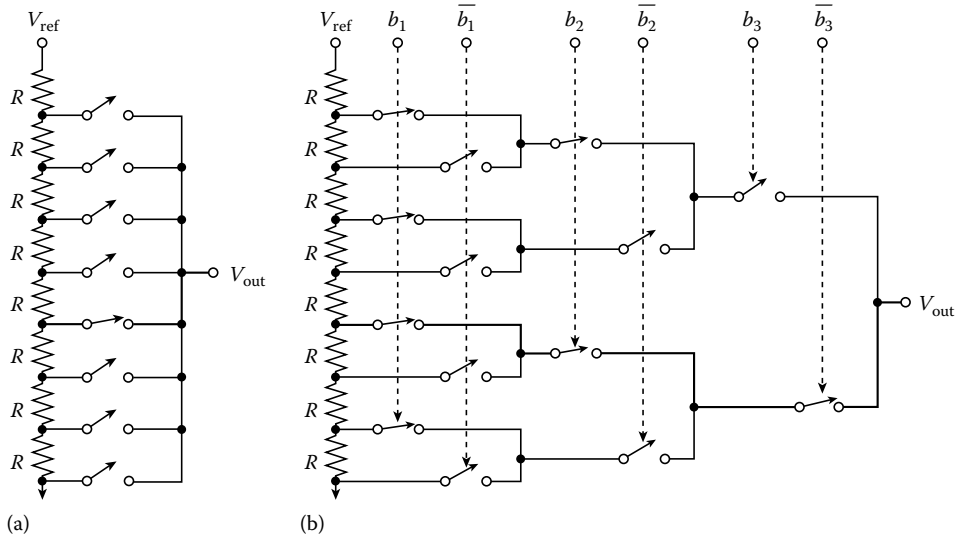


FIGURE 10.4 Resistor-string DAC: (a) with 1-out-of-2^N decoder and (b) with a binary tree decoder.

TABLE 10.1 Resistor in IC Processes

Resistor Type	Sheet R (Ω/sq)	Tolerance (%)	10–20 μm Matching (%)	T.C. (ppm/°C)
Diffusion	100–200	±20	±0.2–0.5	1500
Ion implantation	500–1 k	±5	±0.1	200
Thin film	1 k	±5	±0.1	10–100
Undoped poly	100–500	±20	±0.2	1500

Layout techniques. The use of large geometry devices and/or careful layout is effective in improving the matching marginally. Large geometry devices reduce the random edge effect, and the layout using a common centroid or geometric averaging can reduce the process gradient effect. However, typical matching of resistors in integrated circuits is still limited to an 8–10 bit level due to the mobility and resistor thickness variations. Differential resistor DACs with large feature sizes are reported to exhibit a higher matching accuracy of an 11–12 bit level.

Off-chip adjustment. It is possible to set tap points of a resistor-string to specified voltages by connecting external voltage sources to them, as shown in Figure 10.5a for the 3 bit example. Simply put, the more taps adjusted, the better the integral linearity obtained. An additional benefit of this method is the reduced RC time constant due to the voltage sources at the taps. Instead of using voltage sources, the required voltages can be obtained using parallel trimming, resistors, as shown in Figure 10.5b. However, in addition to external components for trimming, fine adjustments and precision measurement instruments are needed to ensure that voltage levels are correct. Furthermore, due to mismatch in the temperature coefficients between the external components and the on-chip components, retrimming is often required when temperature changes.

Postprocess trimming. The most widely used methods are laser trimming [2], Zener zapping [3], and other electrical trimming using PROM. The trimming method is the same as the parallel resistor trimming shown in Figure 10.5b except for the fact that external trimming resistors are now integrated on the chip. While being trimmed, the resistor string is biased with a constant current. Individual

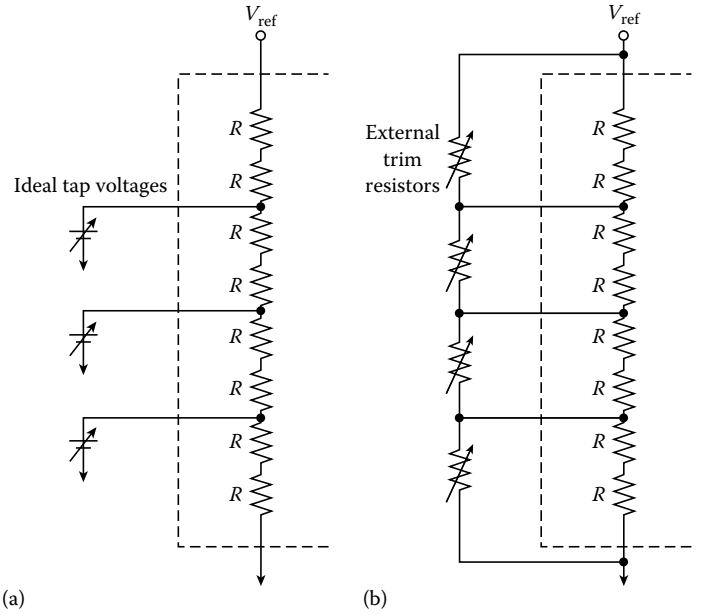


FIGURE 10.5 INL improvements by (a) external voltage sources and (b) parallel resistors.

segments are trimmed to have the same voltage drop. However, during normal conversion, the current source is replaced by a reference voltage source. The focused laser beam for trimming has a finite diameter, and the resistor to be trimmed occupies a large chip area. Both the laser trimming and the Zener zapping processes are irreversible. The long-term stability of trimmed resistors is a major concern, although the electrical and the PROM trimming (if PROM is replaced by EPROM) can be repeated. All trimming methods in this category are time consuming and require precision instruments.

On-chip buffers. The voltage at intermediate taps of the resistor string can be controlled by another resistor string through on-chip unity-gain buffers. This is actually an improved version of the off-chip method. The controlling resistors can be either laser trimmed or electronically controlled by switches. Laser-trimmed controlling resistors have the same problems mentioned earlier. The trimming network can be implemented to electronically control resistor values. In either case buffers with a high open-loop gain, a low output resistance, a large current driving capability, and a wide bandwidth for accurate and fast settling are required.

10.1.3.2 Current-Ratioed DACs

The most popular stand-alone DACs in use today are current-ratioed DACs, of which the two types are a weighted current DAC and an *R*-2*R* DAC.

Binary-weighted current DACs. The weighted current DACs shown in Figure 10.6 are made of an array of switched binary-weighted current sources and the current summing network. In bipolar

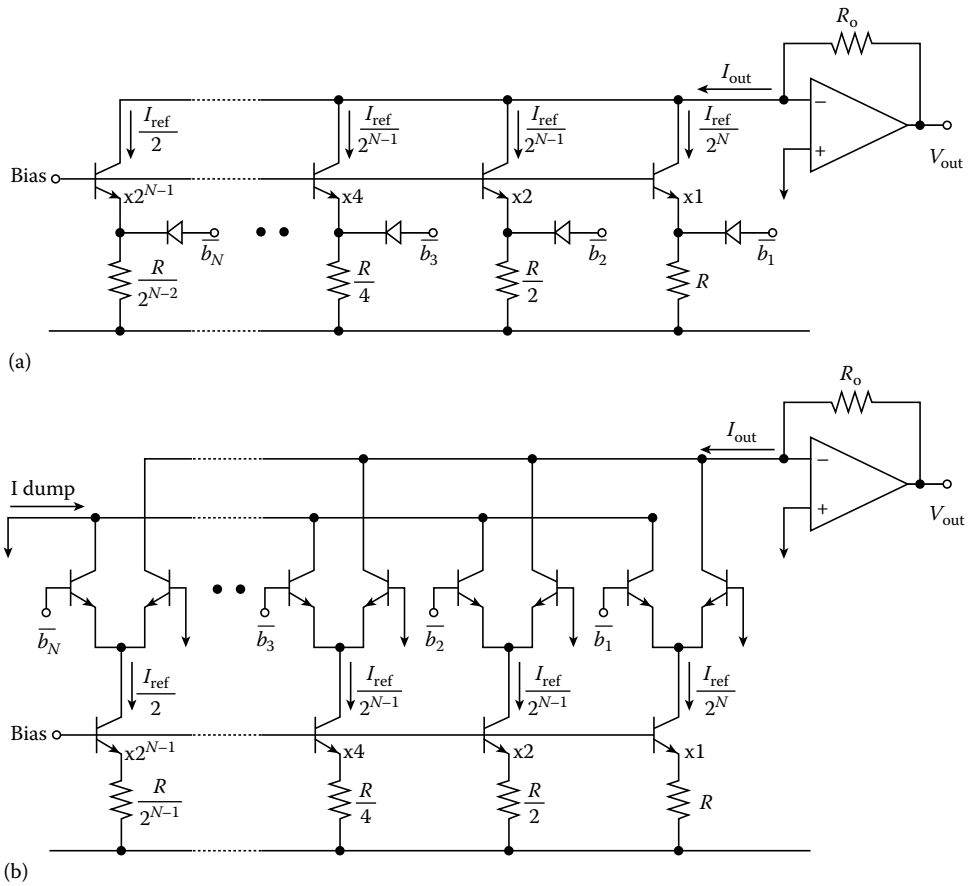


FIGURE 10.6 Binary-weighted current DAC: (a) diode switching and (b) differential pair switching.

technology binary weighting is achieved by ratioed transistors and emitter resistors with binary-related values of R, $R/2$, $R/4$, and so on, while in MOS technology, only ratioed transistors are used. One example is a video random access memory DAC in CMOS, which is made of simple PMOS differential pairs with binary-weighted tail currents. DACs relying on active device matching can achieve an 8 bit level performance with a 0.2%–0.5% matching accuracy using a 10–20 μm device feature size while degeneration with thin-film resistors gives a 10 bit level performance. The current sources are switched on or off by means of switching diodes or emitter-coupled differential pairs (source-coupled pairs in CMOS), as shown in Figure 10.6. The output current summing is done by a wideband transresistance amplifier, but in high-speed DACs, the output current is used directly to drive a resistor load for a maximum speed but with a limited output swing. The weighted current design has the advantage of simplicity and high speed, but it is difficult to implement a high-resolution DAC because a wide range of emitter resistors and transistor sizes is used and very large resistors cause problems with both temperature stability and speed.

***R*-2*R* ladder DACs.** This large resistor ratio problem is alleviated by using a resistor divider known as an *R*-2*R* ladder, as shown in Figure 10.7. The *R*-2*R* network consists of series resistors of value *R* and shunt resistors of value 2*R*. The top of each shunt resistor 2*R* has a single-pole double-throw electronic switch which connects the resistor either to ground or to the output current summing node. The operation of the *R*-2*R* ladder network is based on the binary division of current as it flows down the ladder. At any junction of series resistor *R*, the resistance looking to the right side is 2*R*. Therefore, the input resistance at any junction is *R*, and the current splits into two equal parts at the junction because it sees equal resistances in either direction. The result is binary-weighted currents flowing into each shunt resistor in the ladder. The digitally controlled switches direct the current to either ground or to the summing node. The advantage of the *R*-2*R* ladder method is that only two values of resistors are used, greatly simplifying the task of matching or trimming and temperature tracking. In addition, for high-speed applications relatively low resistor values can be used. Excellent results can be obtained using laser-trimmed thin-film resistor networks. Because the output of the *R*-2*R* DAC is the product of the reference voltage and the digital input word, the *R*-2*R* ladder DAC is often called a multiplying DAC (MDAC).

Both the weighted current DAC and the *R*-2*R* DAC can be used as a current divider to make a sub-DAC. To make a segmented DAC for monotonicity based on the next-segment approach, as discussed earlier, the MSB should be made of thermometer-coded equal currents. Once the MSB is selected, the next segment should be divided further into LSBs as shown in Figure 10.2. INL can be improved by dynamically matching or by self-calibrating the MSB current sources as discussed later.

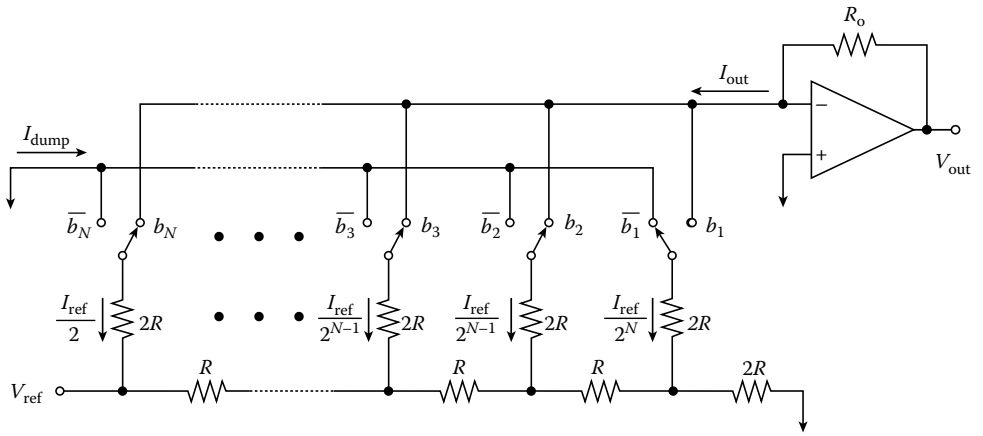


FIGURE 10.7 *R*-2*R* ladder DAC.

Capacitor-array DAC. Capacitors made of double-poly or poly-diffusion in MOS technology are considered one of the most accurate passive components comparable to thin-film resistors in bipolar process, both in the matching accuracy and voltage and temperature coefficients [4]. The only disadvantage in the capacitor-array DAC implementation is the use of a dynamic charge redistribution principle. A switched-capacitor counterpart of the resistor-string DAC is a parallel capacitor array of 2^N unit capacitors (C) with a common top plate. The capacitor-array DAC is not appropriate for stand-alone applications without a feedback amplifier virtually grounding the top plate and an output S/H or deglitcher. The operation of the capacitor-array DAC in Figure 10.8 is based on the thermometer-coded DAC principle, and has a distinct advantage of monotonicity if the system is implemented properly. However, due to the complexity of handling the thermometer-coded capacitor array, a binary-weighted capacitor array is often used, as shown in Figure 10.9, by grouping unit capacitors in binary ratio values. A common centroid layout of the capacitor array is known to give a 10 bit level matching for this application when the unit capacitor size is over 12×12 μm. The matching accuracy of the capacitor in MOS technology depends on the geometry sizes of the capacitor width and length and the dielectric thickness.

As a stand-alone DAC, the top plate of the DAC is precharged either to the offset of the feedback amplifier or to the ground. One smallest capacitor is not necessary for this application. However, as a subblock of an ADC, the total capacitor should be 2^NC, as drawn in Figure 10.9, and the top plate of the

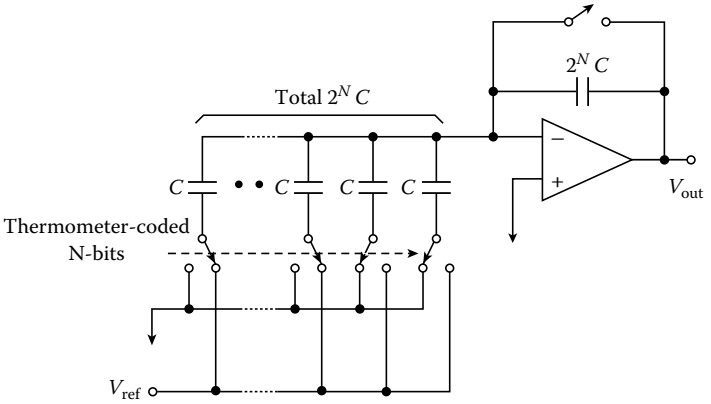


FIGURE 10.8 Thermometer-coded capacitor-array DAC.

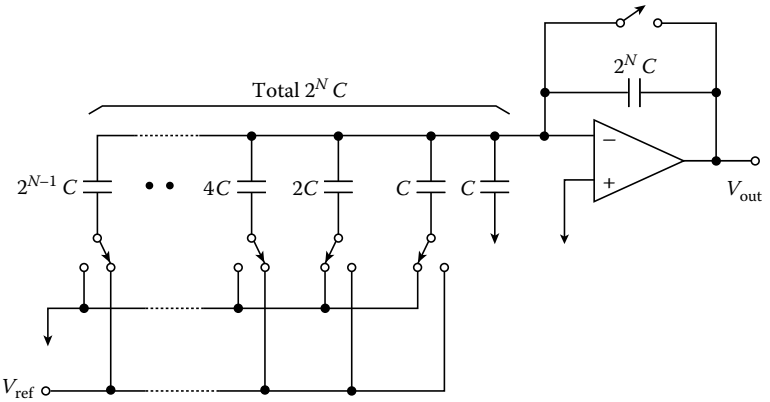


FIGURE 10.9 Binary-weighted capacitor-array DAC.

array is usually connected to the input nodes of comparators or high-gain operational amplifiers, depending on the ADC architectures. As a result, the top plate has a parasitic capacitance, but its effect on the DAC performance is negligible. The capacitor-array DAC requires two-phase nonoverlapping clocks for proper operation. Initially, all capacitors should be charged to ground. After initialization, depending on the digital input, the bottom plates are connected either to V_{ref} or to ground. Consider the case in which the top plate is floating without the feedback amplifier. If the charge at the top plate finishes its redistribution the top plate voltage neglecting the top plate parasitic effect becomes

$$V_0 = \sum_{i=1}^{N} \frac{b_k}{2^i} V_{\text{ref}} \tag{10.6}$$

where $b_N\, b_{N-1}, \ldots, b_2\, b_1$ is the input binary word. For example, switching the MSB capacitor bottom to V_{ref} changes the output voltage by

$$\frac{2^{N-1}C}{\sum_{i=1}^{N} C_i} V_{\text{ref}} = \frac{V_{\text{ref}}}{2} \tag{10.7}$$

where the capacitor C_i, for the ith bit is nominally scaled to $2^{i-1}C$. Therefore, the nonlinearity at the midpoint of the full range is limited by the ratio mismatch of the half sum of the capacitor array to the total sum of the array. Similarly, the nonlinearity at one fourth of the range is limited by the ratio of one fourth of the capacitor array to the total array, and so on.

One important application of the capacitor-array DAC is as a reference DAC for ADCs. As in the case of the *R*-2*R* MDAC, the capacitor-array DAC can be used as an MDAC to amplify residue voltages for multistep ADCs. As shown in Figure 10.9, if the input is sampled on the bottom plates of capacitors instead of the ground, the voltage amplified is the amplified input voltage minus the DAC output. By varying the feedback capacitor size, the MDAC can be used as an interstage residue amplifier in multistep pipelined ADCs. For example, if the feedback capacitor is C and the digital input is the coarse N-bit decision of the sampled analog voltage, the amplifier output is a residue voltage amplified by 2^N for the subsequent LSB conversion.

10.1.3.3 $R + C$ or $C + R$ Combination DACs

Both resistor-string and capacitor-array DACs need 2^N unit elements for N-bits, and the number grows exponentially. Splitting arrays into two, one for MSBs and the other for the LSBs, requires a buffer amplifier to interface between two arrays. Although a floating capacitor connects two capacitors arrays, the parasitic capacitance of the floating node is not well controlled. A more logical combination for high-resolution DAC is between resistor and capacitor DACs. This combination does not require any coupling capacitors or interface buffer amplifiers.

In the $R + C$ combination, the MSB is set by the resistor string, and next segment of the resistor-string DAC supplies the reference voltage of the LSB capacitor DAC, as shown in Figure 10.10. When the top plate is initialized, all capacitor bottom plates are connected to the lower voltage of the next segment of the resistor-string DAC. During the next clock phase, the bottom plates of capacitors are selectively connected to the higher voltage of the segment if the digital bit is ONE, but stays switched to the lower voltage if ZERO. This segmented DAC approach gives an inherent monotonicity as far as the LSB DAC is monotonic within its resolution. Although INL is poor, the fully differential implementation of this architecture benefits from the lack of the even-order nonlinearity, thereby achieving improved INL. On the other hand, in the $C + R$ combination shown in Figure 10.11, the operation of the capacitor DAC is the same. The MSB side reference voltage is fixed, but the reference voltage of the smallest capacitor is supplied by the LSB resistor-string DAC. This approach exhibits nonmonotonicity due to the capacitor

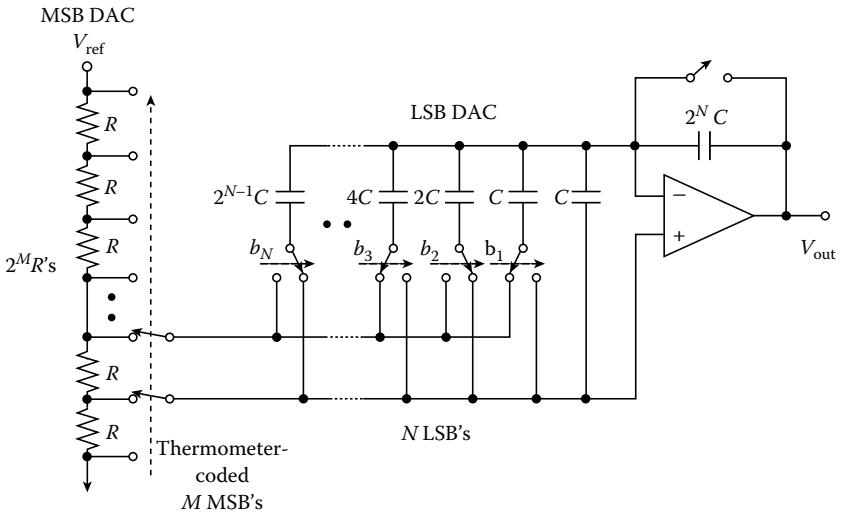


FIGURE 10.10 $R + C$ combination DAC.

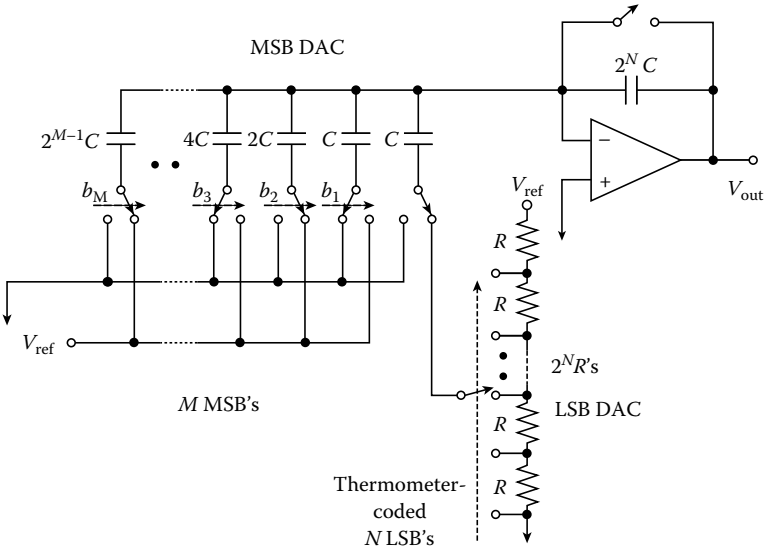


FIGURE 10.11 $C + R$ combination DAC.

DAC matching. Both combination DACs are seldom used as stand-alone DACs due to their limited speed, but are used frequently as subblocks of high-resolution ADCs.

10.1.4 Techniques for High-Resolution DACs

Most DACs are made of passive or active components such as resistors, capacitors, or current sources, and their linearity relies on the matching accuracy of those components. Among frequently used DAC components, diffused resistors and transistors are in general known to exhibit an 8 bit level matching while thin-film resistors and capacitors are matched to a 10 bit level. Trimming or electronic calibration is needed in order to obtain a higher linearity than what is achievable with bare component matching.

The traditional solutions to this have been the wafer-level trimming methods such as laser trimming and Zener zapping. Although many other promising trimming or matching techniques such as polysilicon fuse trimming, electrical trimming techniques using PROM, and large device matching by geometrical averaging have been proposed, conventional factory-set trimming or matching techniques give no flexibility of retrimming. How successfully these techniques can be applied to large-volume production of high-resolution DACs and how the factory-trimmed components will perform over the long term are still in question.

The future trend is toward more sophisticated and intelligent electronic solutions that overcome and complement some of the limitations of conventional trimming techniques. The methods recently developed are dynamic circuit techniques [5] for component matching, switched-capacitor integration [6], electronic calibration [7] of a DAC nonlinearity, and oversampling interpolation techniques [8] which trade speed with resolution. In particular, the oversampling interpolative DACs are used widely in stand-alone applications such as digital audio playback systems or digital communications due to their inherent monotonicity.

10.1.4.1 Dynamic Matching Techniques

In general, a dynamic element matching to improve the accuracy of the binary ratio is a time-averaging process. For simplicity, consider a simple voltage or current divide-by-two element, as shown in Figure 10.12. Due to mismatches in the ratio of resistors, transistors, and capacitors, the divided voltage or current is not exactly $V_{ref}/2$ or $I_{ref}/2$, but their sum is V_{ref} or I_{ref}. The dynamic matching concept is to multiplex these two outputs with complementary errors of Δ and $-\Delta$ so that the errors Δ and $-\Delta$ can be averaged out over time while the average value of $V_{ref}/2$ or $I_{ref}/2$ remains. It is in effect equivalent to the suppressed carrier balanced modulation of the error component Δ. The high-frequency energy can be filtered out using a post low-pass filter. This technique relies on the accurate timing of the duty cycle. Any duty cycle error or timing jitter results in inaccurate matching. The residual matching inaccuracy becomes a second-order error proportional to the product of the original mismatch and the timing error.

The application of dynamic element matching to the binary-weighted current DAC is a straightforward switching of two complementary currents. Its application to the binary voltage divider using two identical resistors or capacitors requires exchanging of resistors or capacitors. This can be achieved easily by reversing the polarity of the reference voltage for the divide-by-two case. However, in the general case of N element matching the current division is inherently simpler in implementation than the voltage division. In general, to match the N independent elements, a switching network with N inputs and N outputs is required. The function of the switching network is to connect any input out of N inputs to

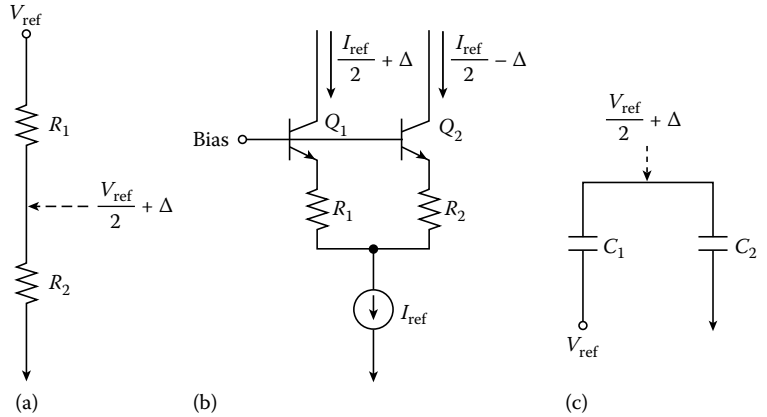


FIGURE 10.12 Divide-by-two elements: (a) resistor, (b) current, and (c) capacitor.

one output with an average duty cycle of $1/N$. The simplest one is a barrel shifter rotating the I–O connections in a predetermined manner [5]. This barrel shifter generates a low-frequency modulated error when N becomes larger because the same pattern repeats every N clocks. A more sophisticated randomizer with the same average duty can distribute the mismatch error over the wider frequency range. The latter technique finds applications as a multibit DAC in the multibit noise-shaping sigma-delta data converter, whose linearity relies on the multibit DAC.

Voltage or current sampling. The voltage or current sampling concept is an electronic alternative to direct mechanical trimming. To sample voltage or current using a voltage or current sampler is equivalent to trimming individual voltage or current sources. The voltage sampler is usually called a S/H circuit, while the current sampler is called a current copier. The voltage is usually sampled on the input capacitor of a buffer amplifier and the current is usually sampled on the input capacitor of a transconductance amplifier such as an MOS transistor gate. Therefore, both voltage and current sampling techniques are ultimately limited by their sampling accuracy.

The idea behind the voltage or current sampling DAC is to use one voltage or current element repeatedly. One example of the voltage sampling DAC is a discrete-time integrator-type DAC with many S/H amplifiers for sampling output voltages. The integrator integrates a constant charge repeatedly, and its output is sampled on a new S/H amplifier every time the integrator finishes an integration as shown in Figure 10.13a. This is equivalent to generating equally spaced reference voltages by stacking identical unit voltages [6]. The fundamental problem associated with this sampling voltage DAC approach is the accumulation of the sampling error and noise in generating larger voltages. Similarly, the current sampling DAC can sample a constant current on current sources made of MOS transistors, as shown in Figure 10.13b [7]. Because one reference current is copied on other identical current samplers, the matching accuracy can be maintained as far as the sampling errors are kept constant. It is not practical to make a high-resolution DAC using voltage or current sampling alone. Therefore, this approach is limited to generating MSB DACs for the segmented DAC or for the subranging ADCs.

10.1.4.2 Electronic Calibration Techniques

Electronic calibration is a general term to describe various circuit techniques, which usually predistort the DAC transfer characteristic so that the DAC linearity can be improved. One such technique is a

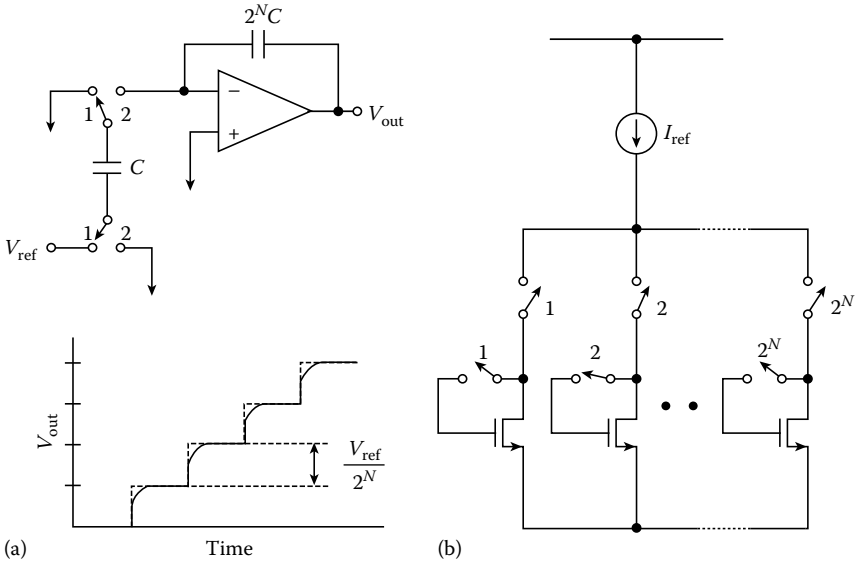


FIGURE 10.13 Voltage and current sampling concepts: (a) integrator and (b) current copier.

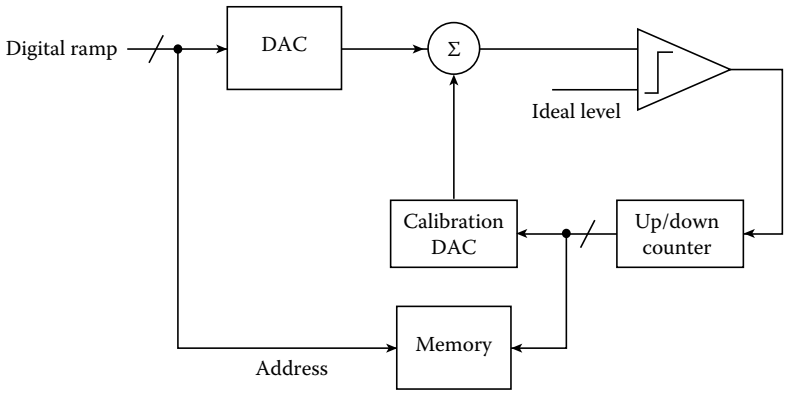


FIGURE 10.14 Code mapping with a calibration DAC.

straightforward code mapping, and the other is a self-calibration. The code-mapping calibration is a very limited technique only for the factory because it requires a precision measurement setup and a large digital memory. The self-calibration is to incorporate all the calibration mechanisms and hardware on the DAC as a built-in function so that users can recalibrate whenever calibrations are needed. The self-calibration is based on an assumption that the segmented DAC linearity is limited by the MSB DAC so that only errors of MSBs may be measured, stored in memory, and recalled during normal operation. There are two ways of measuring the MSB errors. In one method individual-bit nonlinearities, usually appearing as component mismatch errors, are measured digitally [9], and a total error, which is called a code error, is computed from individual-bit errors depending on the output code during normal conversion. On the other hand, the other method measures and stores digital code errors directly and eliminates the digital code-error computation during normal operation [10]. The former requires less digital memory during normal conversion while the latter requires fewer digital computations.

Direct code mapping. The simplified code mapping of a DAC can be done with a calibration DAC, digital memory, and a precision instrument to measure the DAC output, as shown in Figure 10.14. The idea is to measure the DAC error using a calibration DAC so that the DAC output corrected by the calibration DAC can produce an ideal DAC output. The input code of the calibration DAC is stored as a code error in digital memory addressed by the DAC input code. This code error is recalled to predistort the DAC output during normal operation. This technique needs a 2^N memory with a word length corresponding to the number of bits of the calibration DAC. This method can correct any kinds of DAC nonlinearities as long as the calibration DAC has an output range wide enough to cover the whole range of nonlinearity. However, the same method can be implemented without the use of a calibration DAC if the main DAC is monotonic with extra bits of resolution. In this case, the calibration is a simple code mapping, selecting correct input digital codes for correct DAC output voltages among redundant input digital codes.

Self-calibration for individual capacitor errors. The idea of measuring the individual bit errors using a calibration DAC is to quantize the difference Δ in the divide-by-two elements in Figure 10.12 because the ideal divide ration is 1/2. For example, the MSB should be half of the whole range of a DAC, the second MSB is half of the MSB, and so on. Unless buffer amplifiers are used, the ideal calibration DACs for *R* and *C* DACs are *C* and *R* DACs, respectively. The ratio error measurement cycle of 2 bit *C* DAC is illustrated in Figure 10.15. Errors can be quantized using a successive approximation method, but the up/down converter is shown here for simplicity. Initially, the top plate is charged to the comparator offset and the bottom plates of C_1 and C_2 sample 0 and V_{ref}. At the same time, the bottom plate of C_C samples $V_{\text{ref}}/2$ and the up/down counter is reset to make $V_{\text{cal}} = 0$. In the next clock period the charge is redistributed by swapping 0 and V_{ref} on the bottom plates of C_1 and C_2. Then the top plate residual error V_x will be from the charge conservation

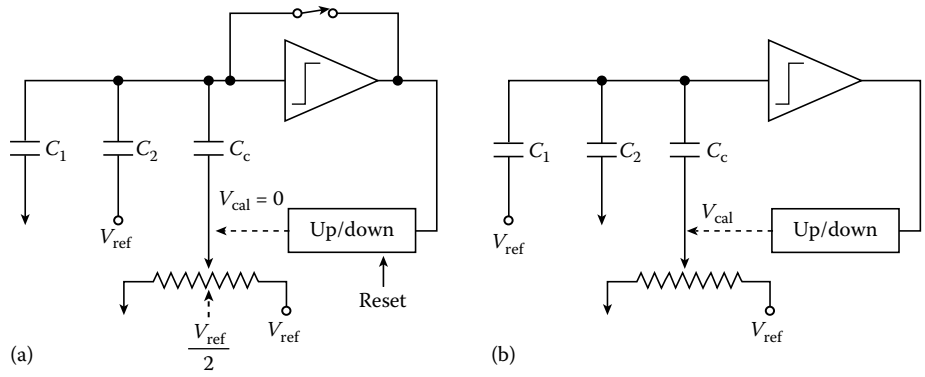


FIGURE 10.15 Capacitor ratio error measurement cycles: (a) initialization and (b) error quantization.

$$V_x = \frac{C_1}{C_1 + C_2 + C_C} V_{ref} - \frac{C_2}{C_1 + C_2 + C_C} V_{ref} = \frac{\Delta C}{2C + C_C} V_{ref} \tag{10.8}$$

If $V_{cal} = 0$, $C_1 = C + \Delta C/2$, and $C_2 = C - \Delta C/2$. This top plate residual voltage can be nulled out by changing the calibration DAC voltage V_{cal}. The measured calibration voltage is approximately

$$V_{cal} = -\frac{\Delta C}{C_C} V_{ref} \tag{10.9}$$

As the actual error V_x is half of the measured value when the V_{ref} is applied to C_1, the actual calibration DAC voltage to be subtracted during normal operation becomes $V_{cal}/2$. Similarly, the multibit calibration can start from the MSB measurement and move down to the LSB side [9].

The extension of this calibration technique to current DACs is straightforward. For example, two identical unipolar currents, I_1 and I_2, can be compared using a voltage comparator and a calibration DAC, as shown in Figure 10.16. After I_1 is switched in, the calibration DAC finds an equilibrium as a null. Then the difference can be measured by interchanging I_1 and I_2 and finding a new equilibrium. Therefore, the current difference error is obtained as

$$I_{cal} = I_2 - I_1 \tag{10.10}$$

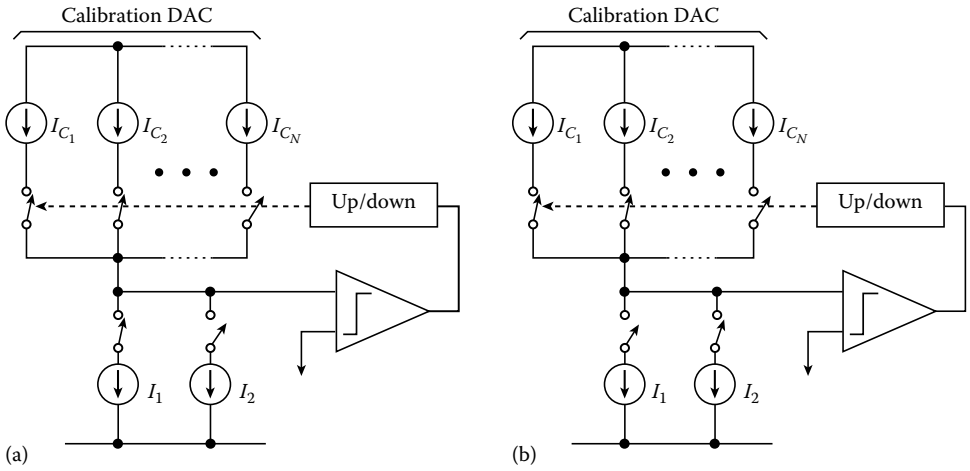


FIGURE 10.16 Current difference measurement cycles: (a) initialization and (b) error quantization.

During normal operation, half of this value should be added to the DAC output using the same calibration DAC every time the current I_1 is switched to the DAC output. Similarly, the same amount is subtracted if I_2 is switched to the output.

Code-error calibration. The code-error calibration is based on the simple fact that the thermometer-coded MSBs of a DAC are made of segments of equal magnitude [10]. Any nonuniform segment will contribute to the overall nonlinearity of a DAC. The segment error between two adjacent input codes is measured by comparing the segment with the ideal segment. Starting from the reference point, 0 or $V_{ref}/2$, the same procedure is repeated until all the segment errors are measured. Therefore, the current code error, Error(j), is obtained by adding the current segment error to the accumulated sum of all the previous segment errors:

$$\text{Error}(j) = \sum_{k=1}^{j} \text{Seg}(k) \tag{10.11}$$

where Seg(k) is the kth segment error from the reference point. These measured code errors are stored in memory addressed by digital codes so that they can be subtracted from uncalibrated raw digital outputs during normal conversion.

The segment error measurement of a current-ratioed DAC thermometer-coded MSBs is similar to the current difference measurement in Figure 10.16. The only difference in measurement is the use of the reference segment current in place of one of the two currents to be compared. That is, each MSB current source is compared to the reference segment. For the capacitive DAC, the kth segment error can be measured in two cycles. After the output of the DAC is initialized to have a negative ideal segment voltage with the input digital code corresponding to $k-1$, the input code is increased by 1 as shown in Figure 10.17. Applying digital codes to the capacitor-array DAC means connecting the bottom plates to either V_{ref} or ground depending on the corresponding digital bits. Then the kth segment error is generated at the output and can be measured digitally using subsequent ADC stages or using a calibration DAC as shown in Figure 10.15.

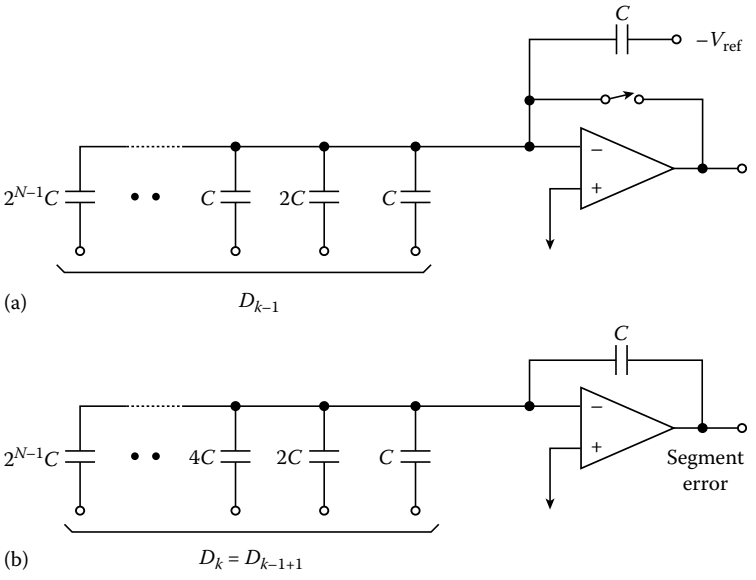


FIGURE 10.17 Code-error measurement cycle: (a) initialization and (b) error quantization.

Digital truncation errors. All calibration methods need extra bits of resolution in the error measurements because digital truncation errors are accumulated during code-error computations. For example, if the truncation errors are random, the additions of n digital numbers will increase the standard deviation of the added number by $n^{1/2}$. This accumulated truncation error affects both DNL and INL of the converter self-calibrated using measured errors of individual bits. On the other hand, if calibrated using measured segment errors, the DNL of the converter is always guaranteed to be within $\pm\frac{1}{2}$ LSB of a target resolution because all segment errors are measured with one extra bit of resolution, but the INL will still be affected by the digital truncation because code errors are obtained by accumulating segment errors. The effect of the digital truncation errors due to n repeated digital additions on the INL can be modeled using uncorrelated and independent random variables, and the standard deviation of INL is calculated in LSB units as

$$\sigma_{\text{INL}} = \sqrt{\frac{(n-i)(i-1)}{12(n-1)}}(\text{LSB}) \quad \text{for } i = 1, 2, \ldots, n \tag{10.12}$$

For example, when $n = 16$, the maximum standard deviation of the INL at the midpoint is about 0.56 LSB.

10.1.4.3 Interpolative Oversampling Techniques

Ordinary DACs generate a discrete output level for every digital word applied to their input, and it is difficult to generate a large number of distinct output levels for long words. The oversampling interpolative DAC achieves fine resolution by covering the signal range with a few widely spaced levels and interpolating values between them. By rapidly oscillating between coarse output levels, the average output corresponding to the applied digital code can be generated with reduced noise in the signal band [8]. The general architecture of the interpolative oversampling DAC is shown in Figure 10.18. A digital filter interpolates sample values of the input signal in order to raise the word rate to a frequency well above the Nyquist rate. The core of the technique is a digital truncator to truncate the input words to shorter output words. These shorter words are then converted into analog form at the high sample rate so that the truncation noise in the signal band may be satisfactorily low. The sampling rate upconversion for this is usually done in stages using two upsampling digital filters. The first filter, usually a two to four times oversampling FIR, is to shape the signal band for sampling rate upconversion and to equalize the passband droop resulting from the second SINC filter for higher-rate oversampling.

A noise-shaping sigma-delta-sigma modulator can be built in digital form to make a digital truncator as shown in Figure 10.19. Using a linearized model, the z-domain transfer function of the modulator is

$$Y(z) = \frac{\alpha H(z)}{1 + \alpha H(z)} X(z) + \frac{1}{1 + \alpha H(z)} Q(z) \tag{10.13}$$

where

$Q(z)$ is the quantization noise

α is the quantizer gain

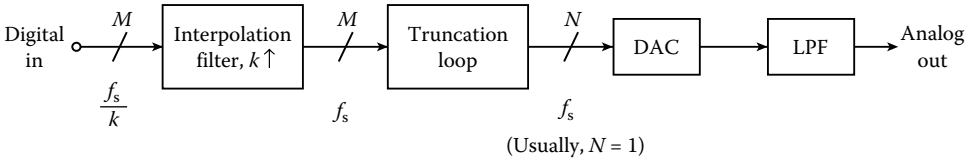


FIGURE 10.18 Interpolative oversampling DAC.

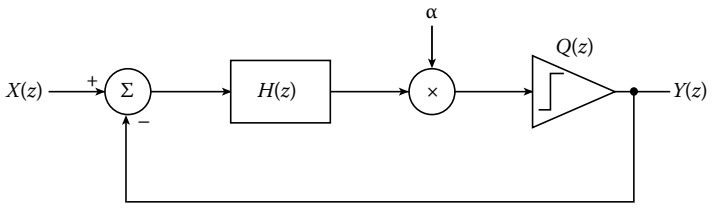


FIGURE 10.19 Delta-sigma modulation as a digital truncator.

The loop filter $H(z)$ can be chosen so that the quantization noise may be high-pass filtered while the input signal is low-pass filtered. For the first-order modulator, the loop filtered is just an integrator with a transfer function of

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{10.14}$$

while for the second-order modulator, the transfer function is

$$H(z) = \frac{z^{-1}(2 - z^{-1})}{(1 - z^{-1})^2} \tag{10.15}$$

However, the standard second-order modulator is implemented, as shown in Figure. 10.20, using a double integration loop. In general, first-order designs tend to produce correlated idling patterns. Second-order designs are vastly superior to first-order designs both in terms of the required OSR to achieve a particular SNR as well as in the improved randomness of the idling patterns. However, even the second-order loop is not entirely free of correlated fixed patterns in the presence of small DC inputs. The second-order loop needs dithering to reduce fixed pattern noises, but loops of a higher order than third do not exhibit fixed pattern noises.

Stability. The quantizer gain α plays an important role in keeping the modulator stable. Considering α, the transfer function of the second-order loop shown in Figure 10.20 becomes

$$Y(z) = \frac{\alpha z^{-2} X(z) + (1 - z^{-1})^2 Q(z)}{1 - 2(1 - \alpha)z^{-1} + (1 - \alpha)z^{-2}} \tag{10.16}$$

The root locus of the transfer function in the z-domain is shown in Figure 10.21. As shown, the second-order loop becomes unstable for $\alpha > 4/3$ because one pole moves out of the unit circle. This in turn implies that the signal at the input of the quantizer becomes too large. Most delta-sigma modulators become unstable if the signal to the quantizer exceeds a certain limit. Higher-order modulators tend to be

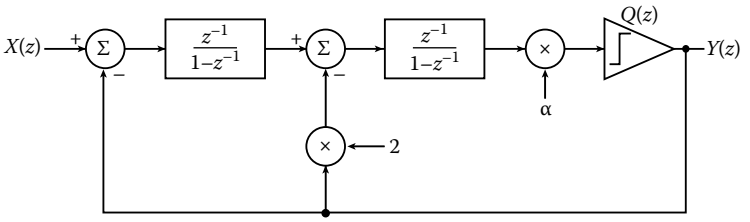


FIGURE 10.20 Second-order 1 bit modulator.

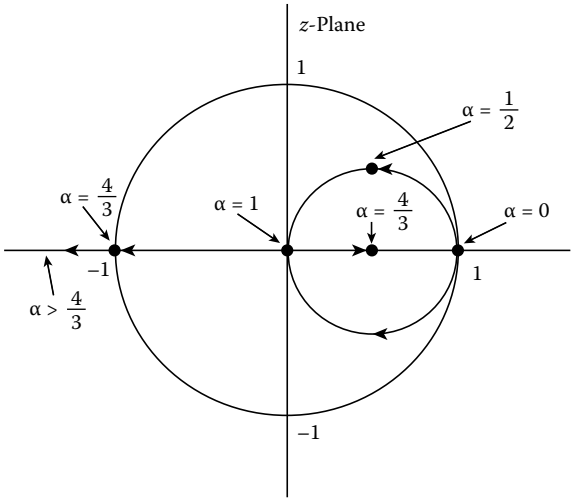


FIGURE 10.21 Root locus of the second-order loop transfer function.

overloaded easily at higher quantizer gain than first-or second-order modulators. Therefore, for the stability reason, the integrator outputs of the loop filter are clamped so that the signal at the input of the quantizer is limited for linear operation. Digital truncators of a higher order than second are feasible in digital circuits because signal levels can be easily detected and controlled. The straightforward third order or higher loop using multiple loops is unstable, but higher order modulators can be built using either the cascaded MASH [11] or the single-bit higher order [12] architecture.

Dynamic range. In general, for the Nth order loop, the noise falls by $6N + 3$ dB for every doubling of the sampling rate, providing $N + 0.5$ extra bits of resolution. Because the advantage of oversampling begins to appear when the OSR is >2, a practically achievable dynamic range by oversampling is approximately

$$\mathrm{DR} > (6N + 3)(v)\mathrm{dB} \tag{10.17}$$

where M is the OSR. For example, a second-order loop with 256 times oversampling can give a dynamic range of >105 dB, but the same dynamic range can be obtained using a third-order loop with only 64 times oversampling. The dynamic range is not a limit in the digital modulator. In practice, the dynamic range is limited in the rear-end analog DAC and postfilter.

One-bit or multibit DAC. The rear end of the interpolative oversampling DAC is an analog DAC. Because the processing in the interpolation filter and truncator is digital instead of analog, achieving precision is easy. Therefore, the oversampling DAC owes its performance to the rear-end analog part because the conversion of the truncated digital words into analog form takes place in the rear-end DAC. The 1 bit quantizer can be easily overloaded and needs clamping to be stable, while multibit quantizers are more stable due to their small quantization errors. However, the multibit system is limited by the accuracy of the multibit DAC. Although the analog techniques such as dynamic matching or self-calibration can improve the performance of the multibit DAC, the 1 bit DAC is simpler to implement and its performance is not limited by component matching. It is true that a continuous time filter can convert the 1 bit digital bitsteam into an analog waveform, but it is difficult to construct an ideal undistorted digital waveform without clock jitter. However, if the bitstream is converted into a charge packet, a high linearity is guaranteed due to the uniformity of the charge packets.

A typical differential 1 bit switched-capacitor DAC with one-pole roll-off can be built as shown in Figure 10.22 using two-phase nonoverlapping clocks 1 and 2. There are many advantages in a fully

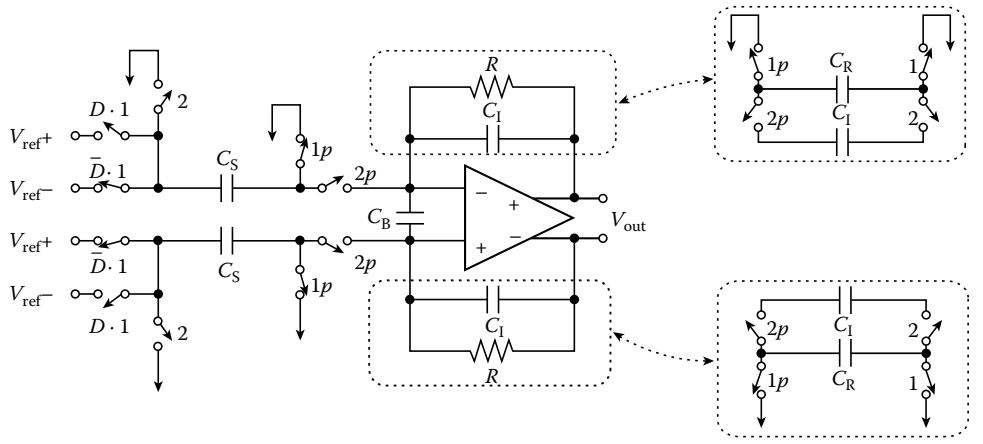


FIGURE 10.22 Switched-capacitor 1 bit DAC/filter.

differential implementation. The dynamic range increases by 3 dB because the signal is doubled (6 dB) but the noise gains by 3 dB. It also rejects most noisy coupling through power supplies or through the substrate as a common-mode signal. Furthermore, the linearity is improved because the even-order nonlinearity components of the capacitors and the op-amp are canceled. In the implementation of Figure 10.22, a resistor as a loss element can be replaced by a capacitor switched in and out at f_c as illustrated. The bandwidths of these filters in both cases are set by $1/RC_I$ and f_cC_R/C_I, respectively, Also, the filter DC gains are defined by Rf_c C_S and C_S/C_R, respectively. The digital bitstream is converted into a charge packet by sampling the reference voltage on the bottom plates of the sampling capacitors (C_S). If the digital bit is ZERO, $-V_{ref}$ is sampled during the clock phase 1 and the charge on C_S is dumped on the lossy integrator during phase 2. On the other hand, if the digital bit is ONE, V_{ref} is sampled instead. To reduce the input-dependent switch-feedthrough component, the switches connected to the top plates should be turned off slightly earlier than the bottom plate switches using $1p$ and $2p$. Alternatively, a slightly different 1 bit DAC is possible by sampling a constant reference voltage by inverting the polarity of the integration depending on the digital bit as shown in Figure 10.23.

The op-amp for this application should have a high DC gain and a fast slew rate. The op-amp DC gain requirement is a little alleviated considering the linear open-loop transfer characteristic of most op-amps

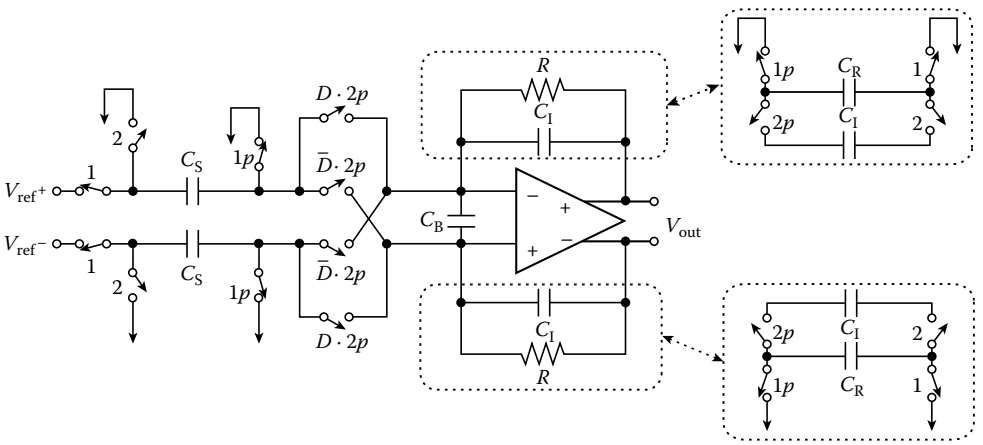


FIGURE 10.23 Alternative 1 bit DAC sampling constant V_{ref}.

within a limited swing range. As discussed earlier, the slew-limited settling generates an error proportional to the square of the magnitude. Therefore, a nonslewing op-amp such as a class AB input op-amp performs better for this application. The op-amp starts to slew when a larger voltage than its linear input range is applied. When the charge packet of the sampled reference voltage is dumped onto the input summing node, it causes a voltage step on the summing node. The bypass capacitor C_B between two summing nodes helps to reduce this voltage step to prevent the op-amp from slewing. The large the C_B, the smaller the voltage step. However, a too-large C_B will narrowband the feedback amplifier, and the settling will take longer as a result.

Postfiltering requirement. Although the one-pole roll-off will substantially attenuate high-frequency components around f_c, the 1 bit DAC should be followed by a continuous time postfilter so that the charge packets can be smoothed out. Unlike the delta-sigma modulator which filters out the out-of-band-shaped noise using digital filters, the demodulator output noise can be filtered only by analog filters. Because the shaped noise is out-of-band, it does not affect the inband performance directly, but the large out-of-band high-frequency noise tends to generate inband intermodulation components and limit the dynamic range of the system. Therefore, the shaped high-frequency noise needs to be filtered with a low-pass filter one order higher than the order of the modulator. It is challenging to meet this postfiltering requirement with analog filtering techniques. Analog filters for this application are often implemented in continuous time using a cascade of Sallen–Key filters made of emitter follower unity-gain buffers, but both switched-capacitor and continuous time filtering techniques have improved significantly to be applied to this application. The other possibility is the hybrid implementation of an FIR filter using digital delays and an analog current swimming network. Because the output is a bitstream, current sources weighted using coefficients of an FIR filter are switched to the current summer depending on the digital bit to make a low-pass FIR filter.

10.1.5 Sources of Conversion Errors

10.1.5.1 Glitch

The basic function of the DAC is the conversion of digital numbers into analog waveforms. A distortion-free DAC creates instantaneously an output voltage that is proportional to the input digital number. In reality, DACs cannot achieve this impossible goal. If the input digital number changes from one value to a different one, the DAC output voltage always reaches a new value sometime later. For DACs, the shape of the transient response is a function governed in large part by two mechanisms, glitch and slew rate limit. The ideal transient response of a DAC to a step is a single-time constant exponential function, which only generates an error growing linearly with the input signal, as explained in Figure 10.3. Any other transient responses give rise to errors that have no bearing on the input signal. The glitch impulse is described in terms of a picovolts times seconds or equivalent unit.

Glitches are caused by small time differences between some current sources turning off and others turning on. Take, for example, the major code transition at half scale from 011 . . . 11 to 100 . . . 00. Here, the MSB current source turns on while all other current sources turn off. The small difference in switching times results in a narrow half-scale glitch, as shown in Figure 10.24. Such a glitch, for example, can produce distorted characters in CRT display applications. To alleviate both glitch and slew-rate problems related to transients, a DAC is followed by a S/H amplifier, usually called a deglitcher. The deglitcher stays in the hold mode while the DAC changes its output value. After the switching transients have settled, the deglitcher is changed to the sampling mode. By making the hold time suitably long, the output of the deglitcher can be made independent of the DAC transient response. Thus, the distortion during transients can be circumvented by using a fast S/H amplifier. However, the slew rate of the deglitcher is on the same order as that of the DAC, and the transient distortion will still be present, now as an artifact of the deglitcher.

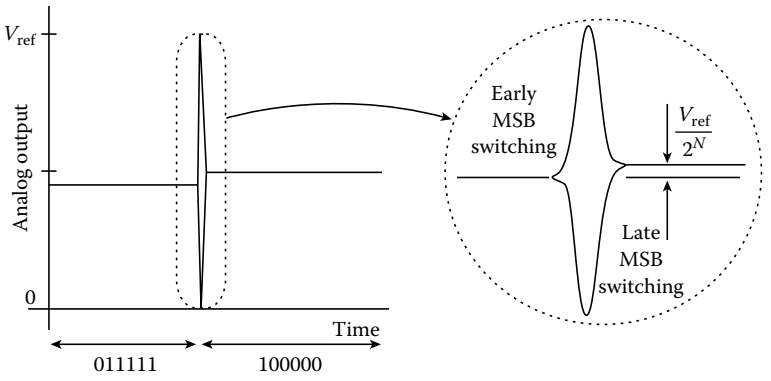


FIGURE 10.24 Glitch impulse at a major carry.

10.1.5.2 Timing Error-Word Clock Jitter

Although a DAC is ideally linear, it needs precise timing to correctly reproduce an analog output signal. If the samples do not generate an analog waveform with the identical timing with which they were taken, distortion will result, as explained in Figure 10.25. Jitter can be loosely defined as timing errors in analog-to-digital and digital-to-analog conversion. When the analog voltage is reconstructed using a DAC with timing variations in the word clock, the sample amplitudes, the ONEs and ZEROs are correct, but they come out at the wrong time. Because the right amplitude at the wrong time is the wrong amplitude, a timing jitter in the word clock produces an amplitude variation in the DAC output, causing the waveform to change shape. This in turn introduces either spurious components related to the jitter frequency or raises the noise floor of a DAC, unless the jitter is periodic. If the jitter has a Gaussian distribution with a root mean square jitter of Δt, the worst-case SNR resulting from this random word clock jitter is

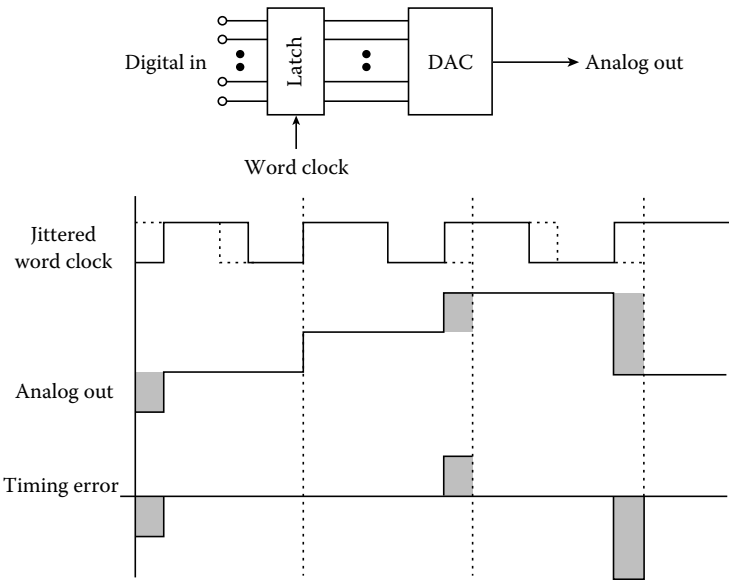


FIGURE 10.25 Word clock jitter effect.

$$\text{SNR} = -20 \times \log \frac{2\pi f \Delta t}{M^{1/2}} \tag{10.18}$$

where f is the signal frequency and M is the OSR. The OSR M is defined as

$$M = \frac{f_c}{2f_n} \tag{10.19}$$

where
f_c is the word clock frequency
f_n is the noise bandwidth

The timing jitter error is more critical in reproducing high-frequency components. In other words, to make an N-bit DAC, an upper limit for the tolerable word clock jitter is

$$\Delta t < \frac{1}{2\pi B 2^N} \left(\frac{2M}{3}\right)^{1/2} \tag{10.20}$$

where B is the bandwidth of the baseband. This implies that the error power induced in the baseband by clock jitter should be no larger than the quantization noise resulting from an ideal N-bit DAC. For example, a Nyquist-sampling 16 bit DAC with a 22 kHz bandwidth should have a word clock jitter of <90 ps.

These timing errors are caused by variation in the clock signal that controls the time when the DAC converts each digital word to an analog voltage. Usually, the term ''clock jitter'' results from everything inside a digital processor (noise from digital circuitry, inductance and capacitance of a clock bus or of a printed circuit board trace) as well as the instability of the clock source. For example, noise in digital circuitry causes the zero-crossing points to shift slightly. If the digital signal has an average slope of 10 V/μs, just 1 mV_{rms} of noise will cause 100 ps of root mean square jitter.

10.1.5.3 Voltage Reference

Ideally, the voltage reference V_{ref} is a constant temperature- and supply-independent voltage with a zero output resistance. The most common voltage reference source is a silicon bandgap voltage reference of about 1.2 V. Depending on the process used, the bandgap reference voltage has a temperature coefficient of typically 20–100 ppm/°C at room temperature when it is set to a voltage ranging from 1.2 to 1.3 V. To generate a different reference voltage other than 1.2 V, op-amps are often used to make inverting and noninverting amplifiers with trimmable feedback gains. Because the bandgap voltage for zero temperature coefficient is not clearly defined and process dependent, it is common to trim this voltage at a wafer level, and it is extremely difficult to achieve an absolute accuracy over a wide range of temperature. However, most DAC applications, except for precision instruments, do not require an absolute accuracy of the reference voltage, and the load-driving capability of the op-amp used in the voltage reference is of paramount interest.

When a DAC is used as a subblock of an ADC, the DAC has only to settle to a final value within a given clock period. In such applications it is not important how the DAC settles. However, the DAC as a stand-along device should have a single-pole response without a glitch and a slew limit. Except for the current-ratioed DAC, the voltage reference is periodically loaded and disturbed by a switched-in load at the clock rate, and the reference output should be restored immediately so that the DAC can settle fast. When disturbed, the voltage reference needs to settle with a time-constant shorter than that of the DAC with an unlimited slew rate. Figure 10.26 illustrates this situation when the voltage reference periodically refreshes the loading capacitor. If the op-amp unity-gain bandwidth is $1/\tau_0$ rad/s, the op-amp restores the output with a time constant $(1 + R_2/R_1)\,\tau_0$ of the feedback amplifier. The large capacitor C_D helps to prevent the op-amp from slewing because the voltage dip of $V_{\text{ref}} C_L/(C_D + C_L)$ decreases as C_D increases.

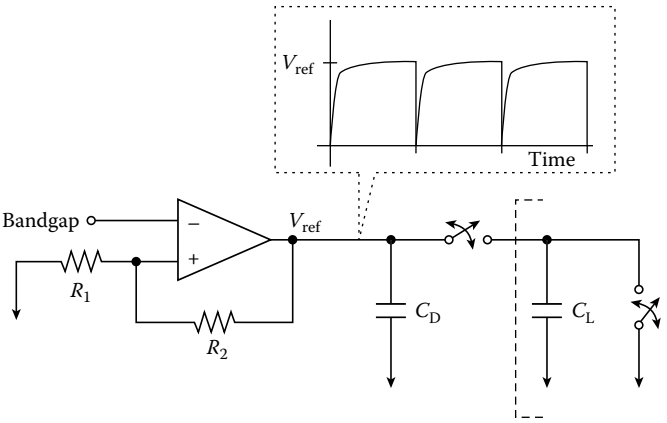


FIGURE 10.26 Voltage reference periodically charging a load capacitor.

10.1.5.4 Noise

Noise is a fundamental limit in high-resolution DACs. The resolution of a DAC is limited by a quantization noise given by Equation 10.2, unless the circuit contributes higher noise. There are many noise sources in real DACs. The DAC output is corrupted by noises directly coupled from bouncing power supplies, bias lines, and ground, as well as the noise sources such as thermal noise, flicker noise, and shot noise. To reduce the former noise sources, it is necessary to carefully separate analog and digital supplies and to eliminate sets of ground loops using a star-ground configuration. The use of a fully-differential architecture helps to reduce such coupling noises. However, the latter set of noise sources is predictable and can be reduced by a low-noise design.

The dominant noise source of a resistor-string DAC shown in Figure 10.4 is a white thermal noise of resistors. If the output resistance of the DAC is R_{out}, the root mean square output noise of the DAC is $(4kTR_{out})^{1/2}$ V/Hz$^{1/2}$, where k is the Boltzman constant and T is the absolute temperature. The thermal noise is about 4 nV/Hz$^{1/2}$ if R_{out} is 1 kΩ. The output resistance of this DAC depends on the digital input code. The worst-case R_{out} is one quarter of the total resistance of the string when the output is connected to the center tap. For lower noise, the total resistance should be minimized at the cost of high power consumption. The noise of the voltage reference appears at the output after being divided by the resistor string. The voltage noise of the output buffer, if used, is directly added to this noise.

On the other hand, the noise source of the current-ratioed DAC of Figures 10.6 and 10.7 is the current noise contributed either by the shot noise of the current source or by the parallel resistor connected to the output node. If the total output current is I_{out} and the total output shunt resistance is R_p, the root mean square output shot noise is 2 qI_{out} A/Hz$^{1/2}$ and the root mean square current noise contributed by the shunt resistor is 4 kT/R_p A/Hz$^{1/2}$, where q is 1.6×10^{-19} C. The shot noise of a 50 μA current and the current noise of a 1 kΩ shunt resistor give the same root mean square noise of 4 pA/Hz$^{1/2}$. The shunt feedback resistance of the transresistance amplifier also contributes to the total noise. If the current DAC is terminated by a low-impedance source of 50–75 Ω, the current noise is usually dominated by this termination resistor.

The main source of the capacitor-array DAC shown in Figures 10.8 and 10.9 is a well-known kT/C noise. The sampled root mean square noise voltage of the capacitor of C is $(kT/C)^{1/2}$ V without regard to the switch-on resistance. The 1 pF sampling capacitor gives a 64 μV sampled noise. This noise is evenly distributed over the Nyquist band (half the sampling frequency), and puts a lower limit on the smallest signal the DAC can handle. Furthermore, the reference and ground noises are divided by the C divider and appear at the output of the DAC. When applied to the oversampling converters either in delta-sigma modulators or in a 1 bit DAC, shown in Figures 10.22 and 10.23, the kT/C noise in the signal band is lower than the Nyquist-rate converters by the OSR of M given by Equation 10.19.

10.1.6 Low-Spurious DAC Design Examples

Modern broadband digital communication systems such as cable modem, cellular baseband, and wireless local area network consist of multiple channels. For high bit rate transmission of 54 Mbit/s over 20 MHz bandwidth, the RF standards such as IEEE 802.11a and Hyperlan2 use the orthogonal frequency division multiplexing scheme that divides the signal band into 52 of 300 kHz narrowband channels. It is challenging to preserve signal integrity of such a complex waveform. Multiple channels are mixed in many different ways when analog signal processing is nonlinear. For example, the multichannel spectrum is perfect when digitally generated, but it is degraded while being converted into analog waveform due to the DAC nonlinearity. To generate an accurate multichannel spectrum without mixing, DAC should be linear both in static and dynamic performance. High spurious-free dynamic range is the most important for frequency-domain applications. Recently, a few techniques such as spatial averaging [13–15], self-trimming [16], and dynamic linearity enhancement [16,17] are reported.

10.1.6.1 Spatial Averaging for Static Linearity

DAC static linearity measured in terms of DNL and INL depends on how DAC components are arranged. In most stand-alone DACs, an array of current sources is predominantly used. It is well known that the binary-weighted array is simple to decode but exhibits poor DNL, while the thermometer-coded array is complex to decode but exhibits good DNL. In fact, the complexity of the thermometer-coded current DAC grows exponentially. Therefore, in typical applications above 8–10 bits, the MSB part is coded in thermometer while the LSB part is coded in binary. The thermometer-coded DAC array occupies a large area, and current source matching is limited by overall gradient effect. In a large thermometer-coded array, systematic and graded errors are far greater in effect than random device mismatch errors. They result from many factors such as edge effect, current source output resistance, supply voltage drop, thermal gradient, and process gradients of doping and thickness across the die, etc. In integrated circuit design, the common-centroid layout scheme has been used to effectively match devices when limited by the gradient effect.

Since the thermometer-coded array is physically large, a two-dimensional common-centroid scheme should be used. To improve INL, the two-dimensional array can be split into four quadrants biased separately using common-centroid bias circuits [13]. To improve INL further, each current source can be split into four small slices connected in parallel, and each slice is placed in one of the four different quadrants [14]. This results in the first-order spatial averaging of the systematic and gradient errors. In general, the gradient effect is first-order, but the second-order effect limits DAC INL performance at 14 bit level or higher. Further partitioning the unit current source into 16 small slices connected in parallel can remove the residual gradient effect [15]. Unlike the first-order centroid scheme, the second-order spatial averaging scheme uses four common-centroid slices placed in four different quadrants. In theory, the matching improves as the array is split into many smaller pieces and the common-centroid scheme is used in each small piece. However, the complexity grows significantly, and the operating speed is inevitably reduced.

10.1.6.2 Self-Trimming for Static Linearity

The two-dimensional geometric averaging is very costly in terms of chip area and the operating speed as the array gets larger. Calibration techniques as discussed are effective in overcoming device mismatch and gradient effects in IC process. One recent trend is self-trimming. The concept is to adjust the current source values electronically like laser trimming. Although laser trimming is a one-time adjustment, the self-trimming is adaptive and continuously engaged. A trimmable floating current source is shown in Figure 10.27, which can be used as one MSB current source in the thermometer-coded array [16]. The floating current source made of transistors N_4 and P_2 is switched from the top side by transistors N_6 and N_7. When the current source needs trimming, the bottom transistor N_2 is turned on, and the current is switched to the measuring resistor R_2. This arrangement of trimming is transparent to the normal

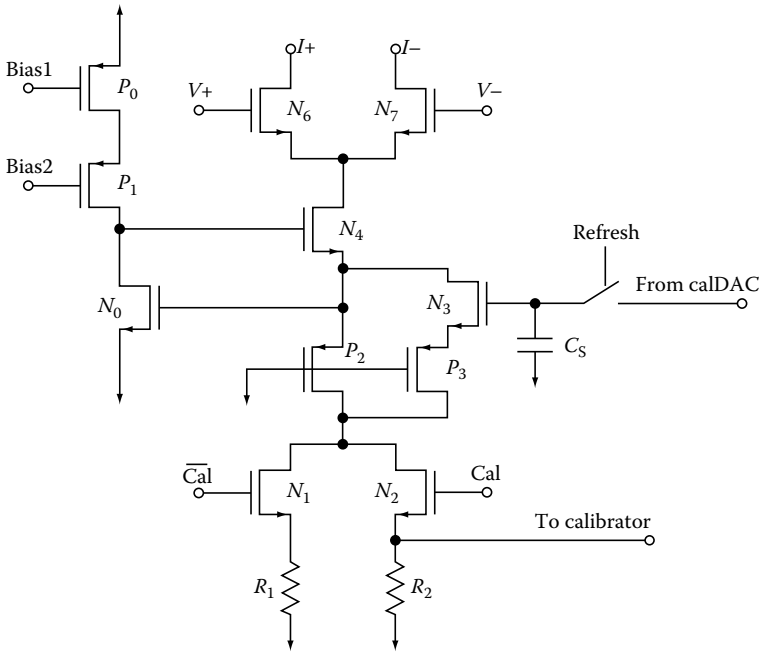


FIGURE 10.27 Trimmable floating current source.

operation and is performed in background. An oversampling calibrator based on delta-sigma modulation repeatedly monitors the voltage across R_2, accurately compares it with the preset reference voltage, and constantly feedbacks the error signal that periodically updates the capacitor C_s to adjust the current source.

10.1.6.3 Dynamic Linearity Enhancement

The static linearity up to over 14 bits can be achieved either with spatial averaging or with self-trimming, but the dynamic linearity heavily depends on how the DAC output settles. For example, the wireless baseband spectrum covers over 20 MHz bandwidth, but generating such a wideband signal with high static and dynamic linearity is not a trivial task. Furthermore, in scaled CMOS, devices are more nonlinear and low supply voltage often limits the DAC linearity. In standard current DACs, each current source is switched to the output at different times due to clock skew, and adds parasitic capacitance to the output node, which results in code-dependent output settling. The clock skew and the code-dependent settling are most prominent among many factors limiting dynamic DAC linearity. Isolating the output node from current sources and synchronizing the code transitions are simple solutions for these, but the clock skew is difficult to deal with high precision. It is often suggested that the DAC's dynamic linearity can be improved if its output is sampled and held using a deglitcher. In fact, the deglitcher is functionally a track and hold circuit, but it is also difficult to implement at high frequencies with high precision.

The return-to-zero (RZ) scheme solves the deglitcher problem as explained in Figure 10.28a [17]. While the error in the track and hold stays for the half period, the error in the RZ exponentially decays. The logic behind this is that rather than sampling analog voltage accurately, it is easier to let the output keep on changing exponentially. Exponential settling is a linear process. The same effect as RZ can be achieved in a simpler manner, called track and attenuate [16]. Rather than resetting the DAC output, just attenuating it achieves the same goal. Figure 10.28b illustrates the DAC output stage performing the track and attenuate function. The differential DAC output current comes into the folded-cascode node V_x, which is isolated from the output node. The output nodes track the signal during the half clock period,

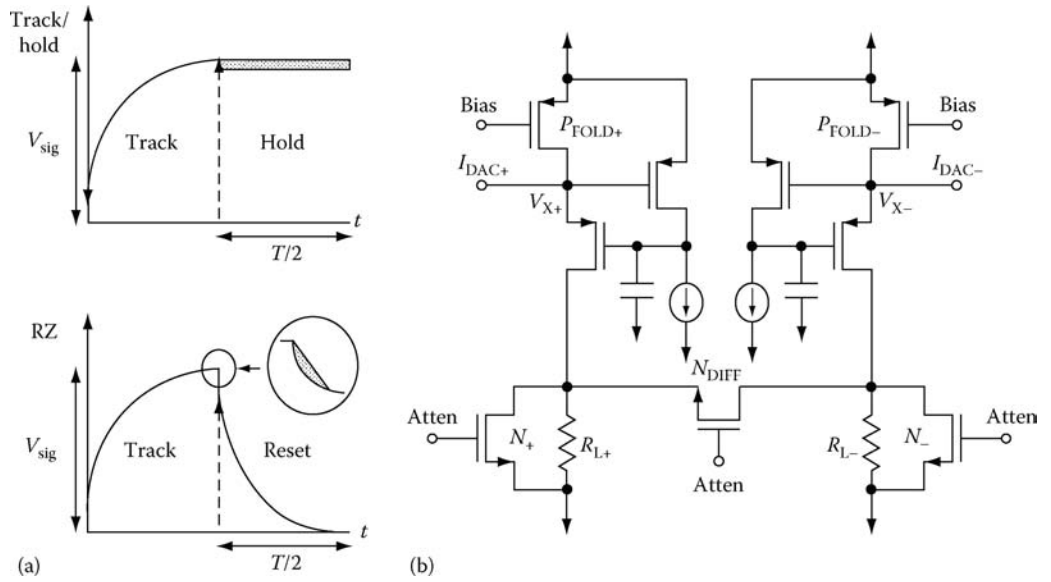


FIGURE 10.28 (a) Track and hold vs. RZ and (b) track/attenuate output stage.

but are bypassed by three transistors N_+, N_-, and N_{DIFF} for attenuation during the remaining half clock period. The difference of the RZ DAC from the conventional DAC is that the output spectrum nulls at multiples of twice the sampling frequency, which in turn provides less attenuation in the passband.

References

1. D. M. Freeman, Slewing distortion in digital-to-analog conversion. *J. Audio Eng. Soc.*, 25, 178–183, 1977.
2. R. B. Craven, An integrated circuit 12-bit D/A converter. *Dig. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, pp. 40, 41, 1975.
3. D. T. Comer, A monolithic 12-bit DAC. *IEEE Trans. Circuits Syst.*, CAS-25, 504–509, 1978.
4. J. M. McCreary and P. R. Gray, All-MOS charge redistribution analog-to-digital conversion techniques I. *IEEE J. Solid-State Circuits*, SC-10, 371–379, 1975.
5. R. J. Van de Plassche, Dynamic element matching for high accuracy monolithic D/A converters. *IEEE J. Solid-State Circuits*, SC–11, 795–800, 1976.
6. M. J. M. Pelgrom and M. Roorda, An algorithmic 15-bit CMOS digital-to-analog converter. *IEEE J. Solid-State Circuits*, SC-23, 1402–1405, 1988.
7. D. W. J. Groeneveld, H. J. Schouwenaars, H. A. H. Termeer, and C. A. A. Bastiaansen, A self-calibration technique for monolithic high-resolution D/A converters. *IEEE J. Solid-State Circuits*, SC-24, 1517–1522, 1989.
8. J. C. Candy and A. N. Huynh, Double interpolation for digital-to-analog conversion. *IEEE Trans. Commun.*, 34, 77–81, 1986.
9. H. S. Lee, D. A. Hodges, and P. R. Gray, A self-calibrating 15 bit CMOS A/D converter. *IEEE J. Solid-State Circuits*, SC-19, 813–819, 1984.
10. S. H. Lee and B. S. Song, Digital-domain calibration of multistep analog-to-digital converters. *IEEE J. Solid-State Circuits*, SC-27, 1679–1688, 1992.
11. T. Hayashi, Y. Inabe, K. Uchimura, and T. Kimura, A multistage delta-sigma modulator without double integration loop. *Dig. IEEE Int. Solid-State Circuits Conf.*, pp. 182, 183, 1986.

12. W. L. Lee and C. G. Sodini, A topology for higher-order interpolative coder. *Proc. Int. Symp. Circuits Syst*., pp. 459–462, 1987.
13. C. H. Lin and K. Bult, A 10-b, 500-MSample/s CMOS DAC in 0.6 mm^2. *IEEE J. Solid-State Circuits*, SC-33, 1948–1958, 1998.
14. J. Bastos, A. M. Marques, M. S. J. Steyaert, and W. Sansen, A 12-bit intrinsic accuracy high-speed CMOS DAC. *IEEE J. Solid-State Circuits*, SC-33, 1959–1969, 1998.
15. G. A. M. Van der Plas, J. Vandenbussche, W. Sansen, M. S. J. Steyaert, and G. G. E. Gielen, A 14 bit intrinsic accuracy Q^2 random walk CMOS DAC. *IEEE J. Solid-State Circuits*, SC-34, 1708–1718, 1999.
16. A. R. Bugeja and B. S. Song, A self-trimming14-b100-MS/s CMOS DAC. *IEEE J. Solid-State Circuits*, SC-35, 1841–1852, 2000.
17. A. R. Bugeja, B. S. Song, P. L. Lakers, and S. F. Gillig, A 14-b, 100MS/s CMOS DAC designed for spectral performance. *IEEE J. Solid-State Circuits*, SC-34, 1719–1732, 1999.

10.2 Analog-to-Digital Converters

Ramesh Harjani

10.2.1 Introduction

With the increased complexity possible in modern-day integrated circuits, analog-to-digital converter (ADC) and digital-to-analog converter (DAC) have become ubiquitous components of mixed-signal integrated circuits. ADCs transform an analog signal, V_A, into an N-bit digital representation, V_d. Such a converter is said to have a resolution of N bits. The digital signal V_d is an approximation of the original analog signal, V_A, and the maximum error during this conversion process for an N-bit converter is equal to $1/2^N$ of the full-scale value. This error is called the quantization error.

A number of topologies exist for ADC. They can be classified as Nyquist rate converters or oversampled converters. Nyquist rate converters sample the input at the minimum sampling rate, i.e., two times the maximum signal frequency. As the name implies, oversampled converters take more samples than is mandated by the Nyquist criterion to generate extremely high resolution. Nyquist rate converters are usually classified as (1) high-speed, (2) medium-speed, and (3) high-resolution converters. However, the different architectures are better categorized by evaluating the number of clock cycles they use to perform the analog-to-digital conversion. For example, for N bits of resolution a high-speed converter performs the conversion in one or two clock cycles, a medium-speed converter performs the conversion in N clock cycles, and a high-resolution converter performs the conversion in 2^N clock cycles. Thus, we classify them here as 1-, N-, and 2^N-clock converters.

Before describing the details of the various ADC architectures we discuss some of the performance characteristics of ADCs in general and describe some test techniques that are used to measure these characteristics.

10.2.1.1 ADC Test Techniques

ADCs are primarily tested using parametric techniques. That is, a key set of parameters that characterize an ADC are verified. An ADC is characterized by its static and dynamic performance. In static performance, we are primarily concerned about the linearity of the I–O transfer characteristics, and in dynamic performance we are concerned about the operation of the converter at full operating speed.

Figure 10.29 shows the transfer characteristics for an ideal 3 bit converter. The dashed line shows the transfer characteristics of an infinite precision converter and the bold line shows the transfer characteristics of a 3 bit version. We note that the least significant bit (LSB) value is equal to $1/2^N$ of the full-scale value. Figures 10.30 through 10.33 show examples of the static performance characteristics of an ADC.

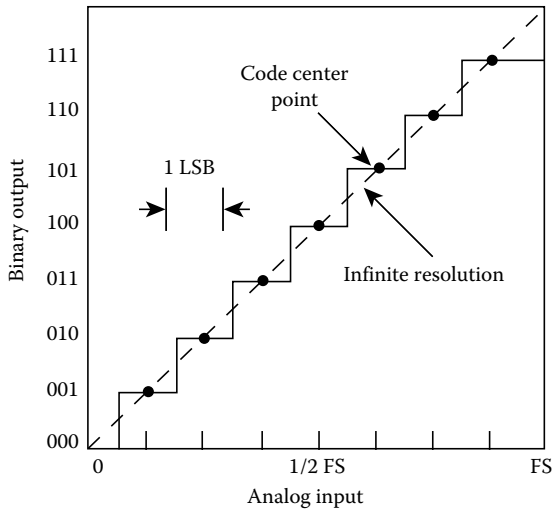


FIGURE 10.29 Transfer characteristics for an ideal 3 bit ADC.

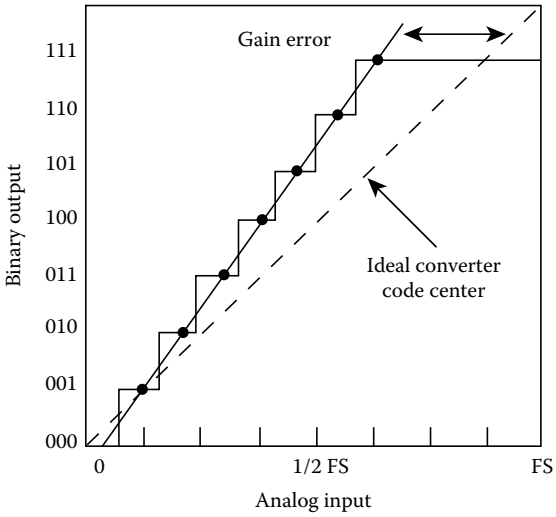


FIGURE 10.30 Gain error in an ADC.

A gain error is said to be present when the maximum digital value does not correspond to the full-scale analog value. An offset error corresponds to a horizontal shift in the transfer characteristics. The integral nonlinearity error specifies the maximum deviation of the transfer characteristics from the ideal code center. Differential nonlinearity specifies the deviation of each stepsize from $1/2^N$ of the full-scale value.

Dynamic ADC performance characteristics include signal-to-noise ratio (SNR), effective bits, aperture errors, and input signal bandwidth. During full-speed operation some additional errors become evident because of the finite settling time and bandwidth limitations of the circuits within the ADC. The SNR and effective bits are dynamic ways of measuring the minimum resolution and errors in the transfer characteristics. Different terms are used to specify similar performance parameters largely because different measurement techniques are used to generate them. The input signal bandwidth specifies the maximum frequency of the input signal.

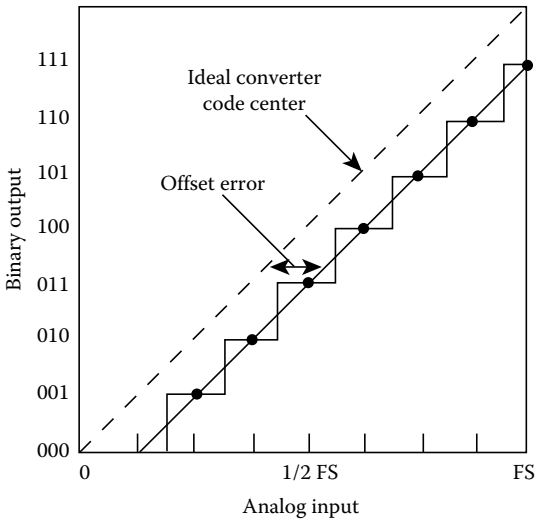


FIGURE 10.31 Offset error in an ADC.

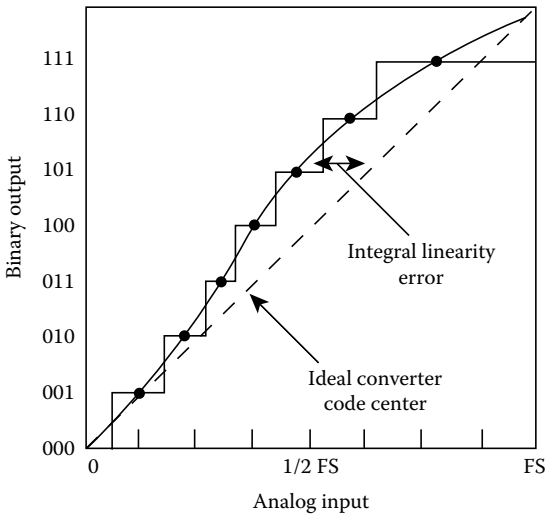


FIGURE 10.32 Integral nonlinearity error.

Because of the large number of performance specifications for ADCs, quite a few techniques are used to test them as well. These testing techniques also reflect the different types of performance characteristics, i.e., static and dynamic. The more traditional methods of testing ADC are primarily concerned with checking the static characteristics of these converters. Examples of such test techniques include analog difference signal methods, crossplot methods, and servo loop code transition measurement methods [3]. Dynamic techniques to test linearity are based on using a well-known and near-full-scale input signal and evaluating the output code probability over a large input sample size. This technique is called the ''code density test'' or the ''histogram test.'' The output code probability for a linear ramp input is uniform for all codes. However, it is difficult to generate extremely accurate ramp signals at high speed, therefore, sine-wave signals are used. Unfortunately, with a sine wave the output code probability is no longer uniform but is instead ''cusp-shaped''. An estimate of the differential nonlinearity can be

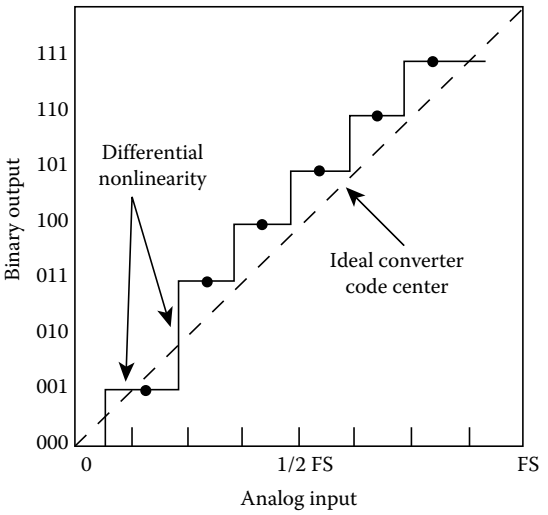


FIGURE 10.33 Differential nonlinearity error.

generated by evaluating the difference between the expected code probability and measured values. The code density test requires a large number of data samples in the range of several hundreds of thousands. Additionally, a small number of large magnitude errors are easily masked in this technique. More recently, frequency domain-based techniques have been used to measure the harmonic content of the converted signal to provide an estimate of the SNR and effective number of bits. In this technique, a discrete time Fourier transform (DTFT) is performed on the output data sequence and is used to measure the data converter performance characteristics. Although the number of data samples required are fewer than the code density approach, this approach still requires a few thousand data samples. Additionally, proper windowing functions and synchronized sampling may be required to reduce spectral leakage.

Data converters are relatively time-consuming to test. However, rather than explain all the techniques that can be used to test data converters, we concentrate on a single method, the crossplot method, to provide the reader with some insight into the complexity of testing ADCs. For additional details about the other methods, readers are referred to [3]. A block diagram for the crossplot method is shown in Figure 10.34. In the crossplot technique, the output of the lowest 2 or 3 bits of an ADC are fed to the

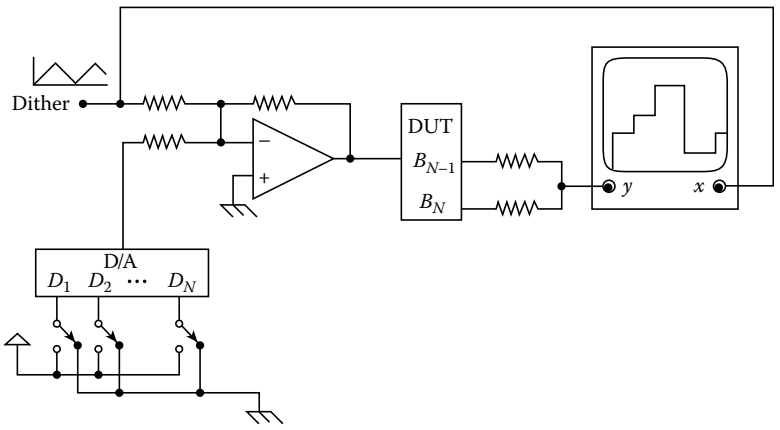


FIGURE 10.34 Crossplot technique used to test ADC static characteristics. (From Demler, M.J., *High-Speed Analog-to-Digital Conversion*, Academic Press, New York, 1991, 162.)

Y input of an oscilloscope and a separate triangular dither signal is fed to the *X* input of the oscilloscope. The input to the ADC is generated using a discrete summing amplifier that adds the output of a DAC, which is itself swept through its input space at a much slower rate, and the dither signal, mentioned earlier. This technique generates a staircase waveform on the oscilloscope. Therefore, the linearity of the last 2 or 3 bits around a fixed bias voltage, which is generated by the DAC, can be seen easily on an oscilloscope. The primary advantage of this technique is that it uses fairly inexpensive equipment. However, it is extremely time consuming to evaluate all 2^N combinations and the technique provides no information about the integral linearity, offset, and gain errors. In general, techniques that are used to test ADCs are fairly complex and either require a large sample size, i.e., high cost, or use extremely complex equipment to perform the tests.

Having described the basic characteristics of ADC, we now describe of the different ADC architectures in detail. We first consider Nyquist rate converter topologies and then describe oversampled converters.

10.2.2 Nyquist Rate Converters

As mentioned earlier, Nyquist rate converters are classified as 1-clock converters, *N*-clock converters, and 2^N-clock converters. Examples of 1-clock converters include the flash architecture, the pipelined architecture [11,12], and the voltage folding architecture. Examples of *N*-clock converters include the successive-approximation architecture [9] and the algorithmic architecture [6]. Examples of 2^N-clock converters include the single-slope and dual-slope architectures [4].

10.2.2.1 1-Clock Converters

Flash converters. Flash or parallel converters are the highest rate converters and are sometimes also called ''video rate converters'' because they operate at rates necessary for video signals. Parallel converters are O(1)-clock converters in that they require one or two clock cycles to perform an *N*-bit conversion. The basic principle of operation is fairly simple and is easily explained with the help of Figure 10.35. This figure is a block diagram for a 3 bit flash converter. The input is compared to $(2^3 - 1)$ comparators. The comparison voltage for the *i*th comparators is set to be equal to $(iV_{FS})/(2^3)$, where V_{FS} is the full-scale voltage. The resulting outputs are then converted from thermometer code to binary code. In general, the time required to perform the analog-to-digital conversion is equal to the comparator resolution time plus the time taken to perform the digital code conversion, i.e., one clock cycle. The reference voltages for the comparators are usually generated using an equally spaced resistor string that is offset by $R/2$, as shown in Figure 10.36. The primary advantage of providing the $R/2$ offset is that it allows the code center to pass through the origin. The resolution of integrated MOS flash converters is limited to approximately 8 bits. This is primarily due to the matching constraints placed on the resistors and also because of the exponential increase in the number of components ($\approx 2^N$) required for higher resolution. The speed for MOS flash converters is usually limited to about 50 MHz. However, if higher speed operation is required, then the process of time interleaving may be employed. Here, *M* (*N*-bit) ADCs are operated in parallel, but each is delayed somewhat, as shown in Figure 10.37 [1]. By starting the conversion operation for A/D_2 before the operation of A/D_1 is complete allows the complete interleaved converter to operate at *M* times the speed of each individual converter. Although the technique allows for higher operating speed, the number of parallel paths has increased. Therefore, due to size limitations the resolution is usually lower than for simple flash converters.

Subranging converters. The exponential area penalty ($\alpha\ 2^N$) for flash converters can be mitigated by using subranging techniques without incurring severe speed penalties. Here, instead of deciding all *N*-bits at one time, only a subset of these is decided during the first clock phase, and the rest are decided in the following clock phases. If two clock phases are used to decide all *N*-bits, this is called a two-step subranging converter topology.

A block diagram for a two-step subranging converter is shown in Figure 10.38. In this converter topology the α most significant bits (MSBs) are decided by the coarse ADC, the results of which are then

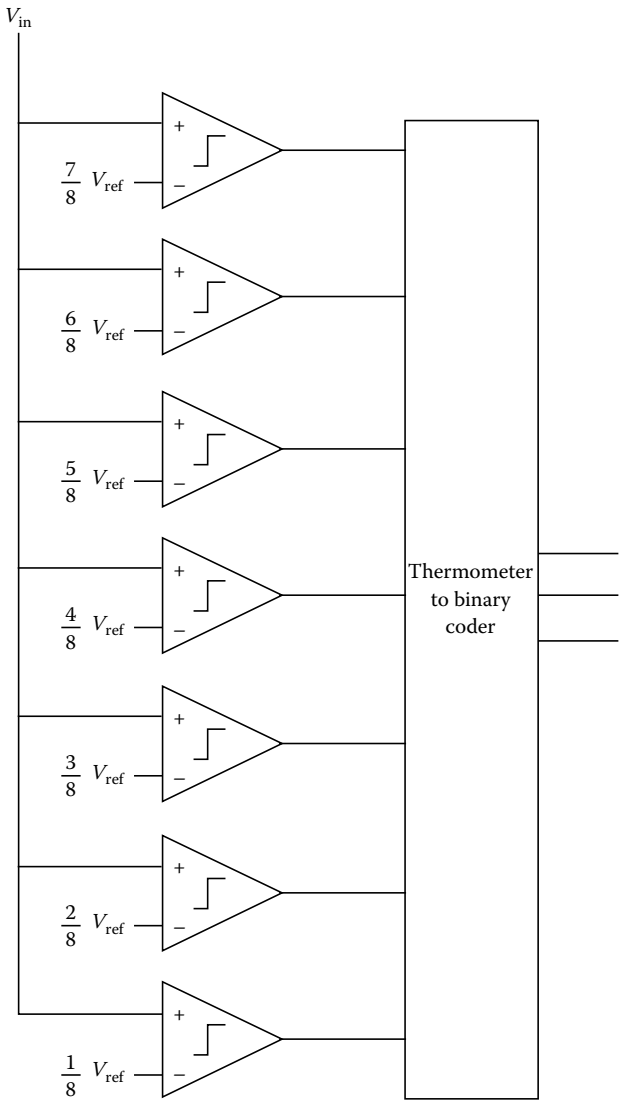


FIGURE 10.35 Three bit flash converter block diagram.

passed on to an α-bit DAC. The output voltage of the DAC is subtracted from the input signal. The resulting voltage is amplified 2α times and then a $(N-\alpha)$-bit fine converter is used to resolve the remaining bits. The entire process—coarse conversion, digital-to-analog subtraction, amplification, and fine conversion—is performed in one clock period. The coarse and fine converters are usually implemented as simple flash converters. Therefore, the operating speed of a pure subranging converter is at least twice as slow as that of a flash converter. However, the total number of comparators (and associated circuits) is reduced from 2^N to $2\alpha + 2^{N-\alpha}$. The savings in area can be substantial. Usually, the value of α is selected to be roughly equal to $N/2$. For example, for $N = 10$ and selecting $\alpha = 5$, the number of comparators reduce from $2^{10} = 1024$ to $2^6 = 64$, which is a savings of 93.75%. This subranging process can be extended further to a larger number of sequential stages; however, the total propagation delay usually limits the operation speed fairly significantly.

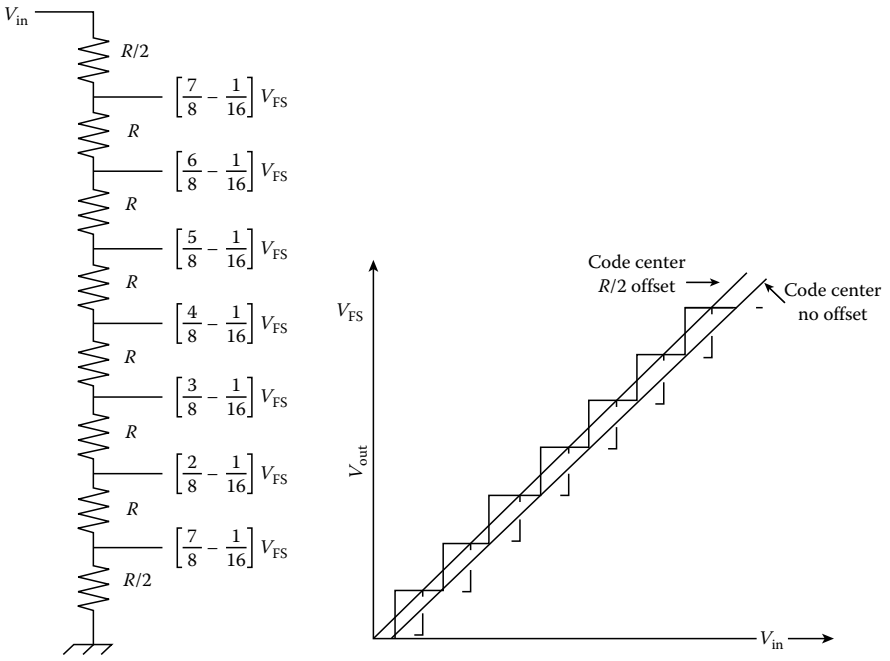


FIGURE 10.36 Flash converter 1/2 LSB offset.

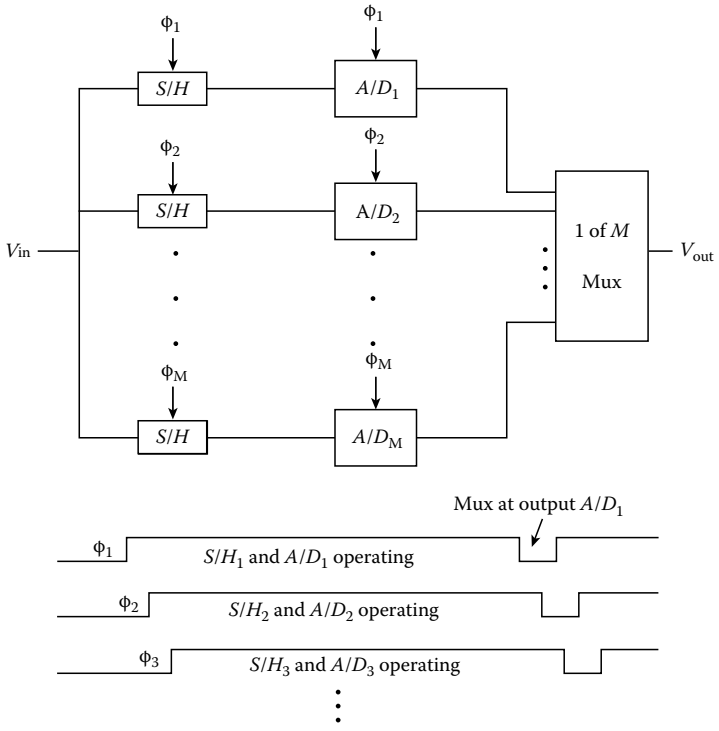


FIGURE 10.37 Time interleaved converter block diagram.

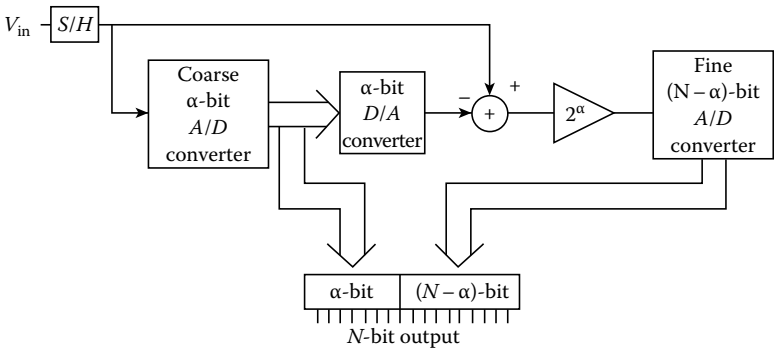


FIGURE 10.38 Subranging converter block diagram.

Pipelined converters. Instead of operating both the coarse and fine converter in the same clock period, a sample-and-hold (S/H) could be added to the circuit in Figure 10.38, such that when the fine converter is resolving the lower $(N-\alpha)$ bits for the first input sample the coarse converter can begin resolving the upper α bits for the next input sample. This kind of converter can be made to operate at speeds comparable to flash converters and is called a pipelined converter. A block diagram for a two-stage pipelined converter is shown in Figure 10.39. Because the α MSBs are decided during the previous clock period, the results from the fine and coarse converters need to be synchronized by delaying the output of the coarse converter by a single clock period. As with the subranging converter the number of comparators have decreased from 2^N to $2^{N-\alpha}+2\alpha$.

Unlike the subranging topology, the pipeline methodology can easily be extended to N sequential stages, each resolving only 1 bit, because of the sample-and-held circuits between each of the stages. The block diagram for such an N-step converter is shown in Figure 10.40. In this figure, note that only one comparator is used per stage and the interstage gain block (also performing the S/H function) has a gain of two. Once again for synchronization purposes, shift registers are used. This pipelined converter [11,12] operates as follows. The input is first compared to see if it is greater than V_{ref}. If the input is greater than V_{ref}, then we set the MSB bit. If the MSB is set, then V_{ref} is subtracted from the input voltage. However, if the MSB is not set, nothing is done. The resulting voltage is then doubled. During the next clock cycle, the resulting voltage is once again compared to V_{ref} to give the MSB-1 bit. While the second stage generates the second MSB for the first input, the first stage processes a new input. When the pipeline is filled, N inputs are being processed simultaneously. Several shift registers are used to ensure that all the data bits

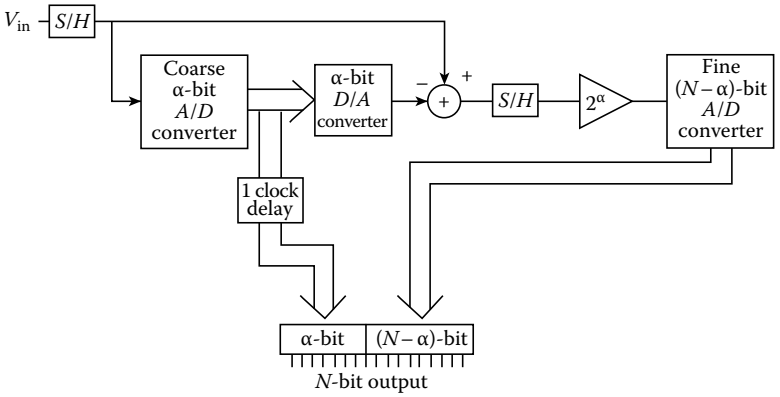


FIGURE 10.39 Two-stage pipelined converter block diagram.

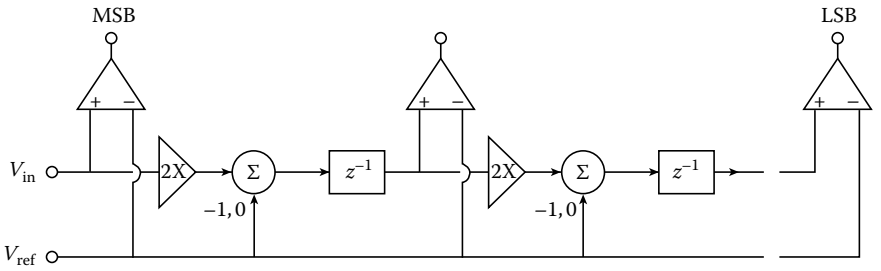


FIGURE 10.40 *N*-stage pipelined converter.

corresponding to a single input are output simultaneously. This process continues until we finally generate the LSB. The analog output of the *i*th stage can be written as

$$V_i = 2[V_{i-1} - b_i V_{\text{ref}}]z^{-1} \tag{10.21}$$

where b_i is equal to 1 if the *i*th bit is set and equal to 0 otherwise.

Figure 10.41 is a simple circuit realization for a pipelined ADC [13]. Other more advanced topologies [12] reduce some of the problems associated with this topology, but operate on the same principle. Only one stage of an *N*-bit converter is shown in this figure. In this first stage, the amplifier A_1 performs the comparison function and the amplifier A_2 performs the 2× multiplication and the conditional subtraction function. Both amplifiers, along with the associated capacitors and switches, also implement a S/H function. Let us first concentrate on the comparison function. For a sufficiently large gain, during ϕ_1 the capacitor C_h is charged to $V_a - V_{off}$, where V_{off} is equal to the amplifier offset voltage. During ϕ_2 the reference voltage, V_{ref}, is connected to the positively charged terminal of capacitor C_h and the feedback around the amplifier is removed. During this period the amplifier A_1 acts as a comparator and compares the voltage $[V_{ref} - (V_a - V_{off})]$ against V_{off}; i.e., the effect of the amplifier offset voltage is completely

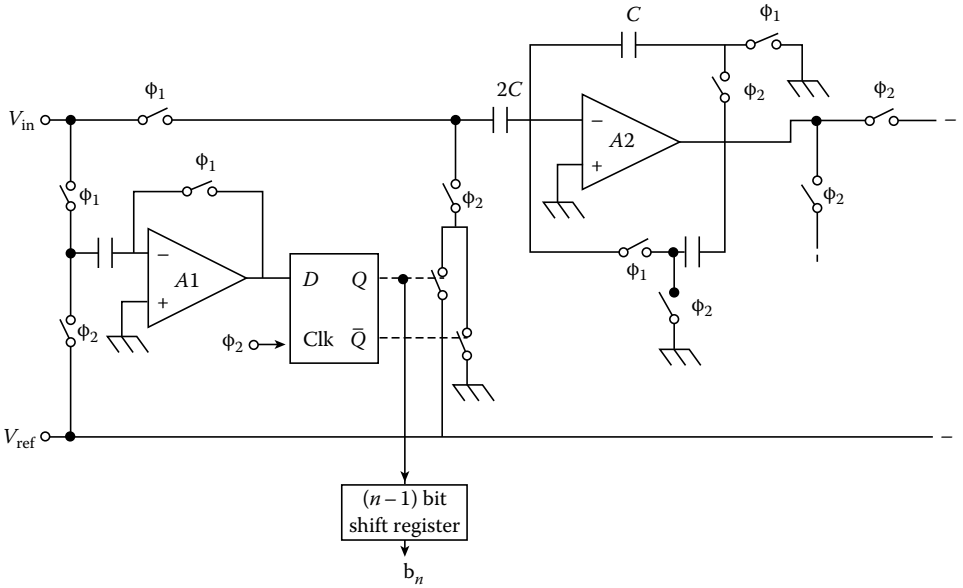


FIGURE 10.41 Circuit realization of an *N*-stage pipelined converter.

canceled. The preceding discussion is valid only for sufficiently large gain. However, if the gain of amplifier A_1 is not sufficiently large then the charge transfer is not complete and an error is introduced in the comparison; i.e., the effective comparison is not with V_{ref} but with $V_{\text{ref}} + \Delta V$. The error in the comparison is dependent on the input signal voltage. Note that the first stage samples the input at ϕ_1 while the next stage samples the input at ϕ_2. Likewise, the third stage samples the input at ϕ_1 and the fourth stage samples the input at ϕ_2, and so forth.

Digital-error correction in multistep converters. Pipelined and subranging converters that have an interstage gain >1 can utilize digital-error correction [14] to improve linearity. We shall illustrate the principle of digital-error correction with the help of a 4 bit two-stage pipelined converter. Each of the stages resolves 2 bits. Digital-error correction can be used to correct for linearity errors in all except the last stage. Additionally, it is unable to correct for digital-to-analog linearity and op-amp settling time errors. Therefore, for our two-stage example we will only be able to correct for errors in the first stage and we shall assume an ideal DAC.

Figure 10.42 is the block diagram for the 4 bit two-stage pipelined converter without digital-error correction. This circuit is a 4 bit version of Figure 10.39. The input signal is sample-and-held by S/H_1. The course MSB bits for the overall converter are generated by the first-stage subconverter (A/D_1). The analog value corresponding to these bits is then generated by the first-stage digital-to-analog subconverter (D/A_1). The difference between the input signal and the digital-to-analog output is called the residue. This residue is amplified by the interstage gain stage ($G = 4$) and passed on to the second-stage analog-to-digital subconverter. The second-stage analog-to-digital subconverter (A/D_1) then generates the lower 2 bits. Because the second stage is working on the signal after one clock delay, an intermediate delay stage is added to synchronize the outputs of the two stages.

The residue for an ideal converter varies from $-\frac{1}{2}$ LSB to $+\frac{1}{2}$ LSB of the first-stage subconverter resolution, as shown in Figure 10.43. In the case of nonlinearity in the first subconverter the residue will have excursions above and below the $\pm\frac{1}{2}$ LSB value, as shown in Figure 10.44. For an ideal digital-to-analog conversion, the residue corresponding to each digital code is still accurate and no data have been lost as yet. In the traditional pipelined converter shown in Figure 10.42 any residue value from the first converter that is greater than $\pm\frac{1}{2}$ LSB of the first stage saturates the second stage and produces errors.

If, however, we change the overall converter topology such that the resolution of the second subconverter is increased by 1 bit, i.e., we double the number of levels, and reduce the interstage gain by half, then we can detect when the residue exceeds the $\pm\frac{1}{2}$ LSB levels and correct for its effect digitally. Figure 10.45 is a block diagram for the pipelined converter in Figure 10.42 with digital-error correction. Whenever the residue from the first stage exceeds $+\frac{1}{2}$ LSB it implies that the digital output of the first

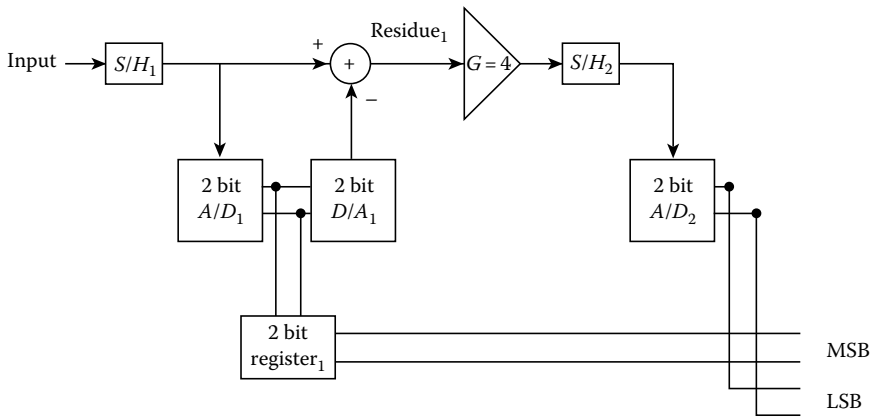


FIGURE 10.42 Four bit, two-stage, pipelined converter block diagram without error correction.

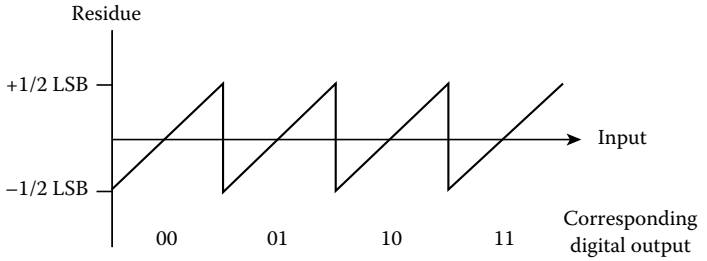


FIGURE 10.43 Ideal subconverter residue.

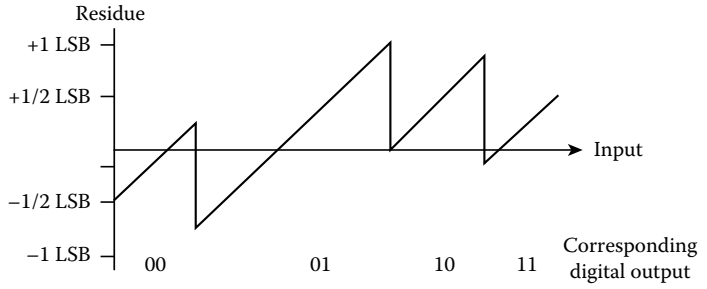


FIGURE 10.44 Nonideal subconverter residue.

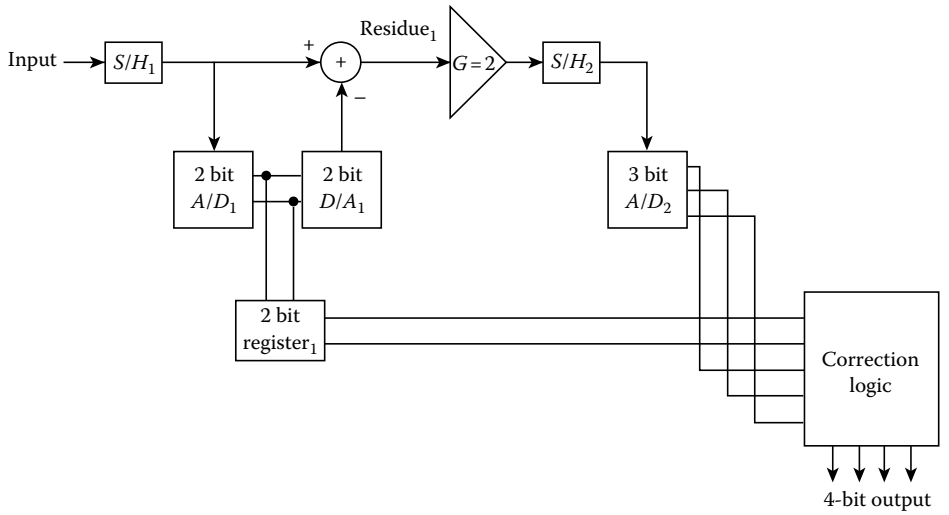


FIGURE 10.45 Four bit, two-stage, pipelined converter block diagram with error correction.

stage subconverter is too small. Likewise, whenever the residue is less than $-\frac{1}{2}$ LSB it implies that the digital output of the first stage is too large. By adding a $\frac{1}{2}$ LSB offset at the input of the first analog-to-digital subconverter and at the output of the first DAC the input to the second subconverter for an ideal first subconverter is restricted between $\frac{1}{4}$ full-scale and $\frac{3}{4}$ full-scale. Any excursion outside this region implies an error in the first analog-to-digital subconverter. The approximate value for this error is measured by the second-stage subconverter and is then subtracted digitally from the final value.

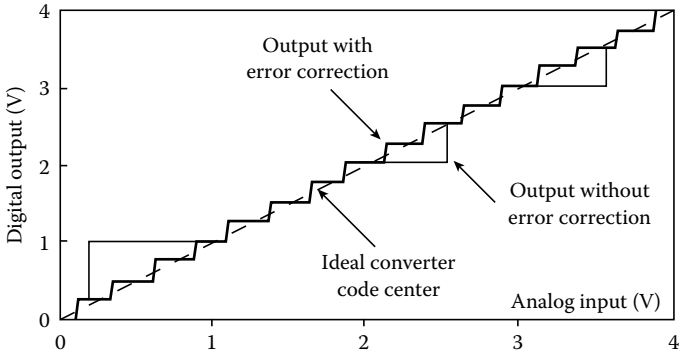


FIGURE 10.46 Digital-error correction simulation results.

Nonlinearities from the second stage are not corrected via this scheme; however, as the interstage gain is greater than 1 the effect of the nonlinearities in the second stage will have a much lower effect than those resulting from the first stage. Nonlinearities in the DAC can be reduced substantially by utilizing reference feedforward compensation [15]. Here, the reference for the second stage changes dynamically and is obtained by amplifying the first-stage digital-to-analog subconverter segment voltage that corresponds to the most current digital output code of the first-stage analog-to-digital subconverter. Figure 10.46 presents the simulation results for the 4 bit two-stage pipelined converter with and without digital error correction. For purposes of clarity the second-stage subconverter is made ideal. In a real converter some nonlinearity would still exist, but would be limited to that introduced by the last stage. Traditionally, even though the resolution of the first subconverter is only 2 bits, it needs to be linear to the overall converter resolution. Digital-error correction can reduce the linearity requirements such that it is commensurate with its resolution.

10.2.2.2 *N*-Clock Converters

Both the successive approximation and algorithmic analog-to-digital topologies require N clock cycles to perform an N-bit conversion. They both perform 1 bit of conversion per clock cycle. The successive approximation converter is a subclass of the subranging converter, in which during each clock cycle only 1 bit of resolution is generated. The algorithmic converter is a variation of the pipelined converter, in which the pipeline is folded back into a loop. Both topologies essentially perform a binary search to generate the digital value. However, in the case of the successive approximation converter the binary search is performed on the reference voltage, while in the case of the algorithmic converter the search is performed on the input signal.

Successive approximation converters. A block diagram for the successive approximation converter is shown in Figure 10.47. Because the conversion requires N clock cycles a S/H version of the input signal is provided to the negative input of the comparator. The comparator controls the digital logic circuit that performs the binary search. This logic circuit is called the successive approximation register (SAR). The output of the SAR is used to drive the DAC that is connected to the positive input of the comparator.

During the first clock period, the input is compared with the MSB, i.e., the MSB is temporarily raised high. If the output of the comparator remains high, then the input lies somewhere between 0 and $V_{ref}/2$, and the MSB is reset to 0. However, if the comparator output is low, then the input signal is somewhere between $V_{ref}/2$ and V_{ref}, and the MSB is set high. During the next clock period the MSB-1 bit is evaluated in the same manner. This procedure is repeated such that at the end of N clock periods all N-bits have been resolved. Figure 10.48 is the binary search procedure for a 4 bit converter and shows the comparator output sequence that corresponds to an input equal to 72% of V_{ref}.

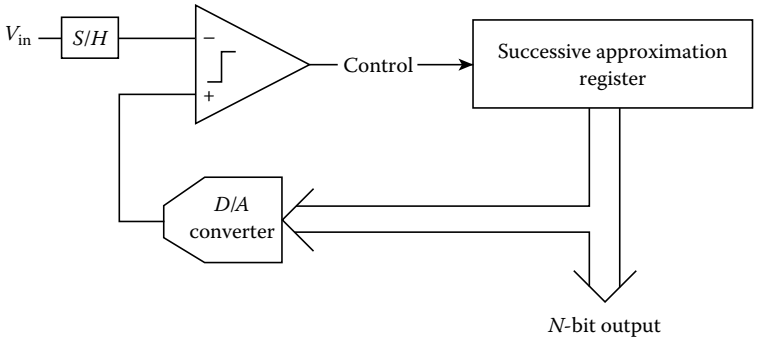


FIGURE 10.47 Successive approximation converter block diagram.

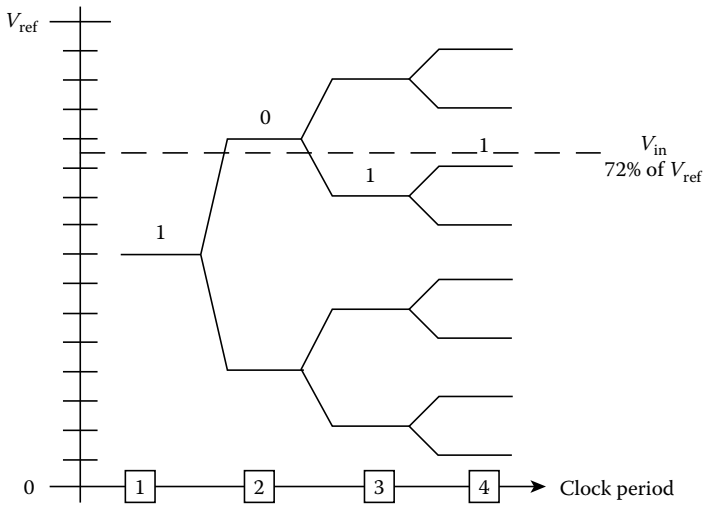


FIGURE 10.48 Binary search process for successive approximation.

The successive approximation converter is one of the most popular topologies in both MOS and bipolar technologies. In MOS technologies the charge-redistribution implementation [9] of the successive approximation methodology is the most commonly used. The circuit diagram of a 4 bit charge redistribution converter is shown in Figure 10.49. In this circuit, the binary weighted capacitors $\{C, C/2, \ldots, C/8\}$ and the switches $\{S_1, S_2, \ldots, S_5\}$ form the 4 bit scaling DAC. For each conversion the circuit operates as a sequence of three phases. During the first phase (sample) switch S_0 is closed and all the other switches $\{S_1, S_2, \ldots, S_6\}$ are connected such that the input voltage V_{in} is sampled onto all the capacitors. During the next phase (hold) S_0 is open and the bottom plates of all the capacitors are connected to ground; i.e., switches $\{S_1, S_2, \ldots, S_5\}$ are switched to ground. The voltage, V_x, at the top plate of the capacitors at this time is equal to $-V_{in}$ and the total charge in all the capacitors is equal to $-2CV_{in}$. The final phase (redistribution) begins by testing the input voltage against the MSB. This is accomplished by keeping the switches $\{S_1, S_2, \ldots, S_5\}$ connected to ground and switching S_1 and S_6 such that the bottom plate of the largest capacitor is connected to V_{ref}. The voltage at the top plate of the capacitor is equal to

$$V_x = \frac{V_{ref}}{2} - V_{in} \tag{10.22}$$

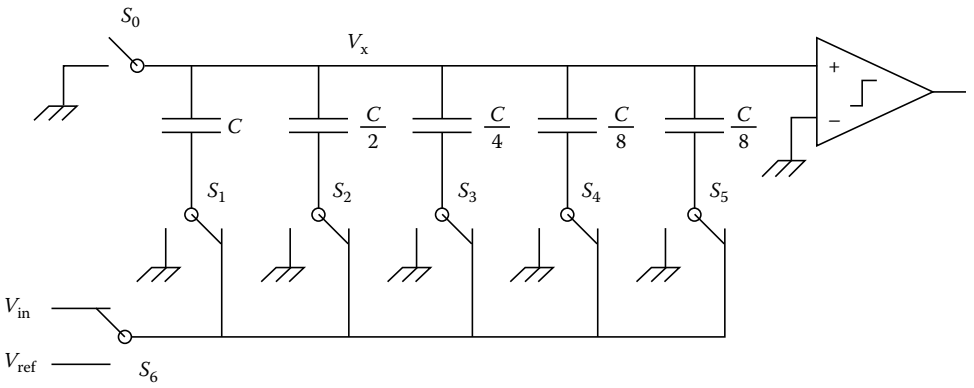


FIGURE 10.49 Charge-distribution implementation of the successive approximation architecture.

If $V_x > 0$ then the comparator output goes high, signifying that $V_{in} < (V_{ref}/2)$ and switch S_1 is switched back to ground. If the comparator output is low, then $V_{in} > (V_{ref}/2)$ and the switch S_1 is left connected to V_{ref} and the MSB is set high. In a similar fashion the next bit, MSB-1, is evaluated. This procedure is continued until all N-bits have resolved. After the conversion process the voltage at the top plate is such that

$$V_x = -V_{in} + \left\{ b_3 \frac{V_{ref}}{2^1} + b_2 \frac{V_{ref}}{2^2} + b_1 \frac{V_{ref}}{2^3} + b_0 \frac{V_{ref}}{2^0} \right\} \tag{10.23a}$$

$$-1 \text{ LSB} < V_x < 0 \tag{10.23b}$$

where b_i is {0, 1} depending upon if bit$_i$ was set to 0 or 1.

One of the advantages of the charge-redistribution topology is that the parasitic capacitance from the switches has little effect on the accuracy. Additionally, the clock feedthrough from switch S_0 only causes an offset and the clock feedthrough from switches $\{S_1, S_2, \ldots, S_5\}$ is input signal independent because they are always connected to either ground or V_{ref}. However, any mismatch in the binary ratios of the capacitors in the array causes nonlinearity, which limits the accuracy to 10 or 12 bits.

Self-calibration successive approximation converters. Fortunately, self-calibrating [7] techniques have been introduced that correct for errors in the binary ratios of the capacitors. Figure 10.50 is the block diagram for a successive approximation-based self-calibrating ADC. The circuit consists of an N-bit binary weighted capacitor array main DAC, an M-bit resistor string sub-DAC, and a calibration DAC. Digital logic is used to control the circuit during calibration and also to store the error voltages.

Let each weighted capacitor C_i have a normalized error in its ratio $(1 + \varepsilon_i)$ from its ideal value:

$$C_i = 2^{i-1} C(1 + \varepsilon_i) \tag{10.24}$$

Each capacitor contributes an error voltage at the top plate which is equal to

$$V_{e_i} = \frac{V_{ref}}{2^N} 2^{i-1} \varepsilon_i \quad i = 1B, 2, \ldots, N \tag{10.25}$$

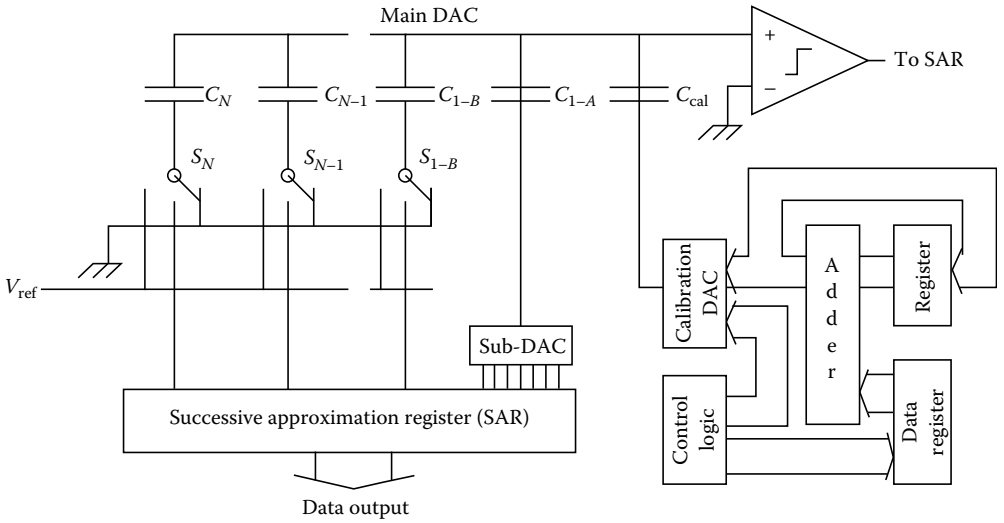


FIGURE 10.50 Self-calibration charge redistribution converter.

Therefore, the total linearity error is equal to

$$V_{\text{error}} = \sum_{i=1B}^{N} V_{\varepsilon_i} b_i \tag{10.26}$$

where b_i is the logic value of the ith bit.

The calibration cycle begins by measuring the error contribution from the largest capacitor and progressing to the smallest. The error from the MSB capacitor is evaluated by closing S_0 and setting switches $\{S_1, S_2, \ldots, S_5\}$ such that all the capacitors except C_{MSB} are charged to V_{ref}. Next, the switch S_0 is opened and switches $\{S_1, S_2, \ldots, S_5\}$ are switched to connect the bottom plates to ground. Under ideal conditions, i.e., $C_{\text{MSB}} = 2^{N-1}C$, the voltage at the top plate is equal to zero. It should be noted that the total capacitance is equal to $2C$. However, because $C_{\text{MSB}} = 2^{N-1}C(1 + \varepsilon_{\text{MSB}})$, the top plate voltage $V_x = (V_{\text{ref}}/2)\varepsilon_{\text{MSB}}$, such that $V_{x_{\text{MSB}}} = 2V_{\varepsilon_{\text{MSB}}}$ Therefore, the error voltage at the top plate is a direct measure of the corresponding error in the capacitor ratio. A successive approximation search using the sub-DAC is used to measure these voltages. The relationship between the measured residual voltage and the error voltage is equal to

$$V_{\varepsilon_i} = \frac{1}{2}\left\{ V_{x_i} - \sum_{j=i+1}^{N} V_{\varepsilon_i} \right\} \tag{10.27}$$

which corresponds to the equivalent error terms on the digital side. These digital correction terms are stored and subsequently added or subtracted during the normal operation cycle. Self-calibration improves the resolution of successive approximation converters to approximately 15 or 16 bits.

Algorithmic converters. As stated earlier, the algorithmic ADC is formed by modifying a pipelined converter. Here, the pipeline has been closed to form a loop. All N-bits are evaluated by a single stage, therefore implying that a N-bit conversion requires N clock cycles. A block diagram for the algorithmic converter is shown in Figure 10.51 [6] and consists of a S/H, a $2\times$ amplifier, a comparator, and a reference subtraction circuit. The circuit operates as follows. The input is first sampled and held by

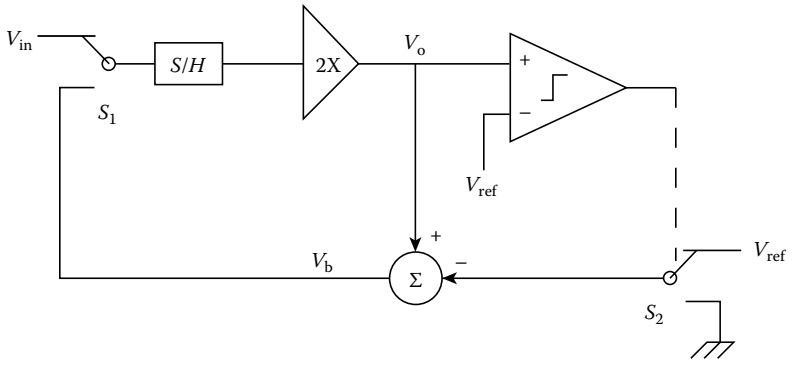


FIGURE 10.51 Algorithmic ADC block diagram.

setting S_1 to V_{in}. This signal is then multiplied by 2 (by the 2× amplifier). The result of this multiplication, V_o, is compared to V_{ref}. If $V_{o_N} > V_{ref}$ then the MSB, b_N, is set to 1 or it is set to 0. If b_N is equal to 1, then S_2 is connected to V_{ref} such that V_{b_N} is equal to

$$V_{b_N} = 2V_{0_N} - b_N V_{ref} \quad b_N = \{0, 1\} \tag{10.28}$$

This voltage is then sample-and-held and used to evaluate the MSB-1 bit. This procedure continues until all N-bits are resolved. The general expression for V_o is equal to

$$V_{0_i} = [2V_{0_{i-1}} - b_i V_{ref}]z^{-1} \tag{10.29}$$

where b_i is the comparator output for the ith evaluation and z^{-1} implies a delay of one clock period.

A circuit implementation for this ADC topology is shown in Figure 10.52 [10]. This circuit uses three amplifiers, five ratio-matched capacitors (C_1 to C_5), an arbitrary valued capacitor, C_6, and a comparator. Two implifiers and the capacitors (C_1 to C_5) form the recirculating register and the gain of two amplifiers. The amplifier A_3 and capacitor C_6 form an offset compensated comparator. The switches controlled by V_3, V_4, and V_5 load the input or selectively subtract the reference voltage. The conversion is started by setting V_1, V_2, and V_3 high. This forces V_x and V_y to 0 and loads V_{in} into C_1. Then, V_1 is set low and V_5 is set high. Therefore, the charge $V_{in} * C_1$ is transferred from C_1 to C_2. C_1 is made to be equal to C_2, therefore, $V_x = V_{in}$(C_3 is also charged to V_x). Because V_1 has been set low the comparator output goes high if $V_{in} > 0$, or else it remains low. This determines the MSB. The MSB-1 is determined by setting V_2 low and setting V_1 high. This forces the charge from C_3 to transfer to C_4($V_4 = V_x$; C_5 is also charged to V_4). During the same period C_1 is connected to ground if MSB = 1, or it is connected to V_{ref}. Next, V_2 is set low and V_1 is set high, while C_1 is switched from ground to V_{ref} for V_{ref} to ground. This transfers a charge equivalent to $C_1 \pm V_{ref}$ from C_1 to C_2 and transfers the charge in C_5, $C_5 * V_y$, to C_2. The capacitor C_5 is made to be twice as large as C_2, therefore, the voltage at V_x is equal to $2 * V_{in} \pm V_{ref}$. This process is repeated and the comparator determines bit MSB-1. This circuit has been shown to provide up to 10 bits of resolution at a maximum conversion rate of 200 kHz.

The maximum resolution of the algorithmic converter is limited by the ratio matching of the capacitors, clock feedthrough, capacitor voltage coefficient, parasitic capacitance, and offset voltages. The previous topology solves the problem of parasitic capacitances and amplifier offset voltage, however, its maximum resolution is limited by the ratio matching of the capacitors that are used to realize the gain of two amplifiers. This problem is partially resolved by using a ratio-independent multiply-by-two algorithm [6] to increase the maximum resolution to the 12 bit level. The ratio-independent multiply-by-two algorithm is easily explained by the circuit shown in Figure 10.53. During ϕ_1, capacitor C_1 is

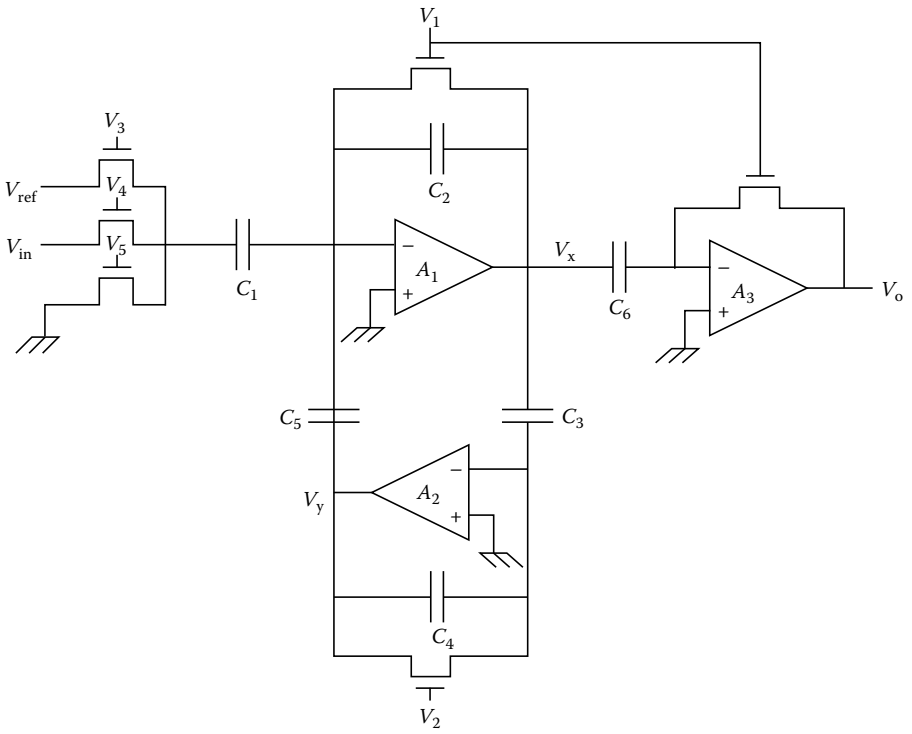


FIGURE 10.52 Example circuit implementation of the algorithmic converter.

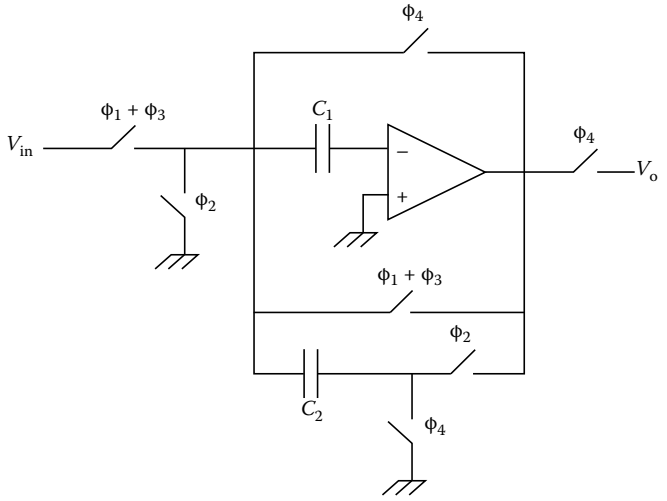


FIGURE 10.53 Ratio-independent multiply-by-two circuit.

charged to V_{in}. This charge is then transferred onto C_2 during ϕ_2. The charge on C_2 is equal to $V_{in} * C_1$. During ϕ_3 C_2 is disconnected from the feedback path and V_{in} is once again sampled onto C_1. During ϕ_4 the charge in C_2 is added to C_1. The total charge in C_1 is now equal to $C_1\ V_{in} + C_1\ V_{in} = 2C_1V_{in}$ and is completely independent of the value of C_2. Therefore, the voltage at the output at ϕ_4 is equal to $2V_{in}$. The only constraint is that the input voltage be held steady, i.e., S/H during ϕ_1 and ϕ_3.

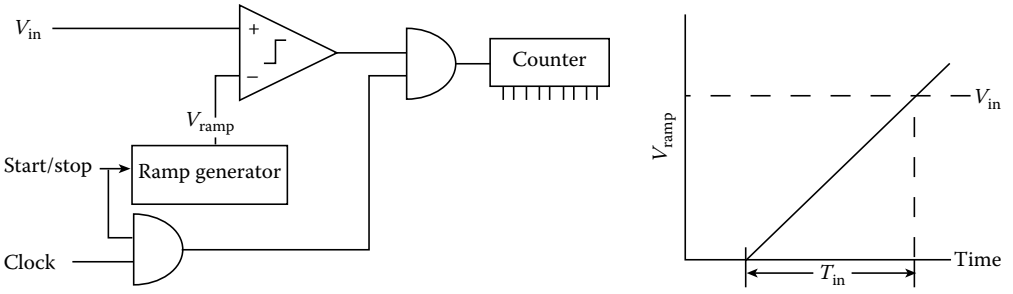


FIGURE 10.54 Single-slope integrating converter.

10.2.2.3 2^N-Clock Converters

The basic principle of the integrating converter can be explained with the help of Figure 10.54. A comparator compares the input signal with the output of a ramp voltage generator. The ramp voltage generator is zeroed after each measurement. The output of this comparator is used to gate the clock to an interval counter. The counter output corresponding to the ramp time T_{in} provides an accurate measure of the input voltage. The input voltage V_{in} is equal to $T_{in} * U$, where U is the ramp rate. Because the absolute values of components are not well controlled and also because of the large offset voltages associated with MOS amplifiers and comparators, a calibration or a reference cycle is usually added to calculate the ramp rate and the offset voltage. A simple circuit for a single-slope integrating converter that includes the calibration cycle is shown in Figure 10.55 [4].

The ramp voltage is generated using a constant current source to charge a capacitor. The ramp voltage, V_{ramp}, is equal to $\int_0^t (I/c)dt$, which is equal to $(I\Delta t)/c$ for a constant current I. The ramp voltage is compared against the analog ground voltages, V_{in} and V_{ref}, respectively. The addition of the third calibration cycle eliminates any offset errors. The final resolution is dependent only on the linearity of the ramp generator, i.e., the linearity of the current source. In the single-slope approach just described, the calibration is done in digital. However, the complete calibration can be performed in analog as well, as in the dual-slope approach. Further improvements include a charge balancing technique [4] that

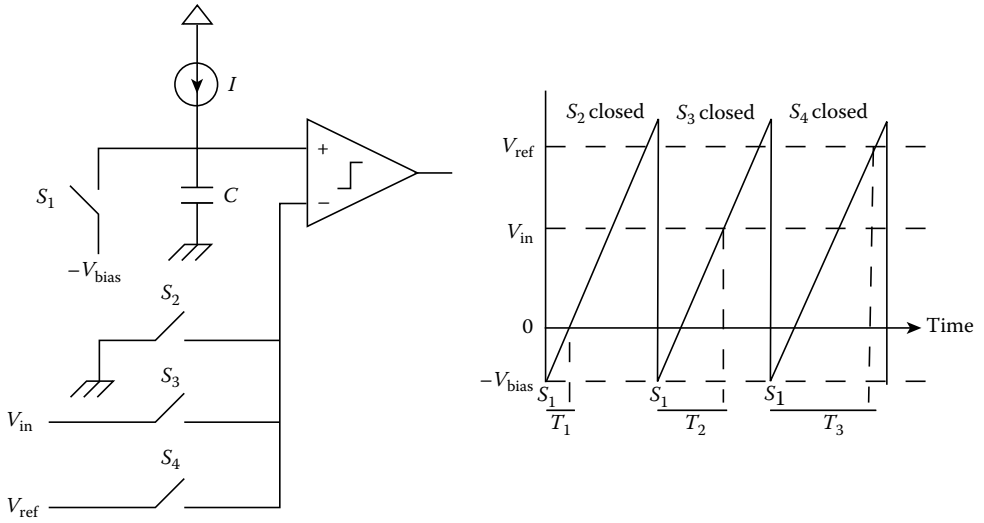


FIGURE 10.55 Single-slope integrating converter with calibration cycle.

uses an oscillating integration process to keep the voltage across the capacitor closer to zero, thereby reducing the linearity constraints on the ramp generator. The primary advantage of the integrating converter is the small number of precision analog components that are required to generate extremely high resolution. However, the primary disadvantage is the conversion time required. It takes 2^N clock cycles to generate a N-bit conversion.

10.2.3 Oversampled Converters

Oversampling converters have the advantage over Nyquist rate converters in that they do not require very tight tolerances from the analog components and also because they simplify the design of the anti-alias filter. Examples of oversampling converters include the noise-shaping architecture and the interpolative architecture. Our discussion centers around noise-shaping converters.

If the analog input signal V_{in} has a frequency spectrum from 0 to f_0 then $2f_0$ is defined as the Nyquist rate. Oversampling converters sample the input at a rate larger than the Nyquist frequency. If f_s is the sampling rate, then $(f_s)/(2f_0) = \text{OSR}$ is called the oversampling ratio. Oversampling converters use "signal averaging" along with a low-resolution converter to provide extremely high resolution. This technique can best be understood by considering the following example in Figure 10.56.

Let the input be exactly in the middle of V_n and V_{n+1} and let it be sampled a number of times. If, in addition to the input signal, we add some random noise, then for a large number of samples the output would fall on V_n 50% of the time and on V_{n+1} the other 50% of the time. If the signal was a littler closer to V_{n+1}, then the percentage of times the output falls on V_{n+1} would increase. Using this averaging technique we can get a better estimate of the input signal. However, in simple oversampling the resolution only increases by $\sqrt{n}$, where n is the number of samples of V_{in} that are averaged. Therefore, to increase the resolution of the converter by one additional bit we are required to increase the number of samples by 4×.

Noise-shaping converters use feedback to generate the necessary noise and additionally perform frequency shaping of the noise spectrum to reduce the amount of oversampling necessary. This can be illustrated with the help of Figure 10.57. The output from H_1 is quantized by an N-bit ADC. This digital value is then converted to an analog value by the N-bit DAC. This value is subtracted from the input and the result is sent to H_1. Here, we assume an N-bit converter for simplicity, however, for the special case in which $N = 1$ the noise-shaping converter is called a sigma-delta converter. The quantization process approximates an analog value by a finite-resolution digital value. This step introduces a quantization error, Q_n. Further, if we assume that the quantization error is not correlated to the input, the system can now be modeled as a linear system, as shown in Figure 10.58. Here, we note that the error introduced by the analog-to-digital process is modeled by Q_n. The output voltage for this system can now be written as

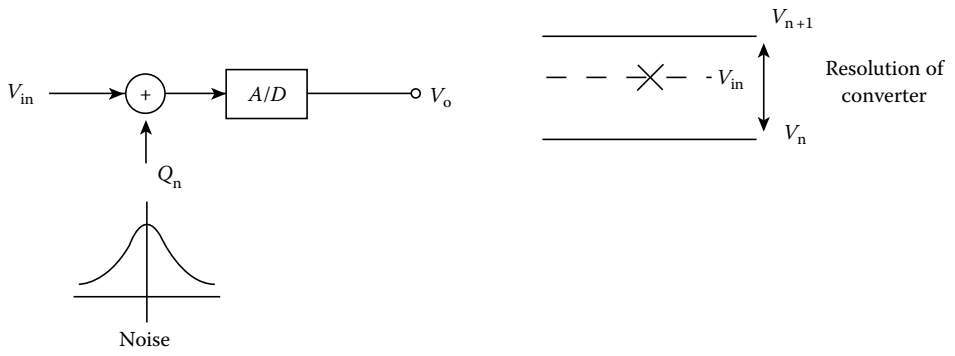


FIGURE 10.56 Higher resolution provided by oversampling.

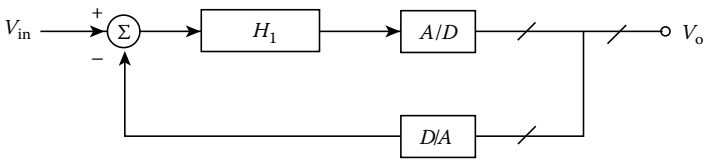


FIGURE 10.57 Noise-shaping oversampling converters.

$$V_0 = \frac{Q_n}{[1+H_1]} + \frac{V_{in}H_1}{[1+H_1]} \tag{10.30}$$

Data converters are sampled data systems, and as such are easier to analyze in the Z-domain. For most sigma-delta converters H_1 has the characteristics of a low-pass filter and is usually implemented as a switched-capacitor integrator. MOS switched-capacitor integrators can be implemented with either a delay in the forward signal path or a delay in the feedback path, and can be modeled in the Z-domain by Figures 10.59 and 10.60, respectively.

We use the first integrator architecture because it simplifies some of the algebra. For a first-order sigma-delta converter H_1 is realized as a simple switched-capacitor integrator, i.e., $H_1 = (z^{-1})/(1-z^{-1})$. Therefore, Figure 10.58 can now be drawn as Figure 10.61. Replacing H_1 by $(z^{-1})/(1-z^{-1})$ in Equation 10.30 we can write the transfer function for the first-order sigma-delta converter as

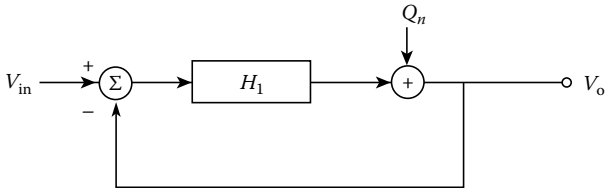


FIGURE 10.58 Linear system model of noise-shaping converter.

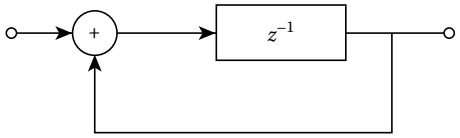


FIGURE 10.59 Forward path delay integrator.

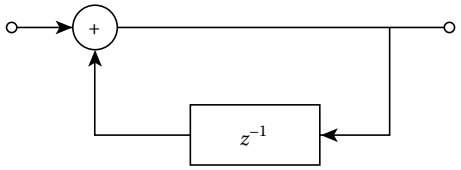


FIGURE 10.60 Feedback path delay integrator.

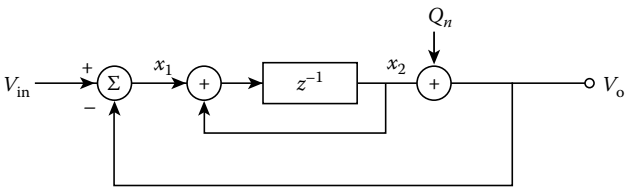


FIGURE 10.61 First-order noise-shaping converter.

$$V_o = V_{in}z^{-1} + Q_n(1 - z^{-1}) \tag{10.31}$$

As can be seen from Equation 10.31, the output is a delayed version of the input plus the quantization noise multiplied by the factor $(1 - z^{-1})$. This function has a high-pass characteristic, as shown in Figure 10.62. We note here that the quantization noise is substantially reduced at lower frequencies and increases slightly at higher frequencies. In this figure, f_o is the input signal bandwidth and $f_s/2 = \pi$ corresponds to the Nyquist rate of the oversampling converter. For simplicity the quantization noise is usually assumed to be white* with a spectral density equal to $e_{rms}\sqrt{2/f_s}$. Therefore, the magnitude of the output noise spectrum can be written as

$$N(f) = e_{rms}\sqrt{\frac{2}{f_s}}\left|1 - z^{-1}\right| = 2e_{rms}\sqrt{\frac{2}{f_s}}\sin\left(\frac{\pi f}{f_s}\right) \tag{10.32}$$

Further, if $f_o \ll f_s$ we can approximate the root mean square noise in the signal band, $(0 < f < f_o)$, by

$$N_{f_0} \approx e_{rms}\frac{\pi}{3}\left(\frac{2f_0}{f_s}\right)^{3/2} \tag{10.33}$$

As the OSR increases the quantization noise in the signal band decreases; i.e., for a doubling of the OSR the quantization noise drops by 20 $\log(2)^{3/2} \approx 9$ dB. Therefore, for each doubling of the OSR we effectively increase the resolution of the oversampling converter by an additional 1.5 bits.

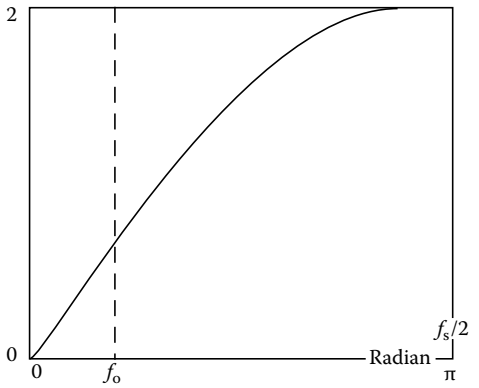


FIGURE 10.62 Magnitude response of the function $(1 - z^{-1})$.

* Quantization noise is clearly not uncorrelated or white for the first-order sigma-delta modulator, but becomes increasingly so for the higher-order systems.

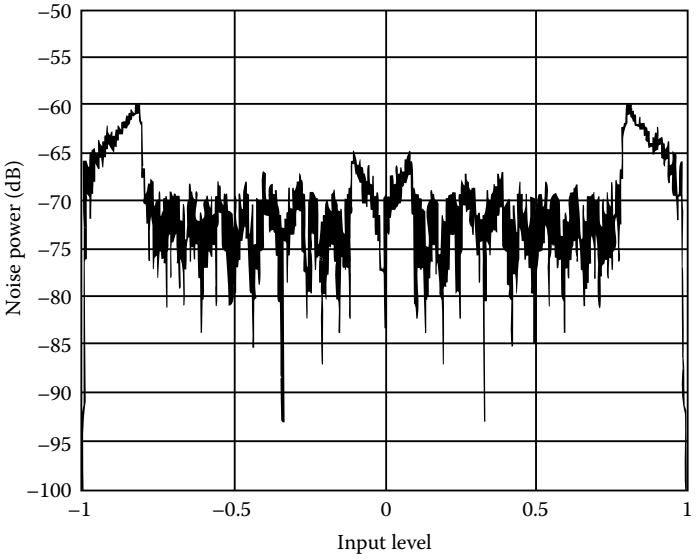


FIGURE 10.63 Pattern noise for a first-order sigma-delta modulator for DC inputs.

The previous analysis was based on the assumption that the quantization noise was not correlated to the input and uniformly distributed across the Nyquist band. We now reexamine these assumptions. The assumption that the quantization noise is not correlated with the input only holds for extremely busy input signals. This is particularly not true for the first-order modulator assumed in the analysis above, such that for extremely low frequency or DC inputs the first-order modulator generates pattern noise (also called tones), as shown in Figure 10.63. The peaks of the pattern noise occur at input voltages that are integer divisors of the quantization step. It is possible to provide a conceptual explanation for this occurrence. For example, for an input that is an integer divisor of the quantization level the digital output of the quantizer repeats itself at an extremely low frequency. This low-frequency repetition causes noise power to be introduced into the signal band. The quantization noise for second- and higher-order models is significantly more uncorrelated and is usually assumed to be white.

The quantization error has a value that is limited to $+\frac{1}{2}$ LSB of the quantizer (the ADC in Figure 10.47). If we assume that the quantization noise is white and uniformly distributed over the quantization level, then the average noise quantization is equal to

$$\int_{-\frac{1}{2}\mathrm{LSB}}^{+\frac{1}{2}\mathrm{LSB}} x^2 \mathrm{d}x = \frac{\mathrm{LSB}^2}{12} = P_n \tag{10.34}$$

Because the quantization noise is sampled at the clock frequency f_s, the entire noise power is aliased back into the overall converter Nyquist ban $[0-(f_s/2)]$. Therefore, the spectral density of the quantization noise is equal to

$$P_n = \frac{\mathrm{LSB}^2}{12} = \int_0^{f_s/2} n_e(f)^2 \delta f = n_e(f)^2 \frac{f_s}{2} \tag{10.35a}$$

$$n_e(f) = P_n\sqrt{\frac{2}{f_s}} \tag{10.35b}$$

The SNR for an ADC is defined as $10\log(P_s/P_n)$, where P_s is the signal power. The signal power is highly waveform dependent. For example, the P_s for a full-scale sine wave input, $(A/2)\sin(\omega T)$, which is applied to an N-bit quantizer, can be written in terms of the quantization level as

$$\frac{A^2}{8} = \frac{[(2^N - 1)\text{LSB}]^2}{8} \tag{10.36}$$

Therefore,

$$\text{SNR} = 10\log\left(\frac{P_s}{P_n}\right) = 10\log\left[\frac{12(2^N - 1)^2}{8}\right] \tag{10.37}$$

10.2.3.1 Higher-Order Modulators

In Figure 10.61, we replaced H_1 for Figure 10.58 with a first-order integrator. Clearly, H_1 can be replaced by other higher-order functions that have a low-pass characteristic.* For example, in Figure 10.64 we show a second-order modulator. This modulator uses one forward delay integrator and one feedback delay integrator to avoid stability problems. The output voltage for this figure can be written as

$$V_0 = V_{\text{in}}z^{-1} + Q_n(1 - z^{-1})^2 \tag{10.38}$$

Note that the quantization noise is shaped by the second-order difference equation. This serves to further reduce the quantization noise at low frequencies. However, a further increase in the noise occurs at higher frequencies. A comparison of the noise shaping offered by the first and second-order modulators is shown in Figure 10.65. Once again, assuming that $f_o \ll f_s$ we can write an expression for the root mean square noise in the signal band for the second-order modulator as

$$N_{f_0} \approx e_{\text{rms}}\frac{\pi^2}{\sqrt{5}}\left(\frac{2f_0}{f_s}\right)^{5/2} \tag{10.39}$$

The noise power in the signal bandwidth falls by 15 dB for every doubling of the OSR. One of the added advantages of the second-order modulator over the first-order modulator is that quantization noise has been shown to be less correlated to the input, therefore, less pattern noise.

From our analysis so far, it would seem that increasing the order of the filter would reduce the necessary OSR for a given resolution. This is true, however, the simple Candy-style modulator (shown in

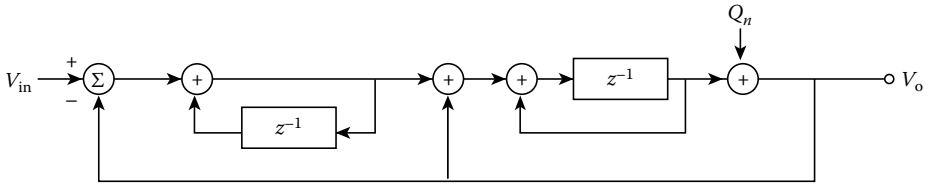


FIGURE 10.64 Second-order modulator block diagram.

* Actually, it is not necessary that they have low-pass characteristics. Bandpass characteristics may be preferred if the input signal is to be bandlimited.

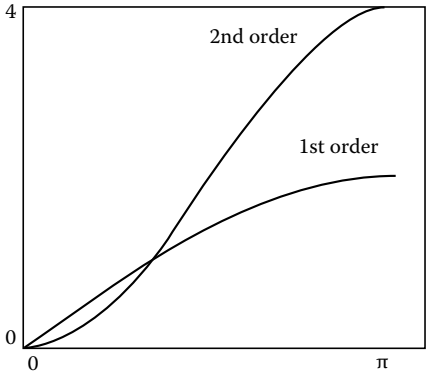


FIGURE 10.65 Noise shaping due to the second-order modulator in Figure 10.62.

Figures 10.61 and 10.64) with orders >2 results in stability problems. This is because for higher-order modulators the later integrator stages are easily overloaded and saturated. This in turn increases the noise in the signal band. However, higher-order modulators can be realized by using a cascade of lower order modulators in the MASH architecture [8]. In the cascaded MASH technique, both the digital output and the output of the integrator of each lower-order modulator is passed on to the next module. A second-order MASH architecture using two cascaded first-order sections is shown in Figure 10.66. It can be shown that the output is equal to

$$Y = z^{-2}X - Q_{n_2}(1 - z^{-1})^2 \tag{10.40}$$

Once again, we note that the quantization noise is multiplied by the second-order difference equation. The sign in front of the noise term is not important. However, for complete cancellation of the quantization noise from the first integrator the gain of the first loop needs to be identical to the gain of the second loop. Therefore, the amplifier gain and capacitor matching become extremely important. It has been shown that a 1% matching and an op-amp gain of 80 dB are sufficient for 16 bits of accuracy [8].

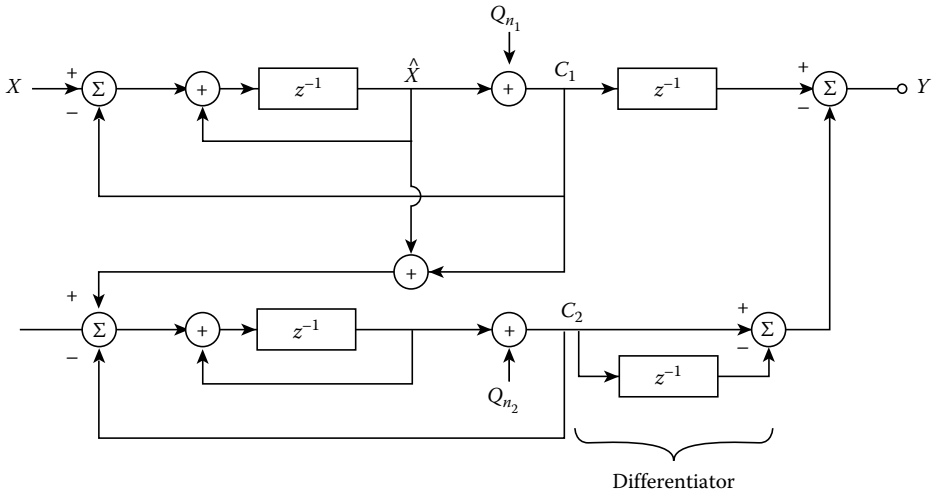


FIGURE 10.66 MASH architecture for a second-order modulator.

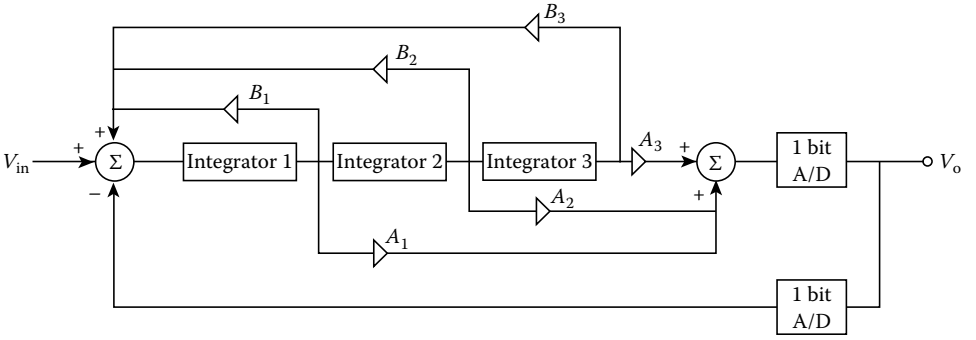


FIGURE 10.67 Finite pole-zero loop filter higher-order modulator.

An alternate methodology to stabilize higher-order oversampled coders is the use of finite poles and zeroes for the loop filter [16], H_1 in Figure 10.57. Up until now, all the loop filters have been integrators with poles at DC and zeroes at extremely high frequencies. The loop filter can be realized using additional feedback and feedforward paths as shown in Figure 10.67. A third-order modulator is shown in this figure. Having finite poles and zeroes serves two purposes: (1) the nonzero poles function to reduce the in-band noise by flattering the quantization noise transfer function at low frequencies, (2) the finite zeroes function to reduce the magnitude of the quantization noise at high frequencies. By reducing the magnitude of the quantization noise at high frequencies, even higher-order modulators can be made stable. Additionally, these modulators have been shown to be devoid of pattern noise artifacts.

10.2.3.2 Multibit Quantizers

The primary reason for using single-bit or two-level quantizers is their inherent perfect linearity. Because only two levels can exist, a straight line can always be drawn between these two levels. On the other hand number of advantages are found in using multibit quantizers in oversampling converters. The quantization noise generated in the multibit-based noise-shaping converter is significantly more "white" and uncorrelated with the input signal, thereby reducing the probability of pattern noise. Additionally, the quantization noise power goes down exponentially as the number of bits in the quantizer increases. However, the primary problem associated with multilevel quantizers is the nonlinearity errors present with the DAC in the modulator loop. This problem can be illustrated with the help of Figure 10.68.

In Figure 10.68, the error resulting from the nonlinearity in the multibit ADC is included as AD_{NL} and the error resulting from the nonlinearity in the multibit DAC is included as DA_{NL}. The output voltage is given by Equation 10.43. Here, note that the analog-to-digital nonlinearity is suppressed by the loop filter, while the digital-to-analog nonlinearity is only subjected to a unit delay. Therefore, any

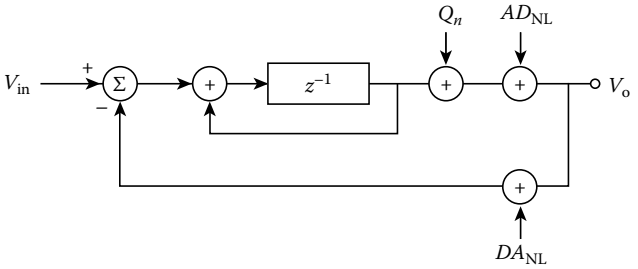


FIGURE 10.68 Model for nonlinearity associated with multibit quantizers.

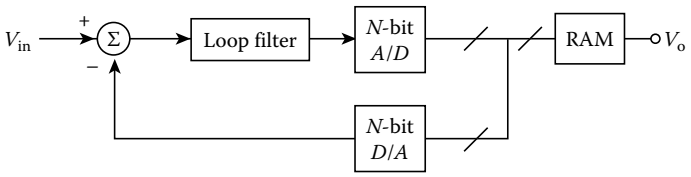


FIGURE 10.69 Digital error correction for multibit quantizers.

digital-to-analog nonlinearity directly appears in the output. A number of methods have been applied to reduce the effects of nonlinearity associated with multibit quantizers. The two most promising methods that have emerged are digital error correction [18] and dynamic element matching [17,19].

$$V_0 = V_{in}z^{-1} - DA_{NL}z^{-1} + Q_n(1 - z^{-1}) + AD_{NL}(1 - z^{-1}) \tag{10.41}$$

A block diagram for digital error correction for multibit quantizer-based noise-shaping converters is shown in Figure 10.69. The random access memory (RAM) and the multibit DAC have the same input signal. Because of the high gain in the loop at low frequencies the output of the DAC is almost identical to the input voltage, V_{in}. Now, if the digital RAM is programmed to generate the exact digital equivalent of the digital-to-analog output for any digital input, then the RAM output and the digital-to-analog output will be identical to each other. Because the output of the DAC is almost identical to the input voltage, the output voltage will also be the exact digital equivalent of the analog input. The RAM can be programmed by reconfiguring the modulator stages and feeding the system with a multibit digital ramp [18].

In the dynamic element matching approach, the various analog elements that are used to generate the different analog voltage levels are dynamically swapped around. The various elements can be swapped randomly [17] or in a periodic fashion [19]. The use of random permutations translates the nonlinearity of the DAC into random noise that is distributed throughout the oversampling converter Nyquist range. This method virtually eliminates errors due to nonlinearity, but unfortunately it also increases the noise level in the signal band. In a variation of this basic technique, the various analog elements are swapped in a periodic fashion such that the nonlinearity in the DAC is translated into noise at higher frequencies. Individual level averaging further eliminates the possibility of pattern noise within the signal band [19].

10.2.3.3 Technology Constraints

One of the primary reasons for using sigma-delta converters is that they do not require good matching among the analog components. Therefore, for the two-level sigma-delta converter the nonidealities are introduced primarily by the integrator loop. To aid in the analysis of the various technology constraints we shall consider a particular implementation of an integrator (Figure 10.70). The ideal transfer function for this circuit is given by $-1/(1 - z^{-1})$. To realize this ideal transfer function the circuit relies on the virtual ground generated at the negative input of the integrator to accomplish complete charge transfer during each clock period. However, limited amplifier gain does not generate a perfect virtual ground,

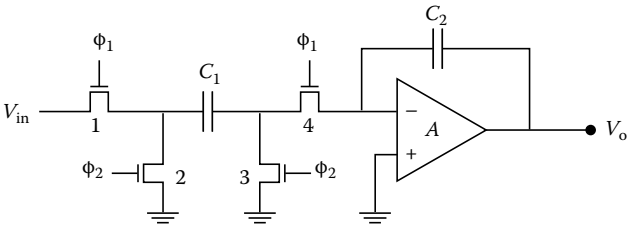


FIGURE 10.70 Example circuit implementation for a switched-capacitor integrator.

thereby not accomplishing the complete transfer of charge during each clock period. The effect of the limited gain is similar to a leaky integrator and the transfer function for the leaky integrator can be written as

$$H(z) = \frac{-1}{1 - \alpha z^{-1}} \tag{10.42a}$$

where

$$\alpha = \frac{1}{1 - \frac{1}{A}\left(1 + \frac{C_1}{C_2}\right)} \approx \frac{1}{1 - \frac{2}{A}} \tag{10.42b}$$

The net effect of finite gain is to increase the modulation noise in the signal band as illustrated by Figure 10.71 for the first-order modulator. In this figure, the *X*-axis is plotted from 0 to 1 rather than the complete Nyquist band to emphasize the signal band. The noise transfer function has been plotted for a number of amplifier gains. When compared to Figure 10.62, note the increase in the noise level in the signal band. The effect of finite gain is felt throughout the input signal magnitude range as shown in Figure 10.72. The graph for Figure 10.71 was generated using the linearized model for the modulator

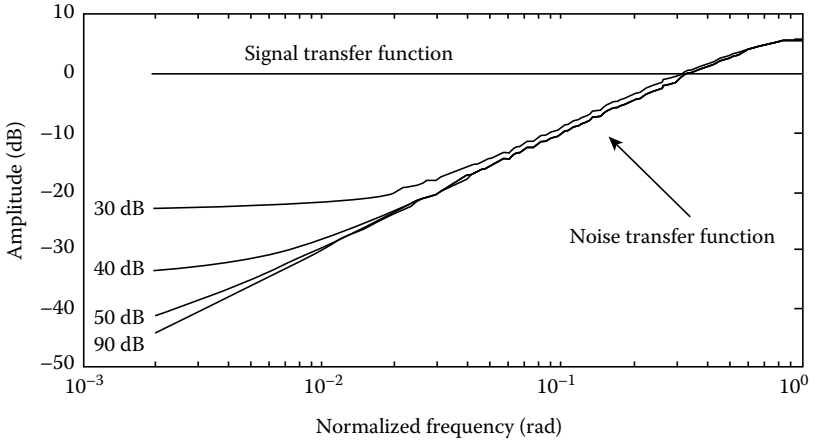


FIGURE 10.71 Effect of finite amplifier gain on noise transfer function using a linear model.

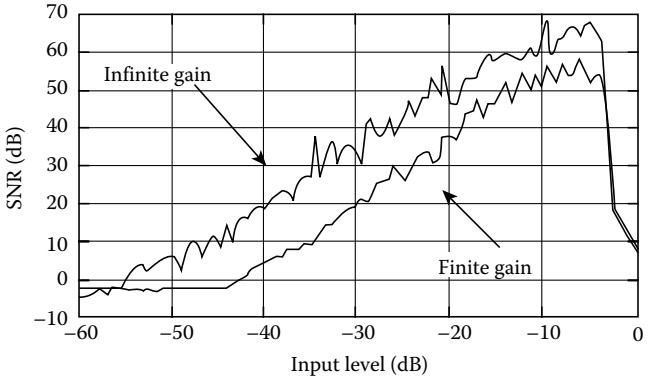


FIGURE 10.72 Effect of finite amplifier gain using difference equation simulations.

presented in Equation 10.21 and the graph in Figure 10.72 was generated using the difference equation method. The difference equation method does not make any assumptions about linearity nor does it assume that the input is uncorrelated with the quantization noise, however, it requires considerably more simulation time. Because of oversampling the bandwidth requirements for the op-amps in the integrators are usually large. Unfortunately, it is extremely difficult to realize extremely high gain and extremely high bandwidth amplifiers in MOS. One solution that attempts to mitigate the finite gain effect is to estimate the amount of incomplete charge transfer and compensate for it [5].

Circuit noise provides additional limitations to the maximum resolution realizable by an oversampled converter. The primary noise sources are the thermal noise generated by the switches in the integrator, amplifier noise, charge injection, and clock feedthrough from the switches. Because of sampling, the thermal noise associated with the finite on resistance of the switches is aliased back into the Nyquist band of the oversampling converter. The total noise aliased into the baseband for large bandwidth amplifiers is equal to kT/C for each switch pair, where k is the Boltzmann constant, T is the temperature in degrees Kelvin, and C is the value of the sampling capacitor in the integrator. For the parasitic insensitive integrator in Figure 10.70, the total noise from this source is equal to $2kT/C$. This noise is evenly spread across the Nyquist band, but only the fraction $2f_o/f_s$ of this noise appears in the signal band. The rest is filtered out by the digital LPF. Using this constraint, for a full-scale sine wave input the minimum sampling capacitance is given by

$$C_{\min} = 16 \cdot k\mathrm{T} \cdot \mathrm{SNR}_{\mathrm{desired}} \tag{10.43}$$

The inband portion of the amplifier noise is also added to the output signal. In general, only the noise of the first amplifier is important for higher-order converters. For MOS amplifiers the flicker noise component is significantly more important as it tends to dominate in the signal band. When necessary, correlated double sampling techniques [5] can be used to reduce the effect of this noise source. Correlated double sampling, or autozeroing as it is sometimes called, has the added benefit that it eliminates any amplifier offset voltages. It is usually important to remove this offset voltage only for data acquisition applications.

Because tight component matching is not required of the analog components, sigma-delta converters are particularly well suited for mixed-signal applications. However, having digital circuits on the sample chip increases the switching noise that is injected into the substrate and into the power supply lines. Any portion of this noise that lies in the signal band is added to the input signal. Therefore, fully differential integrator topologies should be used for high-resolution converters. Substrate and supply noise are common-mode signals and are reduced by the common-mode rejection ratio of the amplifier when using fully differential circuits.

In addition to the amplifier and the switching noise the charge injection from switches also sets a limit on the maximum resolution attainable component. Charge injection from switches has a signal-dependent component and a signal-independent component. The effect of the signal-independent component is to introduce an additional offset error that can easily be calibrated out, if necessary. However, the signal-dependent component, particularly from the input sampling transistor (transistor M_1 in Figure 10.73), cannot be distinguished from input signal. This signal-dependent component is highly nonlinear and can be reduced substantially by using proper clock phasing. Signal-dependent charge injection from transistors M_1 and M_2 in Figure 10.73 can be canceled to first order by delaying the turn off of ϕ_1' slightly [6].

A number of topologies for the digital low-pass filters have been tried. However, it has been shown that simple finite impulse response (sinc) filters are probably the optimal choice. It has been shown that the number of stages of sinc filtering necessary is equal to the modulator order plus 1 [2]. Noise-shaping converters have the ability to provide extremely high resolution. However, care must be used when using simple linear assumptions. Clearly, for the first-order modulator the white noise assumption

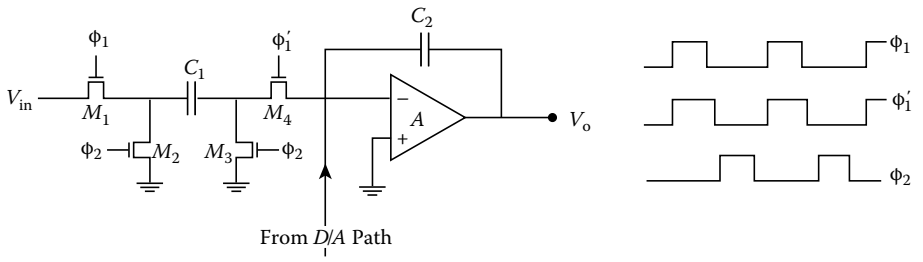


FIGURE 10.73 Proper clock phase to eliminate signal-dependent clock feedthrough.

breaks down. Additionally, it has been shown that the simple linear model overestimates the realizable SNR. For example, the linearized model overestimates the attainable SNR by as much as 14 dB for the second-order modulator.

Acknowledgment

The author acknowledges the help of his students in completing this manuscript, particularly Feng Wang.

References

1. W. C. Black, High-speed CMOS A/D conversion techniques, PhD thesis, Berkeley: University of California, 1980.
2. J. C. Candy and G. C. Temes, *Oversampling Methods for A/D and D/A Conversion*, New York: IEEE Press, 1992.
3. M. J. Demler, *High-Speed Analog-to-Digital Conversion*, New York: Academic Press, 1991.
4. P. R. Gray and D. A. Hodges, All-MOS analog-digital conversion techniques. *IEEE Trans. Circuits Syst*., CAS-25(7), 482–489, 1978.
5. P. J. Hurst and R. A. Levinson, Delta-sigma A/Ds with reduced sensitivity to op-amp noise and gain. *Proceedings of IEEE Int. Symp. Circuits Syst.*, Portland, Oregon, pp. 254–257, 1989.
6. P. W. Li, M. J. Chin, P. R. Gray, and R. Castello, A ratio-independent algorithmic analog-to-digital conversion technique. *IEEE J. Solid-State Circuits*, SC-19(6), 828–836, 1984.
7. H. S. Lee, D. A. Hodges, and P. R. Gray, A self-calibrating 15 bit CMOS A/D converter. *IEEE J. Solid-State Circuits*, SC-19(6), 813–819, 1983.
8. Y. M. Matsuya, K. Uchimura, and A. Iwata, A 16-bit oversampling A-to-D conversion technology using triple integration noise shaping. *IEEE J. Solid-State Circuits*, SC-22, 921–929, 1987.
9. J. L. McCreary and P. R. Gray, All-MOS charge redistribution analog-to-digital conversion techniques I. *IEEE J. Solid-State Circuits*, SC-10, 371–379, 1975.
10. R. H. McCharles, V. A. Saletore, W. C. Black, and D. A. Hodges, An algorithmic analog-to-digital converter. *Proceedings of the IEEE Int. Solid-State Circuits Conf*., San Francisco, California, 1977.
11. S. Masuda, Y. Kitamura, S. Ohya, and M. Kikuchi, A CMOS pipelined algorithmic A/D converter, *Proceedings of the IEEE Custom Integrated Circuits Conf*., San Jose, California, pp. 559–562, 1984.
12. G. C. Temes, F. J. Wang, and K. Watanabe, Novel pipeline data converters. *Proceedings of the IEEE Int. Symp. Circuits Systems*, Helsinki, Finland, pp. 1943–1946, 1988.
13. R. Unbehauen and A. Cichocki, *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems*, New York: Springer-Verlag, 1989.
14. S. H. Lewis, Video-rate analog-to-digital conversion using pipelined architectures, PhD thesis, Berkeley, CA: University of California, 1987.

15. S. Sutarja and P. R. Gray, A pipelined 13-bit, 250-ks/s, 5-V analog-to-digital converter. *IEEE J. Solid-State Circuits*, SC-23, 1316–1323, 1988.
16. K. Chao, S. Nadeem, W. Lee, and C. Sodini, A higher-order topology for interpolative modulators for oversampling A/D converters. *IEEE Trans. Circuits Syst.*, CAS–37, 309–318, March 1990.
17. R. Carley, A noise-shaping coder topology for 15+ bit converters. *IEEE J. Solid-State Circuits*, SC–24, 267–273, April 1989.
18. R. Walden et al., Architectures for higher-order multibit sigma-delta modulators, *Proceedings of the IEEE Int. Symp. Circuits Syst.*, Portland, Oregon, pp. 895–898, 1990.
19. B. Leung and S. Sutarja, Multibit sigma-delta A/D converter incorporating a novel class of dynamic element matching techniques. *IEEE Trans. Circuits Syst.*, CAS-39(1), 35–51, 1992.

Further Information

Max Hauser provides an extremely good overview of oversampling converters in ''Principles of oversampling A/D conversion,'' *J. Audio Eng. Soc.*, 39(1/2), 3–26, 1991.

Sources for further reading about Nyquist rate converters include D.J. Dooley, *Data Conversion Integrated Circuits*. New York: IEEE Press, 1980, and R.J. van der Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Boston: Kluwer Academic, 1994.

IEEE Journal of Solid-State Circuits, particularly the December issues, and the *IEEE Transactions on Circuits and Systems* are good sources for more recent research on data converters.

Index

A

Amplifier impulse response, 6-15
Amplifier transfer function, 6-18
Analog circuit cells
  balanced differential amplifier
    engineering constraints, 2-45–2-47
    generalized system diagram, 2-44–2-45
    monolithic fabrication process, 2-45
    single-ended output voltage, 2-44
  common-base amplifier
    circuit broadbanding, 2-32
    common-emitter–common-base cascode, 2-30–2-32
    current buffering purpose, 2-26–2-27
    diode resistance, 2-28
    driving point input resistance, 2-29
    driving point output resistance, 2-29–2-30
    equivalent circuit, 2-27–2-28
    Kirchhoff's current law constraint, 2-27
    Miller multiplication, 2-32
    Norton transconductance, 2-31
    NPN and PNP AC schematic diagrams, 2-26–2-27
    small-signal analysis, 2-27
    voltage divider, 2-31
  common bipolar junction transistor (BJT) biasing circuits
    cascode current mirror, 2-6
    improved bandgap reference, 2-9–2-10
    low-bias current mirror, 2-5–2-6
    simple band-gap reference, 2-9
    simple current mirror, beta helper, 2-2–2-3
    simple current mirror, emitter degeneration, 2-4
    V_{BE} multiplier circuit, 2-8–2-9
    V_{BE} referenced current mirror, 2-6–2-8
    well-defined temperature coefficient, 2-9
    Widlar current mirror, 2-4–2-5
    Wilson current mirror, 2-3–2-4
    Zener diode reference, 2-8
  common-collector amplifier
    AC schematic diagrams, 2-33
    active load, 2-35–2-36
    driving point input and output resistances, test circuit, 2-34
    Thévenin load resistance, 2-37
    voltage gain, 2-33
  common-emitter amplifier
    active current source load, 2-24–2-26
    driving point output resistance, 2-23
    macromodel, 2-19–2-20, 2-23
    Norton current and equivalent circuit, 2-19
    NPN and PNP AC schematic diagrams, 2-17, 2-20–2-21
    operation, 2-17
    output coupling capacitance, 2-24
    small-signal test structure, 2-18–2-19
    Thévenin source voltage, 2-18
  common-mode input voltage, 2-41–2-42
  Darlington connection
    forward transconductance, 2-41
    schematic diagram, 2-38–2-39
    small-signal equivalent circuit, 2-39–2-40
    transconductance amplifier, 2-38
    transconductance frequency response, 2-38–2-39
    voltage gain, 2-40–2-41
    Wilson mirror load, 2-39
  differential input source voltage, 2-41

diode-connected transistor
KVL analysis, 2-15
small-signal transistor model, 2-14–2-15
static common-base current gain, 2-15–2-16
subject diagram, 2-14
V_{BE} multiplier, 2-15–2-17
volt–ampere characteristics, 2-14
performance index, 2-43
small-signal model, 2-11–2-14
system-level diagram, differential amplifier, 2-41–2-42
Thévenin equivalent I/O circuits
balanced bipolar differential amplifier, 2-52–2-53
driving point common-mode output resistance, 2-49–2-50
driving point differential-mode output resistance, 2-49–2-50
Kirchhoff's current and voltage laws, 2-47
open-circuit differential-mode gain, 2-51
pertinent test cell, 2-48–2-49
test circuit, 2-47–2-49
Thévenin model, 2-51
two-port model, 2-54
zero input signal excitation, 2-49
Analog integrated circuits
chip parasitic
circuit model, 1-143
effects on circuits, 1-136–1-137
inductance, 1-139
measurement, 1-145
modeling technique, 1-135
nonlinear interconnects, 1-141–1-142
overlap capacitance, 1-133–1-134
packaged IC, 1-142–1-143
parallel line capacitance, 1-133–1-134, 1-136
PSPICE ac simulation, 1-143–1-144
resistance, 1-137–1-139
substrate capacitance, 1-133–1-134
transmission line behavior
effects on circuits, 1-140–1-141
modeling, 1-140
types, 1-139–1-140
Analog-to-digital converters (ADCs)
Nyquist rate type
1-clock type, 10-33–10-40
N-clock type, 10-40–10-46
2^N-clock type, 10-46–10-47
oversampled type
feedback path delay integrator, 10-48
first-order noise-shaping converter, 10-48–10-49
forward path delay integrator, 10-48
higher resolution, 10-47
linear system model, 10-47–10-48
magnitude response, 10-49
noise-shaping type, 10-47–10-48
oversampling ratio (OSR), 10-47
pattern noise, 10-50
SNR, 10-51
test techniques
crossplot technique, 10-32–10-33
differential nonlinearity error, 10-32
gain error, 10-30
integral nonlinearity error, 10-31
offset error, 10-31
transfer characteristics, 10-29–10-30
Anceau's PLL scheme, 9-64–9-65
Avalanche noise, 3-52–3-53

B

Backward difference transformation, 5-21
Balanced binary tree (BBT), 9-59–9-61
Barrel shifter, 9-76–9-77
BiCMOS amplifiers, 2-95–2-96
Bilateral *z*-transform (BZT), 7-30–7-31
Bipolar integrated circuit design, 2-1
Bipolar junction transistor
current gain
base–emitter voltage, 1-4, 1-6
base transport and emitter injection efficiency, 1-6
gain–current relationship, 1-6–1-7
Ebers–Moll model, 1-2–1-3
Gummel–Poon model, 1-4–1-5
high-current phenomena, 1-7–1-8
integrated NPN transistor, 1-11–1-12
lateral and vertical PNP transistor, 1-12
second-order effects, 1-14–1-15
SiGe HBTs
collector and base currents *vs.* EB voltage, 1-17
cutoff frequency *vs.* collector current, 1-18
energy band diagram, 1-17
industry practice and fabrication technology, 1-19–1-20
measured doping and Ge profile, 1-17–1-18
operation principle and performance advantage, 1-18–1-19
small-signal model, 1-9–1-10
SPICE model, 1-15–1-16
thermal sensitivity, 1-13–1-14
Bipolar noise
avalanche noise, 3-52–3-53
burst noise–RTS noise, 3-51–3-52
$1/f$ noise, 3-50–3-51
generation–recombination noise, 3-49–3-50
noise characterization
equivalent noise resistance and noise temperature, 3-54–3-55
equivalent noise voltage and current, 3-53–3-54
noise figure, 3-55
noise $1/f^2$, 3-51
shot noise, 3-49
thermal noise, 3-48–3-49
Bipolar transistor, 3-53–3-54

Broadband bipolar networks
bipolar transistor modeling, high frequency
hybrid π model, 3-2–3-3
Miller approximation, 3-3
modified equivalent circuit, 3-4
simplified high-frequency model, 3-3
split current source, 3-4
broadband amplifier stability
classical feedback system review, 3-31
criteria, 3-31–3-32
grounded capacitor compensation, 3-35–3-36
high-frequency performance, 3-36
input capacitance to ground inversion, 3-40
Miller compensation and pole separation, 3-33
op-amp internal compensation strategy, 3-31
phase lag neutralization, 3-38–3-39
power supply impedance, 3-37
resistive and capacitive load effects, 3-37–3-38
single-stage op-amp compensation, 3-34–3-35
two-stage op-amp architecture, 3-32–3-33
C_μ neutralization, 3-9–3-10
current conveyor
applications, 3-12–3-13
broadband analog amplifier, 3-12
current–voltage transfer relationship, 3-12
first-generation current conveyor (CCI), 3-12, 3-15–3-16
second-generation current conveyor (CCII), 3-12, 3-16–3-18
single BJT CCII–, 3-13–3-14
supply-current sensing, voltage op-amp, 3-14–3-15
current-feedback operation amplifier
analysis, 3-22–3-23
architecture, 3-18–3-19
basic current mirror, 3-26–3-28
closed-loop inverting operation, 3-21–3-22
closed-loop noninverting operation, 3-20–3-21
design and development, 3-18
differential-mode operation, 3-19–3-20
feedback current, 3-20
high-speed performance, 3-18
improved broadband current mirror, 3-28–3-29
input stage, 3-44–3-45
phase linearity, 3-29
pole frequency comparison, 3-23–3-24
practical considerations, broadband design, 3-30–3-31
R_2 value, 3-29–3-30
slow rate, 3-24–3-25
wideband and high-gain, 3-25–3-26
Miller's theorem, 3-2
negative feedback, 3-10–3-11
RF bipolar transistor layout, 3-11
single-gain stages
common-base (CB) stage, 3-8–3-9
common-collector (CC) stage, 3-6–3-8
common-emitter (CE) stage, 3-4–3-6
transfer function and bandwidth characteristic
current-feedback, 3-41–3-43
voltage-feedback, 3-43–3-44
Widlar current mirror, 3-45–3-47
Bulk–drain depletion capacitance, 1-54
Bulk-induced modulation, 1-37
Bulk transconductance, 1-60
Butterfly operations, 7-14

C

Cascade voltage switch logic (CVSL) gate, 9-80–9-81
Cauer form, 6-13
CC–CE stage, 3-10
CE–CB cascode stage, 3-9
Channel length modulation effect
drain current equation, 1-88–1-89
JFET technology and device, 1-87
super MOS transistor, 2-90
Circular convolution theorem, 7-11–7-12
Circular spiral inductors
concentric type, 1-127
mutual inductance, 1-129–1-130
self-inductance, 1-127–1-129
total inductance, 1-127
Class *AB* current conveyor, 3-15
Classical feedback system, 3-51
Clocked CMOS logic (C^2MOS), 9-78–9-79
Clocking schemes
current-steered logic and dedicated third-layer, 9-69
distribution
Anceau's PLL scheme, 9-64–9-65
balanced binary tree (BBT), 9-59–9-61
clock ring configuration, 9-56–9-57
clock trunk concepts, 9-53–9-56
delay, skew, and rise time comparison, 9-59
four-quadrant approaches, 9-50
Grover's interval-halving PLL scheme, 9-66–9-68
H-trees, 9-57–9-59
phase-locked loops, 9-61–9-63
PLLs, CMOS, 9-63–9-64
single-driver configurations, 9-48–9-49
symmetric and generalized clock buffer trees, 9-50–9-53
tuning, large systems, 9-68–9-69
mixed technology, 9-69

optoelectronic clock distribution, 9-69–9-70
principles
clock signal manipulation, 9-46–9-47
controlled skew introduction, 9-45–9-46
delay minimization, 9-47–9-48
dynamic logic, 9-42–9-44
flip-flops, 9-42
isochronic/equipotential regions, 9-35–9-36
latches, 9-41–9-42
multiple-phase overlapping clocks, 9-40
overlapping clock phase generator, 9-40–9-41
single-phase type, 9-37–9-38
skew and delay, 9-34–9-35
skew on-chip nature, 9-36–9-37
synchronizers and metastability, 9-44–9-45
two-phase clock generator circuit, 9-39–9-40
two-phase type, 9-38–9-39
$\overline{Q}$ elimination, 9-69
reconfigurable clock nets, 9-69–9-70
voltage swing, 9-69
Clock recovery architecture
early–late block diagram, 5-28–5-29
edge-detection-based method, 5-29
integrator, 5-28–5-29
quadricorrelator, 5-29–5-30
waveforms, 5-28–5-29
Closed-loop transfer function, PLL, 5-3–5-4
CMOS
amplifier, 2-94–2-95
mixer topology
linear MOS mixers, 4-22–4-23
nonlinearity and LO-feedthrough analysis, 4-23–4-25
switching modulators, 4-21–4-22
RF integration, 4-11
Code-error calibration, DAC, 10-17
Common-base (CB) stage, 3-8–3-9
Common-collector (CC) amplifier
AC schematic diagram, 2-33
active load, 2-35–2-36
driving point input and output resistance, test circuit, 2-34
equivalent circuit, 3-6
equivalent high-frequency model, 3-8
output impedance, 3-7–3-8
Thévenin load resistance, 2-37
transfer function, 3-7
transform property, 3-8
voltage gain, 2-33
Common-emitter (CE) stage
equivalent circuit model, 3-5
high-frequency model, 3-4
Miller approximation, 3-4–3-6
right-hand-plane (RHP) zero, 3-6
second-order characteristic equation, 3-6
Complementary bipolar technology (CBT), 2-6
Complementary pass-transistor logic (CPL) gate, 9-80
Composite super NMOS transistors, 2-91–2-92
Compound semiconductor FET technologies
HEMT device
cross section structure, 1-99
drain current–drain voltage characteristic, GaN, 1-98–1-99
efficiency, 1-99
GaAs MESFET, 1-95–1-96
gate connected field plate, GaN, 1-100
microwave and mm-wave performance, 1-97
microwave power amplifier performance, 1-100
recessed gate AlGaAs/GaAs structure, 1-96
Schottky gate characteristics, 1-96
heterojunctions, AlGaAs and GaAs transition, 1-94–1-95
III–V compound semiconductors, 1-92–1-93
wide bandgap compound semiconductors, 1-97–1-98
Continuous-time Fourier transform
arbitrary aperiodic signal, 7-15
frequency-selective behavior, 7-17–7-18
properties, 7-16
spectral density magnitude, 7-16–7-17
spectral density, sinusoidal pulse, 7-17
Current-controlled current source (CCCS), 1-71–1-72
Current-controlled voltage source (CCVS), 1-72
Current conveyor
applications, 3-12–3-13
broadband analog amplifier, 3-12
current–voltage transfer relationship, 3-12
first-generation current conveyor (CCI), 3-12, 3-15–3-16
second-generation current conveyor (CCII), 3-12, 3-16–3-18
single BJT CCII–, 3-13–3-14
supply-current sensing, voltage op-amp, 3-14–3-15
Current crowding effect, 1-8
Current-feedback operation amplifier
analysis, 3-22–3-23
architecture, 3-18–3-19
basic current mirror, 3-26–3-28
closed-loop inverting operation, 3-21–3-22
closed-loop noninverting operation, 3-20–3-21
design and development, 3-18
differential-mode operation, 3-19–3-20
feedback current, 3-20
high-speed performance, 3-18
improved broadband current mirror, 3-28–3-29
phase linearity, 3-29
pole frequency comparison, 3-23–3-24
practical considerations, broadband design, 3-30–3-31
R_2 value, 3-29–3-30

slow rate, 3-24–3-25
wideband and high-gain, 3-25–3-26
Current-ratioed DAC, 10-8–10-11

D

Data converters
analog-to-digital converters (ADCs)
Nyquist rate type, 10-33–10-47
oversampled type, 10-47–10-57
test techniques, 10-29–10-33
digital-to-analog converters (DACs)
architectures, 10-6–10-12
design issues, 10-3–10-5
error sources, 10-22–10-25
high resolution techniques, 10-13–10-22
low-spurious design examples, 10-26–10-28
signal-to-noise ratio and dynamic range, 10-2
transfer characteristics, 10-1–10-2
Delay-locked loop (DLL)
block diagram, 5-30
modern digital system, synchronous communication, 5-31
PLL bandwidth control, 5-26
timing relationships, 5-31–5-32
voltage-controlled delay line (VCDL), 5-30
Differential amplifier
C_μ neutralization, gain stage, 3-10
voltage gain cells, 2-96–2-97
Differential nonlinearity (DNL), 10-3
Diffused resistors
avalanche breakdown mechanism, 1-107
isolation region, 1-106
n^+ diffusion layer, 1-108–1-109
normalized frequency response, 1-107–1-108
n-type emitter-diffused resistor, 1-107–1-109
p-type resistor and n-type epitaxial (epi) region, 1-106–1-107
Digital circuits and systems
architecture, 9-33
clocking schemes
current-steered logic and dedicated third-layer, 9-69
distribution, 9-48–9-69
mixed technology, 9-69
optoelectronic clock distribution, 9-69–9-70
principles, 9-33–9-48
$\overline{Q}$ elimination, 9-69
reconfigurable clock nets, 9-69–9-70
voltage swing, 9-69
microprocessor-based design
architecture, 9-92–9-95
features, 9-83–9-86
with general purpose microprocessor, 9-95–9-100
guidelines, 9-110–9-111
interfacing, 9-100–9-105
memory, 9-86–9-92
with microcontroller, 9-105–9-110
MOS logic circuits
CMOS inverter, 8-8–8-10
digital inverter, 8-4–8-6
dynamic CMOS logic gates, 8-14–8-15
MOSFET models, 8-1–8-4
nMOS logic gates, 8-6–8-8
static CMOS logic gates, 8-11–8-13
MOS storage circuits
dynamic charge storage, 9-72–9-75
dynamic CMOS logic, 9-78–9-82
shift register, 9-75–9-77
programmable logic devices (PLDs)
combinational logic (CL) PAL devices, 9-11–9-16
combinational logic (CL) PLD classification, 9-10
complexity device ladder, 9-1–9-2
design process, 9-24–9-27
FPGA architectures, 9-22–9-25
notation, 9-4–9-5
programmable array logic (PAL), 9-8–9-10
programmable logic array (PLA), 9-5–9-6
programmable macrocell outputs, 9-18–9-21
programmable read only memory (PROM), 9-7–9-8
sequential PAL devices, 9-16–9-18
state machines synthesis, 9-30–9-31
technologies, 9-2–9-4
VHDL synthesis style, FPGA, 9-27–9-30
systolic arrays
concurrency, parallelism, pipelining, 9-111–9-117
digital filters, 9-117–9-122
eigenvalue and SVDs, 9-137–9-142
Kalman filtering (KF), 9-132–9-137
recursive LSs estimation, 9-126–9-132
systolic word and bit-level designs, 9-122–9-126
transmission gates (TG)
analog processing, 8-24–8-33
complementary transistor version, 8-17–8-18
continuous time filters, 8-27–8-28
digital processing, 8-15–8-24
MOS operational amplifier compensation, 8-24–8-26
pass-transistor logic, 8-18–8-24
single transistor version, 8-15–8-17
switched-capacitor circuits, 8-28–8-33
transimpedance compensation, 8-26–8-27
Digital signal processing
bilateral z-transform (BZT), 7-30–7-31
continuous-time Fourier transform
arbitrary aperiodic signal, 7-15
frequency-selective behavior, 7-17–7-18

properties, 7-16
sinusoidal pulse, spectral density, 7-17
spectral density magnitude, 7-16–7-17
convolution operation, 7-26
discrete Fourier transform (DFT)
aliasing error, 7-9–7-11
applications, 7-11–7-12
discrete Fourier series (DFS) pair, 7-8
Hann window, 7-11
inverse DFT (IDFT), 7-7–7-8
leakage error, 7-10
magnitude spectrum, bandlimited signal, 7-8
periodic discrete-time signal, 7-6
quantization error, 7-6
rectangular window, 7-11
sampling frequency, 7-9
spectral properties, continuous-time periodic signal, 7-8
X(k) magnitude spectrum, 7-9–7-10
discrete-time Fourier transform (DTFT)
Dirac impulse function, 7-20
periodic function and FS coefficients, 7-19
properties, 7-23
sampling process model, 7-18, 7-20
discrete-time signals, 7-5
fast Fourier transform, 7-13–7-14
Fourier series, continuous-time periodic signals
approximation error, 7-2
complex FS coefficients, 7-3–7-4
cyclical phenomena, 7-4
Dirichlet conditions, 7-2
Euler's identity, 7-3
fundamental frequency, 7-2
Gibbs oscillation, 7-4–7-5
mean square error minimization, 7-2
periodic signal, 7-1–7-2
frequency response, 7-26–7-28
ideal digital filters, 7-29–7-30
linear and time invariant discrete-time systems
block diagram, 7-23–7-24
characteristic equation, 7-25
initial conditions, 7-24–7-25
off-line processing, 7-24
unit pulse response, 7-25–7-26
sampling theorem
aliasing error, 7-21–7-22
antialiasing filter, 7-21–7-22
data reconstruction formula, 7-21
DTFT magnitude, 7-20–7-22
exponentially weighted sinusoidal pulse, 7-22
FT magnitude, 7-22
low-pass filter magnitude frequency response, 7-21
sampling frequency, 7-20–7-21
spectral density magnitude, bandlimited signal, 7-20
stability, 7-26
transfer function, 7-33
unilateral *z*-transform, 7-33–7-35
z-plane, 7-31–7-32
Digital-to-analog converters (DACs)
architectures
current-ratioed type, 10-8–10-11
R+C/C+R combination type, 10-11–10-12
resistor-string type, 10-6–10-8
design issues
conversion speed, 10-5
linearity, 10-3
monotonicity, 10-3–10-5
error sources
glitch, 10-22–10-23
noise, 10-25
timing error-word clock jitter, 10-23–10-24
voltage reference, 10-24–10-25
high resolution techniques
dynamic matching type, 10-13–10-14
electronic calibration type, 10-14–10-18
interpolative oversampling type, 10-18–10-22
low-spurious design examples
dynamic linearity enhancement, 10-27–10-28
self-trimming, 10-26–10-27
spatial averaging, 10-26
signal-to-noise ratio and dynamic range, 10-2
transfer characteristics, 10-1–10-2
Digital truncation errors, 10-18
Diode equation, 1-2
Direct code mapping, 10-15
Direct frequency synthesizer, 4-17
Discrete-time index, 7-4
Discrete-time transfer functions, 5-21
Down converter, 4-15–4-16
Drain–source channel resistance, 1-60

E

Ebers–Moll model, 1-2–1-3
Edge detection, NRZ data, 5-29
Electronic calibration type, DAC
capacitor ratio error measurement cycles, 10-16
code-error calibration, 10-17
current difference measurement cycles, 10-16–10-17
digital truncation errors, 10-18
direct code mapping, 10-15
self-calibration for individual capacitor errors, 10-15–10-16
Elmore's approach, 6-15
Emitter follower, *see* Common-collector (CC) amplifier
Epitaxial resistors, 1-111–1-112
Extended RF transistor model, 4-9–4-10

F

Fast Fourier transform (FFT), 7-13–7-14
FET/current mirror bias circuit, 2-63
Fialkov condition, 6-12
Field programmable gate arrays (FPGAs)
architectures
high-level layout, 9-22
LUT, 9-24
minimal CLB, 9-23–9-24
programmable elements, 9-23

XC4010XL chip, CLB, 9-24–9-25
technologies, 9-22
VHDL synthesis style
combinational logic, 9-29
latches, 9-30
registers and flip-flops, 9-27–9-29
Finite impulse response (FIR) filters, 7-28, 9-117–9-118
First-order infinite impulse response filter, 5-22
1/*f* noise, 3-50–3-51
Folded cascode amplifiers, biasing circuits, 2-72–2-73
Forward transconductance, 1-60
Fourier series, 5-7
Fractional-*N* synthesizers, 4-19–4-20
Frequency locked-loop (FLL), 5-9
Frequency synthesizer
block diagrams, 5-32–5-33
dual-modulus prescalar, 5-33–5-34
output frequency, 5-32

G

Gain-boosting principle, 2-90–2-91
Generation–recombination noise, 3-49–3-50
Generation–recombination phenomena, 1-4
Gibbs oscillation, 7-4–7-5
Grover's interval-halving PLL scheme, 9-66–9-68
Gummel–Poon model, 1-4–1-5

H

Hestenes algorithm, 9-140–9-141
High-speed voltage buffer, 3-37

I

Ideal digital filters, 7-29–7-30
Indirect convolution method, 7-12
Infinite impulse response (IIR) filters, 7-28–7-29
bidirectional systolic arrays, 9-121
clustered look-ahead, 9-119–9-120
I–O relationship, 9-118
overlapped subfilter scheme, 9-122
scattered look-ahead, 9-120–9-121
systolic ring scheme, 9-122
Input-referred third-order intercept point (IIP3), 4-5
Integral nonlinearity (INL), 10-3
Integrated PNP transistors, 1-12
Integrator-type DACs, 10-4–10-5
Intermodulation distortion (IMD), 4-4–4-5
Interpolative oversampling technique
alternative 1 bit DAC sampling constant, 10-21–10-22
delta-sigma modulation, 10-18–10-19
dynamic range, 10-20
one-bit/multibit, 10-20
postfiltering requirement, 10-22
stability, 10-19–10-20
switched-capacitor 1 bit DAC/filter, 10-20–10-21
Ion-implanted resistors, 1-112–1-113

J

JFET technology and devices
channel-length modulation effect, 1-87
ion implanted silicon JFET, IC process, 1-91
large-signal model, drain current equations, 1-88–1-89
operating regions
cutoff and subthreshold current regions, 1-86–1-87
ohmic and pinch-off region, 1-85–1-86
static current–voltage characteristics, 1-85
small-signal model, 1-89–1-90
static *I–V* characteristics, 1-84
temperature effects, 1-87–1-88
JK-flipflop PD, 5-13–5-14
Junction capacitors
abrupt pn junction, 1-115–1-116
base–collector capacitor structure, 1-117–1-118
base–emitter capacitor structure, 1-117–1-119
depletion width, 1-116–1-117
permittivity, 1-116

K

Kalman filtering (KF)
Faddeev algorithm, 9-134–9-137
model, 9-132–9-133
other forms, 9-133–9-134
systolic matrix implementation, 9-134
Kirchhoff's current law, 1-3, 3-21
Kirchhoff's voltage law, 2-5
Kirk effect, 1-7–1-8

L

Linear bipolar technology, canonic cells
balanced differential amplifier
engineering constraints, 2-45–2-47
generalized system diagram, 2-44–2-45
monolithic fabrication process, 2-45
single-ended output voltage, 2-44
common-base amplifier
circuit broadbanding, 2-32
common-emitter–common-base cascode, 2-30–2-32
current buffering purpose, 2-26–2-27
diode resistance, 2-28
driving point input/output resistance, 2-29–2-30
equivalent circuit, 2-27–2-28
Kirchhoff's current law constraint, 2-27
Miller multiplication, 2-32
Norton transconductance, 2-31
NPN and PNP AC schematic diagrams, 2-26–2-27
small-signal analysis, 2-27
voltage divider, 2-31

common-collector amplifier
AC schematic diagram, 2-33
active load, 2-35–2-36
driving point input and output resistance, test circuit, 2-34
Thévenin load resistance, 2-37
voltage gain, 2-33
common-emitter amplifier
active current source load, 2-24–2-26
driving point output resistance, 2-23
macromodel, 2-19–2-20, 2-23
Norton current and equivalent circuit, 2-19
NPN and PNP AC schematic diagrams, 2-17, 2-20–2-21
operation, 2-17
output coupling capacitance, 2-24
small-signal test structure, 2-18–2-19
Thévenin source voltage, 2-18
common-mode input voltage, 2-41–2-42
Darlington connection
forward transconductance, 2-41
schematic diagram, 2-38–2-39
small-signal equivalent circuit, 2-39–2-40
transconductance amplifier, 2-38
transconductance frequency response, 2-38–2-39
voltage gain, 2-40–2-41
Wilson mirror load, 2-39
differential input source voltage, 2-41
diode-connected transistor
KVL analysis, 2-15
small-signal transistor model, 2-14–2-15
static common-base current gain, 2-15–2-16
subject diagram, 2-14
V_{BE} multiplier, 2-15–2-17
volt–ampere characteristics, 2-14
performance index, 2-43
small-signal model, 2-11–2-14
system-level diagram, differential amplifier, 2-41–2-42
Thévenin equivalent I/O circuits
balanced bipolar differential amplifier, 2-52–2-53
driving point common-mode/differential mode output resistance, 2-49–2-50
Kirchhoff's current and voltage laws, 2-47
open-circuit differential-mode gain, 2-51
pertinent test cell, 2-48–2-49
test circuit, 2-47–2-49
Thévenin model, 2-51
two-port model, 2-54
zero input signal excitation, 2-49
Linear convolution method, 7-12
Load capacitance neutralization, 3-38–3-39
Long-tail pair input transconductance, 3-24–3-25
Loop filter (LF) gain, 5-8
Low-noise amplifier (LNA)
cutoff frequency, 4-14
drain current equation, 4-14
IIP2 and IIP3 plots, 4-15–4-16
noise and source impedance matching, 4-14
noise figure, 4-12–4-14
third-order intermodulation, 4-14
topology, 4-12–4-13
transit voltage, 4-15
Low-pass filter (LPF)
PLL techniques, 5-2–5-3
two-stage op-amp architecture, 3-32

M

Matched transistors, 2-2–2-3
Metal–oxide–silicon field effect transistor (MOSFET) technology
charge storage, 1-24–1-25
cutoff regime, 1-38
depletion capacitance, 1-53–1-54
depletion zone analysis
body effect voltage, 1-35–1-36
charge density, 1-33–1-34
electric field intensity, 1-34–1-35
Gauss' law, 1-33
volt–ampere characteristics, 1-32–1-33
design-oriented analysis strategy
biasing, 1-77–1-78
circuit structure, 1-78
comments, 1-80–1-81
forward static transfer characteristic, 1-79–1-81
parameterization process, 1-79
transconductance coefficient, mobility degradation, 1-77–1-78
gate–bulk capacitance
characteristics, 1-32–1-33
depletion layer, 1-31
N-channel MOSFET, 1-30
pertinent equivalent circuit, 1-29
silicon dielectric constant, 1-29, 1-31
surface capacitance density, 1-29
Kirchhoff 's voltage law, 1-24
large-signal model, 1-56–1-57
lateral electric fields
carrier mobility degradation, 1-48–1-49
carrier velocity, 1-46–1-47
critical electric field, 1-46
drain saturation voltage, 1-48
Level 49 HSPICE model, 1-51
modulation voltage, 1-48
voltage and current correction factors, 1-49–1-50
ohmic regime
channel potential, 1-39–1-40
cross section, 1-38–1-39
drain saturation voltage, 1-38
pinched off channel, 1-40–1-41
static circuit model, 1-41–1-42
transconductance coefficient, 1-40

saturation regime
built-in potential, 1-44
channel length modulation voltage, 1-43
common-source volt–ampere characteristic curves, 1-45
drain current, 1-42, 1-44
large-signal circuit model, 1-44
strong inversion, 1-42–1-43
small-signal model
analysis, 1-58–1-59, 1-71
bulk-gate transconductance, 1-68
bulk modulation factor, 1-62, 1-69
bulk transconductance, 1-60, 1-62
common-source interconnection, 1-70
equivalent circuit, 1-70
forward transadmittance, 1-71
forward transconductance, 1-60–1-62
HSPICE model, 1-66
N-channel MOSFET operation, 1-59–1-60
P-channel MOSFET operation, 1-59–1-61
radial signal frequency, 1-68
scattering parameters, 1-66–1-67
short circuit admittance parameters, y_{ij}, 1-66–1-67
signal drain current, 1-69
simulating results, 1-74–1-76
three-port network, 1-66
VCCS synthesis, 1-71–1-72
surface charge density
Boltzmann's constant, 1-26
equilibrium condition, 1-27
Fermi potential, 1-26, 1-28
Gauss' law, 1-27
magnitude, 1-28
surface electron concentration, 1-27–1-28
temperature effects, 1-52–1-53
threshold condition, 1-36–1-37
unity gain frequency
radio frequency choke (RFC), 1-63
small-signal equivalent model, 1-63–1-64
voltage-controlled current source (VCCS), 1-64–1-65
vertical electric fields, 1-51–1-52
Metal–oxide–silicon (MOS) technology
capacitors, 1-119–1-120
current bias circuit, 2-68
M68HC11 microcontroller
circuit diagram, 9-106
memory map and sample ROM content, 9-107–9-108
operational modes, 9-106
programming model, 9-105–9-106
three button and four-digit LCD display device, 9-109–9-110
timing diagram, 9-106–9-107
Microprocessor-based design
architecture
bit level activity, ALU, 9-93
data paths, 9-93–9-94
other registers, 9-93–9-94
register-to-register transfer activity, accumulator, 9-92
binary cell (BC)
2 bits/word programmable ROM, 9-88–9-89
cascade and parallel construction, RAM module, 9-88
circuit, 9-86
EEPROM, 9-91
8K × 8 EPROM and package pin assignment, 9-89–9-90
map, sample ROM content, and assembly language source, 9-91–9-92
module, 9-87–9-88
multiword read/write memory circuit, 9-86–9-87
PROM, 9-90
unidirectional to bidirectional bus conversion, 9-86–9-87
features
block diagram, 9-85–9-86
conceptual diagram, 9-83–9-84
with general purpose microprocessor (Z80)
clock cycles, 9-99–9-100
control signals tasks, 9-97–9-98
pin assignment, 9-96–9-97
programming model, 9-95–9-96
schematic diagram, 9-99
timing diagrams, 9-98–9-99
guidelines, 9-110–9-111
interfacing
D/A converter, 9-102
daisy chain, 9-104
I/O ports with handshaking, 9-103
opto-isolated power control circuits, 9-101–9-102
switches, 9-102
timer-counter circuit, 9-104–9-105
wire OR'd circuit, 9-104
write controlled A/D, 9-102
with microcontroller (M68HC11)
circuit diagram, 9-106
memory map and sample ROM content, 9-107–9-108
operational modes, 9-106
programming model, 9-105–9-106
three button and four-digit LCD display device, 9-109–9-110
timing diagram, 9-106–9-107
Miller effect, 3-9
Monolithic device models
bipolar junction transistor
base–emitter voltage, 1-4, 1-6
base transport efficiency, 1-6
collector and base currents *vs.* EB voltage, 1-17
cutoff frequency *vs.* collector current, 1-17–1-18
Ebers–Moll model, 1-2–1-3
emitter injection efficiency, 1-6

energy band diagram, 1-17
gain–current relationship, 1-6–1-7
Gummel–Poon model, 1-4–1-5
high-current phenomena, 1-7–1-8
industry practice and fabrication technology, 1-19–1-20
integrated NPN transistor, 1-11–1-12
lateral and vertical PNP transistor, 1-12
measured doping and Ge profile, 1-17–1-18
operation principle and performance advantage, 1-18–1-19
second-order effects, 1-14–1-15
small-signal model, 1-9–1-10
SPICE model, 1-15–1-16
thermal sensitivity, 1-13–1-14
HEMT devices
cross section structure, 1-99
drain current–drain voltage characteristic, GaN, 1-98–1-99
GaAs MESFET, 1-95–1-96
gate connected field plate, GaN, 1-100
microwave and mm-wave performance, 1-97
microwave power amplifier performance, 1-100
recessed gate AlGaAs/GaAs structure, 1-96
Schottky gate characteristics, 1-96
heterojunctions, AlGaAs and GaAs transition, 1-94–1-95
III–V compound semiconductors, 1-92–1-93
JFET technology and devices
channel-length modulation effect, 1-87
cutoff and subthreshold current regions, 1-86–1-87
ion implanted silicon JFET, IC process, 1-91
large-signal model, drain current equations, 1-88–1-89
ohmic and pinch-off region, 1-85–1-86
silicon, operation, 1-82–1-83
small-signal model, 1-89–1-90
static current–voltage characteristics, 1-85
static *I–V* characteristics, 1-84
temperature effects, 1-87–1-88
metal–oxide–silicon field effect transistor (MOSFET) technology
biasing, 1-77–1-78
body effect voltage, 1-35–1-36
carrier mobility degradation, 1-48–1-49
carrier velocity, 1-46–1-47
charge density, 1-33–1-34
charge storage, 1-24–1-25
circuit structure, 1-78
critical electric field, 1-46
cutoff regime, 1-38
depletion capacitance, 1-53–1-54
depletion layer, 1-31
drain saturation voltage, 1-48
electric field intensity, 1-34–1-35
forward static transfer characteristic, 1-79–1-81
Gauss' law, 1-33
Kirchhoff 's voltage law, 1-24
large-signal model, 1-56–1-57
Level 49 HSPICE model, 1-51
modulation voltage, 1-48
N-channel MOSFET, 1-30
parameterization process, 1-79
pertinent equivalent circuit, 1-29
radio frequency choke (RFC), 1-63
silicon dielectric constant, 1-29, 1-31
small-signal equivalent model, 1-63–1-64
surface capacitance density, 1-29
temperature effects, 1-52–1-53
threshold condition, 1-36–1-37
transconductance coefficient, mobility degradation, 1-77–1-78
vertical electric fields, 1-51–1-52
voltage and current correction factors, 1-49–1-50
voltage-controlled current source (VCCS), 1-64–1-65
volt–ampere characteristics, 1-32–1-33
wide bandgap compound semiconductors, 1-97–1-98
MOSFET biasing circuits
CMOS technology
BJT, 2-57–2-58
circuit design, 2-55
diffusion resistors realization, 2-58–2-59
integrated circuit implementation, 2-56
N-well and P-well processes, 2-58
principal devices, 2-56–2-57
voltage reference, 2-56
device models and parameter variability
proportional to absolute temperature (PTAT), 2-62
temperature dependence, 2-59, 2-61
threshold voltage, 2-60–2-61
dynamic biasing, 2-75–2-76
low power supply voltage, 2-74–2-75
N- and P-doped polysilicon gate threshold, 2-70
simple amplifiers and other circuits
current mirrors, 2-73
folded cascode amplifiers, 2-72–2-73
single-stage amplifier, cascode loads, 2-71–2-72
two-stage amplifier, 2-70–2-71
voltage and current references
bandgap principle, bipolar technology, 2-66
BJT V_{be}-based references, 2-64–2-65

BJT V_T-based references, 2-65–2-66
curvature-compensated bandgap references, 2-67
discrete time bandgap references, 2-67–2-68
enhancement and depletion-mode threshold voltage difference, 2-69–2-70
lateral bipolar devices, 2-68–2-69
MOSFET threshold, 2-63
operational amplifier, 2-66
subthreshold region, 2-68
supply-voltage, 2-62–2-63
MOSFET technology, canonical cells
BiCMOS amplifiers, 2-95–2-96
CMOS amplifier, 2-94–2-95
composite transistors
BiCMOS, 2-84–2-85
bidirectional, 2-86–2-87
body effect, threshold voltage, 2-84
diode leakage current, 2-89
drain current, 2-87–2-88
equivalent circuit, 2-86–2-87
I_D curves, 2-86, 2-88
MOS-folded composite transistors, 2-85–2-86
physical cross section, 2-86–2-87
simulation program with integrated circuit emphasis (SPICE) model, 2-86, 2-90
various bulk connections, 2-86–2-87
V_{S1} curves, 2-87–2-88
differential amplifier, 2-96–2-97
folded-cascode operational amplifier, 2-98
matched device pairs
composite MOSFET (COMFET) circuits, 2-79
current mirrors, 2-79
differential pairs, 2-78–2-79
drain and differential current, 2-78
operation, saturation region, 2-78–2-79
transistor pairs operation, triode region, 2-79–2-81
voltage follower, 2-78–2-79
NMOS amplifier, 2-92–2-94
super MOS transistors
composite super NMOS transistors, 2-91–2-92
output impedance, 2-90–2-91
regulated current mirrors, 2-90
simulated I_D curves, 2-91, 2-93
unmatched device pairs
CMOS composite transistor, 2-82–2-83
CMOS inverter, 2-82, 2-84
drain current, 2-82
parallel and series composite NMOS transistors, 2-81–2-82
MOS logic circuits
CMOS inverter
circuit and switch model, 8-8–8-9
power dissipation, 8-10
VTC, 8-9–8-10
digital inverter
switching times, 8-5–8-6
symbol and electronic parameters, 8-4
voltage transfer characteristics (VTC), 8-5
dynamic CMOS logic gates
charge sharing problem, 8-14–8-15
three-input NAND gate and timing intervals, 8-14
MOSFET models
capacitances, 8-2–8-3
primary device voltages, 8-1–8-2
switching models, 8-3–8-4
symbols, 8-3
nMOS logic gates
AOI gates, 8-7–8-8
configurations, 8-6–8-7
NOR and NAND gates, 8-7–8-8
threshold voltage loss, 8-7
static CMOS logic gates
AOI gate and XOR circuit, 8-11–8-12
NAND and NOR gates, 8-11–8-12
pseudo-nMOS logic circuits, 8-12
TG-based 2:1 multiplexer, 8-12–8-13
transmission gate (TG), 8-12–8-13
XOR and XNOR gates, 8-13
MOS storage circuits
dynamic charge storage
charge sharing, 9-74–9-75
nMOS–nMOS, 9-73
nMOS–pMOS, 9-73–9-74
pMOS–pMOS, 9-73
source–drain connection storage nodes, 9-72–9-73
source–gate connection, 9-74
dynamic CMOS logic
cascade voltage switch logic (CVSL) gate, 9-80–9-81
clocked CMOS logic (C^2MOS), 9-78–9-79
complementary pass-transistor logic (CPL) gate, 9-80
domino CMOS logic, 9-79–9-80
NORA, 9-81–9-82
precharge–evaluate logic gate, 9-78
shift register
clocked barrel shifter, 9-76–9-77
parallel type, 9-76
simple type, 9-75–9-76
MOS transistor, 4-15

N

Natural frequency, PLL, 5-23–5-24
N-channel MOSFET (NMOS), 1-22
Network realization, 6-4–6-5
NMOS amplifier
common-source, enhancement load, 2-92–2-93
gain, 2-94
small-signal equivalent circuit, 2-94
Noise spectral density, 3-50–3-51
Noise temperature, 3-55
Noninverting amplifier, voltage-feedback op-amp, 3-23–3-24
Non-return-to-zero (NRZ)
data format, 5-27–5-28
phase and frequency detectors, 5-15

NORA CMOS dynamic logic, 9-81–9-82
Nyquist equation, 3-48
Nyquist rate converters
1-clock type
digital-error correction, 10-38–10-40
flash/parallel type, 10-33–10-35
pipelined type, 10-36–10-38
subranging converters, 10-33–10-34, 10-36
N-clock type
algorithmic type, 10-43–10-46
self-calibration successive approximation type, 10-42–10-43
successive approximation type, 10-40–10-42
2^N-clock type, 10-46–10-47

O

Ohmic regime
channel potential, 1-39–1-40
cross section, 1-38–1-39
drain saturation voltage, 1-38
pinched off channel, 1-40–1-41
static circuit model, 1-41–1-42
transconductance coefficient, 1-40
Operational amplifier (op-amp), rail-to-rail operation, 2-74
Optoelectronic clock distribution, 9-69–9-70
Output-referred third-order intercept point (OIP3), 4-5
Oversampled converters, ADC
feedback path delay integrator, 10-48
first-order noise-shaping converter, 10-48–10-49
forward path delay integrator, 10-48
higher resolution, 10-47
linear system model, 10-47–10-48
magnitude response, 10-49
noise-shaping type, 10-47–10-48
oversampling ratio (OSR), 10-47
pattern noise, 10-50
SNR, 10-51

P

Passive components
circular spiral inductors
concentric type, 1-127
mutual inductance, 1-129–1-130
self-inductance, 1-127–1-129
total inductance, 1-127
conductivity, 1-103–1-104
diffused resistors
avalanche breakdown mechanism, 1-107
isolation region, 1-106
n^+ diffusion layer, 1-108–1-109
normalized frequency response, 1-107–1-108
n-type emitter-diffused resistor, 1-107–1-109
p-type resistor and n-type epitaxial (epi) region, 1-106–1-107
electron and hole mobility *vs.* impurity concentration, 1-104–1-105
epitaxial resistors, 1-111–1-112
ion-implanted resistors, 1-112–1-113
junction capacitors
abrupt pn junction, 1-115–1-116
base–collector capacitor structure, 1-117–1-118
base–emitter capacitor structure, 1-117–1-119
depletion width, 1-116–1-117
permittivity, 1-116
MOS capacitors, 1-119–1-120
n-type nonuniformly doped resistor, 1-104
parallel-plate capacitor, 1-114
pinched resistors, 1-110–1-111
polysilicon capacitors, 1-120–1-121
rectangular spiral inductors
electrical model, 1-124–1-125
mutual inductance, 1-122–1-124
self-inductance, 1-122–1-123
self-resonant frequency, 1-125
total inductance, 1-122
transformer structure, 1-125–1-126
sheet resistance, 1-103–1-104, 1-106
thin-film resistors, 1-113–1-114
uniformly doped resistor, 1-103
Passive lag filter, 5-4
P-channel MOSFET (PMOS), 1-22–1-23
Phase detector (PD), 5-1–5-2
Phase-error transfer function, 5-4
Phase-frequency detector (PFD), 5-2
Phase-locked loop (PLL) circuits
applications
clock recovery architecture, 5-28–5-30
data conversion, 5-28
data format, 5-27–5-28
delay-locked loop, 5-30–5-32
frequency synthesizer, 5-32–5-34
basic operation concepts, 5-1–5-2
charge-pump
block diagram, 5-22
closed-loop transfer function, 5-23
LF schematic, 5-22–5-23
normalized natural frequency, 5-23
purpose, 5-22
static phase error, 5-23–5-24
classification, 5-2
continuous-time loop filter (LF)
Bode plots, 5-16–5-17
frequency response, 5-17–5-18
high and low gain loop, 5-17
second-order loop *vs.* damping factors, 5-17–5-18
transfer function, 5-16–5-17
types, 5-16
definition, 5-1
design considerations
adaptive-bandwidth, 5-25

linear model, charge-pump PLL, 5-26
process, voltage, and temperature variations, 5-24–5-25
timing recovery, 5-26
typical procedures, 5-24
wide-range LC-tank VCO, 5-25
discrete-time loop filter (LF), 5-21–5-22
frequency synthesizer, 4-17–4-18
phase and frequency detectors
JK-flipflop, phase tracking, 5-13–5-14
operation, 5-15
waveforms, 5-15
XOR, 5-13–5-14
s-domain to *z*-domain transformations
backward difference method, 5-19–5-20
bilinear transformation method, 5-20–5-21
rectangular area approximation, 5-19–5-20
trapezoidal integration method, 5-20–5-21
techniques
acquisition process, 5-8–5-9
basic topology, 5-2–5-3
lock-in process, 5-7–5-8
loop orders, 5-3–5-4
noise performance, 5-9–5-10
tracking process, 5-4–5-6
voltage-controlled oscillators
harmonic oscillator, 5-11
resistive tuning, 5-12
ring oscillator topology, 5-11
Poisson's equation
JFET static *I–V* characteristics, 1-84
surface charge density, 1-26
Pole frequency
current-feedback operations amplifier, 3-23–3-24
two-stage op-amp architecture, 3-33
Polysilicon capacitors, 1-120–1-121
Potential divider, 2-62
Power added efficiency (PAE), 4-27
Power amplifier (PA)
CMOS RF amplifier
linearization techniques, 4-27–4-29
technology, 4-25
switching class E amplifier, 4-25–4-27
Prescaler, 4-19
Programmable array logic (PAL)
architecture, 9-8–9-9
example, 9-9–9-10
Programmable logic array (PLA)
architecture, 9-5
programmed chip, 9-6
Programmable logic devices (PLDs)
combinational logic (CL) classification, 9-10
combinational logic PAL devices
examples, 9-14–9-16
function implementation, 9-11, 9-13
implementation range, 9-14–9-15
PAL16L8 chip logic diagram, 9-11–9-12
complexity device ladder, 9-1–9-2
design process
design entry modes, 9-25–9-26
flow diagram, 9-24–9-26
logic synthesis, 9-26–9-27
mapping and simulation, 9-27
FPGA architectures
high-level layout, 9-22
LUT, 9-24
minimal CLB, 9-23–9-24
programmable elements, 9-23
XC4010XL chip, CLB, 9-24–9-25
FPGA technologies, 9-22
notation, 9-4–9-5
programmable array logic (PAL)
architecture, 9-8–9-9
example, 9-9–9-10
programmable logic array (PLA)
architecture, 9-5
programmed chip, 9-6
programmable macrocell outputs, PAL
circuit diagram, 9-18–9-19
circuit sizes, 9-21
macrocell architecture, 9-19–9-20
PAL22V10 chip architecture, 9-18
switch settings, 9-20–9-21
programmable read only memory (PROM)
conceptual diagram, 9-7
truth table, 9-7–9-8
sequential PAL devices
implementation range, 9-16, 9-18
PAL16R4 chip logic diagram, 9-16–9-17
state machines synthesis, 9-30–9-31
technologies
process type, 9-2–9-3
programming type, 9-3–9-4
VHDL synthesis style, FPGA
combinational logic, 9-29
latches, 9-30
registers and flip-flops, 9-27–9-29
Pull-in process, 5-8–5-9
Pulse-forming network output response, 6-7

Q

$\overline{Q}$ elimination, 9-69
Q-factor, 4-11–4-12

R

Radio frequency (RF) front-ends, 4-1–4-2
Radix-2 FFT algorithm, 7-13
R+C/C+R combination DAC, 10-11–10-12
Reactance pulse-forming network synthesis
delayed output pulse
algebraic ratio, 6-11
Cauer form, 6-13
delayed quasi-rectangular pulse, 6-9, 6-12
deterioration, 6-13–6-15
Fialkov condition, 6-12
Laplace transform, 6-10–6-11
normalized time, 6-9–6-10
pulse parameter, 6-12
slew rate, fronts, 6-9, 6-11

step response, 6-12–6-13
transfer function realization, 6-13
voltage source efficiency coefficient, 6-9–6-10, 6-13–6-14
non-delayed output pulses, 6-7–6-8
quasi-rectangular output pulse and laplace transform approximation, 6-4
impulse excitation, 6-2
impulse response, 6-4
rise and fall times, 6-3
shift theorem, 6-4
realization requirements, 6-4–6-5
second approximation step impulse response $V_o(s)$, 6-5
spectrum bandwidth ω_m, 6-6
transfer function, 6-5–6-6
sinusoidal pulse forming approximation, realization, 6-21–6-22
transfer function, 6-20–6-21
wideband amplifiers, transfer functions
approximation, 6-16–6-17
design, 6-18–6-19
Elmore's approach, 6-15
step response parameters and Laplace transform, 6-15–6-16
tabulated results, 6-19
Real feedback amplifier, 3-36
Realizable transfer function, 6-18
Rectangular spiral inductors
electrical model, 1-124–1-125
mutual inductance, 1-122–1-124
self-inductance, 1-122–1-123
self-resonant frequency, 1-125
total inductance, 1-122
transformer structure, 1-125–1-126
Region of convergence (ROC), 7-32
Relaxation oscillators, 5-11
Resistor/current mirror bias circuits, 2-63
Resistor-string DAC, 10-6–10-8
RF communication circuits
frequency synthesizer
fractional-*N* synthesis, 4-19–4-20
oscillator, 4-18–4-19
prescaler, 4-19
topology, 4-17–4-18
receiver
down converter, 4-15–4-16
LNA, 4-12–4-15
signal interference, 4-3
system performance metrics
first- and third-order intermodulation, 4-5
gain compression, 4-5
LNA, 4-4
noise figure, 4-3–4-4
nonlinearity performance, 4-4–4-5
receiver sensitivity, 4-3
signal-to-noise and distortion ratio (SNDR), 4-4
signal-to-noise ratio (SNR), baseband, 4-3–4-4
technology
active devices, 4-9–4-10
passive devices, 4-11–4-12
transceiver architectures
direct/zero-IF receiver, 4-7–4-8
down-conversion process, 4-6–4-7
heterodyne receiver, 4-6–4-7
image/mirror signal, 4-6
image rejection, 4-7–4-8
image suppression, 4-7
low output impedance driver, 4-6–4-7
quadrature amplitude modulation (QAM), 4-8
transmission medium, 4-2–4-3
transmitter
CMOS power amplification, 4-25
CMOS RF PA linearization, 4-27–4-29
linear MOS mixers, 4-22–4-23
nonlinearity and LO-feedthrough analysis, 4-23–4-25
switching class E amplifier, 4-25–4-27
switching modulators, 4-21–4-22
up *vs.* down conversion, 4-20–4-21
Ring oscillator jitter, 5-9–5-10

S

Saturation current, 1-4–1-5
Saturation regime
built-in potential, 1-44
channel length modulation voltage, 1-43
common-source volt–ampere characteristic curves, 1-45
drain current, 1-42, 1-44
large-signal circuit model, 1-44
strong inversion, 1-42–1-43
Second-generation current conveyor (CCII), 3-16–3-18
Segmented DACs, 10-4
Self-biased V_{BE} referenced current source, 2-7
Self-biased V_T referenced current source, 2-7–2-8
Self-calibration for individual capacitor errors, DAC, 10-15–10-17
Series R–C snubber, 3-39
Shot noise, 3-49
Signal interference, 4-3, 4-5
Signal-to-noise and distortion ratio (SNDR), 4-4
Simple current mirror, beta helper, 2-2–2-3
Simulation program with integrated circuit emphasis (SPICE) model
bipolar transistor parameters, 1-15–1-16
Gummel–Poon model, 1-5
small-signal model, 1-9
thermal sensitivity, 1-13–1-14
Single-stage op-amp architecture, 3-34–3-35
Sinusoidal envelope, output response, 6-20
Skew and delay, 9-34–9-35
Small-signal model
BJT
Boltzmann voltage, 2-12
emitter–base junction diffusion resistance, 2-13
equivalent circuit, 2-10–2-11
forward Early resistance and transconductance, 2-13
large-signal model, 2-11–2-12
monolithic fabrication process, 2-12

quiescent-operating point, 2-13
static common-emitter current, 2-12
MOSFET technology
analysis, 1-58–1-59, 1-71
bulk-gate transconductance, 1-68
bulk modulation factor, 1-62, 1-69
bulk transconductance, 1-60, 1-62
common-source interconnection, 1-70
equivalent circuit, 1-70
forward transadmittance, 1-71
forward transconductance, 1-60–1-62
HSPICE model, 1-66
N-channel MOSFET operation, 1-59–1-60
P-channel MOSFET operation, 1-59–1-61
radial signal frequency, 1-68
scattering parameters, 1-66–1-67
short circuit admittance parameters, y_{ij}, 1-66–1-67
signal drain current, 1-69
simulating results, 1-74–1-76
three-port network, 1-66
VCCS synthesis, 1-71–1-72
Square-law current model, 1-88, 1-90
Statz model, 1-89
Step response parameters, 6-19
Stray capacitance, 3-40
Stripe geometry, 3-11
Supply decoupling circuitry, 3-37
Synchronizers and metastability, 9-44–9-45
Systolic arrays
concurrency, parallelism, pipelining
definitions, 9-111–9-112
design techniques, 9-114–9-117
linear and rectangular type, 9-112
linear correlation, 9-113–9-114
uniprocessor system, 9-112
digital filters
finite impulse response (FIR) filters, 9-117–9-118
infinite impulse response (IIR) filters, 9-118–9-122
eigenvalue and SVDs
rectangular matrix, Hestenes algorithm, 9-140–9-141
rectangular nonsymmetric matrix, 9-141–9-142
spatial filtering problem, 9-137–9-138
symmetric matrix, 9-139
Kalman filtering (KF)
Faddeev algorithm, 9-134–9-137
model, 9-132–9-133
other forms, 9-133–9-134
systolic matrix implementation, 9-134
recursive LS estimation
Givens orthogonal transformation, 9-128–9-129
optimal residual and solutions, 9-129
QRD, 9-127
sliding window and forgetting factor approach, 9-127
technique, 9-126–9-127
triangular implementation, 9-129–9-132
systolic word and bit-level designs
advantages, 9-123
hierarchical approach, 9-122–9-123
serial convolver, 9-123–9-126

T

Table look-up synthesizer
fast Fourier transform, 7-13
topology, 4-17
Thermal noise, 3-48–3-49
Thermal voltage, V_T, 2-4–2-5
Thin-film resistors, 1-113–1-114
Tracking process, PLL
acceleration error, 5-5–5-6
final value theorem, 5-5
hold range, 5-6
Laplace transform, 5-5
step phase error, 5-5–5-6
velocity error/static phase error, 5-5
Transfer function poles, 6-19
Transmission gates (TG)
analog processing, 8-24–8-33
complementary transistor version
resistances, 8-18
voltage transmission properties, 8-17–8-18
continuous time filters, 8-27–8-28
digital processing, 8-15–8-24
MOS operational amplifier compensation
fully differential folded-cascode op-amp, 8-25–8-26
two-stage op-amp, 8-24–8-25
pass-transistor logic
adder, 8-21–8-22
CMOS *D* latch, 8-21
CPL circuit and modules, 8-22–8-23
CPL full adder circuit, 8-23
Karnaugh map, XOR function, 8-18–8-19
model, 8-18–8-19
OR gates, 8-20
16 ×16 bit multiplier, 8-23–8-24
SRAM and DRAM cells, 8-21
truth table, XOR function, 8-18–8-19
two-input multiplexer, 8-20
XOR gates, 8-19–8-20
single transistor version
I–O characteristics, 8-16–8-17
nMOS and pMOS, 8-15–8-16
switched-capacitor circuits
bottom-plate differential-input lossless digital integrator (LDI), 8-29–8-30
differential bilinear integrator, 8-29–8-30
direct digital integrator (DDI), 8-28–8-29
switch charge injection analysis, 8-30–8-33

transimpedance compensation
neuron and synapse operation, 8-27
preamplifier circuit, 8-26
Typical switched capacitor integrator, 2-75

U

Ultra-wideband (UWB) technology, 6-19–6-20
Unit step function, 7-5

V

VHDL, programming language
combinational logic, 9-29
latches, 9-30
registers and flip-flops
asynchronous reset code, 9-28
synchronous reset code, 9-27
synthesis tools, 9-28
state machine, 9-30–9-31
Voltage-controlled current source (VCCS), 1-64–1-65
Voltage-controlled oscillator (VCO)
frequency synthesizer, 4-17
LC-tank, 4-18
passive device, 4-12
RF transceiver architectures, 4-9
Voltage-controlled voltage source (VCVS), 1-71–1-72
Voltage follower, 3-14–3-15
V_t-referenced current bias circuit, 2-64

W

Wideband amplifiers, transfer functions
approximation, 6-16–6-17
design, 6-18–6-19
Elmore's approach, 6-15
step response parameters and Laplace transform, 6-15–6-16
tabulated results, 6-19
Wide bandgap compound semiconductors, 1-97–1-98
Widlar current mirror
bipolar junction transistor (BJT) biasing circuit, 2-4–2-5
current-feedback operations amplifier, 3-26–3-27
Wilson current mirror, 2-3–2-4

X

XOR PD, 5-13–5-14

Z

Z80 microprocessor-based design
clock cycles, 9-99–9-100
control signals tasks, 9-97–9-98
pin assignment, 9-96–9-97
programming model, 9-95–9-96
schematic diagram, 9-99
timing diagrams, 9-98–9-99

Electrical Engineering

Upon its initial publication, *The Circuits and Filters Handbook* broke new ground. It quickly became **the** resource for comprehensive coverage of issues and practical information that can be put to immediate use. Not content to rest on his laurels, in addition to updating the second edition, editor Wai-Kai Chen divided it into tightly focused texts that made the information easily accessible and digestible. These texts have been revised, updated, and expanded so that they continue to provide solid coverage of standard practices and enlightened perspectives on new and emerging techniques.

Analog and VLSI Circuits draws together international contributors who provide the latest information on analog and VLSI circuits, omitting extensive theory and proofs in favor of numerous examples throughout each chapter. The first part of the text focuses on analog integrated circuits, presenting up-to-date knowledge on monolithic device models, analog circuit cells, high performance analog circuits, RF communication circuits, and PLL circuits. In the second half of the book, well-known contributors offer the latest findings on VLSI circuits, including digital circuits, digital systems, and data converters.

Features

- Presents the fundamental biasing building blocks used in bipolar integrated circuit technology and the refinements of them that have evolved over time
- Provides a tutorial on techniques for integrated circuit amplifiers
- Discusses circuit techniques suitable for developing a single-chip CMOS solution compatible with all standards and capable of adapting itself

This volume will undoubtedly take its place as the engineer's first choice in looking for solutions to problems encountered when working with analog and VLSI circuits.

58916

ISBN: 978-1-4200-5891-8
90000
9 781420 058918

CRC Press
Taylor & Francis Group
an **informa** business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

The Circuits and Filters Handbook

Third Edition

Computer Aided Design and Design Automation

Edited by

Wai-Kai Chen



CRC Press
Taylor & Francis Group

Computer Aided Design and Design Automation

The Circuits and Filters Handbook

Third Edition

Edited by

Wai-Kai Chen

Fundamentals of Circuits and Filters

Feedback, Nonlinear, and Distributed Circuits

Analog and VLSI Circuits

Computer Aided Design and Design Automation

Passive, Active, and Digital Filters

The Circuits and Filters Handbook

Third Edition

Computer Aided Design and Design Automation

Edited by

Wai-Kai Chen

University of Illinois

Chicago, U. S. A.

CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the

Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2009 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4200-5918-2 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Computer aided design and design automation / Wai-Kai Chen.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-1-4200-5918-2
ISBN-10: 1-4200-5918-1
1. Engineering design. 2. Electric engineering--Computer-aided design. I. Chen, Wai-Kai, 1936- II. Title.

TA174.C58135 2009
621.3--dc22 2008048129

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface vii

Editor-in-Chief ix

Contributors xi

SECTION I Computer Aided Design and Optimization

1 Modeling of Circuit Performances **1**-1
Sung-Mo Kang and Abhijit Dharchoudhury

2 Symbolic Analysis Methods **2**-1
Benedykt S. Rodanski and Marwan M. Hassoun

3 Numerical Analysis Methods **3**-1
Andrew T. Yang

4 Design by Optimization **4**-1
Sachin S. Sapatnekar

5 Statistical Design Optimization **5**-1
Maciej A. Styblinski

6 Physical Design Automation **6**-1
Naveed A. Sherwani

7 Design Automation Technology **7**-1
Allen M. Dewey

8 Computer-Aided Analysis **8**-1
J. Gregory Rollins and Peter Bendix

9 Analog Circuit Simulation **9**-1
J. Gregory Rollins

SECTION II Design Automation

10 Internet-Based Microelectronic Design Automation Framework **10**-1
Moon-Jung Chung and Heechul Kim

11 System-Level Design **11**-1
Alice C. Parker, Yosef Tirat-Gefen, and Suhrid A. Wadekar

12 Performance Modeling and Analysis Using VHDL and SystemC **12**-1
Robert H. Klenke, Jonathan A. Andrews, and James H. Aylor

13 Embedded Computing Systems and Hardware/Software Codesign **13**-1
Wayne Wolf

14 Design Automation Technology Roadmap **14**-1
Donald R. Cottrell

Index **IN-1**

Preface

As system complexity continues to increase, the microelectronic industry must possess the ability to adapt quickly to the market changes and new technology through automation and simulations. The purpose of this book is to provide in a single volume a comprehensive reference work covering the broad spectrum of computer aided design and optimization techniques, which include circuit performance modeling, design by optimization, statistical design optimization, physical design automation, computer aided analysis and circuit simulations; and design automation, which includes system-level design, performance modeling and analysis, and hardware/software codesign. This book is written and developed for the practicing electrical engineers and computer scientists in industry, government, and academia. The goal is to provide the most up-to-date information in the field.

Over the years, the fundamentals of the field have evolved to include a wide range of topics and a broad range of practice. To encompass such a wide range of knowledge, this book focuses on the key concepts, models, and equations that enable the design engineer to analyze, design, and predict the behavior of large-scale systems. While design formulas and tables are listed, emphasis is placed on the key concepts and theories underlying the processes.

This book stresses fundamental theories behind professional applications and uses several examples to reinforce this point. Extensive development of theory and details of proofs have been omitted. The reader is assumed to have a certain degree of sophistication and experience. However, brief reviews of theories, principles, and mathematics of some subject areas are given. These reviews have been done concisely with perception.

The compilation of this book would not have been possible without the dedication and efforts of Professor Steve Sung-Mo Kang, and most of all the contributing authors. I wish to thank them all.

Wai-Kai Chen

Editor-in-Chief



Wai-Kai Chen is a professor and head emeritus of the Department of Electrical Engineering and Computer Science at the University of Illinois at Chicago. He received his BS and MS in electrical engineering at Ohio University, where he was later recognized as a distinguished professor. He earned his PhD in electrical engineering at the University of Illinois at Urbana–Champaign.

Professor Chen has extensive experience in education and industry and is very active professionally in the fields of circuits and systems. He has served as a visiting professor at Purdue University, the University of Hawaii at Manoa, and Chuo University in Tokyo, Japan. He was the editor-in-chief of the *IEEE Transactions on Circuits and Systems, Series I and II*, the president of the IEEE Circuits and Systems Society, and is the founding editor and the editor-in-chief of the *Journal of Circuits, Systems and Computers*. He received the Lester R. Ford Award from the Mathematical Association of America; the Alexander von Humboldt Award from Germany; the JSPS Fellowship Award from the Japan Society for the Promotion of Science; the National Taipei University of Science and Technology Distinguished Alumnus Award; the Ohio University Alumni Medal of Merit for Distinguished Achievement in Engineering Education; the Senior University Scholar Award and the 2000 Faculty Research Award from the University of Illinois at Chicago; and the Distinguished Alumnus Award from the University of Illinois at Urbana–Champaign. He is the recipient of the Golden Jubilee Medal, the Education Award, and the Meritorious Service Award from the IEEE Circuits and Systems Society, and the Third Millennium Medal from the IEEE. He has also received more than a dozen honorary professorship awards from major institutions in Taiwan and China.

A fellow of the Institute of Electrical and Electronics Engineers (IEEE) and the American Association for the Advancement of Science (AAAS), Professor Chen is widely known in the profession for the following works: *Applied Graph Theory* (North-Holland), *Theory and Design of Broadband Matching Networks* (Pergamon Press), *Active Network and Feedback Amplifier Theory* (McGraw-Hill), *Linear Networks and Systems* (Brooks/Cole), *Passive and Active Filters: Theory and Implements* (John Wiley), *Theory of Nets: Flows in Networks* (Wiley-Interscience), *The Electrical Engineering Handbook* (Academic Press), and *The VLSI Handbook* (CRC Press).

Contributors

Jonathan A. Andrews
Department of Electrical and Computer Engineering
Virginia Commonwealth University
Richmond, Virginia

James H. Aylor
Department of Electrical Engineering
University of Virginia
Charlottesville, Virginia

Peter Bendix
Technology Modeling Associates, Inc.
Palo Alto, California

Wai-Kai Chen
Department of Electrical and Computer Engineering
University of Illinois at Chicago
Chicago, Illinois

Moon-Jung Chung
Department of Computer Science
Michigan State University
East Lansing, Michigan

Donald R. Cottrell
Silicon Integration Initiative, Inc.
Austin, Texas

Allen M. Dewey
Department of Electrical and Computer Engineering
Duke University
Durham, North Carolina

Abhijit Dharchoudhury
Beckman Institute
University of Illinois at Urbana–Champaign
Urbana, Illinois

Marwan M. Hassoun
Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa

Sung-Mo Kang
Office of Chancellor
University of California, Merced
Merced, California

Heechul Kim
Department of Computer Science and Engineering
Hankuk University of Foreign Studies
Yongin, Korea

Robert H. Klenke
Department of Electrical and Computer Engineering
Virginia Commonwealth University
Richmond, Virginia

Alice C. Parker
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

Benedykt S. Rodanski
Faculty of Engineering
University of Technology
Sydney, New South Wales, Australia

J. Gregory Rollins
Technology Modeling Associates, Inc.
Palo Alto, California

Sachin S. Sapatnekar
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, Minnesota

Naveed A. Sherwani
Intel Corporation
Hillsboro, Oregon

Maciej A. Styblinski
Department of Electrical Engineering
Texas A&M University
College Station, Texas

Yosef Tirat-Gefen
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

Suhrid A. Wadekar
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, California

Wayne Wolf
Department of Electrical Engineering
Princeton University
Princeton, New Jersey

Andrew T. Yang
Apache Design Solutions, Inc.
San Jose, California

I

Computer Aided Design and Optimization

Sung-Mo Kang
University of California, Merced

1 Modeling of Circuit Performances *Sung-Mo Kang and Abhijit Dharchoudhury* **1**-1
Introduction • Circuit Performance Measures • Input–Output Relationships • Dependency of Circuit Performances on Circuit Parameters • Design of Experiments • Least-Squares Fitting • Variable Screening • Stochastic Performance Models • Other Performance Modeling Approaches • Example of Performance Modeling • References

2 Symbolic Analysis Methods *Benedykt S. Rodanski and Marwan M. Hassoun* **2**-1
Introduction • Symbolic Modified Nodal Analysis • Solution Methods • Ideal Operational Amplifiers • Applications of Symbolic Analysis • Symbolic Analysis Software Packages • References

3 Numerical Analysis Methods *Andrew T. Yang* **3**-1
Equation Formulation • Solution of Linear Algebraic Equations • References

4 Design by Optimization *Sachin S. Sapatnekar* **4**-1
Introduction • Optimization Algorithms • Transistor Sizing Problem for CMOS Digital Circuits • Analog Design Centering Problem • References

5 Statistical Design Optimization *Maciej A. Styblinski* **5**-1
Introduction • Problems and Methodologies of Statistical Circuit Design • Underlying Concepts and Techniques • Statistical Methods of Yield Optimization • Conclusion • References

6 Physical Design Automation *Naveed A. Sherwani* **6**-1
Introduction • Very Large-Scale Integration Design Cycle • Physical Design Cycle • Design Styles • Partitioning • Other Partitioning Algorithms • Placement • Routing • Classification of Global Routing Algorithms • Classification of Detailed Routing Algorithms • Compaction • Summary • References

7 Design Automation Technology *Allen M. Dewey* **7**-1
Introduction • Design Entry • Conceptual Design • Synthesis • Verification • Test • Frameworks • Summary • References

8 Computer-Aided Analysis *J. Gregory Rollins and Peter Bendix* **8**-1
Circuit Simulation Using SPICE and SUPREM • Appendix A • References • Parameter Extraction for Analog Circuit Simulation • References

9 Analog Circuit Simulation *J. Gregory Rollins* **9**-1
Introduction • Purpose of Simulation • Netlists • Formulation of the Circuit Equations • Modified Nodal Analysis • Active Device Models • Types of Analysis • Verilog-A • Fast Simulation Methods • Commercially Available Simulators • References

1

Modeling of Circuit Performances

1.1 Introduction **1**-1
1.2 Circuit Performance Measures **1**-2
1.3 Input–Output Relationships **1**-3
Memoryless (Static) Relationship • Dynamic Relationships
1.4 Dependency of Circuit Performances on Circuit Parameters **1**-3
1.5 Design of Experiments **1**-5
Factorial Design • Central Composite Design • Taguchi's Orthogonal Arrays • Latin Hypercube Sampling
1.6 Least-Squares Fitting **1**-11
1.7 Variable Screening **1**-13
1.8 Stochastic Performance Models **1**-14
1.9 Other Performance Modeling Approaches **1**-14
1.10 Example of Performance Modeling **1**-15
References **1**-16

Sung-Mo Kang
University of California, Merced

Abhijit Dharchoudhury
University of Illinois at Urbana–Champaign

1.1 Introduction

The domain of computer-aided design and optimization for circuits and filters alone is very broad, especially since the design objectives are usually multiple. In general, the circuit design can be classified into

- Electrical design
- Physical design

Physical design deals with concrete geometrical parameters while electrical design deals with the electrical performances of the physical object. Strictly speaking, both design aspects are inseparable since any physical design is a realization of a particular electrical design. On the other hand, electrical design without proper abstraction of the corresponding physical design can be futile in the sense that its implementation may not meet the design goals. Thus, electrical and physical design should go hand in hand. Most literature deals with both designs using weighted abstractions. For example, in the design of a digital or analog filter, various effects of physical layout are first captured in the parasitic models for interconnects and in the electrical performance models for transistors. Then, electrical design is performed using such models to predict electrical performances. In physical design optimization, geometrical parameters are chosen such that the predicted electrical performances can meet the design

objectives. For example, in the timing-driven physical layout of integrated circuits, circuit components will be laid out, placed and interconnected to meet timing requirements. In order to make the optimization process computationally efficient, performance models are used during the optimization process instead of simulation tools. In this chapter, the focus of design optimization is on the electrical performance. The computer-aided electrical design can be further classified into performance analysis and optimization. Thus, this chapter first discusses the modeling of electrical performances, symbolic and numerical analysis techniques, followed by optimization techniques for circuit timing, and yield enhancement with transistor sizing and statistical design techniques.

1.2 Circuit Performance Measures

The performance of a particular circuit can be measured in many ways according to application and design goals. In essence, each performance measure is an indication of the circuit output which the designer is interested in for a particular operating condition. The circuit performance, in general, is dependent on the values of the designable parameters such as the transistor sizes, over which the designer has some degree of control, and the values of the noise parameters such as the fluctuation in power supply voltage or the threshold voltages of the transistors, which are random in nature. Also, the performance measure used is dependent on the type of the circuit. The performance measures of digital circuits are usually quite different from those of analog circuits or filters. Some of the important performance measures of digital circuits are

- Signal propagation delay from an input to an output
- Rise and fall times of signals
- Clock signal skew in the circuit
- Power dissipation
- Manufacturing yield
- Failure rate (reliability)

Important performance measures for analog circuits are

- Small-signal gain and bandwidth in frequency domain
- Sensitivities to various nonideal factors (noise)
- Slew rate of operational amplifiers
- Power dissipation
- Manufacturing yield
- Failure rate (reliability)

The modeling of digital circuit performances is usually more difficult to derive than the modeling of analog circuit performances, mainly due to the nonlinear nature of digital circuits. Thus, simple transistor models have been used often to obtain simple qualitative models for propagation delays, rise and fall times, and power dissipation. On the other hand, due to the small-signal nature, the device models required by analog designers need to be much more accurate. Another important performance measure is the manufacturing yield, which determines the product cost. The yield depends on both the quality of manufacturing process control and the quality of the design. A challenging design problem is how to come up with a design which can be robust to nonideal manufacturing conditions and thereby produce a good manufacturing yield. The model for failure rate is difficult to derive due to several complex mechanisms that determine the lifetime of chips. In such cases, empirical models, which are not physical, have been introduced.

In order to develop models for analog circuits, transistors in the circuit are first DC-biased at the proper biasing points. Then, small-signal circuit models are developed at those DC operating points. The resulting linear circuits can be analyzed by either using mathematical symbols (symbolic analysis) or by using simulators.

1.3 Input–Output Relationships

The circuit functionally depends on the input–output relationships and thus circuit specifications and test specifications are made on the basis of the input–output relationships. For modeling purposes, the input–output relationships can be classified into static relationships and dynamic relationships.

1.3.1 Memoryless (Static) Relationship

A given input–output pair is said to have a memoryless relationship if it can be described statically without the use of any ordinary or partial differential equation. It is important to note that the choice of input and output variables can determine the nature of the relationship. For instance, the relationship between input charge q and the output voltage v in a 1 F capacitor is static. On the other hand, the relationship between an input current i and the output voltage v, in the same capacitor is nonstatic since the voltage is dependent on the integral of the input current. Ideally, basic circuit elements such as resistor, capacitor, inductor, and source should be characterized by using static relationships, so-called constitutive relationships, with proper choice of input and output variables. However, for circuits and systems design, since their input–output relationships are specified in terms of specific variables of interest, such choice of variables cannot be made arbitrarily. Often, both input and output variables are voltages, especially in digital circuits.

1.3.2 Dynamic Relationships

A given input–output pair is said to have a dynamic relationship if the modeling of the relationship requires a differential equation. In most cases, since the circuits contain parasitic capacitors or inductors or both, the input–output relationship is dynamic. For instance, the input–output relationship of an inverter gate with capacitive load C_{load} can be described as

$$C_{\text{load}} \frac{\mathrm{d}V_{\text{out}}}{\mathrm{d}t} = i(V_{\text{out}}, V_{\text{in}}, V_{\text{DD}}) \tag{1.1}$$

where

V_{out} and V_{in} denote the output and input voltages
i denotes the current through the pull-up or pull-down transistor
V_{DD} denotes the power supply voltage

Although the relationship is implicit in this case, Equation 1.1 can be solved either analytically or numerically to find the relationship. Equation 1.1 can be generalized by including noise effects due to variations in the circuit fabrication process.

1.4 Dependency of Circuit Performances on Circuit Parameters

Circuit performances are functions of circuit parameters, which are usually not known explicitly. In order to design and analyze analog or digital circuits, the dependency of the circuit performances on the various circuit parameters need to be modeled.

The actual value of a circuit parameter in a manufactured circuit is expected to be different from the nominal or target value due to inevitable random variations in the manufacturing processes and in the environmental conditions in which the circuit is operated. For example, the actual channel width W of an MOS transistor can be decomposed into a nominal component W^0 and a statistically varying component ΔW, i.e., $W = W^0 + \Delta W$. The nominal component is under the control of the designer and can be set to a particular value. Such a nominal component is said to be designable or controllable, e.g., W^0.

The statistically varying component, on the other hand, is not under the control of the designer and is random in nature. It is called the noise component, and it represents the uncontrollable fluctuation of a circuit parameter about its designable component, e.g., ΔW. For certain circuit parameters, e.g., device model parameters (like the threshold voltages of MOS transistors) and operating conditions (like temperature or power supply voltage), the nominal values are not really under the control of the circuit designer and are set by the processing and operating conditions. For these circuit parameters, the nominal and random components are together called the noise component. In general, therefore, a circuit parameter x_i can be expressed as follows:

$$x_i = d_i + s_i \tag{1.2}$$

where
d_i is the designable component
s_i is the noise component

It is common to group all the designable components to form the set of designable parameters, denoted by **d**. Similarly, all the noise parameters are grouped together to form the noise parameters, denoted by the random vector **s**. Vectorially, Equation 1.2 can be rewritten as

$$\mathbf{x} = \mathbf{d} + \mathbf{s} \tag{1.3}$$

A circuit performance measure is a function of the circuit parameters. For example, the propagation delay of a CMOS inverter circuit is a function of the nominal channel lengths and widths of the NMOS and PMOS transistors. It is also a function of the threshold voltages of the transistors and the power supply voltage. Thus, in general, a circuit performance is a function of the designable as well as the noise parameters:

$$y = y(\mathbf{x}) = y(\mathbf{d}, \mathbf{s}) \tag{1.4}$$

If this function is known explicitly in terms of the designable and noise parameters, then optimization methods can be applied directly to obtain a design which is optimal in terms of the performances or the manufacturing yield. More often than not, however, the circuit performances cannot be expressed explicitly in terms of the circuit parameters. In such cases, the value of a circuit performance for given values of circuit parameters can be obtained by circuit simulation. Circuit simulations are computationally expensive, especially if the circuit is large and transient simulations are required. A compact performance model in terms of the circuit parameters is an efficient alternative to the circuit simulator. Two factors determine the utility of the performance model. First, the model should be computationally efficient to construct and evaluate so that substantial computational savings can be achieved. Second, the model should be accurate.

Figure 1.1 shows the general procedure for constructing the model for a performance measure y in terms of the circuit parameters **x**. The model-building procedure consists of four steps. In the first stem, *m training points* are selected from the **x**-space. The ith training point is denoted by $\mathbf{x}_i$, $i = 1, 2, \ldots, m$. In the second step, the circuit is simulated at these m training points and the values of the performance measure are obtained from the circuit simulation results as $y(\mathbf{x}_1)$, $y(\mathbf{x}_2)$, $\ldots, y(\mathbf{x}_m)$. In the third step, a preassigned function of y in terms of **x** is "fitted" to the data. In the fourth and final step, the model is validated for accuracy. If the model accuracy is deemed inadequate, the modeling procedure is repeated with a larger number of training points or with different models.

The model is called the response surface model (RSM) [1] of the performance and is denoted by $\hat{y}(\mathbf{x})$. The computational cost of modeling depends on the number of training points m, and the procedure of fitting the model to the data. The accuracy of the RSM is quantified by computing error measures which

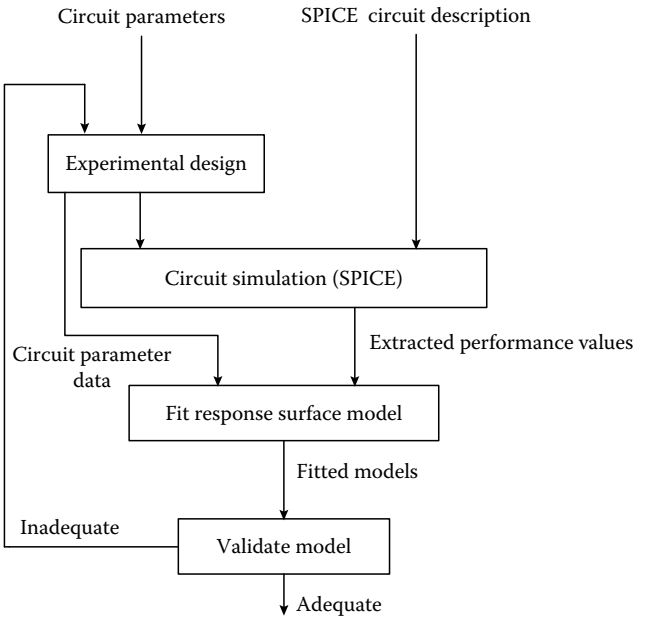


FIGURE 1.1 Performance modeling procedure.

quantify the ''goodness of fit.'' The accuracy of the RSM is greatly influenced by the manner in which the training points are selected from the **x**-space. Design of experiment (DOE) [2] techniques are a systematic means of obtaining the training points such that the maximum amount of information about the model can be obtained from these simulations or experimental runs. In the next section, we review some of the most commonly used experimental design techniques in the context of performance optimization and statistical design. To illustrate certain points in the discussion below, we will assume that the RSM of the performance measure y is the following quadratic polynomial in terms of the circuit parameters:

$$\hat{y}(\mathbf{x}) = \hat{\alpha}_0 + \sum_{i=1}^{n} \hat{\alpha}_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{\alpha}_{ij} x_i x_j \tag{1.5}$$

where the $\hat{\alpha}$s are the coefficients of the model. Note, however, that the discussion is valid for all RSMs in general.

1.5 Design of Experiments

1.5.1 Factorial Design

In this experimental design, each of the circuit parameters $x_1, x_2, \ldots, x_n$ is quantized into two levels or settings, which are denoted by -1 and $+1$. A full factorial design contains all possible combinations of levels for the n parameters, i.e., it contains 2^n training points or experimental runs. The design matrix for the case of $n = 3$ is shown in Table 1.1 and the design is pictorially described in Figure 1.2. The value of the circuit performance measure for the kth run is denoted by y_k.

Much information about the relationship between the circuit performance y and the circuit parameters x_i, $i = 1, 2, \ldots, n$ can be obtained from a full factorial experiment. The main or individual effect of a

TABLE 1.1 Full Factorial Design for $n = 3$

	Parameter Levels			Interaction Levels				
Run	x_1	x_2	x_3	$x_1 \times x_2$	$x_1 \times x_3$	$x_2 \times x_3$	$x_1 \times x_2 \times x_3$	y
1	−1	−1	−1	+1	+1	+1	−1	y_1
2	−1	−1	+1	+1	−1	−1	+1	y_2
3	−1	+1	−1	−1	+1	−1	+1	y_3
4	−1	+1	+1	−1	−1	+1	−1	y_4
5	+1	−1	−1	−1	−1	+1	+1	y_5
6	+1	−1	+1	−1	+1	−1	−1	y_6
7	+1	+1	−1	+1	−1	−1	−1	y_7
8	+1	+1	+1	+1	+1	+1	+1	y_8

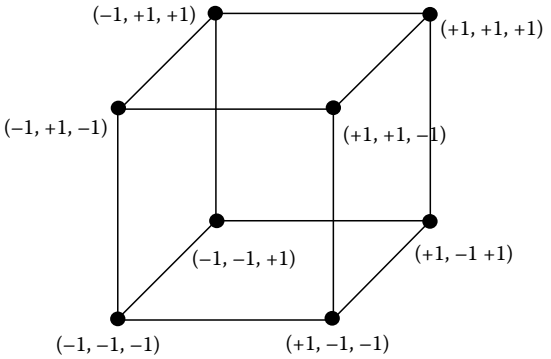


FIGURE 1.2 Pictorial representation of full factorial design for $n = 3$.

parameter quantifies how much a parameter affects the performance singly. The main effect of x_i, denoted by v_i, is given by

$$v_i = \frac{1}{2^n} \sum_{k=1}^{2^n} x_{ik} \times y_k \tag{1.6}$$

where

x_{ik} is the value of x_i in the kth run

y_k is the corresponding performance value

Thus, the main effect is the difference between the average performance value when the parameter is at the high level (+1) and the average performance value when the parameter is at the low level (−1). The main effect of x_i is the coefficient of the x_i term in the polynomial RSM of Equation 1.5. Similarly, the interaction effect of two or more parameters quantities how those factors jointly affect the performance. The two-factor interaction effect is computed as the difference between the average performance value when both factors are at the same level and the average performance value when they are at different levels. The two-factor interaction effect of parameters x_i and x_j, denoted by $v_{i\times j}$, $i \neq j$, is defined as follows:

$$\begin{aligned} v_{i\times j} &= \frac{1}{2}\left\{v_{i|j=+1} - v_{i|j=-1}\right\} \\ &= \frac{1}{2^n} \sum_{k=1}^{2^n} x_{ik} \times x_{jk} \times y_k \end{aligned} \tag{1.7}$$

The two-factor interaction effect of x_i and x_j, $v_{i\times j}$, is the coefficient of the $x_i x_j$ terms in the quadratic RSM of Equation 1.5. Note that $v_{i\times j}$ is the same as $v_{j\times i}$. Moreover, higher-order multifactor interaction effects can be computed recursively as before. Note that the interaction columns of Figure 1.2 are obtained by the aggregated "sign-multiplication" given in Equation 1.7.

Thus, the full factorial design allows us to estimate the coefficients of all the first-order and cross-factor second-order coefficients in the RMS of Equation 1.5. However, it does not allow us to estimate the coefficients of the pure quadratic term x_i^2. Moreover, the number of experimental runs increases exponentially with the number of circuit parameters; this may become impractical for a large number

TABLE 1.2 Half Fraction of a Full Factorial Design for $n = 3$

	Parameter Levels			Interaction Levels				
Run	x_1	x_2	x_3	$x_1 \times x_2$	$x_1 \times x_3$	$x_2 \times x_3$	$x_1 \times x_2 \times x_3$	y
1	−1	−1	+1	+1	−1	−1	+1	y_1
2	−1	+1	−1	−1	+1	−1	+1	y_2
3	+1	−1	−1	−1	−1	+1	+1	y_3
4	+1	+1	+1	+1	+1	+1	+1	y_4

of circuit parameters. In many cases, high-order multifactor interactions are small and can be ignored. In such cases, it is possible to reduce the size of the design by systematically eliminating some interactions and considering only some of the runs. The accuracy with which the main effects and the low-order interaction effects are estimated does not have to be compromised. Such designs are called fractional factorial designs. The most popular class of fractional factorial designs are the 2^{n-p} designs, where a $1/2^p$th fraction of the original 2^n design is used. One of the half fractions of the 2^3 design shown in Table 1.1, i.e., a 2^{3-1} design is shown in Table 1.2.

We observe that some of the columns of Table 1.2 are now identical. It can be seen from Equations 1.6 and 1.7 that we cannot distinguish between effects that correspond to identical columns. Such effects are said to be confounded or aliased with each other. From Table 1.2, we see that v_3, the main effect of x_3, and $v_{1\times2}$, the interaction effects of x_1 and x_2, are confounded with each other. Moreover the three-factor interaction effect $v_{1\times2\times3}$ is confounded with the grand average of the performance (corresponding to a column of 1's). Confounding, however, is not really a problem, since in most applications, high-order interaction effects are negligible. For example, in the quadratic RSM of Equation 1.5, only main effects and two-factor interaction effects are important, and it can be assumed that all higher-order interaction effects are absent.

There is a systematic way of obtaining fractional factorial designs by first isolating a set of basic factors (circuit parameters) and then defining the remaining nonbasic factors in terms of the interaction of the basic factors. These interactions that define the nonbasic factors in terms of the basic factors are called generators. The set of basic factors and the generator functions completely characterize a fractional factorial design. For example, the generator function used in the example of Table 1.2 is

$$x_3 = x_1 \times x_2 \tag{1.8}$$

Note that, once the generator function is known for a fractional design, the confounding pattern is also completely known. Moreover, note that the generator functions are not unique. The generator functions $x_1 = x_2 \times x_3$ and $x_2 = x_1 \times x_3$ also give rise to the design of Table 1.2. This can be understood by realizing that a column corresponding to a parameter sign-multiplied by itself results in a column of all 1s, i.e., the identity column, denoted by I. Thus, multiplying both sides of Equation 1.8 by x_2, we get $x_3 \times x_2 = x_1 \times x_2 \times x_2 = x_1 \times I = x_1$, which is also a generator function. Thus, an appropriate set of generator functions must be defined.

Another important concept related to confounding is called the resolution of a design, which indicates the clarity with which individual effects and interaction effects can be evaluated in a design. A *resolution III* design is one in which no two main effects are confounded with each other (but they may be confounded with two-factor and/or higher-order interaction effects). An important class of resolution *III* designs is the saturated fractional factorial designs. Suppose that the number of factors (parameters), n, can be expressed as $n = 2^q - 1$, for some integer q. Then, a resolution *III* design can be obtained by first selecting q basic factors and then saturating the 2^q design by assigning a nonbasic factor to each of the possible combinations of the q basic factors. There are $p = n - q$ nonbasic factors and generator functions, and these

designs are called 2^{n-p}_{III} designs, since $2^q = n+1$ runs are required for n factors. Note that the fractional factorial design of Table 1.2 is a 2^{n-p}_{III} design, with $n=3$ and $p=1$. In the saturated design for $n=2^q-1$ factors, the confounding patterns for the basic and nonbasic factors are the same, which implies that any of the q factors may be chosen as the basic factors. However, if $n+1$ is not an integral power of 2, then q is chosen as $q=[\log_2(n+1)]$. Next, we define q basic factors and generate a 2^q design. Next, we assign $p=n-q$ nonbasic factors to p generators and form the 2^{n-p}_{III} design. Note that, in this case, only p of the 2^q-q-1 available generators are used, and the confounding pattern depends on the choice of the q basic factors and the p generators. The number of runs in the saturated design is considerably less than 2δ.

Another important class of designs commonly used is called *resolution V* designs. In these designs, no main effect or two-factor interaction effect is confounded with another main effect or two-factor interaction effect. Such designs are denoted by 2^{n-p}_{V} designs, where as before, n is the number of factors and p is the number of generators. These designs can be used to estimate the coefficients of all but the pure quadratic terms in the RSM of Equation 1.5. Since the number of such coefficients (including the constant term) is $C=1+n+n(n-1)/2$, there must be at least C runs. Note that C is still substantially less than 2^n. Another important advantage of the factorial designs is that they are orthogonal, which implies that the model coefficients can be estimated with minimum variances (or errors). An algorithmic method for deriving resolution V designs is given in [3].

1.5.2 Central Composite Design

As mentioned previously, one of the problems of factorial designs in regard to the RMS of Equation 1.5 is that the coefficients of the pure quadratic terms cannot be estimated. These coefficients can be estimated by using a central composite design. A central composite design is a combination of a "cube" and a "star" subdesign. Each factor in a central composite design takes on five values: 0, ± 1, and $\pm\gamma$. Figure 1.3 shows the central composite design for the case of $n=3$. The cube subdesign, shown in dotted lines in Figure 1.3, is a fractional factorial design with the factor levels set to ± 1. This design is of resolution V so that linear and cross-factor quadratic terms in the RSM of Equation 1.5 can be estimated. The star

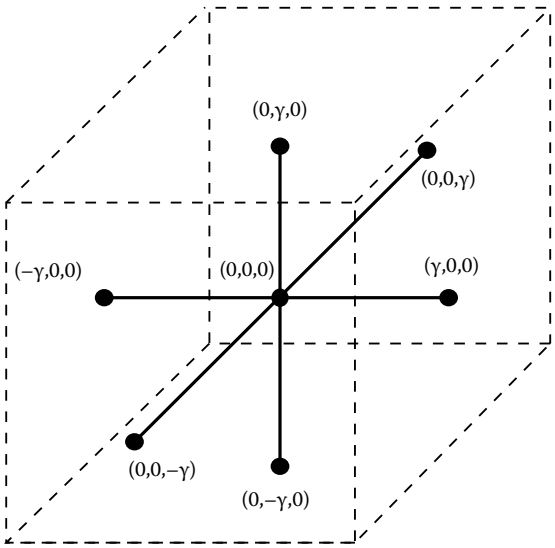


FIGURE 1.3 Central composite design for $n=3$.

subdesign, shown in solid lines in Figure 1.3, is used to estimate the pure quadratic terms, x_1^2, in the RMS, and consists of

- One *center* point, where all the parameters are set to 0
- 2*n Axial* points, one pair for each parameter by setting its value to $-\gamma$ and $+\gamma$ and setting all other parameters to 0

The parameter γ (chosen by the user) is selected so that the composite plan satisfies the rotatability property [1]. The main advantage of the central composite design is that all the coefficients of Equation 1.5 can be estimated using a reasonable number of simulations.

1.5.3 Taguchi's Orthogonal Arrays

Taguchi's method using orthogonal arrays (OA) [4] is another popular experimental design technique. An orthogonal array is a special kind of fractional factorial design. These orthogonal arrays can be classified into two types. The first category of OAs corresponds to two-level designs, i.e., the parameters are quantized to two levels each, while the second corresponds to three-level designs, i.e., the parameters are quantized to three levels each. These arrays are often available as tables in books which discuss Taguchi techniques [4]. As an example, we show the L18 OA in Table 1.3. The number in the designation of the array refers to the number of experimental runs in the design. The L18 OA belongs to the second category of designs, i.e., each parameter has three levels. In the Taguchi technique, the experimental design matrix for the controllable or designable parameters is called the inner array, while that for the noise parameters is called the outer array. The L18 design of Table 1.3 can be used as an inner array as well as an outer array.

1.5.4 Latin Hypercube Sampling

The factorial, central composite, and Taguchi experimental designs described earlier set the parameters to certain levels or quantized values within their ranges. Therefore, most of the parameters space remains

TABLE 1.3 Taguchi's L18 OA

Run	Parameter Levels							
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	1	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

unsampled. It is therefore desirable to have a more "space-filling" sampling strategy. The most obvious method of obtaining a more complete coverage of the parameter space is to perform random or Monte Carlo sampling [5]. In random sampling, each parameter has a particular statistical distribution and the parameters are considered to be statistically independent. The distribution of the circuit parameters can often be obtained from test structure measurements followed by parameter extraction. Often, however, based on previous experience or reasonable expectation, the circuit parameters are considered to be normal (or Gaussian). Since the designable parameters are deterministic variables, they are considered to be independent and uniformly distributed for the purposes of random sampling. Random samples are easy to generate and many inferences can be drawn regarding the probability distribution of the performances. The problem with random sampling, however, is that a large sample is usually required to estimate quantities with sufficiently small errors. Moreover, circuit parameters are usually not statistically independent. In particular, the noise parameters are statistically correlated because of the sequential nature of semiconductor wafer processing. Further, a random sample may not be very space filling either. This can be understood by considering the bell-shaped Gaussian density curve. In random sampling, values near the peak of the bell-shaped curve would be more likely to occur in the sample since such values have higher probability of occurrence. In other words, values away from the central region would not be well represented in the sample.

Latin hypercube sampling (LHS) [6,7] is a sampling strategy that alleviates this problem to a large extent. For each parameter x_i, all portions of its distribution are represented by sample values. If S is the desired size of the sample, then the range of each x_i, $i = 1, 2, \ldots, n$ is divided into S nonoverlapping intervals of equal marginal probability $1/S$. Each such interval is sampled once with respect to the probability density in that interval to obtain S values for each parameter. Next, the S values for one parameter are randomly paired with the S values for another parameter, and so on. Figure 1.4 illustrates this process for the case of two circuit parameters x_1 and x_2, where x_1 is a uniform random variable, and x_2 is a Gaussian random variable, and $S = 5$. The marginal probability in an interval is defined as the area under the probability density curve for that interval. Therefore, intervals with equal probability are intervals with equal areas under the probability density curve. For the uniformly distributed x_1, equal probability intervals are also of equal length. For x_2, however, the intervals near the center are smaller (since the density there is higher) than the intervals away from the center (where the density is lower). Figure 1.4 shows that $S = 5$ values are chosen from each region for x_1 and x_2 (shown as open circles). These values are then randomly paired so as to obtain the sample points (shown as filled dots). If the circuit parameters are correlated, then the pairing of the sample values can be controlled so that

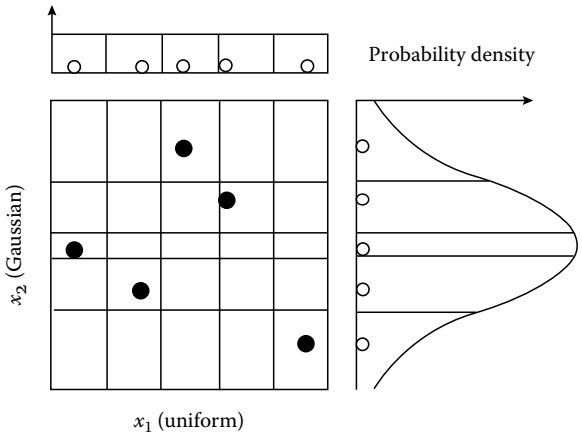


FIGURE 1.4 Latin hypercube sampling for a uniform and a Gaussian random variable.

the sample correlation matrix is close to the user-provided correlation matrix for the circuit parameters. Note that LHS will provide a more uniform coverage of the parameter space than other experimental design techniques. Moreover, a sample of any size can be easily generated and all probability densities are handled.

1.6 Least-Squares Fitting

After the experimental design has been used to select the training points in the parameter space, the circuit is simulated at the training points and the values of the performance measure are extracted from the simulation results. Let S denote the number of experimental runs, and y_k, $k = 1, \ldots, S$ denote the corresponding performance values. The aim in least-squares fitting is to determine the coefficients in the RSM so that the fitting error is minimized. Let C denote the number of coefficients in the RSM. For example, the value of C for the RSM of Equation 1.5 is $C = (n+1)(n+2)/2$, where n is the number of circuit parameters. There are interpolation-based methods that can be used to determine the coefficients when there are a smaller number of data points than coefficients, i.e., when $S < C$. One such method is called maximally flat quadratic interpolation and is discussed later in this chapter. We will restrict our discussion to the case of $S \geq C$. If $S = C$, then there are as many data points as coefficients, and simple simultaneous equation solving will provide the values of the coefficients. In this case, the model interpolates the data points exactly, and there is no fitting error to be minimized. When $S > C$, the coefficients can be determined in such a way that the fitting error is minimized. This method is called least-squares fitting (in a numerical analysis context) or linear regression (in a statistical context). The error measure is called the sum of squared errors and is given by

$$\in = \sum_{i=1}^{S} (y_k - \hat{y}_k)^2 \tag{1.9}$$

where

y_k denotes the simulated performance value
$\hat{y}_k$ denotes the model-predicted value at the kth data point

Least-squares fitting can then be stated as

$$\min_{\hat{\alpha}_i} \in = \sum_{i=1}^{s} (y_k - \hat{y}_k)^2 \tag{1.10}$$

where $\hat{\alpha}_i$ represents the (unknown) coefficients of the RSM. The error ε is used to determine the adequacy of the model. If this error is too large, the modeling procedure should be repeated with a larger number of training points or a different sampling strategy, or a different model of the performance may have to be hypothesized.

From a statistical point of view (linear regression), the actual performance value is considered to be

$$y_i = \hat{y}_i + e_i, \quad i = 1, 2, \ldots, S \tag{1.11}$$

where e_i denotes the random error in the model. In least-squares fitting, this random error is implicitly assumed to be independent and identically distributed normal random variables with zero mean and constant variance. In particular, we assume that

$$\begin{aligned} E(e_i) &= 0 \\ \mathrm{Var}(e_i) &= \sigma^2 \\ \text{and} \quad \mathrm{Cov}(e_i, e_j) &= 0, \quad i \neq j \end{aligned} \tag{1.12}$$

If the RSM of the performance measure is linear, then it can be written as

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{C} \alpha_i f_i(\mathbf{x}) \tag{1.13}$$

where
$f_i(\mathbf{x})$ are the known basis functions of the circuit parameters
C denotes the number of basis functions

Note that the RSM is said to be linear (and least-squares fitting can be used) if it is linear in terms of the model coefficients α_i. The basis functions $f_i(\mathbf{x})$ need not be linear. For instance, the basis functions used in the quadratic RSM of Equation 1.5 are 1, x_i, and $x_i x_j$.

Let $\mathbf{y}$ and $\mathbf{e}$ be S-vectors: $\mathbf{y} = [y_1, \ldots, y_S]^{\mathrm{T}}$ and $\mathbf{e} = [e_1, \ldots, e_S]^{\mathrm{T}}$, where S is the number of data points. Also, let us define the model coefficients as $\alpha = [\alpha_1, \ldots, \alpha_C]$. Finally, let $\mathbf{X}$ denote the following $S \times C$ matrix:

$$X = \begin{bmatrix} f_1(\mathbf{x}_1) & \cdots & f_C(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{x}_S) & \cdots & f_C(\mathbf{x}_S) \end{bmatrix} \tag{1.14}$$

Then, Equation 1.11 can be vectorially written as

$$\mathbf{y} = \mathbf{X}\alpha + \mathbf{e} \tag{1.15}$$

We choose α to minimize the function

$$\mathrm{RSS}(\alpha) = (\mathbf{y} - \mathbf{X}\alpha)^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\alpha) \tag{1.16}$$

The least squares estimate $\hat{\alpha}$ of α is given by the following formula:

$$\hat{\alpha} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{1.17}$$

The function Equation 1.16 evaluated at $\hat{\alpha}$ is called the residual sum of squares, abbreviated RSS, and is given by

$$\mathrm{RSS} = (\mathbf{y} - \mathbf{X}\hat{\alpha})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\hat{\alpha}) \tag{1.18}$$

It can be shown that an estimate of the variance of the random error is

$$\hat{\sigma}^2 = \frac{\mathrm{RSS}}{S - C} \tag{1.19}$$

and that $\hat{\alpha}$ is unbiased, i.e., $E(\hat{\alpha}) = \alpha$, and the variance of the estimate is

$$\mathrm{var}(\hat{\alpha}) = \hat{\alpha}^2(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} \tag{1.20}$$

Another figure of merit of the linear regression is called the coefficient of determination, denoted by R^2, and given by

$$R^2 = 1 - \frac{\text{RSS}}{\sum_{i=1}^{s} (\text{y}_1 - \bar{y})^2} \tag{1.21}$$

where $\bar{y}$ is the sample average of the y_is, $i = 1, \ldots, S$. The coefficient of determination measures the proportion of variability in the performance that is explained by the regression model. Values of R^2 close to 1 are desirable.

1.7 Variable Screening

Variable screening is used in many performance modeling techniques as a preprocessing step to select the significant circuit parameters that are used as independent variables in the performance model. Variable screening can be used to substantially reduce the number of significant circuit parameters that must be considered, and thereby reduce the complexity of the analysis. The significant circuit parameters are included in the performance model, while the others are ignored. Several variable screening approaches have been proposed; in this chapter, we discuss a simple yet effective screening strategy that was proposed in [8].

The variable screening method of [8] is based on the two-level fractional factorial designs discussed earlier. We assume that the main effects (given by Equation 1.6) of the circuit parameters dominate over the interaction effects and this assumption enables us to use a resolution *III* design. Suppose that there are n original circuit parameters, denoted by $x_1, \ldots, x_n$. Then, we can use the simulation results from the fractional factorial design to compute the following quantities for each of the circuit parameters x_i, $i = 1, \ldots, n$:

1. Main effect v_i computed using Equation 1.6
2. *High deviation* d_i^{h} from the nominal or center point, computed by

$$d_i^{\text{h}} = \frac{2}{n_{\text{r}}} \sum_{k \in K_i^+} y_k - y_{\text{c}} \tag{1.22}$$

   where
   n_{r} is the number of simulations
   K_i^+ is the set of indices where x_i is $+1$
   y_{c} is the performance value at the center point

3. The *low deviation* d_i^{l} from the center point, computed using a formula similar to Equation 1.22, with K_i^+ replaced by K_i^-

Note that only two of the preceding quantities are independent, since the third can be obtained as a linear combination of the other two. A statistical significance test is now used to determine if x_i is significant. To this end, we assume that the quantities v_i, d_i^{h}, and d_i^{l} are sampled from a normal distribution with a sample size of n (since there are n circuit parameters). Denoting any one of the above quantities v_i, d_i^{h}, or d_i^{l} by δ_i, the hypothesis for the significance test is

$$\begin{aligned} &\text{Null hypothesis } H_0: \quad |\delta_i| = 0, \text{ i.e., } x_i \text{ is insignifiant} \\ &\qquad\qquad\qquad\quad H_0 \text{ is rejected if } |\delta_i|/\sigma_\delta > t_{\alpha_s/2, n-1} \end{aligned} \tag{1.23}$$

In Equation 1.23, a two-sided t-test is used [9] with significance level α_s and σ_δ denotes the estimate of the standard deviation of $\delta_i \cdot t_{\alpha_s/2, n-1}$ denotes the value of the t statistic with significance level $\alpha_s/2$ and degrees of freedom $n - 1$. We declare a parameter x_i to be insignificant if it is accepted in the tests for at least two of v_i, d_i^{h}, and d_i^{l}. This screening procedure is *optimistic* in the sense that an insignificant

parameter may be wrongly deemed significant from the preceding significance tests. The values of the *t*-variable can be read off from standard *t*-distribution tables available in many statistics text books.

1.8 Stochastic Performance Models

As stated earlier, the objective of performance modeling is to obtain an inexpensive surrogate to the circuit simulator. This performance model can then be used in a variety of deterministic and statistical optimization tasks in lieu of the expensive circuit simulator. The circuit simulator is a computer model, which is deterministic in the sense that replicate runs of the model with identical inputs will always result in the same values for the circuit performances. This lack of random error makes computer-based experiments fundamentally different from physical experiments. The experimental design and response surface modeling techniques presented earlier have the tacit assumption that the experiments are physical and the difference between the observed performance and the regression model behaves like white noise. In the absence of random error, the rationale for least-squares fitting of a response surface is not clear even though it can be looked upon solely as curve fitting. Stochastic performance modeling [10,11] has been used recently to model the outputs (in our case, the circuit performances) of computer codes or models (in our case, the circuit simulator). The basic idea is to consider the performances as realizations of a stochastic process and this allows one to compute a predictor of the response at untried inputs and allows estimation of the uncertainty of prediction.

The model for the deterministic performance y treats it as a realization of a stochastic process $Y(\mathbf{x})$ which includes a regression model:

$$Y(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i f_i(\mathbf{x}) + Z(\mathbf{x}) \tag{1.24}$$

The first term in Equation 1.24 is the simple regression model for the performance and the second term represents the systematic departure of the actual performance from the regression model as a realization of a stochastic process. This model is used to choose a design that predicts the response well at untried inputs in the parameter space. Various criteria based on functionals of the means square error matrix are used to choose the optimal design. One the design is selected, the circuit simulator is exercised at those design points, and the circuit performance values are extracted. Next, a linear (in the model coefficients $\hat{\alpha}_i s$) predictor is used to predict the values of the performance at untried points. This predicted value can be shown to be the sum of two components: the first component corresponds to the first term in Equation 1.24 and uses the generalized least squares estimate of the coefficients α, and the second component is a smoothing term, obtained by interpolating the residuals at the experimental design points. The main drawback of the stochastic performance modeling approach is the excessive computational cost of obtaining the experimental design and of predicting the performance values.

1.9 Other Performance Modeling Approaches

The most popular performance modeling approach combines experimental design and least-squares-based response surface modeling techniques described previously [8,12–15]. A commonly used assumption valid for MOS circuits that has been used frequently is based on the existence of four critical noise parameters [16–18]. These four critical parameters are the flat-band voltage, the gate-oxide thickness, the channel length reduction, and the channel width reduction.

Several other performance modeling approaches have been proposed, especially in the context of statistical design of integrated circuits. One such approach is based on self-organizing methods [19] and is called the group method of data handling (GMDH) [20]. An interesting application of GMDH for performance modeling is presented in [21,22]. Another interesting approach is called maximally flat

quadratic interpolation (MFQI) [23]. A modeling approach combining the advantages of MFQI and GMDH is presented in [24]. It has been tacitly assumed in the discussion so far that a single RSM is sufficiently accurate to approximate the circuits performances over the entire circuit parameter space. This may not be true for many circuits. In such cases, a piecewise modeling approach may be used advantageously [8]. The circuit parameter space is divided into smaller regions, in each of which a single performance model is adequate. The continuous performance models in the various regions are then *stitched* together to preserve continuity across the regions.

1.10 Example of Performance Modeling

Below, we show an application of performance modeling applied to the clock skew of a clock distribution circuit. Consider the clock driver circuit shown in Figure 1.5. The top branch has three inverters and the lower branch has two inverters to illustrate the problem of signal skew present in many clock trees. We define the clock skew to be the difference between the times at which CLK and its complement cross a threshold of 0.5 V_{DD}. Two skews occur, as shown in Figure 1.6, one corresponding to the rising edge of the CLK, ΔS_r, and another corresponding to the falling edge of CLK, ΔS_f. The performance measure of interest is the signal skew ΔS defined as the larger of the two skews.

The designable circuit parameters of interest in this example are the nominal channel widths of each of the transistors in the second and third inverters in the top branch and the last inverter in the lower

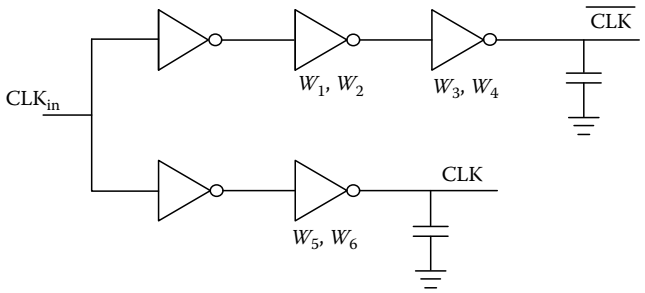


FIGURE 1.5 Clock distribution circuit.

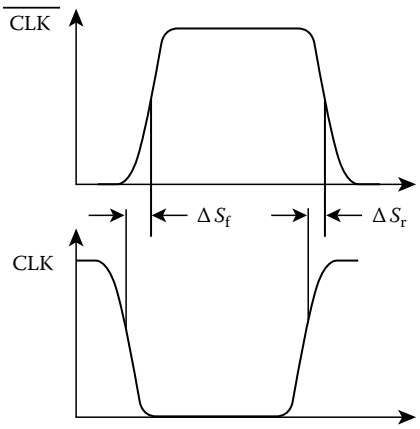


FIGURE 1.6 Definition of rising and falling clock skews.

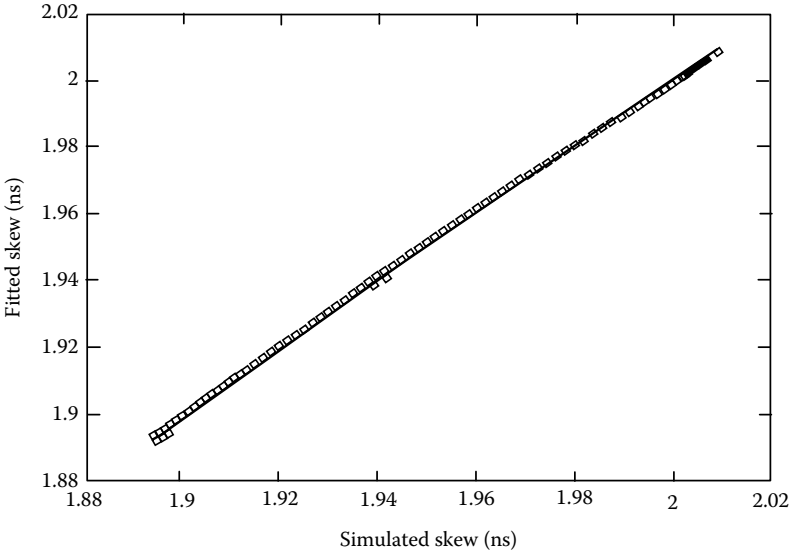


FIGURE 1.7 Plot of predicted versus simulated values for RSM of Equation 1.25.

branch. The designable parameters are denoted by W_1, W_2, W_3, W_4, W_5, and W_6 and are marked in Figure 1.5. The noise parameters of interest are the random variations in the channel widths and lengths of all nMOS and pMOS transistors in the circuit, and the common gate oxide thickness of all the transistors. These are denoted by ΔW_n (channel width variation of nMOS transistors), ΔL_n (channel length variation nMOS transistors), ΔW_p (channel width variation of pMOS transistors), ΔL_p (channel length variation of pMOS transistors), and t_{ox} (gate oxide thickness of all nMOS and pMOS transistors). We hypothesize a linear RSM for the clock skew in terms of the designable and noise parameters of the circuit. To this end, we generate an experimental design of size 30 using Latin hypercube sampling. The circuit is simulated at each of these 30 training points, and the value of the clock skew ΔS is extracted from the simulation results. Then, we use regression to fit the following linear RSM:

$$\begin{aligned}\Delta S = {} & 1.93975 - 0.0106W_1 + 0.0121W_2 + 0.0045W_3 - 0.1793W_4 \\ & + 0.0400W_5 + 0.0129W_6 - 0.0149\Delta W_n - 0.0380\Delta W_p \\ & - 0.0091\Delta L_n + 0.0115\Delta L_p + 0.161t_{ox}\end{aligned} \tag{1.25}$$

This RSM has an R^2 of 0.999. The prediction accuracy for the preceding RSM is in Figure 1.7 where the values of ΔS predicted from the RSM are plotted against the simulated values for a set of checking points (different from the training points used to fit the RSM). Most of the points lie near the 45° line indicating that the predicted values are in close agreement with the simulated values.

References

1. G. E. P. Box and N. R. Draper, *Empirical Model Building and Response Surfaces*, New York: Wiley, 1987.
2. G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*, New York: Wiley, 1987.
3. N. R. Draper and T. J. Mitchell, The construction of saturated 2_R^{k-p} designs, *Ann. Math. Stat.*, 38: 1110–1126, August 1967.

4. P. J. Ross, *Taguchi Techniques for Quality Engineering*, New York: McGraw-Hill, 1988.
5. J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, London: Methuen and Co. Ltd., 1964.
6. M. D. McKay, R. J. Beckman, and W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, 21: 239–245, May 1979.
7. R. L. Iman and W. J. Conover, A distribution-free approach to inducing rank correlations among input variables, *Commn. Stat*., B11(3): 311–334, 1982.
8. K. K. Low and S. W. Director, An efficient methodology for building macromodels of IC fabrication processes, *IEEE Trans. Comput. Aided Des*., 8: 1299–1313, December 1989.
9. D. C. Montgomery and E. A. Peck, *Introduction to Linear Regression Analysis*, New York: Wiley, 1982.
10. J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Design and analysis of computer experiments, *Stat. Sci*., 4(4): 409–435, 1989.
11. M. C. Bernardo, R. Buck, L. Liu, W. A. Nazaret, J. Sacks, and W. J. Welch, Integrated circuit design optimization using a sequential strategy, *IEEE Trans. Comput. Aided Des*., 11: 361–372, March 1992.
12. T. K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick, Statistical modeling of VLSI circuit performances, *Proceedings of the IEEE International Conference Computer-Aided Design*, Santa Clara, CA, November 1986, pp. 224–227.
13. T. K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick, Statistical performance modeling and parametric yield estimation of MOS VLSI, *IEEE Trans. Comput. Aided Des*., CAD-6: 1013–1022, November 1987.
14. T. K. Yu, S. M. Kang, W. Welch, and J. Sacks, Parametric yield optimization of CMOS analog circuits by quadratic statistical performance models, *Int. J. Circuit Theory Appl*., 19: 579–592, November 1991.
15. A. Dharchoudhury and S. M. Kang, An integrated approach to realistic worst-case design optimization of MOS analog circuits, *Proceedings of the 29th ACM/IEEE Design Automation Conference*, Anaheim, CA, June 1992, pp. 704–709.
16. P. Cox, P. Yang, S. S. Mahant-Shetti, and P. Chatterjee, Statistical modeling for efficient parametric yield estimation, *IEEE J. Solid State Circuits*, SC-20: 391–398, February 1985.
17. P. Yang and P. Chatterjee, Statistical modeling of small geometry MOSFETs, *IEEE International Electron Device Meeting*, San Franscisco, CA, December 1982, pp. 286–289.
18. P. Yang, D. E. Hocevar, P. Cox., C. Machala, and P. Chatterjee, An integrated and efficient approach for MOS VLSI statistical circuit design, *IEEE Trans. Comput. Aided Des*., 5: 5–14, January 1986.
19. S. J. Farlow, Ed., *Self-Organizing Methods in Modeling: GMDH Type Algorithms*, New York: Marcel Dekker, 1984.
20. A. G. Ivakhnenko, The group method of data handling, a rival of the method of stochastic approximation, *Sov. Autom. Control*, 13(3): 43–45, 1968.
21. A. J. Strojwas and S. W. Director, An efficient algorithm for parametric fault simulation of monolithic IC's, *IEEE Trans. Comput. Aided Des*., 10: 1049–1058, August 1991.
22. S. Ikeda, M. Ochiai, and Y. Sawaragi, Sequential GMDH and its application to river flow prediction, *IEEE Trans. Syst. Man Cybern*., SMC-6: 473–479, July 1976.
23. R. M. Biernacki and M. A. Styblinski, Statistical circuit design with a dynamic constraint approximation scheme, *Proceedings of the IEEE ISCAS*, San Jose, CA, 1986, pp. 976–979.
24. M. A. Styblinski and S. A. Aftab, Efficient circuit performance modeling using a combination of interpolation and self organizing approximation techniques, *Proceedings of the IEEE ISCAS*, New Orleans, LA, 1990, pp. 2268–2271.

2

Symbolic Analysis Methods

2.1 Introduction **2**-1
2.2 Symbolic Modified Nodal Analysis **2**-3
2.3 Solution Methods **2**-7
Determinant-Based Solutions • Parameter Reduction Solutions
2.4 Ideal Operational Amplifiers **2**-12
Nullator, Norator, and Nullor • Ideal Op-Amp Model
2.5 Applications of Symbolic Analysis **2**-16
Frequency Response Evaluation • Circuit Response Optimization • Sensitivity Analysis • Circuit Sizing • Parameter Extraction in Device Modeling • Statistical Analysis • Fault Diagnosis of Analog Circuits • Insight into Circuit Operation • Education
2.6 Symbolic Analysis Software Packages **2**-18
References **2**-19

Benedykt S. Rodanski
University of Technology, Sydney

Marwan M. Hassoun
Iowa State University

2.1 Introduction

The late 1980s and 1990s saw a heightened interest in the development of symbolic analysis techniques and their applications to integrated circuits analysis and design. This trend continues in the new millennium. It resulted in the current generation of symbolic analysis methods, which include several software implementations like ISSAC [Gie89], SCAPP [Has89], ASAP [Fer91], EASY [Som91], SYNAP [Sed88], SAPWIN [Lib95], SCYMBAL [Kon88], GASCAP [Hue89], SSPICE [Wie89], STAINS [Pie01], and Analog Insydes [Hen00]. This generation is characterized by an emphasis on the production of usable software packages and a large interest in the application of the methods to the area of analog circuit design. This includes extensive efforts to reduce the number of symbolic expression generated by the analysis through hierarchical methods and approximation techniques [Fer92,Sed92,Wam92,Hsu94, Yu96]. This generation also includes attempts at applying symbolic analysis to analog circuit synthesis [Gie91,Cab98], parallel processor implementations [Has93b,Mat93,Weh93], time-domain analysis [Has91,Als93,Lib93], and behavioral signal path modeling [Ley00].

The details of traditional symbolic analysis techniques as they apply to linear circuit theory are given in [Rod08]. This chapter addresses symbolic analysis as a computer-aided design tool and its application to integrated circuit design. Modern symbolic analysis is mainly based on either a topological approach or a modified nodal analysis (MNA) approach. The basic topological (graph-based) methodologies are discussed in [Rod08] because the methods are part of traditional symbolic analysis techniques. The methodology addressed in this chapter is based on the concept of MNA [Ho75] as it applies to symbolic analysis.

Traditional symbolic circuit analysis is performed in the frequency domain where the results are in terms of the frequency variable s. The main goal of performing symbolic analysis on a circuit in the frequency domain is to obtain a symbolic transfer function of the form:

$$H(s,\mathbf{x}) = \frac{N(s,\mathbf{x})}{D(s,\mathbf{x})}, \quad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_p \end{bmatrix}, \quad p \leq p_{\text{all}} \tag{2.1}$$

The expression is a rational function of the complex frequency variable s, and the variables x_1 through x_p representing the variable circuit elements, where p is the number of variable circuit elements and p_{all} is the total number of circuit elements. Hierarchical symbolic analysis approaches are based on a decomposed form of Equation 2.1 [Sta86,Has89,Has93a,Pie01]. This hierarchical representation is referred to as a sequence of expressions representation to distinguish it from the single expression representation of Equation 2.1. The major advantage of having a symbolic transfer function in single expression form is the insight that can be gained by observing the terms in both the numerator and the denominator. The effects of the different terms can, perhaps, be determined by inspection. This process is very valid for the cases where there are relatively few symbolic terms in the expression.

The heart of any circuit simulator, symbolic or numeric, is the circuit analysis methodology used to obtain the mathematical equations that characterize the behavior of the system. The circuit must satisfy these equations. These equations are of three kinds: Kirchhoff's current law (KCL), Kirchhoff's voltage law (KVL), and the branch relationship (BR) equations [Chu75]. KCL and KVL are well known and have been discussed earlier in this book. The definition of BR is ''For independent branches, a branch relationship defines the branch current in terms of the branch voltages.'' The most common independent branches are resistors, capacitors, and inductors. For the case of dependent branches, the branch current or voltage is defined in terms of other branches' currents or voltages. The four common types of dependent branches are the four types of controlled sources: voltage-controlled voltage source (VCVS), voltage-controlled current source (VCCS), current-controlled voltage source (CCVS), and current-controlled current source (CCCS). A linear element would produce a BR equation that is linear, and a nonlinear element would produce a nonlinear BR equation. The scope of this section is linear elements. If a nonlinear element is to be included in the circuit, a collection of linear elements could be used to closely model the behavior of that element around a dc operating point (quiescent point) in a certain frequency range [Gie94]. BR equations are not used on their own to formulate a system of equations to characterize the circuit. They are, however, used in conjunction with KCL or KVL equations to produce a more complete set of equations to characterize the circuit. Hybrid analysis [Chu75] uses a large set of BR equations in its formulation.

A system of equations to solve for the circuit variables, a collection of node voltages and branch currents, does not need to include all of the above equations. Several analysis methods are based on these equations or a subset thereof. Mathematically speaking, an equal number of circuit unknowns and linearly independent equations is required in order for a solution for the system to exist. The most common analysis methods used in circuit simulators are as follows:

1. A modification of nodal analysis as used in the numeric circuit simulators SPICE2 [Nag75], and SLIC [Idl71], and in the formulation of the symbolic equations in [Ald73]. Nodal analysis [Chu75] uses KCL to produce the set of equations that characterize the circuit. Its advantage is its relatively small number of equations. This results in a smaller matrix, which is very desirable in computer implementations. Its limitation, however, is that it only allows for node-to-datum voltages as variables. No branch current variables are allowed. Also the only type of independent power sources it allows are independent current sources. However, voltage sources are handled by using their Norton equivalent. A further limitation to this method is that only VCCSs are allowed, none of the other dependent sources are allowed in the analysis. To overcome these limitations, the MNA method was proposed in [Ho75]. It allows branch current as variables in the analysis,

which in turn leads to the ability to include voltage sources and all four types of controlled sources in the analysis. The MNA method is described in more detail in the next section because it is used in symbolic analysis after further modification to it. These modifications are necessary in order for the analysis to accept ideal operational amplifiers (op-amps), and to produce a compact circuit characterization matrix. This matrix is referred to as the reduced MNA matrix (RMNAM).

2. Hybrid analysis [Chu75] is used in the numeric simulators ASTAP [Wee73] and ECAP2 [Bra71]. The method requires the selection of a tree and its associated co-tree. It uses cutset analysis (KCL) and loop analysis (KVL) to formulate the set of equations. The equations produced solve for the branch voltages and branch currents. The method is able to accommodate voltage sources and all four types of dependent sources. It utilizes a much larger matrix than the nodal analysis methods; however, it is a very sparse matrix (lots of zero entries). The main drawback is that the behavior of the equations is highly dependent on the tree selected. A bad tree could easily result in an ill-conditioned set of equations. Some methods of tree selection can guarantee a well-conditioned set of hybrid equations [Nag75]. Therefore, special rules must be observed in the tree selection process that constitutes a large additional overhead on the analysis.
3. Sparse tableau analysis [Hac71] has been used in symbolic analysis to employ the parameter extraction method (this chapter). It uses the entire set of circuit variables: node voltages, branch voltages, and branch currents, for the symbolic formulation in addition to capacitor charges and inductor fluxes in the case of numeric simulations. This results in a huge set of equations and in turn a very large matrix but a very sparse one. This method uses all the above stated equation formulation methods: KCL, KVL and the BR. The entire system of equations is solved for all the circuit variables. The disadvantage of the method is its inherent problem of ill-conditioning [Nag75].

It has been found that the sparse tableau method only produces a very minor improvement in the number of mathematical operations, the nodal analysis methods despite its large overhead, stemming from a large matrix, and the large and rigid set of variables that have to be solved for. The MNA method gives the choice of what branch current variables to be solved for and equations are added accordingly. The tableau method would always have a fixed size matrix and produce unneeded information to the user.

For the purpose of symbolic analysis, as highlighted in this chapter, the entire analysis is performed in the frequency domain with the aid of the complex frequency variable s. The Laplace transform representation of the value of any circuit element is therefore used in the formulation. Also, since a modification of the nodal admittance matrix is used, the elements may be represented by either their admittance or impedance values.

2.2 Symbolic Modified Nodal Analysis

The initial step of MNA is to formulate the nodal admittance matrix **Y** [Chu75] from the circuit. The circuit variables considered here are all the node-to-datum voltages, referred to simply as node voltages; there are n_v of them. They are included in the variable vector **V**. So **V** has the dimensions of $(n_v \times 1)$. The vector **J**, also of dimensions $(n_v \times 1)$, represents the values of all independent current sources in the circuit. The ith entry of **J** is the sum of all independent current sources entering node i. The linear system of node equations can be represented in the following matrix form:

$$\mathbf{YV} = \mathbf{J} \tag{2.2}$$

Row i of **Y** represents the KCL equation at node i. The coefficient matrix, **Y**, called the node admittance matrix, is constructed by writing KCL equations at each node, except for the datum node. The ith

equation then would state that the sum of all currents leaving node *i* is equal to zero. The equations are then put into the matrix form of Equation 2.2. The following example illustrates the process.

Example 2.1

Consider the circuit in Figure 2.1.

Collecting the node voltages would produce the following **V**:

$$\mathbf{V} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{2.3}$$

Considering all the independent current sources in the circuit, the vector **J** becomes

$$\mathbf{J} = \begin{bmatrix} J_1 \\ J_4 \\ 0 \end{bmatrix} \tag{2.4}$$

Notice that the last entry of **J** is a zero because no independent current sources are connected to node 3.

Now, writing KCL equations at the three non-datum nodes produces

$$\begin{aligned} &G_2 v_1 + G_3(v_1 - v_2) + \frac{1}{sL_7}(v_1 - v_3) = J_1 \\ &G_5 v_2 + G_3(v_2 - v_1) + sC_6(v_2 - v_3) = J_4 \\ &sC_6(v_3 - v_2) + \frac{1}{sL_7}(v_3 - v_1) + g_8 V_{R_3} = 0 \end{aligned} \tag{2.5}$$

Substituting $V_{R_3} = v_1 - v_2$ in the third equation of the set Equation 2.5 and rearranging the variables results in

$$\begin{aligned} &\left(G_2 + G_3 + \frac{1}{sL_7}\right)v_1 - G_3 v_2 - \frac{1}{sL_7} v_3 = J_1 \\ &-G_3 v_1 + (G_3 + G_5 + sC_6)v_2 - sC_6 v_3 = J_4 \\ &\left(g_8 - \frac{1}{sL_7}\right)v_1 - (g_8 + sC_6)v_2 + \left(\frac{1}{sL_7} + sC_6\right) = 0 \end{aligned} \tag{2.6}$$

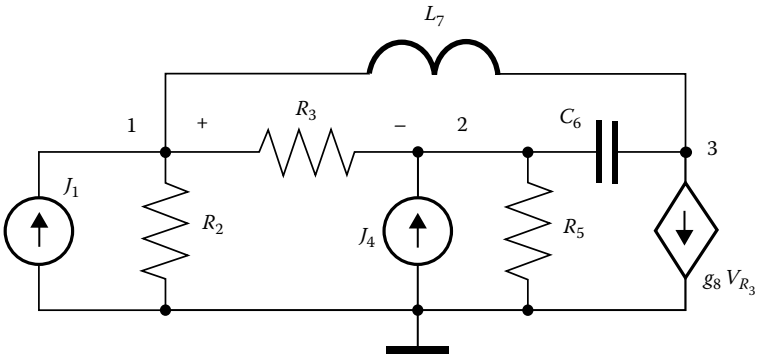


FIGURE 2.1 MNA example circuit.

Finally, rewriting Equation 2.6 in a matrix form of Equation 2.2 yields

$$\begin{bmatrix} G_2+G_3+\dfrac{1}{sL_7} & -G_3 & -\dfrac{1}{sL_7} \\ -G_3 & G_3+G_5+sC_6 & -sC_6 \\ g_8-\dfrac{1}{sL_7} & -(g_8+sC_6) & \dfrac{1}{sL_7}+sC_6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} J_1 \\ J_4 \\ 0 \end{bmatrix} \tag{2.7}$$

An automatic technique to construct the node admittance matrix is the element stamp method [Ho75]. The method consists of going through each circuit component and adding its contribution to the nodal admittance matrix in the appropriate positions. It is an easy way to illustrate the impact of each element on the matrix.* Resolving Example 2.1 using the element stamps would produce exactly the same **Y** matrix, **V** vector, and **J** vector as in Equation 2.7.

The MNA technique, introduced in [Ho75], expands on the above nodal analysis in order to readily include independent voltage sources and the three other types of controlled sources: VCVS, CCCS, and CCVS. This is done by introducing some branch currents as variables into the system of equations, which in turn allows for the introduction of any branch current as an extra system variable. Each extra current variable introduced would need an additional equation to solve for it. The extra equations are obtained from the BRs for the branches whose currents are the extra variables. The effect on the nodal admittance matrix is the deletion of any contribution in it due to the branches whose currents have been declared as variables. This matrix is referred to as $\mathbf{Y}_n$. The addition of new variables and a corresponding number of equations to the system results in the need to append extra rows and columns to $\mathbf{Y}_n$. The augmented $\mathbf{Y}_m$ matrix is referred to as the MNA matrix (MNAM). The new system of equations, in matrix form, is

$$\begin{bmatrix} \mathbf{Y}_n & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ \mathbf{E} \end{bmatrix} \tag{2.8}$$

where

I is a vector of size n_i whose elements are the extra branch current variables introduced
E contains the values of the independent voltage sources
C and **D** correspond to BR equations for the branches whose currents are in **I**

An important feature of the MNAM approach is its ability to include impedance elements in the formulation. Let Z_x be an impedance connected between nodes i and j. Introduce the current through Z_x as new variable i_x. The extra equation needed is provided by the corresponding BR: $v_i - v_j - Z_x i_x = 0$. The usefulness of this approach can be appreciated when one needs to handle short circuits ($R = 0$) or inductors at dc. It becomes indispensable in approximation techniques that require the circuit matrix to contain only entries of the type $a = \alpha + s\beta$.

According to [Ho75], the MNA current variables should include all branch currents of independent voltage sources, controlled voltage sources, and all controlling currents. Further modifications of this procedure, like RMNA formulations [Has89], compacted MNA (CMNA) formulations [Gie89], and supernode analysis [Som93] relax these constraints and result in fewer MNA equations.

Example 2.2

As an example of a MNAM formulation, consider the circuit of Figure 2.2. The extra current variables are the branch currents of the independent voltage source E_1 (branch 1), the CCVS (branch 5), and the

* Program STAINS has the animate option that allows viewing the animated process of matrix formulation.

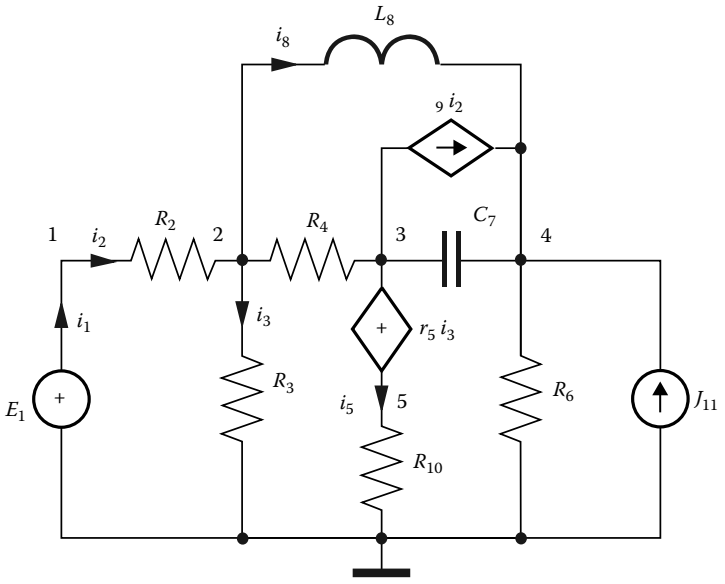


FIGURE 2.2 Circuit of Example 2.2.

inductor L_8 (branch 8). They are referred to as i_1, i_5, and i_8, respectively. The CCCS $\beta_9 i_2$ can be transformed to a VCCS by noticing that $i_2 = G_2(v_1 - v_2)$, so we have $\beta_9 i_2 = \beta_9 G_2(v_1 - v_2)$. Similarly, the CCVS $r_5 i_3$ becomes a VCVS $r_5 G_3 v_2$.

Using element stamps, the MNA system of equations becomes

$$\left[\begin{array}{ccccc|ccc}
G_2 & -G_2 & 0 & 0 & 0 & 1 & 0 & 0 \\
-G_2 & G_2+G_3+G_4 & 0 & 0 & 0 & 0 & 0 & 1 \\
\beta_9 G_2 & -G_4-\beta_9 G_2 & G_4+sC_7 & -sC_7 & 0 & 0 & 1 & 0 \\
-\beta_9 G_2 & \beta_9 G_2 & -sC_7 & G_6+sC_7 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & G_{10} & 0 & -1 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -r_5 G_3 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & -sL_8
\end{array}\right]
\left[\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ \hline i_1 \\ i_5 \\ i_8 \end{array}\right]
=
\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ J_{11} \\ 0 \\ \hline E_1 \\ 0 \\ 0 \end{array}\right]$$

The above system of linear algebraic equations can be written in the following matrix form:

$$\mathbf{Ax} = \mathbf{b} \tag{2.9}$$

where

A is a symbolic matrix of dimension $n \times n$
x is a vector of circuit variables of length n
b is a symbolic vector of constants of length n
n is the number of circuit variables: currents, voltages, charges, or fluxes

The analysis proceeds by solving Equation 2.9 for **x**.

2.3 Solution Methods

2.3.1 Determinant-Based Solutions

Here the basic idea is to apply an extension of Cramer's rule, symbolically, to find a transfer function from the coefficient matrix of Equation 2.9. According to Cramer's rule, the value of a variable x_i in Equation 2.9 is calculated from the following formula:

$$x_i = \frac{\left|\mathbf{A}^{(i)}\right|}{|\mathbf{A}|} \tag{2.10}$$

where $\mathbf{A}^{(i)}$ is obtained from matrix $\mathbf{A}$ by replacing column i with the vector $\mathbf{b}$ and $|\bullet|$ denotes the determinant. Any transfer function, which is a ratio of two circuit variables, can be found by application of Equation 2.10:

$$\frac{x_i}{x_j} = \frac{\left|\mathbf{A}^{(i)}\right|/|\mathbf{A}|}{\left|\mathbf{A}^{(j)}\right|/|\mathbf{A}|} = \frac{\left|\mathbf{A}^{(i)}\right|}{\left|\mathbf{A}^{(j)}\right|} \tag{2.11}$$

Several symbolic algorithms are based on the concept of using Cramer's rule and calculating the determinants of an MNAM. Most notably is the program ISSAC [Gie89], which uses a recursive determinant expansion algorithm to calculate the determinants of Equation 2.11. Although there are many other algorithms to calculate the determinant of a matrix, like elimination algorithms and nested minors algorithms [Gie91], recursive determinant expansion algorithms were found the most suitable for sparse linear matrices [Wan77]. These algorithms are cancellation-free given that all the matrix entries are different, which is a very desirable property in symbolic analysis.

Let $\mathbf{A}$ be a square matrix of order n. Delete row i and column j from $\mathbf{A}$ and denote the resulting matrix of order $n-1$ by $\mathbf{A}_{-i-j}$. Its determinant, $M_{ij} = |\mathbf{A}_{-i-j}|$, is called a *minor* of $\mathbf{A}$ of order $n-1$. The quantity $\Delta_{ij} = (-1)^{i+j} M_{ij}$ is called a *cofactor* of a matrix element a_{ij}. The determinant of $\mathbf{A}$ can be calculated by recursive application of the following formula:*

$$|\mathbf{A}| = \begin{cases} \sum_{j=1}^{n} a_{ij}\Delta_{ij} = \sum_{j=1}^{n} a_{ji}\Delta_{ji} & \text{for } n > 1 \\ a_{11} & \text{for } n = 1 \end{cases} \tag{2.12}$$

where i is an arbitrary row or column of matrix $\mathbf{A}$.

As an example, the determinant of a 3×3 matrix is calculated by this method as

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = (-1)^{1+1} a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + (-1)^{1+2} a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + (-1)^{1+3} a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

Here, i was chosen to be, arbitrarily, row 1. The expression is cancellation-free in this example because all the matrix entries are unique. Note that for matrices with duplicated entries, the algorithm is not cancellation-free.

* Single application of Equation 2.12 is known as the Laplace expansion of the determinant.

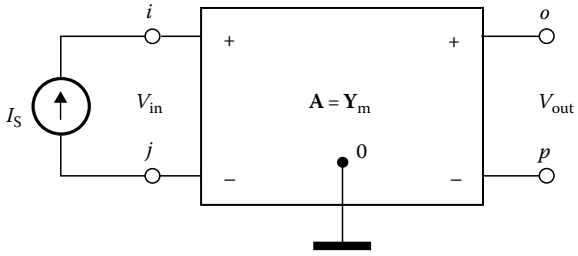


FIGURE 2.3 Two-port representation of a circuit.

Most symbolic analysis problems deal with two-port circuits of the form shown in Figure 2.3, where the signal source, I_S, is the only independent source in the circuit.

Let the two-port be described by the modified nodal equation of the form as in Equation 2.8. The right-hand side of this equation will have only two nonzero entries: $+I_S$ at position i and $-I_S$ at position j. Applying Cramer's rule to calculate potential at any node k, we get

$$v_k = \frac{I_S \Delta_{ik} - I_S \Delta_{jk}}{\Delta} \tag{2.13}$$

Now, if we wish, for example, to calculate the voltage transmittance of the two-port in Figure 2.3, using Equation 2.13, we obtain

$$T_v = \frac{v_o - v_p}{v_i - v_j} = \frac{\Delta_{io} - \Delta_{jo} - \Delta_{ip} + \Delta_{jp}}{\Delta_{ii} - \Delta_{ji} - \Delta_{ij} + \Delta_{jj}} \tag{2.14}$$

Equation 2.14 simplifies considerably if, as it is often the case, nodes j and p are identical with the reference node. We then have

$$T_v = \frac{v_o}{v_i} = \frac{\Delta_{io}}{\Delta_{ii}} \tag{2.15}$$

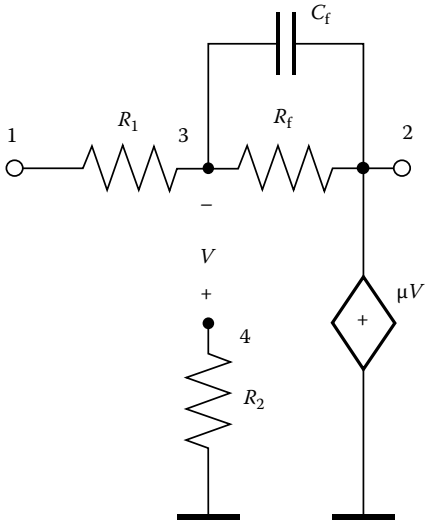


FIGURE 2.4 Circuit for Example 2.3.

Example 2.3

As an example, the voltage transfer function $T_v = v_2/v_1$ for the circuit in Figure 2.4 is to be calculated. The MNAM of this circuit is

$$\mathbf{Y}_m = \left[\begin{array}{cccc|c} G_1 & 0 & -G_1 & 0 & 0 \\ 0 & G_f + sC_f & -G_f - sC_f & 0 & 1 \\ -G_1 & -G_f - sC_f & G_1 + G_f + sC_f & 0 & 0 \\ 0 & 0 & 0 & G_2 & 0 \\ \hline 0 & 1 & \mu & -\mu & 0 \end{array}\right]$$

Using Equation 2.15 and expanding the cofactors according to Equation 2.12, we get

$$T_v = \frac{v_2}{v_1} = \frac{\Delta_{12}}{\Delta_{11}} = \frac{(-1)^{1+2}\begin{vmatrix} 0 & -G_f - sC_f & 0 & 1 \\ -G_1 & G_1 + G_f + sC_f & 0 & 0 \\ 0 & 0 & G_2 & 0 \\ 0 & \mu & -\mu & 0 \end{vmatrix}}{(-1)^{1+1}\begin{vmatrix} G_f + sC_f & -G_f - sC_f & 0 & 1 \\ -G_f - sC_f & G_1 + G_f + sC_f & 0 & 0 \\ 0 & 0 & G_2 & 0 \\ 1 & \mu & -\mu & 0 \end{vmatrix}}$$

$$= \frac{-G_1 G_2 \mu}{-G_2[\mu(-G_f - sC_f) - (G_1 + G_f + sC_f)]} = -\frac{G_1 \mu}{G_1 + G_f + sC_f + \mu(G_f + sC_f)}$$

Note that cancellation of a common term (G_2) in the numerator and denominator occurred. Although not present in the above example, cancellation of terms in each determinant expansion may occur in general if matrix entries are not unique.

2.3.2 Parameter Reduction Solutions

These methods use basic linear algebra techniques, applied symbolically, to find the solution to Equation 2.9. The goal here is to reduce the size of the system of equations by manipulating the entries of matrix **A** down to a 2×2 size. The two variables remaining in **x** are the ones for which the transfer function is to be calculated. The manipulation process, which in simple terms is solving for the two desired variables in terms of the others, is done using methods like Gaussian elimination [Nob77]. This methodology is used in the symbolic analysis programs SCAPP [Has89] and STAINS [Pie01]. Both programs use successive application of a modified Gaussian elimination process to produce the transfer function of the solution. The main difference between the techniques used in SCAPP and STAINS is in pivot selection strategy. While SCAPP selects pivots from the diagonal of **A**, STAINS employs a locally optimal pivot selection [Mar57,Gus70] that minimizes the number of arithmetic operations at each elimination step, thus producing more compact sequences of expressions. The process is applied to the MNAM and the result is a RMNAM ($\mathbf{Y}_R$). The process of reducing a circuit variable, whether a node voltage or a branch current, is referred to as variable suppression.

If the signal (input) source is the only independent source in the circuit, then the reduction process leads to the final RMNAM being identical with the well-known two-port admittance matrix. Its elements are known as the short-circuit admittance parameters or simply the *y*-parameters and have been used in circuit analysis from the 1940s. Since the two-port representation is the most commonly encountered in symbolic analysis problems, for the rest of this section we concern ourselves with terminated two-ports, depicted in Figure 2.5. The advantage of such a representation is the fact that once the *y*-parameters and the terminating admittances, Y_S and Y_L, are known, all other parameters of interest, like input/output impedances and various gains, can be readily calculated (e.g., [Gha65]). With no loss of generality we can assign numbers 1 and 2 to the input and output nodes, respectively. All other circuit variables are numbered from 3 to *n*. To focus our attention, we will only consider a (very typical) case when the input and output ports have a common terminal, which is the reference node.

The circuit of Figure 2.5 can be described by the set of symbolic equations of the form as in Equation 2.9, rewritten here more explicitly:

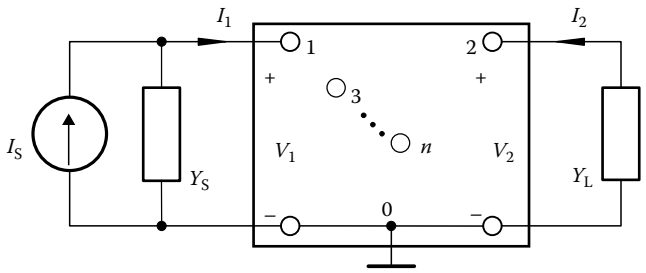


FIGURE 2.5 Terminated two-port for symbolic analysis.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.16}$$

The elements a_{ij} are symbolic entries of general form: $a = \alpha + s\beta$.

Since we are only interested in the relationship between the input and output variables (V_1, I_1 and V_2, I_2 in our case), all other variables can be suppressed. Suppose that we wish to suppress the variable x_p. To achieve this, we can use any equation from the set (Equation 2.16), except the first two equations that has a nonzero coefficient at x_p. Let us choose it to be equation $q > 2$. The qth equation can be written in the expanded form as follows:

$$a_{q1}V_1 + a_{q2}V_2 + \cdots + a_{qp}x_p + \cdots + a_{qn}x_n = 0 \tag{2.17}$$

Provided that $|a_{qp}| \neq 0$, we can calculate x_p from Equation 2.17 as

$$x_p = -\frac{a_{q1}}{a_{qp}}V_1 - \frac{a_{q2}}{a_{qp}}V_2 - \frac{a_{q3}}{a_{qp}}x_3 - \cdots - \frac{a_{qn}}{q_{qp}}x_n \tag{2.18}$$

Substituting Equation 2.18 into Equation 2.16 will eliminate the variable x_p and equation q from the set. During the elimination, each element a_{ij} of **A** undergoes the transformation:

$$a_{ij} \leftarrow a_{ij} - \frac{a_{qj}a_{ip}}{a_{qp}}; \quad i, j = 1, 2, \ldots, n; \quad i \neq q, \quad j \neq p \tag{2.19}$$

This process of suppression of a variable is the very well-known Gaussian elimination. The only difference from the usual appearance of the elimination formula (Equation 2.19) in the literature is the fact that the pivot, a_{qp}, may be off-diagonal ($p \neq q$). In practice, the transformation (Equation 2.19) is only applied to the matrix elements a_{ij} for which $|a_{qj}||a_{ip}| \neq 0$. When any of the updated elements of **A**

is initially zero, a new nonzero element, a fill-in, is created in **A**. The pivot is chosen as to minimize either the number of arithmetic operations* or the number of fill-ins created at each elimination step.

All internal variables are successively eliminated using identical procedure. At the end of this process, we are left with a set of two equations:

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} I_S - Y_S V_1 \\ -Y_L V_2 \end{bmatrix} \tag{2.20}$$

Note that the two entries in the right-hand side vector, I_1 and I_2, have not been updated during the elimination process, thus coefficients y_{ij} in Equation 2.20 are the short-circuit admittance parameters of the original two-port, described by Equation 2.16. Various gain and immittance relations (network functions) can be expressed in terms of the two-port parameters and the termination admittances [Gha65]. For example, the voltage transmittance, V_2/V_1, and the current transmittance, $-I_2/I_1$, can be calculated as

$$T_v = \frac{V_2}{V_1} = -\frac{y_{21}}{y_{22} + Y_L}, \quad T_i = \frac{-I_2}{I_1} = \frac{-y_{21} Y_L}{y_{11}(y_{22} + Y_L) - y_{12} y_{21}}$$

A simple example serves to illustrate and easily verify the formulation of the RMNAM and advantages of a proper pivoting strategy.

Example 2.4

Consider the circuit in Figure 2.6.

Using element stamps, the modified nodal equation of the circuit in Figure 2.4 can be written as

$$\begin{bmatrix} G_1 & 0 & -G_1 & 0 \\ 0 & G_4 + sC_3 & -sC_3 & 0 \\ -G_1 & -sC_3 & G_1 + sC_3 & 1 \\ -\mu & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ v_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ 0 \\ 0 \end{bmatrix} \tag{2.21}$$

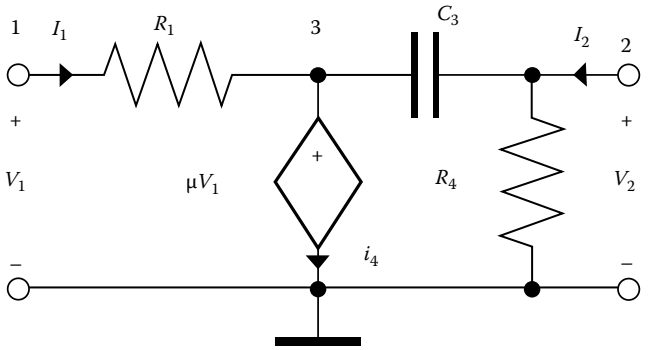


FIGURE 2.6 Circuit of Example 2.4.

* Minimization of the number of arithmetic operations is equivalent with minimization of the number of matrix updates via Equation 2.19. If the numbers of nonzero elements in the pivot row and column are n_q and n_p, respectively, then the number of updates required to suppress x_p using equation q is equal to $(n_q - 1)(n_p - 1)$.

To reduce the coefficient matrix in Equation 2.21 to the two-port admittance matrix, the variables v_3 and i_4 must be eliminated. If we select pivots from the main diagonal only, we have no choice but to first take the element $a_{33} = G_1 + sC_3$. Since row 3 and column 3 have four nonzero elements each, elimination of v_3 using the third equation requires $(4 - 1)(4 - 1) = 9$ updates. Moreover, six fill-ins will be created, making the matrix full. Four more updates are required to suppress the current variable, i_4, increasing the total number of updates to 13. On the other hand, if we are not restricted to diagonal pivots, the best choice for the first one is the element $a_{34} = 1$. Inspection of Equation 2.21 reveals that elimination of the fourth variable (i_4) using equation 3 requires no updates at all ($n_p - 1 = 0$). With no computational effort, we obtain

$$\begin{bmatrix} G_1 & 0 & -G_1 \\ 0 & G_4 + sC_3 & -sC_3 \\ -\mu & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ 0 \end{bmatrix} \tag{2.22}$$

Now, using the element $a_{33} = 1$ of Equation 2.22 as a pivot, only two updates are required to suppress variable v_3. The final equation becomes

$$\begin{bmatrix} G_1 - \mu G_1 & 0 \\ -\mu sC_3 & G_4 + sC_3 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

By unrestricted pivot selection, we have achieved the desired result with only 2 updates, compared to 13 updates if the pivot was chosen from the main diagonal only.

2.4 Ideal Operational Amplifiers

Neither the classical nodal analysis, nor the MNA in its original form [Ho75], accounts for ideal op-amps in their equation formulation. The way op-amps have usually been handled in those analyses is by modeling each op-amp by a VCVS with a large gain in an attempt to characterize the infinite gain of the ideal op-amp.

The ideal op-amp is a two-port device for which one of the output terminals is usually assumed to be grounded.* Op-amps are used extensively in the building of a large number of analog circuits, especially analog filters, where symbolic circuit simulators have found extensive applications in obtaining transfer functions for the filters. It is therefore necessary for a modern symbolic analysis program to handle ideal op-amps. Both SCAPP and STAINS include ideal op-amps in their lists of available circuit components.

There are two ways by which the ideal op-amps can be handled in an automatic formulation of circuit equation. The first method is simply an extension of the MNA, which accounts for op-amps by adding the constraining equations to the MNA set, introducing an element stamp for the op-amp [Vla94]. The other approach reduces the size of the MNA set by removing variables that are known to be identical (node voltages at both op-amp inputs) and by removing unnecessary equations (KCL equation at the op-amp output) [Has89,Lin91]. The latter method is known as compacting of the MNAM due to op-amps; the resulting matrix is termed compacted MNA matrix (CMNAM). Both approaches are presented below.

2.4.1 Nullator, Norator, and Nullor

Before the characterization of the ideal op-amp is attempted, the concepts of the nullator, the norator, and the nullor are explored [Bru80]. They are not real elements. These ideal components are tools to introduce some mathematical constraints into a circuit. They are used as an aid to the development of insight into the behavior of ideal devices like the ideal op-amp.

The symbol for the nullator is shown in Figure 2.7. A nullator is defined as follows:

* If neither of the output terminals of the op-amp are grounded, such op-amp is said to be floating [Vla94].

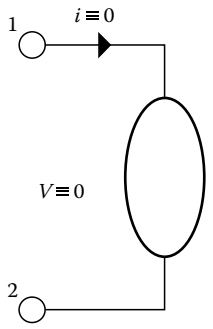


FIGURE 2.7 Nullator.

Definition 2.1: A nullator is a two-terminal element defined by the constraints

$$\begin{aligned} v_1 - v_2 &\equiv 0 \\ i &\equiv 0 \end{aligned} \tag{2.23}$$

The symbol for the norator is presented in Figure 2.8. A norator is defined as follows:

Definition 2.2: A norator is a two-terminal element for which the voltage and current are not constrained. That is,

$$\begin{aligned} v_1 - v_2 &= \text{arbirtary} \\ i &= \text{arbirtary} \end{aligned} \tag{2.24}$$

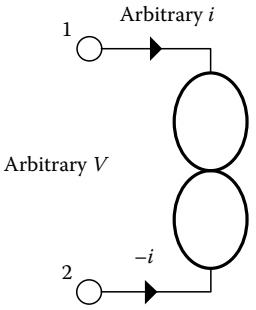


FIGURE 2.8 Norator.

A norator in a circuit introduces freedom from some constraints on the nodes it is connected to.

A circuit must contain an equal number of nullators and norators; in other words, a nullator and a norator must appear in a circuit as a pair. The combination of a nullator and a norator to produce a two-port, as shown in Figure 2.9, is referred to as a nullor. The equations characterizing the nullor are represented then by Equations 2.23 and 2.24.

In the MNA formulation, each nullator introduces additional equation: $v_i - v_j = 0$ into the set. The required extra variable is the corresponding norator's current. Nullor's contribution to the MNAM can be represented as its unique element stamp in similar way as it was done for VCVS, CCVS, etc. This is best illustrated in an example.

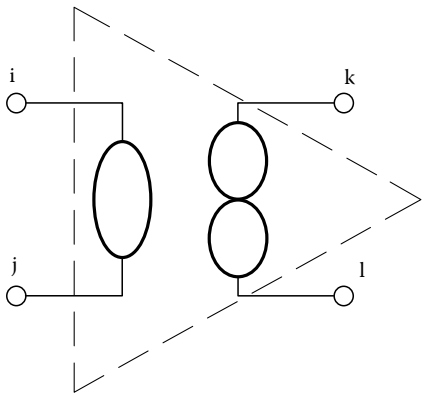


FIGURE 2.9 Nullor.

Example 2.5

Write the MNA equation for the circuit containing a nullor, as shown in Figure 2.10.

To formulate the MNAM, needed in the equation, we start with a 6×6 matrix and use element stamps to include the capacitor and the conductances in the usual manner. The nullator equation, $v_2 - v_5 = 0$, creates two entries in row 6. The norator current, our extra variable i_6, must be included in the KCL equations for the nodes 3 and 4, creating two entries in column 6. This results in the equation

$$\left[\begin{array}{ccccc|c} G_1 + sC_1 & -G_1 & 0 & 0 & -sC_1 & 0 \\ -G_1 & G_1 + G_2 & -G_2 & 0 & 0 & 0 \\ 0 & -G_2 & G_2 & 0 & 0 & 1 \\ 0 & 0 & 0 & G_3 & -G_3 & -1 \\ -sC_1 & 0 & 0 & -G_3 & G_3 + G_4 + sC_1 & 0 \\ \hline 0 & 1 & 0 & 0 & -1 & 0 \end{array}\right] \left[\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ i_6 \end{array}\right] = \left[\begin{array}{c} I_S \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \tag{2.25}$$

The sixth row and the sixth column of the MNAM in Equation 2.25 represent the element stamp of the nullor.

2.4.2 Ideal Op-Amp Model

The two characteristics of an ideal op-amp are

1. Zero voltage differential and no current flowing between its input nodes (the virtual short-circuit)
2. Arbitrary current that the output node supplies (to satisfy the KCL at this node)

Both properties are perfectly modeled using a nullor (Figure 2.9), with terminal l grounded. Adding a nullor stamp to our repertoire of element stamps creates a natural extension of the MNA method.

There is, however, another way of looking at ideal op-amp's (nullor's) contribution to the set of nodal equations, describing a circuit. Consider first the input of an ideal op-amp. Due to property 1, potentials at both input nodes must be identical. So, one of the two variables (node potentials) can be eliminated without any loss of information about the circuit. With one less variable, we must also remove one equation, for the system to have a unique solution. Since the ideal op-amp will always supply the exact current required to satisfy the KCL at the output node, the KCL equation at this node is unnecessary and can be simply deleted from the set. This process results in a MNAM that is more compact than the matrix obtained with the use of ideal op-amp stamp. Thus, the resulting matrix is called the CMNAM.

The rules for formulating CMNAM for a circuit containing ideal op-amps are as follows:

1. Remove the ideal op-amps from the circuit.
2. Write the MNAM $\mathbf{Y}_m$ for this circuit; label all columns and rows of $\mathbf{Y}_m$ with numbers corresponding to variables.

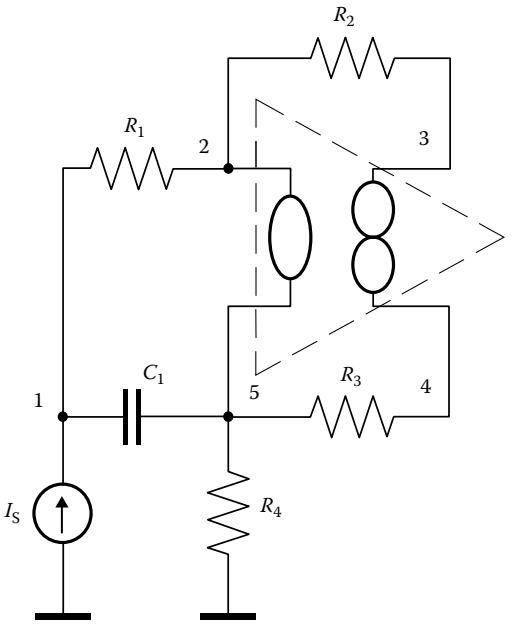


FIGURE 2.10 Circuit with nullor.

3. Repeat for every ideal op-amp in the circuit:
   a. For an ideal op-amp with the input terminals connected to nodes i $(-)$ and j $(+)$, add the column containing label j to the column containing label i and delete the former column; append all labels of the former column to the labels of the latter column. If one of the input nodes is the reference node, simply delete the column and the variable corresponding to the other node.
   b. For an ideal op-amp with the output terminal connected to node k, delete row with label k from the matrix.

Example 2.6

We wish to obtain the CMNAM of the gyrator circuit with two ideal op-amps, as shown in Figure 2.11. Following the compacting rules, we first formulate the NAM of the circuit with op-amps removed and label the rows and columns as shown below:

$$\mathbf{Y}_m = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} G_4 & 0 & 0 & 0 & -G_4 \\ 0 & G_1+G_2 & -G_2 & 0 & 0 \\ 0 & -G_2 & G_2+G_3 & -G_3 & 0 \\ 0 & 0 & -G_3 & G_3+sC_1 & -sC_1 \\ -G_4 & 0 & 0 & -sC_1 & G_4+sC_1 \end{bmatrix} \end{array}$$

Next, according to Rule 3a, for the first op-amp, we add the column containing label 1 to the column containing label 2, delete the former column, and append its labels to the labels of the latter column. Output of O_1 is connected to node 3. Rule 3b requires that we delete row with label 3. This results in the following matrix:

$$\mathbf{Y}_m = \begin{array}{c} \\ 1 \\ 2 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccc} 2,1 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} G_4 & 0 & 0 & -G_4 \\ G_1+G_2 & -G_2 & 0 & 0 \\ 0 & -G_3 & G_3+sC_1 & -sC_1 \\ -G_4 & 0 & -sC_1 & G_4+sC_1 \end{bmatrix} \end{array}$$

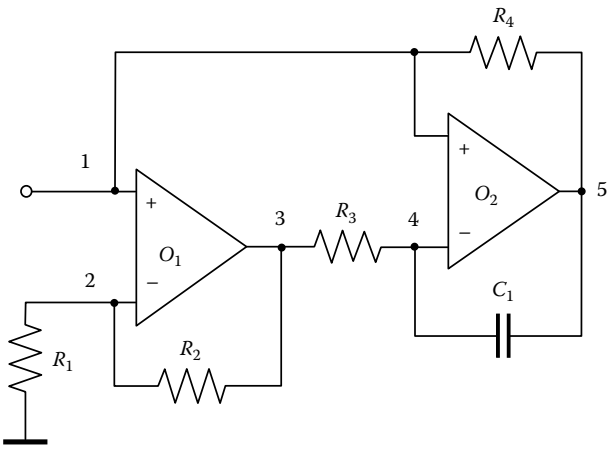


FIGURE 2.11 Gyrator circuit with two ideal op-amps for Example 2.6.

Applying Rule 3 to the op-amp O_2, we add column containing label 1 to the column containing label 4, delete the former column, and append its labels (2,1) to the labels of column 4. According to Rule 3b, we delete row with label 5. The resulting matrix is shown below:

$$\mathbf{Y}_{CM} = \begin{array}{c} \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 3 & 4,2,1 & 5 \end{array} \\ \begin{bmatrix} 0 & G_4 & -G_4 \\ -G_2 & G_1+G_2 & 0 \\ -G_3 & G_3+sC_1 & -sC_1 \end{bmatrix} \end{array}$$

It is customary to rearrange the columns of the CMNAM in ascending order of their smallest labels. After exchanging columns 1 and 2, the final matrix becomes

$$\mathbf{Y}_{CM} = \begin{array}{c} \\ 1 \\ 2 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccc} 4,2,1 & 3 & 5 \end{array} \\ \begin{bmatrix} G_4 & 0 & -G_4 \\ G_1+G_2 & -G_2 & 0 \\ G_3+sC_1 & -G_3 & -sC_1 \end{bmatrix} \end{array}$$

2.5 Applications of Symbolic Analysis

There are many specific applications for which symbolic analysis algorithms have been developed over the years. The following is an attempt to categorize the uses of symbolic analysis methods. For a given application, some overlap of these categories might exist. The goal here is to give a general idea of the applications of symbolic analysis methods. It must be noted that most of these applications cover both *s*-domain and *z*-domain analyses.

It is now widely accepted that in all applications where symbolic formula is generated in order to provide means for subsequent repetitive exact numerical calculation, use of sequential formulas is most effective. For the purpose of interpretation, that is, gaining deeper understanding of circuit behavior, only a single formula can be used. Approximate symbolic analysis emerges here as a very promising approach.

2.5.1 Frequency Response Evaluation

This is an obvious application of having the symbolic transfer function stated in terms of the circuit variable parameters. The process of finding the frequency response curve over a frequency range for a given circuit involves the repetitive evaluation of Equation 2.1 with all the parameters numerically specified and sweeping the frequency over the desired range. A numerical simulator would require a simulation run for each frequency point.

2.5.2 Circuit Response Optimization

This process involves the repetitive evaluation of the symbolic function generated by a symbolic simulator [Lin91,Gie94]. The response of the circuit is repetitively evaluated by substituting different values for the circuit parameters in the equation until a desired numerical response is achieved. The concept, of course, requires a good deal of management in order to reduce the search space and the number of evaluations needed. Such a method for filter design by optimization is discussed in [Tem77] and [Bas72], and a method for solving piecewise resistive linear circuits is discussed in [Chu75]. The idea here is that for a given circuit topology, only one run through the symbolic circuit simulator is necessary.

2.5.3 Sensitivity Analysis

Sensitivity analysis is the process of finding the effect of the circuit performance due to changes in an element value [Lin92,Bal98,Bal01]. There are two types of sensitivities of interest to circuit designers: small-change (or differential) sensitivity and large-change sensitivity. The normalized differential sensitivity of a transfer function H with respect to a parameter x is given as

$$S_x^H = \frac{\partial(\ln H)}{\partial(\ln x)} = \frac{\partial H}{\partial x}\frac{x}{H}$$

The above expression can be found symbolically and then evaluated for the different circuit parameter values. For the transfer function H in a single expression form, the differentiation process is quite straightforward. When H is given in a sequence of expressions form, the most compact formula is obtained by the application of the two-port transimpedance concept and an efficient method of symbolic calculations of the elements of the inverse of the reduced modified node admittance matrix [Bal98]. Similar approach is used to obtain the large-change sensitivity [Bal01], defined as

$$\delta_x^H = \frac{\Delta H}{H}\frac{x}{\Delta x}$$

Once the sequence of expressions for the network function has been generated, only a few additional expressions are needed to determine the sensitivity (both differential and large-change). Additional effort to calculate sensitivities is practically independent of circuit size, making this approach attractive for large-scale circuits.

2.5.4 Circuit Sizing

Circuit sizing refers to the process of computing values for the elements of a given circuit topology such that some set of behavioral circuit specifications is satisfied [Hen00]. This process can be supported by symbolic techniques in two ways. Computer algebra can help to solve a system of design equations symbolically for the circuit parameters in order to obtain a set of generic analytic sizing formulas, or to set up and preprocess a system of design equations for subsequent numerical solution.

2.5.5 Parameter Extraction in Device Modeling

This process involves the repeated comparison of measured data from fabricated devices with the simulation results using the mathematical models for these devices [Kon93,Avi98]. The goal of the process is to update the device models to reflect the measured data. The model parameters are incrementally adjusted and the evaluation process is repeated until the difference between the measured and the simulated results are minimized. Such approaches are reported in [Kon89] and [Kon93] (active devices) and [Avi98] (passive devices).

2.5.6 Statistical Analysis

A widely used statistical analysis method is through Monte Carlo simulations [Sty95]. The circuit behavior has to be repetitively evaluated many times in order to evaluate the statistical variation of a circuit output in response to parameter mismatches due to, for instance, integrated circuits process variations.

2.5.7 Fault Diagnosis of Analog Circuits

The process reported in [Man93] takes measurements from the faulty fabricated circuit and compares it to simulation results. The process is continuously repeated with the parameter values in the simulations

changed until the faulty element is detected. Symbolic techniques have also been applied to multiple fault detection [Fed98].

2.5.8 Insight into Circuit Operation

The insight that can be provided by obtaining the symbolic transfer function versus its numerical counterpart is very evident. The simple example in Section 2.5 illustrates this powerful application. The effect of the different elements on the behavior of the circuit can be observed by inspecting the symbolic expression. This, of course, is possible if the number of symbolic parameters is small, that is, the circuit is small. Insight, however, can also be obtained by observing an approximate symbolic expression that reduces the number of symbolic terms to a manageable figure. Several approximation techniques are reported in [Wam92], [Fer92], [Sed92], [Hsu94], [Yu96], and [Hen00].

2.5.9 Education

Symbolic analysis is most helpful for students as a supplement to linear circuit analysis courses. These courses require the derivation of expressions for circuit impedances, gains, and transfer functions. A symbolic simulator can serve as a check for the correctness of the results in addition to aiding instructors in verifying solutions and making up exercises and examples [Hue89].

Symbolic methods in conjunction with mathematical software are used in teaching analog electronics in addition to numerical simulators. This adds another dimension to standard analysis and design process, gives students greater flexibility, and encourages creativeness [Bal00].

2.6 Symbolic Analysis Software Packages

Several stand-alone symbolic simulators available for public use exist nowadays. A comparison of these software packages based on their functionality is given in Table 2.1. The list is by no means exhaustive; it should be treated just as a representative sample. The software packages that were compared are

TABLE 2.1 Comparison between Some Symbolic Simulation Programs

	ISSAC	ASAP	SYNAP	SAPWIN	SSPICE	SCYMBAL	SCAPP	STAINS
Analysis domains	*s* and *z*	*s*	dc and *s*	*s*	*s*	*z*	*s*	s
Primitive elements	All	All	All	All	All	—	All	All
Small-signal linearization	Yes	Yes	Yes	No	Yes	No	Yes	No
Mismatching	Yes	Yes	Yes	No	No	No	No	No
Approximation	Yes	Yes	Yes	Yes	Yes	No	No	No
Weakly nonlinear analysis	Yes	No	No	No	No	No	No	No
Hierarchical analysis	No	No	No	No	Limited	No	Yes	Yes
Pole/zero extraction	No	Limited	No	Yes	Limited	No	No	No
Graphical interface	No	Yes	No	Yes	No	No	Yes	No
Equation formulation	CMNA	SFG	MNA	MNA	**Y**	SFG	RMNA SFG	RMNA
Language	LISP	C	C++	C++	C	FORTRAN	C	VBA

Source: Adapted from Gielen, G. and Sansen, W., *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Kluwer Academic, Boston, MA, 1991.

Note: SFG, signal flowgraph; **Y**, admittance matrix; VBA, Visual Basic for Applications™.

ISSAC [Gie89], SCAPP [Has89], ASAP [Fer91], SYNAP [Sed88], SCYMBAL [Kon88], STAINS [Pie01], SAPWIN [Lib95], and SSPICE [Wie89].

Add-on packages (toolboxes) that run in symbolic mathematics software environments, like Mathematica, belong to another class of symbolic analysis programs. Analog Insydes [Hen00] is a good example of such an add-on. The toolbox includes approximation algorithms, netlist management, linear and nonlinear device and behavioral modeling, setting up and solving circuit equations, numerical circuit analysis in the frequency and time domains, two- and three-dimensional graphing of analysis results, and data exchange with other circuit simulators. Some routines in the toolbox were written in C to overcome Mathematica's limitations in efficiently handling large sparse systems.

References

[Ald73] G.E. Alderson and P.M. Lin, Computer generation of symbolic network functions: A new theory and implementation, *IEEE Transactions on Circuit Theory*, 20(1): 48–56, January 1973.

[Als93] B. Alspaugh and M.M. Hassoun, A mixed symbolic and numeric method for closed-form transient analysis, *Proceedings of the 5th International Workshop on European Conference on Circuit Theory and Design*, Davos, Switzerland, pp. 1687–1692, September 1993.

[Avi98] G. Avitabile et al., Parameter extraction in electronic device modelling using symbolic techniques, *Proceedings of the 5th International Workshop on Symbolic Methods and Applications in Circuit Design*, Kaiserslautern, Germany, pp. 253–259, October 1998.

[Bal98] F. Balik and B. Rodanski, Calculation of first-order symbolic sensitivities in sequential form via the transimpedance method, *Proceedings of the 5th International Workshop on Symbolic Methods and Applications in Circuit Design*, Kaiserslautern, Germany, pp. 169–172, October 1998.

[Bal00] F. Balik and B. Rodanski, Symbolic analysis in the classroom, *Proceedings of the 6th International Workshop on Symbolic Methods and Applications in Circuit Design*, Lisbon, Portugal, pp. 17–20, October 2000.

[Bal01] F. Balik and B. Rodanski, Obtaining large-change symbolic sensitivities for large-scale circuits, *Proceedings of the European Conference on Circuit Theory and Design*, Espoo, Finland, pp. 205–208, August 2001.

[Bas72] S. Bass, The application of a fast symbolic analysis routine in a network optimization program, *Proceedings of the Midwest Symposium on Circuit Theory*, Rolla, MO, May 1972.

[Bra71] F.H. Branin et al., ECAP II: A new electronic circuit analysis program, *IEEE Journal of Solid-State Circuits*, 6(4): 146–165, August 1971.

[Bru80] L.T. Bruton, *RC Active Circuits: Theory and Design*. Englewood Cliffs, NJ: Prentice-Hall, 1980.

[Cab98] R. Cabeza and A. Carlosena, The use of symbolic analysers in circuit synthesis, *Proceedings of the 5th International Workshop on Symbolic Methods and Applications in Circuit Design*, Kaiserslautern, Germany, pp. 146–149, October 1998.

[Chu75] L.O. Chua and P.M. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[Fed98] G. Fedi et al., On the application of symbolic techniques to the multiple fault location in low testability analog circuits, *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, 45(10): 1383–1388, October 1998.

[Fer91] F.V. Fernandez, A. Rodriguez-Vazquez, and J.L. Huertas, An advanced symbolic analyzer for the automatic generation of analog circuit design equations, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Singapore, pp. 810–813, June 1991.

[Fer92] F.V. Fernandez et al., On simplification techniques for symbolic analysis of analog integrated circuits, *Proceedings of the IEEE International Symposium on Circuits and Systems*, San Diego, CA, pp. 1149–1152, May 1992.

[Gha65] M.S. Ghausi, *Principles and Design of Linear Active Circuits*. New York: McGraw-Hill, 1965.

[Gie89] G. Gielen, H. Walscharts, and W. Sansen, ISSAC: A symbolic simulator for analog integrated circuits, *IEEE Journal of Solid-State Circuits*, 24(6): 1587–1597, December 1989.

[Gie91] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*, Boston, MA: Kluwer Academic, 1991.

[Gie94] G. Gielen, P. Wambacq, and W. Sansen, Symbolic analysis methods and applications for analog circuits: A tutorial overview, *Proceeding of the IEEE*, 82(2): 287–301, February 1994.

[Gus70] F.G. Gustavson, W. Liniger, and R. Willoughby, Symbolic generation of an optimal Crout algorithm for sparse systems of linear equations, *Journal of ACM*, 17(1): 87–109, 1970.

[Hac71] G.D. Hachtel et al., The sparse tableau approach to network analysis and design, *IEEE Transactions on Circuit Theory*, 18(1): 101–113, January 1971.

[Has89] M.M. Hassoun and P.M. Lin, A new network approach to symbolic simulation of large-scale networks, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Portland, OR, pp. 806–809, May 1989.

[Has91] M.M. Hassoun and J.E. Ackerman, Symbolic simulation of large-scale circuits in both frequency and time domains, *Proceedings of the IEEE Midwest Symposium on Circuits and Systems*, Calgary, Canada, pp. 707–710, August 1990.

[Has93a] M.M. Hassoun and K. McCarville, Symbolic analysis of large-scale networks using a hierarchical signal flowgraph approach, *International Journal on Analog Integrated Circuits and Signal Processing*, Kluwer Academic, 3(1): 31–42, January 1993.

[Has93b] M.M. Hassoun and P. Atawale, Hierarchical symbolic circuit analysis of large-scale networks on multi-processor systems, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Chicago, IL, pp. 1651–1654, May 1993.

[Hen00] E. Henning, Symbolic approximation and modeling techniques for analysis and design of analog circuits, PhD dissertation, University of Kaiserslautern. Aachen, Germany: Shaker Verlag, 2000.

[Ho75] C. Ho, A.E. Ruehli, and P.A. Brennan, The modified nodal approach to network analysis, *IEEE Transactions on Circuits and Systems*, 25(6): 504–509, June 1975.

[Hsu94] J.J. Hsu and C. Sechen, DC small-signal symbolic analysis of large analog integrated circuits, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 41(12): 817–828, December 1994.

[Hue89] L.P. Huelsman, Personal computer symbolic analysis programs for undergraduate engineering courses, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Portland, OR, pp. 798–801, May 1989.

[Idl71] T.E. Idleman et al., SLIC: A simulator for linear integrated circuits, *IEEE Journal of Solid-State Circuits*, 6(4): 188–204, August 1971.

[Kon88] A. Konczykowska and M. Bon, Automated design software for switched-capacitor ICs with symbolic simulator SCYMBAL, *Proceedings of the Design Automation Conference*, Anaheim, CA, pp. 363–368, June 1988.

[Kon89] A. Konczykowska and M. Bon, Symbolic simulation for efficient repetitive analysis and artificial intelligence techniques in CAD, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Portland, OR, pp. 802–805, May 1989.

[Kon93] A. Konczykowska et al., Parameter extraction of semiconductor devices electrical models using symbolic approach, *Alta Frequenza Rivista Di Elettronica*, 5(6): 3–5, November 1993.

[Ley00] F. Leyn, G. Gielen, and W. Sansen, Towards full insight in small-signal behavior of analog circuits: Assessment of different symbolic analysis approaches, *Proceedings of the 7th International Workshop on Symbolic Methods and Applications in Circuit Design*, Lisbon, Portugal, pp. 89–93, October 2000.

[Lib93] A. Liberatore et al., Simulation of switching power converters using symbolic techniques, *Alta Frequenza Rivista Di Elettronica*, 5(6): 16–23, November 1993.

[Lib95] A. Liberatore et al., A new symbolic program package for the interactive design of analog circuits, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seattle, WA, pp. 2209–2212, May 1995.

[Lin91] P.M. Lin, *Symbolic Network Analysis*. Amsterdam, the Netherlands: Elsevier Science, 1991.

[Lin92] P.M. Lin, Sensitivity analysis of large linear networks using symbolic programs, *Proceedings of the IEEE International Symposium on Circuits and Systems*, San Diego, CA, pp. 1145–1148, May 1992.

[Man93] S. Manetti and M. Piccirilli, Symbolic simulators for the fault diagnosis of nonlinear analog circuits, *International Journal on Analog Integrated Circuits and Signal Processing*, Kluwer Academic, 3(1): 59–72, January 1993.

[Mar57] H.M. Markowitz, The elimination form of the inverse and its application to linear programming, *Management Science*, 3(3): 255–269, April 1957.

[Mat93] T. Matsumoto, T. Sakabe, and K. Tsuji, On parallel symbolic analysis of large networks and systems, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Chicago, IL, pp. 1647–1650, May 1993.

[Nag75] L.W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Memo ERL-M520, Electronics Research Laboratory, University of California, Berkeley, CA, May 1975.

[Nob77] B. Noble and J. Daniel, *Applied Linear Algebra*, 2nd edn. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[Pie01] M. Pierzchala and B. Rodanski, Generation of sequential symbolic network functions for large-scale networks by circuit reduction to a two-port, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 48(7): 906–909, July 2001.

[Rod08] B. Rodanski and M. Hassoun, Symbolic analysis, *Fundamentals of Circuits and Filters*, W.-K. Chen, Ed. CRC Press: Boca Raton, 2008.

[Sed88] S. Seda, M. Degrauwe, and W. Fichtner, A symbolic analysis tool for analog circuit design automation, *1988 International Conference on Computer-Aided Design*, Santa Clara, CA, pp. 488–491, 1988.

[Sed92] S. Seda, M. Degrauwe, and W. Fichtner, Lazy-expansion symbolic expression approximation in SYNAP, *1992 International Conference on Computer-Aided Design*, Santa Clara, CA, pp. 310–317, 1992.

[Som91] R. Sommer, EASY: An experimental analog design system framework, *Proceedings of the International Workshop on Symbolic Methods and Applications in Circuit Design*, Paris, France, October 1991.

[Som93] R. Sommer, D. Ammermann, and E. Hennig, More efficient algorithms for symbolic network analysis: Supernodes and reduced loop analysis, *International Journal on Analog Integrated Circuits and Signal Processing*, Kluwer Academic, 3(1), 73–83, January 1993.

[Sta86] J.A. Starzyk and A. Konczykowska, Flowgraph analysis of large electronic networks, *IEEE Transactions on Circuits and Systems*, 33(3): 302–315, March 1986.

[Sty95] M.A. Styblinski and M. Qu, Comparison of symbolic analysis, approximation and macro-modeling techniques for statistical design for quality of analog integrated circuits, *Proceedings of the IEEE International Symposium on Circuits and Systems*, Seattle, WA, pp. 2221–2224, May 1995.

[Tem77] G. Temes, *Introduction to Circuit Synthesis and Design*, New York: McGraw Hill, 1977.

[Vla94] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, 2nd Ed. New York: Van Nostrand Reinhold, 1994.

[Wan77] P. Wang, On the expansion of sparse symbolic determinants, *Proceedings of the International Conference on System Sciences*, Honolulu, HI, 1977.

[Wee73] W.T. Weeks et al., Algorithms for ASTAP: A network analysis program, *IEEE Transactions on Circuit Theory*, 20(11): 628–634, November 1973.

[Weh93] E. Wehrhahn, Symbolic analysis on parallel computers, *Proceedings of the European Conference on Circuit Theory and Design*, Davos, Switzerland, pp. 1693–1698, September 1993.

[Wie89] G. Wiezba et al., SSPICE: A symbolic SPICE program for linear active circuits, *Proceedings of the Midwest Symposium on Circuits and Systems*, Urbana, IL, pp. 1197–1201, August 1989.

[Wam92] P. Wambacq, G. Gielen, and W. Sansen, A cancellation free algorithm for the symbolic simulation of large analog circuits, *Proceedings of the IEEE International Symposium on Circuits and Systems*, San Diego, CA, pp. 1157–1160, May 1992.

[Yu96] Q. Yu and C. Sechen, A unified approach to the approximate symbolic analysis of large analog integrated circuits, *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 43(8): 656–669, August 1996.

3

Numerical Analysis Methods

3.1 Equation Formulation **3**-1
Implications of KCL, KVL, and the Element Branch Characteristics • Sparse Tableau Formulation • Nodal Analysis • Modified Nodal Analysis • Nodal Formulation by Stamps • Modified Nodal Formulation by Stamps

3.2 Solution of Linear Algebraic Equations **3**-11
Factorization • Forward Substitution • Backward Substitution • Pivoting • Computation Cost of *LU* Factorization

References **3**-16

Andrew T. Yang
Apache Design Solutions, Inc.

3.1 Equation Formulation

The method by which circuit equations are formulated is essential to a computer-aided circuit analysis program. It affects significantly the setup time, the programming effort, the storage requirement, and the performance of the program.

A linear time-invariant circuit with n nodes and b branches is completely specified by its network topology and branch constraints. The fundamental equations that describe the equilibrium conditions of a circuit are the Kirchhoff's current law (KCL) equations, the Kirchhoff's voltage law (KVL) equations, and the equations which characterize the individual circuit elements. Two methods are popular: the sparse tableau approach and the modified nodal approach.

3.1.1 Implications of KCL, KVL, and the Element Branch Characteristics

Given an example with $n = 4$ and $b = 6$ (as shown in Figure 3.1) we can sum the branch currents leaving each node to zero and obtain

$$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} i_{b_1} \\ i_{b_2} \\ i_{b_3} \\ i_{b_4} \\ i_{b_5} \\ i_{b_6} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

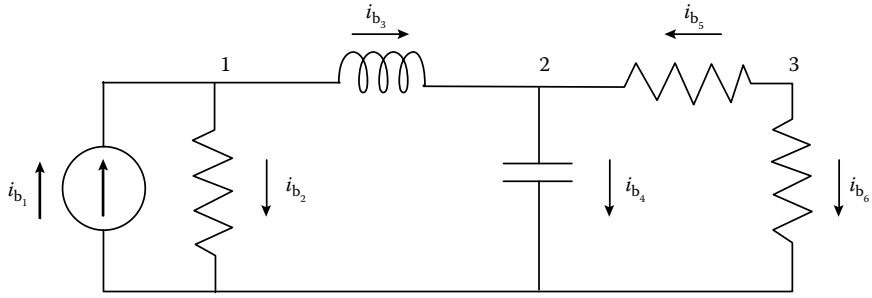


FIGURE 3.1 Network used to illustrate KCL, KVL, and the element branch characteristics.

or

$$A_a \times i_b = 0$$

where

A_a is an $n \times b$ incidence matrix contains $+1$, -1, and 0 entries
i_b is the branch current

Note that A_a is linearly dependent since the entries in each column add up to zero. A unique set of equations can be obtained by defining a datum node and eliminating its corresponding row of A_a. Hence, KCL results in $n-1$ equations and b unknowns. It implies

$$A \times i_b = 0 \tag{3.1}$$

where A is called the *reduced* incidence matrix.

KVL results in b equations with $b+n-1$ unknowns. It implies

$$v_b = A^T \times V_n \tag{3.2}$$

where

A^T is the transpose of the reduced incidence matrix
v_b is the branch voltage
V_n is the node-to-datum (nodal) voltage

Define the convention as follows (Figure 3.2):

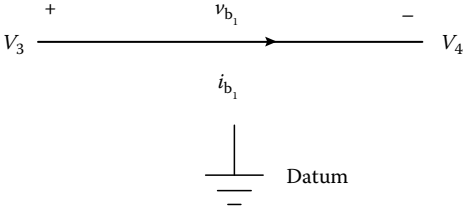


FIGURE 3.2 Reference convention for voltages and current.

One can sum the voltages around the loop using KVL (Figure 3.3):

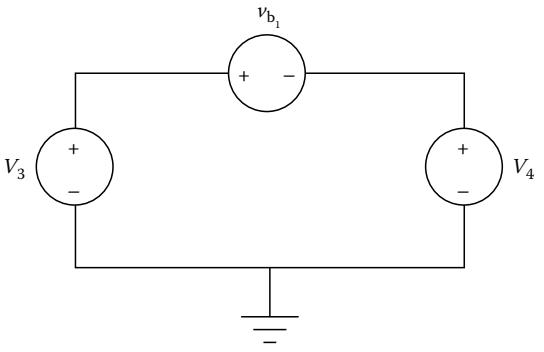


FIGURE 3.3 Voltage around the loop using KVL.

or

$$v_{b_1} = V_3 - V_4$$

Apply KVL to the example above and we obtain

$$\begin{bmatrix} v_{b_1} \\ v_{b_2} \\ v_{b_3} \\ v_{b_4} \\ v_{b_5} \\ v_{b_6} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

or

$$v_b = A^T \times V_n$$

where v_b and V_n are unknowns.

The element characteristic results in generalized constraints equations in the form of

$$Y_b \cdot v_b + Z_b \cdot i_b = S_b \tag{3.3}$$

For example,
Resistor:

$$\frac{1}{R} v_b - i_b = 0$$

Voltage source:

$$v_b = E$$

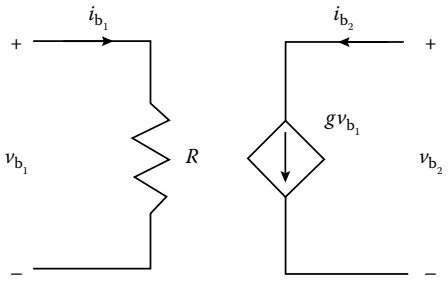


FIGURE 3.4 Voltage-controlled current source (VCCS).

VCCS (Figure 3.4):

$$\begin{bmatrix} 1/R & 0 \\ g & 0 \end{bmatrix} \begin{bmatrix} v_{b_1} \\ v_{b_2} \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i_{b_1} \\ i_{b_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3.1.2 Sparse Tableau Formulation

The sparse tableau method simply combines Equations 3.1 through 3.3. The equation and unknown count can be summarized as follows:

	Number of Equations	Number of Unknowns
(3.1)	$n-1$	b
(3.2)	b	$b+n-1$
(3.3)	b	0
Total	$2b+\mathrm{n}-1$	$2b+n-1$

The sparse tableau in the matrix form is shown next:

$$\begin{bmatrix} 0 & A & 0 \\ A^{\mathrm{T}} & 0 & -1 \\ 0 & Z_{\mathrm{b}} & Y_{\mathrm{b}} \end{bmatrix} \begin{bmatrix} V_{\mathrm{n}} \\ i_{\mathrm{b}} \\ v_{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ S_{\mathrm{b}} \end{bmatrix} \tag{3.4}$$

Equation 3.4 is a system of linear equations.

Note that the matrix formulated by the sparse tableau approach is typically sparse with many zero value entries. The main advantage is its generality (i.e., all circuit unknowns including branch current/voltage and node-to-datum voltage can be obtained in one pass). Since the number of equations is usually very large, an efficient sparse linear system solver is essential.

3.1.3 Nodal Analysis

The number of equations in Equation 3.4 can be reduced significantly by manipulating Equations 3.1 through 3.3. Motivated by the fact that the number of branches in a circuit is generally larger than the number of nodes, nodal analysis attempts to reduce the number of unknowns to V_{n}. As we will see, this is achieved by a loss of generality (i.e., not all types of linear elements can be processed).

We can eliminate the branch voltages v_b by substituting Equation 3.2 into Equation 3.3. This yields

$$Y_b A^T V_n + Z_b i_b = S_b \tag{3.5}$$

$$A \times i_b = 0 \tag{3.6}$$

Combining Equations 3.5 and 3.6 to eliminate the branch currents, i_b, we obtain a set of equations with V_n as unknowns:

$$A \cdot Z_b^{-1} \cdot \left(-Y_b A^T V_n + S_b\right) = 0 \tag{3.7}$$

Since the Z_b matrix may be singular, not all elements can be processed. For example, a voltage-controlled voltage source, as shown in Figure 3.5, can be cast in the form of Equation 3.3:

$$\begin{bmatrix} 0 & 0 \\ u & -1 \end{bmatrix} \begin{bmatrix} v_{b_1} \\ v_{b_2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_{b_1} \\ i_{b_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Note that Z_b is singular and it cannot be inverted. Consider a special case where Equation 3.3 can be expressed as

$$Y_b v_b - i_b = 0 \tag{3.8}$$

or $-Z_b$ is a unit matrix and $S_b = 0$. This condition is true if the circuit consists of

- Resistor
- Capacitor
- Inductor
- Voltage-controlled current source

For example, Figure 3.6 is a VCCS.

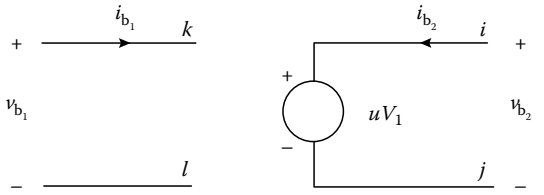


FIGURE 3.5 Voltage-controlled voltage source (VCVS).

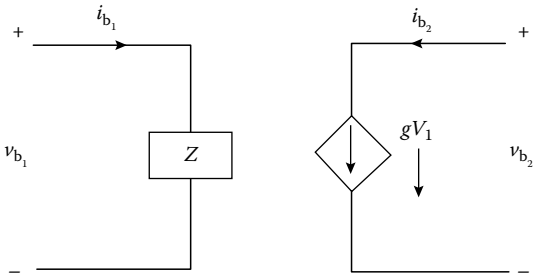


FIGURE 3.6 VCCS.

The branch constraints can be cast as

$$\begin{bmatrix} 1/z & 0 \\ g & 0 \end{bmatrix} \begin{bmatrix} v_{b_1} \\ v_{b_2} \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} i_{b_1} \\ i_{b_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and $z = 1/j\omega C$ for the capacitor, $z = j\omega L$ for an inductor, and $z = R$ for a resistor. For this type of circuit, the nodal analysis results in $n-1$ equations with V_n as unknowns, or

$$A Y_b A^T \times V_n = 0$$

A circuit of this type contains no excitation (voltage/current source). The current sources can be included in the formulation by letting the corresponding edges of the current sources be numbered last. Hence, we can partition the reduced incidence matrix A into

$$A = [A_b | A_J] \tag{3.9}$$

where A_J corresponds to the subincidence matrix of the current source branches. Then, Equations 3.1, 3.2, and 3.8 can be expressed as

$$A_b i_b + A_J J = 0 \tag{3.10}$$

$$v_b = A_b^T V_n \tag{3.11}$$

$$i_b = Y_b v_b \tag{3.12}$$

Rearranging these equations yields

$$A_b Y_b A_b^T \times V_n = -A_J J \tag{3.13}$$

where J is a vector containing the current source values. Voltage sources can also be included in the formulation by simple source transformation which requires moderate preprocessing. The nodal analysis approach, therefore, can be applied to formulate equations for circuits consisting of

- Resistor
- Capacitor
- Inductor
- Voltage source (with preprocessing)
- Current source
- Voltage-controlled current source

Nodal analysis, however, when applied to an inductor, can cause numerical problems when the frequency is low:

$$\left(\frac{1}{j\omega L}\right)(v_{b_1}) + (-1)(i_{b_1}) = 0$$

or

$$\omega \to 0, \quad \frac{1}{j\omega L} \to \infty$$

To sum up, the nodal analysis must be extended to process the following linear elements (without proprocessing):

- Inductor
- Voltage source
- Current-controlled current source
- Current-controlled voltage source
- Voltage-controlled voltage source

3.1.4 Modified Nodal Analysis

A set of self-consistent modifications to the nodal analysis are proposed and the resultant formulation is called the modified nodal analysis (MNA). The MNA resolves the limitations of the nodal analysis method while preserving its advantages. In this section, we present the basic theory of MNA.

Divide all types of elements into three groups:
Group 1: Elements that satisfy

$$i_b = Y_b v_b$$

such as resistor, capacitor, and VCCS.
Group 2: Elements that satisfy

$$Y_b v_b + Z_b i_b = S_b$$

such as voltage source, inductor, VCVS, CCVS, and CCCS.
Group 3: Current source only.

Apply the partitioning technique to Group 1 and Group 2 elements. We can write

$$[A_1|A_2]\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = 0 \Rightarrow A_1 i_1 + A_2 i_2 = 0$$

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix} V_n \Rightarrow v_1 = A_1^T V_n, \quad v_2 = A_2^T V_n$$

$$i_1 = Y_1 v_1$$

$$Y_2 \cdot v_2 + Z_2 \cdot i_2 = S_2$$

Eliminating v_1, i_1, and v_2 from the preceding equations, we derive a system of linear equations with V_n and i_2 as unknowns:

$$A_1 Y_1 A_1^T V_n + A_2 i_2 = 0 \tag{3.14}$$

$$Y_2 A_2^T V_n + Z_2 i_2 = S_2 \tag{3.15}$$

Casting them in matrix form gives

$$\begin{bmatrix} A_1 Y_1 A_1^T & A_2 \\ Y_2 A_2^T & Z_2 \end{bmatrix}\begin{bmatrix} V_n \\ i_2 \end{bmatrix} = \begin{bmatrix} 0 \\ S_2 \end{bmatrix} \tag{3.16}$$

Finally, we apply the partitioning technique to include the current source (Group 3):

$$\begin{bmatrix} A_1 Y_1 A_1^{\mathrm{T}} & A_2 \\ Y_2 A_2^{\mathrm{T}} & Z_2 \end{bmatrix} \begin{bmatrix} V_{\mathrm{n}} \\ i_2 \end{bmatrix} = \begin{bmatrix} -A_J J \\ S_2 \end{bmatrix} \tag{3.17}$$

or

$$Y_{\mathrm{n}} \times x = J_{\mathrm{n}} \tag{3.18}$$

where

Y_{n} is the node admittance matrix
J_{n} is the source vector
x is the unknown vector

Implementing Equation 3.18 by matrix multiplication is difficult. Stamping methods have been developed to stamp in the entries of Y_{n}, J_{n} *element by element*. It is a very efficient way to implement Equation 3.18 into a network analysis program.

3.1.5 Nodal Formulation by Stamps

In this section, we developed the stamping rules for Group 1 and Group 3 elements only. Given a circuit consisting of Group 1 and Group 3 elements only:

We write the nodal equations at nodes 1, 2, and 3 (Figure 3.7):

$$-I + j\omega C(V_1 - V_3) + 1/R_1(V_1 - V_3) = 0 \qquad \text{(node 1)}$$

$$g(V_1 - V_3) + 1/R_2(V_2 - V_3) = 0 \qquad \text{(node 2)}$$

$$I + j\omega C(V_3 - V_1) + \frac{1}{R(V_3 - V_1)} - g(V_1 - V_3) + \frac{1}{R_2(V_3 - V_2)} = 0 \qquad \text{(node 3)}$$

Cast them in matrix form:

$$\begin{bmatrix} j\omega C + 1/R_1 & 0 & -j\omega C - 1/R_1 \\ g & 1/R_2 & -1/R_2 - g \\ -j\omega C - 1/R_1 - g & -1/R_2 & j\omega C + 1/R_1 + 1/R_2 + g \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix}$$

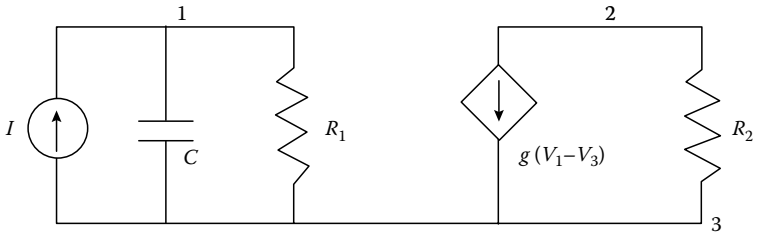


FIGURE 3.7 Network used to illustrate nodal formulation by stamps.

Note that for the left-hand side (LHS), we can write

$$\begin{bmatrix} 1/R_1 & 0 & -1/R_1 \\ 0 & 0 & 0 \\ -1/R_1 & 0 & 1/R_1 \end{bmatrix} + \begin{bmatrix} j\omega C & 0 & -j\omega C \\ 0 & 0 & 0 \\ -j\omega C & 0 & j\omega C \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ g & 0 & -g \\ -g & 0 & g \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/R_2 & -1/R_2 \\ 0 & -1/R_2 & 1/R_2 \end{bmatrix}$$

Therefore, the sampling rule for a resistor is

$$Y_n(i,i) = Y_n(i,i) + \frac{1}{R}$$

$$Y_n(j,j) = Y_n(j,j) + \frac{1}{R}$$

$$Y_n(i,j) = Y_n(i,j) - \frac{1}{R}$$

$$Y_n(j,i) = Y_n(j,i) - \frac{1}{R}$$

If node i is grounded, the corresponding row and column can be eliminated from the node admittance matrix. We then obtained only

$$Y_n(j,j) = Y_n(j,j) + \frac{1}{R}$$

Similar stamping rules can be derived for a capacitor and a VCCS. The stamping rule for a current source J, flowing from i to j is

$$J_n(i) = J_n(i) - J$$

$$J_n(j) = J_n(j) + J$$

3.1.6 Modified Nodal Formulation by Stamps

Given a circuit including Group 2 elements only:

Let us first define auxiliary branch current unknowns i_v, i_L, and i_β for each type of element in Group 2. From nodal analysis, we obtain the following nodal equations (Figure 3.8):

$$i_v + i_L = 0 \qquad \text{(node 1)}$$

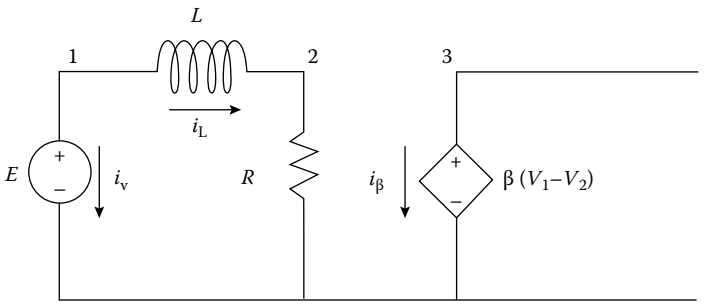


FIGURE 3.8 Network used to illustrate modified nodal formulation by stamps.

$$-i_{\mathrm{L}} + \frac{1}{R(V_2 - V_4)} = 0 \qquad \text{(node 2)}$$

$$i_{\beta} = 0 \qquad \text{(node 3)}$$

$$-i_{\mathrm{v}} + \frac{1}{R(V_4 - V_2)} - i_{\beta} = 0 \qquad \text{(node 4)}$$

For each auxiliary unknown, one auxiliary equation must be provided.

$$\begin{aligned} V_1 - V_4 &= E \\ V_1 - V_2 &= j\omega L i_{\mathrm{L}} \\ V_3 - V_4 &= \beta(V_1 - V_2) \end{aligned}$$

Cast these equations in matrix form:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1/R & 0 & -1/R & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1/R & 0 & 1/R & -1 & 0 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -j\omega L & 0 \\ -\beta & \beta & 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ i_{\mathrm{v}} \\ i_{\mathrm{L}} \\ i_{\beta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ E \\ 0 \\ 0 \end{bmatrix}$$

Hence, the following stamping rules are derived:
Voltage source:
LHS:

$$\begin{array}{c} \\ i \\ j \\ i_{\mathrm{v}} \end{array} \begin{array}{c} \begin{array}{ccc} i & j & i_{\mathrm{v}} \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} \end{array}$$

Right-hand side (RHS):

$$\begin{array}{c} \\ i \\ j \\ i_{\mathrm{v}} \end{array} \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix}$$

Inductor:
LHS:

$$\begin{array}{c} \\ i \\ j \\ i_{\mathrm{L}} \end{array} \begin{array}{c} \begin{array}{ccc} i & j & i_{\mathrm{L}} \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & -1 & -j\omega L \end{bmatrix} \end{array}$$

Note that the numerical problem associated with an inductor is avoided when $\omega \to 0$.

VCVS (Figure 3.5):
LHS:

$$\begin{array}{c} \\ i \\ j \\ i_v \end{array} \begin{array}{c} \begin{array}{ccccc} i & j & k & l & i_v \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -\mu & \mu & 0 \end{bmatrix} \end{array}$$

The stamping rule for a CCVS and a CCCS can be developed following the same arguments. For DC analysis, set $\omega = 0$. In SPICE, the MNA is employed to formulate network equations. To probe a branch current, a user needs to insert a zero value voltage source between the adjacent nodes. The solution i_v is then the branch current. The implementation of MNA can be summarized as follows:

1. Implement an input parser to read in a circuit description file. A ''free'' input format does not restrict a user to follow column entry rules.
2. Build an internal node table which has a one-to-one relationship with user-defined nodes. Hence, a user needs not number the network nodes consecutively.
3. Number the auxiliary nodes for Group 2 elements last. For Group 2 element, one extra node is needed.
4. Solve the system of linear equations to find output voltages and currents.

3.2 Solution of Linear Algebraic Equations

The methods of solving a set of linear equations are basic to all computer-aided network analysis problems. If the network is linear, the equations are linear. Nonlinear networks lead to a system of nonlinear equations which can be linearized about some operating point. Transient analysis involves solving these linearized equations at many iteration points. Frequency domain analysis (small-signal AC analysis) requires the repeated solution of linear equations at specified frequencies.

The discussion in this section will be an introduction to the direct solution method based on *LU* decomposition, a variant of Gaussian elimination. This method is frequently used because of its efficiency, robustness, and ease of implementation. More advanced topics such as the general sparse matrix techniques are not discussed.

Consider a set of n linear algebraic equations of the form:

$$Ax = b \tag{3.19}$$

where
A is an $n \times n$ nonsingular real matrix
x and b are n-vectors

For the system to have a unique solution, A must be nonsingular (i.e., the determinant of A must not be zero). Equation 3.19 can be solved efficiently by first factorizing A into a product of two matrices L and U, which are respectively lower and upper triangular. This so-called *LU* decomposition method for solving a system of linear equations is similar to the Gaussian elimination except b is not required in advance. All operations performed on b using the Gaussian elimination are eliminated to save computation cost. The procedures are expressed as follows:

1. Step 1: Factorization/decomposition
2. Step 2: Forward substitution
3. Step 3: Backward substitution

3.2.1 Factorization

We factor A into a product LU, where L is a lower triangular matrix and U is an upper triangular matrix:

$$A = LU \tag{3.20}$$

Either L or U can have a diagonal of ones. The factors of A, being upper and lower triangular, can be stored in one matrix B, i.e., $B = L + U - I$. In practice, B is stored in place of A to save memory storage. There are two widely used algorithms for factorization: Crout's algorithm (setting the diagonal elements of U to 1) [1] and Doolittle's algorithm (setting the diagonal elements of L to 1) [3]. In the following, we will use a 4×4 matrix to illustrate the direct finding of L and U by the Crout's algorithm:

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Multiplying the two matrices on the LHS of the preceding equations gives

$$\begin{bmatrix} l_{11} & l_{11}u_{12} & l_{11}u_{13} & l_{11}u_{14} \\ l_{21} & l_{21}u_{12} + l_{22} & l_{21}u_{13} + l_{22}u_{23} & l_{21}u_{14} + l_{22}u_{24} \\ l_{31} & l_{31}u_{12} + l_{32} & l_{31}u_{13} + l_{32}u_{23} + l_{33} & l_{31}u_{14} + l_{32}u_{24} + l_{33}u_{34} \\ l_{41} & l_{41}u_{12} + l_{42} & l_{41}u_{13} + l_{42}u_{23} + l_{43} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + l_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

The solution sequences, indicated by the superscripts, for solving the 16 unknowns ($n = 4$) are

$$\begin{matrix} l_{11}^{1} & u_{12}^{5} & u_{13}^{6} & u_{14}^{7} \\ l_{21}^{2} & l_{22}^{8} & u_{23}^{11} & u_{24}^{12} \\ l_{31}^{3} & l_{32}^{9} & l_{33}^{13} & u_{34}^{15} \\ l_{41}^{4} & l_{42}^{10} & l_{43}^{14} & l_{44}^{16} \end{matrix}$$

or

1st column: $l_{11} = a_{11}$, $l_{21} = a_{21}$, $l_{31} = a_{31}$, $l_{41} = a_{41}$
1st row: $u_{12} = \frac{a_{12}}{l_{11}}$, $u_{13} = \frac{a_{13}}{l_{11}}$, $u_{14} = \frac{a_{14}}{l_{11}}$

2nd column: $l_{22} = a_{22} - l_{21}u_{12}$, $l_{32} = a_{32} - l_{31}u_{12}$, $l_{42} = a_{42} - l_{41}u_{12}$
2nd row: $u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}}$, $u_{24} = \frac{a_{24} - l_{21}u_{14}}{l_{22}}$

3rd column: $l_{33} = a_{33} - l_{13}u_{13} - l_{32}u_{23}$, $l_{43} = a_{43} - l_{41}u_{13} - l_{42}u_{13}$
3rd row: $u_{34} = \frac{a_{34} - l_{31}u_{14} - l_{32}u_{24}}{l_{33}}$

4th column: $l_{44} = a_{44} - l_{41}u_{14} - l_{42}u_{24} - l_{43}u_{34}$

Note that l_{11}, l_{22}, and l_{33} are elements by which we divide and they are called *pivots*. Division by a zero pivot is not allowed. We now derive the Crout's algorithm for LU decomposition based on the solution procedures described previously:

1. $l_{ji} = a_{ji}$, $j = 1, 2, \ldots, n$ (1st column)
2. $u_{1j} = a_{1j}/l_{11}$, $j = 2, \ldots, n$ (1st row)
3. At the kth step, for column k,

$$l_{jk} = a_{jk} - l_{j1}u_{1k} - l_{j2}u_{2k} - \cdots = a_{jk} - \sum_{m=1}^{k-1} l_{jm}u_{mk}, \quad j = k, \ldots, n$$

4. At the kth step, for row k,

$$u_{kj} = \left(\frac{1}{l_{kk}}\right)(a_{kj} - l_{k1}u_{1j} - l_{k2}u_{2j} - \cdots) = \left(\frac{1}{l_{kk}}\right)\left(a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}\right), \quad j = k+1, \ldots, n$$

The algorithm can be summarized in a compact form:

1. Set $k = 1$
2. $l_{jk} = a_{jk} - \sum_{m=1}^{k-1} l_{jm}u_{mk}, \quad j = k, \ldots, n$
3. If $k = n$, stop
4. $u_{kj} = \left(\frac{1}{l_{kk}}\right)\left(a_{kj} - \sum_{m=1}^{k-1} l_{km}u_{mj}\right), \quad j = k+1, \ldots, n$
5. $k = k+1$, go to step 2

3.2.2 Forward Substitution

Once A has been factored into L and U, the system of equations is written as follows:

$$LUx = b$$

Define an auxiliary vector y, which can be solved by

$$L \cdot y = b \tag{3.21}$$

Due to the special form of L, the auxiliary vector γ can be solved very simply:

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Starting from the first equation we write the solution as follows:

$$\begin{aligned} y_1 &= \frac{b_1}{l_{11}} \\ y_2 &= \frac{(b_2 - l_{21}y_1)}{l_{22}} \\ y_3 &= \frac{(b_3 - l_{31}y_1 - l_{32}y_2)}{l_{33}} \\ y_4 &= \frac{(b_4 - l_{41}y_1 - l_{42}y_2 - l_{43}y_3)}{l_{44}} \end{aligned}$$

and, in general

$$y_i = \frac{\left(b_i = \sum_{j=1}^{i-1} l_{ij}y_i\right)}{l_{ii}}, \quad i = 1, \ldots, n$$

This is called the *forward substitution* process.

3.2.3 Backward Substitution

Once y has been solved, we can proceed to solve for the unknown x by

$$Ux = y \tag{3.22}$$

Again, due to the special form of U, the unknown vector x can be solved very simply:

$$\begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Starting from the first equation, we write the solution as follows:

$$\begin{aligned} x_4 &= y_4 \\ x_3 &= y_3 - u_{34}x_4 \\ x_2 &= y_2 - u_{23}x_3 - u_{24}x_4 \\ x_1 &= y_1 - u_{12}x_2 - u_{13}x_3 - u_{14}x_4 \end{aligned}$$

and, in general,

$$x_i = y_i - \sum_{j=i+1}^{n} u_{ij}x_j, \quad i = n, n-1, \ldots, 1$$

This is called the *backward substitution* process.

3.2.4 Pivoting

If in the process of factorization the pivot (l_{kk}) is zero, it is then necessary to interchange rows and possibly columns to put a nonzero entry in the pivot position so that the factorization can proceed. This is known as *pivoting*.

If $a_{ll} = 0$

1. Need to find another row i which has $a_{il} \neq 0$. This can always be done. Otherwise, all entries of column 1 are zero and hence, det $|A| = 0$. The solution process should then be aborted.
2. Interchange row i and row 1. Note that this must be done for both A and b.

If $l_{kk} = 0$

1. Find another row $r (r = k+1, \ldots, n)$, which has

$$l_{rk} = a_{rk} - \sum_{m=1}^{k-1} l_{rm}u_{mk} \neq 0$$

2. Interchange row r and row k in A and b.

Pivoting is also carried out for numerical stability (i.e., minimize round-off error). For example, one would search for the entry with the maximum absolute value of l_{rR} in columns below the diagonal and perform row interchange to put that element on the diagonal. This is called partial pivoting. Complete

pivoting involves searching for the element with the maximum absolute value in the unfactorized part of the matrix and moving that particular element to the diagonal position by performing both row and column interchanges. Complete pivoting is more complicated to program than partial pivoting. Partial pivoting is used more often.

3.2.5 Computation Cost of *LU* Factorization

In this section, we derive the multiplication/division count for the *LU* decomposition process. As a variation, we derive the computation cost for the Doolittle's algorithm (setting the diagonal elements of L to 1):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

LHS after multiplication:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} & l_{21}u_{13} + u_{23} & l_{21}u_{14} + u_{24} \\ l_{31}u_{11} & l_{31}u_{12} + l_{32}u_{22} & l_{31}u_{13} + l_{32}u_{23} + u_{33} & l_{31}u_{14} + l_{32}u_{24} + u_{34} \\ l_{41}u_{11} & l_{41}u_{12} + l_{42}u_{22} & l_{41}u_{13} + l_{42}u_{23} + l_{43}u_{33} & l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34} + u_{44} \end{bmatrix}$$

$$\begin{aligned} &\text{Column 1: } (l_{21}, l_{31}, l_{41})u_{11} \\ &\text{Column 2: } (l_{21}, l_{31}, l_{41})u_{12} + (l_{32}, l_{42})u_{22} \\ &\text{Column 3: } (l_{21}, l_{31}, l_{41})u_{13} + (l_{32}, l_{42})u_{23} + l_{43}u_{33} \\ &\text{Column 4: } (l_{21}, l_{31}, l_{41})u_{14} + (l_{32}, l_{42})u_{24} + l_{43}u_{34} \end{aligned} \tag{3.23}$$

Let the symbol $\langle \cdot \rangle$ denote the number of nonzero elements of a matrix or vector. Or

- $\langle U \rangle$: number of nonzeros of U
- $\langle L \rangle$: number of nonzeros of L
- $\langle A_{i\cdot} \rangle$: number of nonzeros in row i of matrix A
- $\langle A_{\cdot j} \rangle$: number of nonzeros in column j of matrix A

From Equation 3.23, the total number of nonzero multiplications and divisions for *LU* factorization is given by

$$(\langle L_{\cdot 1} \rangle - 1)(\langle U_{1\cdot} \rangle) + (\langle L_{\cdot 2} \rangle - 1)(\langle U_{2\cdot} \rangle) + (\langle L_{\cdot 3} \rangle - 1)(\langle U_{3\cdot} \rangle) \tag{3.24}$$

Let α be the total number of multiplications and divisions. Express Equation 3.24 as a summation:

$$\alpha = \sum_{k=1}^{n} (\langle L_{\cdot k} \rangle - 1)(\langle U_{k\cdot} \rangle) \tag{3.25}$$

or for $n = 4$,

$$\alpha = 3 \times 4 + 2 \times 3 + 1 \times 2$$

If L and U are full, Equation 3.25 can be simplified as follows:

$$\begin{aligned}\alpha &= \sum_{k=1}^{n}(n-k)(n-k+1)\\ &= \sum_{k=1}^{n}(n^2+k^2-2kn+n-k)\\ &= n_3 + \sum_{k=1}^{n}k^2 - 2n \times \frac{n(n+1)}{2} + n^2 - \frac{n(n+1)}{2}\\ &= \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2}\\ &= \frac{n^3-n}{3}\end{aligned} \tag{3.26}$$

The total number of mul/div for forward substitution is equal to the total number of nonzeros for L or $\langle L\rangle$. The total number of mul/div for backward substitution is equal to the total number of nonzeros for U for $\langle U\rangle$. Let β be the mul/div count for the forward and backward substitutions,

$$\beta = \langle L\rangle + \langle U\rangle$$

It follows that

$$\beta = n^2 \tag{3.27}$$

If L and U are full, combining Equations 3.26 and 3.27, we obtain the computation cost of solving a system of linear algebraic equations using direct LU decomposition:

$$\begin{aligned}\text{Total} &= \alpha + \beta\\ \text{Total} &= \alpha + \beta = \frac{n^3}{3} + n^2 - \frac{n}{3}\end{aligned} \tag{3.28}$$

References

1. Hachtel et al., The sparse tableau approach to network analysis and design, *IEEE Trans. Circuit Theory*, CT-18: 101–113, January 1971.
2. L. W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Memo No. ERL-M520, Electronic Research Laboratory, University of California, Berkeley, CA, May 1975.
3. C. W. Ho et al., The modified nodal approach to network analysis, *IEEE Trans. Circuits Syst.*, CAS-22: 504–509, June 1975.
4. J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, New York, 1983.
5. K. S. Kundert, *Advances in CAD for VLSI: Circuit Analysis, Simulation and Design*, Vol. 3, Part 1, Chapter 6, North-Holland, Amsterdam, the Netherlands, 1986.

4

Design by Optimization

4.1 Introduction **4**-1

4.2 Optimization Algorithms **4**-2
Nonlinear Optimization Problems • Basic Definitions • Constrained Optimization Methods • Multicriterion Optimization and Pareto Criticality

4.3 Transistor Sizing Problem for CMOS Digital Circuits ... **4**-10
Problem Description • Delay Modeling • Area Model • Sensitivity-Based TILOS Algorithm • Transistor Sizing Using the Method of Feasible Directions • Lagrangian Multiplier Approaches • Two-Step Optimization • Convex Programming-Based Approach • Statistical Optimization

4.4 Analog Design Centering Problem **4**-19
Problem Description • Simplicial Approximation Method • Ellipsoidal Method • Convexity-Based Approaches • Concluding Remarks

References **4**-24

Sachin S. Sapatnekar
University of Minnesota

4.1 Introduction

In many integrated circuit (IC) design situations, a designer must make complex trade-offs between conflicting behavioral requirements, dealing with functions that are often nonlinear. The number of parameters involved in the design process may be large, necessitating the use of algorithms that provide qualitatively good solutions in a computationally efficient manner.

The theory and utilization of optimization algorithms in computer-aided design (CAD) of ICs are illustrated here. The form of a general nonlinear optimization problem is first presented, and some of the commonly used methods for optimization are overviewed. It is frequently said that setting up an optimization problem is an art, while (arguably) solving it is an exact science. To provide a flavor for both of these aspects, case studies on the following specific design problems are examined:

- *Transistor sizing*: The delay of a digital circuit can be tuned by adjusting the sizes of transistors within it. By increasing the sizes of a few selected transistors from the minimum size, significant improvements in performance are achievable. However, one must take care to ensure that the area overhead incurred in increasing these sizes is not excessive. The area-delay trade-off here is the transistor size optimization problem. We address the solution of this problem under deterministic and statistical delay models.
- *Analog design centering*: The values of design parameters of an analog circuit are liable to change due to manufacturing variations. This contributes to a deviation in the behavior of the circuit from the norm, and may lead to dysfunctional circuits that violate the performance parameters that they

were designed for. The problem of design centering attempts to ensure that under these variations, the probability of a circuit satisfying its performance specifications is maximized by placing the circuit in the center of its feasible region.

4.2 Optimization Algorithms

4.2.1 Nonlinear Optimization Problems

The standard form of a constrained nonlinear optimization problem is

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}): \mathbf{R}^n \rightarrow \mathbf{R} \\ \text{subject to} & \mathrm{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathrm{g}: \mathbf{R}^n \rightarrow \mathbf{R}^m,\ \mathbf{x} \in \mathbf{R}^n \end{array} \tag{4.1}$$

representing the minimization of a function f of n variables under constraints specified by inequalities determined by functions $\mathbf{g} = [g_1 \cdots g_m]^{\mathrm{T}}$. f and g_i are, in general, nonlinear functions, so that the linear programming problem is a special case of the above. The parameters $\mathbf{x}$ may, for example, represent circuit parameters, and $f(\mathbf{x})$ and $g_i(\mathbf{x})$ may correspond to circuit performance functions. Note that "$\geq$" inequalities can be handled under this paradigm by multiplying each side by -1, and equalities by representing them as a pair of inequalities. The maximization of an objective function function $f(\mathbf{x})$ can be achieved by minimizing $-f(\mathbf{x})$.

The set $\mathcal{F} = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x}) \leq 0\}$ that satisfies the constraints on the nonlinear optimization problem is known as the feasible set, or the feasible region. If $\mathcal{F}$ is empty (nonempty), then the optimization is said to be unconstrained (constrained).

Several mathematical programming techniques can be used to solve the optimization problem above; some of these are outlined here. For further details, the reader is referred to a standard text on optimization, such as [1]. Another excellent source for optimization techniques and their applications to IC design is a survey paper by Brayton et al. [2].

The above formulation may not directly be applicable to real-life design problems, where, often, multiple conflicting objectives must be optimized. In such a case, one frequently uses techniques that map on the problem to the form in Equation 4.1 (see Section 4.2.4).

4.2.2 Basic Definitions

In any discussion on optimization, it is virtually essential to understand the idea of a convex function and a convex set, since these have special properties, and it is desirable to formulate problems as convex programming problems, wherever it is possible to do so without an undue loss in modeling accuracy. (Unfortunately, it is not always possible to do so!)

Definition 4.1: A set C in $\mathbf{R}^n$ is said to be a convex set if, for every $\mathbf{x}_1, \mathbf{x}_2 \in C$, and every real number α, $0 \leq \alpha \leq 1$, the point $\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in C$.

This definition can be interpreted geometrically as stating that a set is convex if, given two points in the set, every point on the line segment joining the two points is also a member of the set. Examples of convex and nonconvex sets are shown in Figure 4.1.

Two examples of convex sets that are referred to later are the following geometric bodies:

1. An ellipsoid $E(\mathbf{x}, \mathcal{B}, r)$ centered at point $\mathbf{x}$ is given by the following equation:

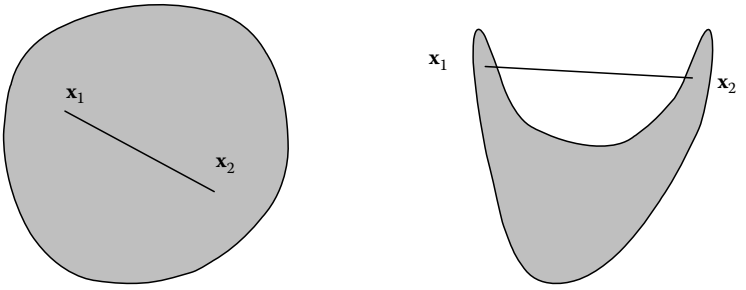


FIGURE 4.1 Convex sets.

$$\{\mathbf{y}|(\mathbf{y}-\mathbf{x})^{\mathrm{T}}\mathcal{B}(\mathbf{y}-\mathbf{x}) \leq r^2\}. \tag{4.2}$$

If $\mathcal{B}$ is a scalar multiple of the unit matrix, then the ellipsoid is called a hypersphere.

2. A (convex) polytope is defined as an intersection of half-spaces, and is given by the following equation:

$$\mathcal{P} \;=\; \{\mathbf{x}|A\mathbf{x} \geq \mathbf{b}\}, A \in \mathbf{R}^{m\times n}, \mathbf{b} \in \mathbf{R}^m, \tag{4.3}$$

corresponding to a set of m inequalities $\mathbf{a}_i^{\mathrm{T}}\,\mathbf{x} \geq b_i$, $\mathbf{a}_i \in \mathbf{R}^n$.

Definition 4.2: The convex hull of m points, $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbf{R}^n$, denoted co $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, is defined as the set of points $\mathbf{y} \in \mathbf{R}^n$ such that

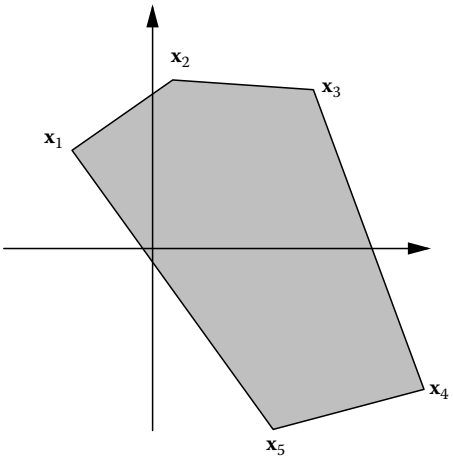


FIGURE 4.2 Convex hull of five points.

$$\mathbf{y} = \sum_{i=1}^{m} \alpha_i \mathbf{x}_i; \quad \alpha_i \geq 0 \;\forall\; i, \quad \sum_{i=0}^{m} \alpha_i = 1. \tag{4.4}$$

The convex hull is the smallest convex set that contains the m points. An example of the convex hull of five points in the plane is shown by the shaded region in Figure 4.2. If the set of points $\mathbf{x}_i$ is of finite cardinality (i.e., m is finite), then the convex hull is a polytope. Hence, a polytope is also often described as the convex hull of its vertices.

Definition 4.3: A function f defined on a convex set Ω is said to be a convex function if, for every $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$, and every α, $0 \leq \alpha \leq 1$,

$$f[\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2] \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2). \tag{4.5}$$

f is said to be strictly convex if the inequality in Equation 4.5 is strict for $0 < \alpha < 1$.

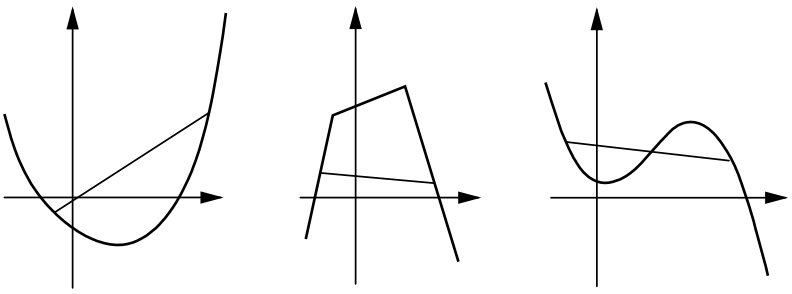


FIGURE 4.3 Convex functions.

Geometrically, a function is convex if the line joining two points on its graph is always above the graph. Examples of convex and nonconvex functions on $\mathbf{R}^n$ are shown in Figure 4.3.

Definition 4.4: A function g defined on a convex set Ω is said to be a concave function if the function $f = -g$ is convex. The function g is strictly concave if $-g$ is strictly convex.

Definition 4.5: The convex programming problem is stated as follows:

$$\text{minimize} \quad f(\mathbf{x}) \tag{4.6}$$

$$\text{such that} \quad \mathbf{x} \in S \tag{4.7}$$

where f is a convex function and S is a convex set. This problem has the property that any local minimum of f over S is a global minimum. An excellent reference on convex programming is [3].

Definition 4.6: A posynomial is a function h of a positive variable $\mathbf{x} \in \mathbf{R}^n$ that has the form

$$h(\mathbf{x}) = \sum_j \gamma_j \prod_{i=1}^{n} x_i^{\alpha_{ij}} \tag{4.8}$$

where the exponents $\alpha_{ij} \in \mathbf{R}$ and the coefficients $\gamma_j > 0$.

For example, the function $3.7x_1^{1.4}x_2^{\sqrt{3}} + 1.8x_1^{-1}x_3^{2.3}$ is a posynomial in the variables x_1, x_2, and x_3. Roughly speaking, a posynomial is a function that is similar to a polynomial, except that the coeffcients γ_j must be positive, and an exponent α_{ij} could be any real number, and not necessarily a positive integer, unlike a polynomial. A posynomial has the useful property that it can be mapped onto a convex function through an elementary variable transformation, $(x_i) = (e^{z_i})$. Such functional forms are useful since in the case of an optimization problem where the objective function and the constraints are posynomial, the problem can easily be mapped onto a convex programming problem.

The family of posynomials is expanded to the superset of generalized posynomials in Ref. [4], and these functions can also be mapped to convex functions under the above transform.

4.2.3 Constrained Optimization Methods

Most problems in IC design involve the minimization or maximization of a cost function subject to certain constraints. In this section, a few prominent techniques for constrained optimization are presented. The reader is referred to [1] for details on unconstrained optimization.

4.2.3.1 Linear Programming

Linear programming is a special case of nonlinear optimization, and is the convex programming problem where the objective and constraints are all linear functions. The problem is stated as

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^{\mathrm{T}}\,\mathbf{x} \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\ \text{where} \quad & \mathbf{c}, \mathbf{x} \in \mathbf{R}^n,\ \mathbf{b} \in \mathbf{R}^m,\ A \in \mathbf{R}^{m \times n} \end{aligned} \tag{4.9}$$

It can be shown that any solution to a linear program must necessarily occur at a vertex of the constraining polytope. The most commonly used technique for solution of linear programs, the simplex method [1], is based on this principle. The computational complexity of this method can show an exponential behavior for pathological cases, but for most practical problems, it has been observed to grow linearly with the number of variables and sublinearly with the number of constraints. Algorithms with polynomial time worst-case complexity do exist; these include Karmarkar's method and the Shor–Khachiyan ellipsoidal method. The computational complexity of the latter, however, is often seen to be impractical from a practical standpoint.

Some examples of CAD problems that are posed as linear programs include placement, gate sizing, clock skew optimization, layout compaction, etc. In several cases, the structure of the problem can be exploited to arrive at graph-based solutions, for example, in layout compaction.

4.2.3.2 Lagrange Multiplier Methods

These methods are closely related to the first-order Kuhn–Tucker necessary conditions on optimality, which state that given an optimization problem of the form in Equation 4.1, if f and $\mathbf{g}$ are differentiable at $\mathbf{x}^\star$, then there is a vector $\boldsymbol{\lambda} \in \mathbf{R}^m$, $(\boldsymbol{\lambda})_i \geq 0$, such that

$$\nabla f(\mathbf{x}^\star) + \boldsymbol{\lambda}^{\mathrm{T}} \nabla \mathbf{g}(\mathbf{x}^\star) = 0 \tag{4.10}$$

$$\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{g}(\mathbf{x}^\star) = 0 \tag{4.11}$$

These correspond to $m+1$ equations in $m+1$ variables; the solution to these provides the solution to the optimization problem. The variables $\boldsymbol{\lambda}$ are known as the Lagrange multipliers.

Note that since $g_i(\mathbf{x}^\star) \leq 0$, and because of the nonnegativity constraint on the Lagrange multipliers, $\boldsymbol{\lambda}$, it follows from Equation 4.11 that $(\boldsymbol{\lambda})_i = 0$ for inactive constraints (constraints with $g_i(\mathbf{x}) < 0$).

4.2.3.3 Penalty Function Methods

These methods convert the constrained optimization problem in Equation 4.1 into an equivalent unconstrained optimization problem, since such problems are easier to solve than constrained problems, as

$$\text{minimize } h(\mathbf{x}) = f(\mathbf{x}) + c \cdot P(\mathbf{x}) \tag{4.12}$$

where

$P(\mathbf{x})$: $\mathbf{R}^n \rightarrow \mathbf{R}$ is known as a penalty function

c is a constant

The value of $P(\mathbf{x})$ is zero within the feasible region, and positive outside the region, with the value becoming larger as one moves farther from the feasible region; one possible choice when the $g_i(\mathbf{x})$'s are continuous is given by

$$P(\mathbf{x}) = \sum_{i=1}^{m} \max(0, -g_i(\mathbf{x})) \tag{4.13}$$

For large c, it is clear that the minimum point of Equation 4.12 will be in a region where P is small. Thus, as c is increased, it is expected that the corresponding solution point will approach the feasible region and minimize f. As $c \to \infty$, the solution of the penalty function method converges to a solution of the constrained optimization problem.

In practice, if one were to begin with a high value of c, one may not have very good convergence properties. The value of c is increased in each iteration until c is high and the solution converges.

4.2.3.4 Method of Feasible Directions

The method of feasible directions is an optimization algorithm that improves the objective function without violating the constraints. Given a point $\mathbf{x}$, a direction $\mathbf{d}$ is feasible if there is a step size $\bar{\alpha} > 0$ such that $\mathbf{x} + \alpha \cdot \mathbf{d} \in \mathcal{F} \ \forall \ 0 \leq \alpha \leq \bar{\alpha}$, where $\mathcal{F}$ is the feasible region. More informally, this means that one can take a step of size up to $\bar{\alpha}$ along the direction $\mathbf{d}$ without leaving the feasible region. The method of feasible direction attempts to choose a value of α in a feasible direction $\mathbf{d}$ such that the objective function f is minimized along the direction, and α is such that $\mathbf{x} + \alpha \cdot \mathbf{d}$ is feasible.

One common technique that uses the method of feasible directions is as follows. A feasible direction at $\mathbf{x}$ is found by solving the following linear program:

$$\begin{aligned} &\text{minimize} && \varepsilon \\ &\text{subject to} && \langle \nabla f(\mathbf{x}) \cdot \mathbf{d} \rangle \leq \varepsilon && (4.14) \\ & && \langle \nabla g_i(\mathbf{x}) \cdot \mathbf{d} \rangle \leq \varepsilon && (4.15) \\ &\text{and} && \text{normalization requirements on } \mathbf{d} \end{aligned}$$

where the second set of constraints are chosen for all $g_i \geq -b$, where b serves to incorporate the effects of near-active constraints to avoid the phenomenon of jamming (also known as zigzagging) [1]. The value of b is brought closer to 0 as the optimization progresses. One common method that is used as a normalization requirement is to set $\mathbf{d}^{\mathrm{T}}\mathbf{d} = 1$. This constraint is nonlinear and nonconvex, and is not added to the linear program as an additional constraint; rather, it is exercised by normalizing the direction $\mathbf{d}$ after the linear program has been solved. An appropriate step size in this direction is then chosen by solving a one-dimensional optimization problem.

Feasible direction methods are popular in finding engineering solutions because the value of $\mathbf{x}$ at each iteration is feasible, and the algorithm can be stopped at any time without waiting for the algorithm to converge, and the best solution found so far can be used.

4.2.3.5 Vaidya's Convex Programming Algorithm

As mentioned earlier, if the objective function and all constraints in Equation 4.1 are convex, the problem is a convex programming problem, and any local minimum is a global minimum. The first large-scale practical implementation of this algorithm is described in Ref. [5].

Initially, a polytope $\mathcal{P} \in \mathbf{R}^n$ that contains the optimal solution, $\mathbf{x}_{\text{opt}}$, is chosen. The objective of the algorithm is to start with a large polytope, and in each iteration, to shrink its volume while keeping the optimal solution, $\mathbf{x}_{\text{opt}}$, within the polytope, until the polytope becomes sufficiently small. The polytope $\mathcal{P}$ may, for example, be initially selected to be an n-dimensional box described by the set

$$\{\mathbf{x} | x_{\min} \leq x_i \leq x_{\max}\} \tag{4.16}$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of each variable, respectively. The algorithm proceeds iteratively as follows:

Step 1. A center $\mathbf{x}_c$ deep in the interior of the polytope $\mathcal{P}$ is found.

Step 2. An oracle is invoked to determine whether the center $\mathbf{x}_c$ lies within the feasible region $\mathcal{F}$. This may be done by verifying that all of the constraints of the optimization problem are met at $\mathbf{x}_c$.

If the point $\mathbf{x}_c$ lies outside $\mathcal{F}$, it is possible to find a separating hyperplane passing through $\mathbf{x}_c$ that divides $\mathcal{P}$ into two parts, such that $\mathcal{F}$ lies entirely in the part satisfying the constraint

$$\mathbf{c}^T\mathbf{x} \geq \beta \tag{4.17}$$

where $\mathbf{c} = -[\nabla_{g_p}(\mathbf{x})]^T$ is the negative of the gradient of a violated constraint, g_p, and $\beta = \mathbf{c}^T\mathbf{x}_c$. The separating hyperplane above corresponds to the tangent plane to the violated constraint.

If the point $\mathbf{x}_c$ lies within the feasible region $\mathcal{F}$, then there exists a hyperplane (Equation 4.17) that divides the polytope into two parts such that $\mathbf{x}_{opt}$ is contained in one of them, with

$$\mathbf{c} = -[\nabla f(\mathbf{x})]^T \tag{4.18}$$

being the negative of the gradient of the objective function, and β being defined as $\beta = \mathbf{c}^T\mathbf{x}_c$ once again.

Step 3. In either case, Equation 4.17 is added to the current polytope to give a new polytope that has roughly half the original volume.

Step 4. Go to Step 1; the process is repeated until the polytope is sufficiently small.

Example 4.1

The algorithm is illustrated by using it to solve the following problem:

$$\text{minimize } f(x_1, x_2) \text{ such that } (x_1, x_2) \in S \tag{4.19}$$

where S is a convex set and f is a convex function. The shaded region in Figure 4.4a is the set S, and the dotted lines show the level curves of f. The point $\mathbf{x}_{opt}$ is the solution to this problem.

1. The expected solution region is bounded by a rectangle, as shown in Figure 4.4a.
2. The center, $\mathbf{x}_c$, of this rectangle is found.
3. The oracle is invoked to determine whether or not $\mathbf{x}_c$ lies within the feasible region. In this case, it can be seen that $\mathbf{x}_c$ lies outside the feasible region. Hence, the gradient of the constraint function is used to construct a hyperplane through $\mathbf{x}_c$, such that the polytope is divided into two parts of roughly equal volume, one of which contains the solution $\mathbf{x}_{opt}$. This is illustrated in Figure 4.4b, where the shaded region corresponds to the updated polytope.
4. The process is repeated on the new smaller polytope. Its center lies inside the feasible region; hence, the gradient of the objective function is used to generate a hyperplane that further shrinks the size of the polytope, as shown in Figure 4.4c.
5. The result of another iteration is illustrated in Figure 4.4d. The process continues until the polytope has been shrunk sufficiently.

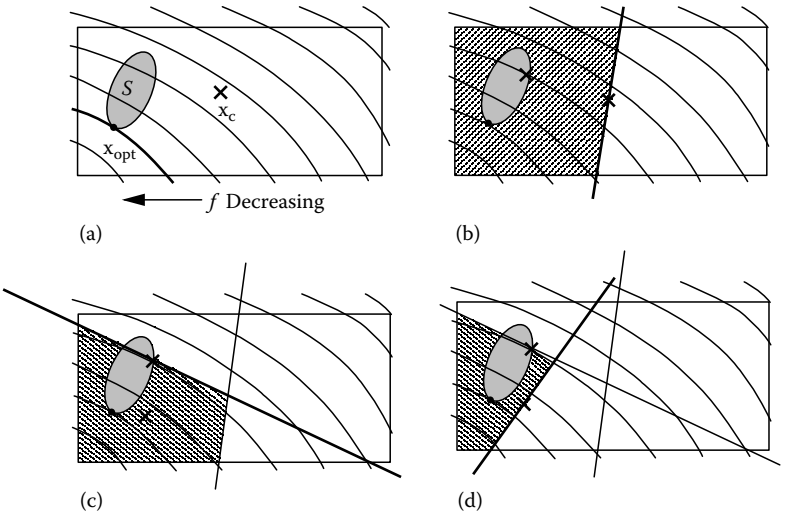


FIGURE 4.4 Example illustrating the convex programming algorithm. (From Sapatnekar, S. S., Rao, V. B., Vaidya, P. M., and Kang, S. M., *IEEE Trans. Comput. Aided Des.*, 12, 1621, 1993. With permission.)

4.2.3.6 Other Methods

The compact nature of this book makes it infeasible to enumerate or describe all of the methods that are used for nonlinear optimization. Several other methods, such as Newton's and modified Newton/quasi-Newton methods, conjugate gradient methods, etc., are often useful in engineering optimization. For these and more, the reader is referred to a standard text on optimization, such as [1].

4.2.4 Multicriterion Optimization and Pareto Criticality

Most IC design problems involve trade-offs between multiple objectives. In cases where one objective can be singled out as the most important one and a reasonable constraint set can be defined in terms of the other objectives, the optimization problem can be stated using the formulation in Equation 4.1. This is convenient since techniques for the solution of a problem in this form have been extensively studied, and a wide variety of optimization algorithms are available.

Let $\mathbf{f}$ be a vector of design objectives that is a function of the design variables $\mathbf{x}$, where

$$\mathbf{f}(\mathbf{x}):\ \mathbf{R}^n \rightarrow \mathbf{R}^m = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \tag{4.20}$$

It is extremely unlikely in a real application that all of the f_i's will be optimal at the same point, and hence one must trade off the values of the f_i's in a search for the best design point.

In this context, we note that at a point $\mathbf{x}$, we are interested in taking a step δ in a direction $\mathbf{d}$, $\|\mathbf{d}\| = 1$, so that

$$f_i(\mathbf{x} + \delta \cdot \mathbf{d}) \leq f_i(\mathbf{x}) \ \forall\ 1 \leq i \leq m \tag{4.21}$$

A Pareto critical point is defined as a point $\mathbf{x}$ where no such small step of size less than δ exists in any direction. If a point is Pareto critical for any step size from the point $\mathbf{x}$, then $\mathbf{x}$ is a Pareto point. The notion of a Pareto critical point is, therefore, similar to that of a local minimum, and that of a Pareto point to a global minimum. In computational optimization, one is concerned with the problem of finding a local minimum since, except in special cases, it is the best that one can be guaranteed of finding without

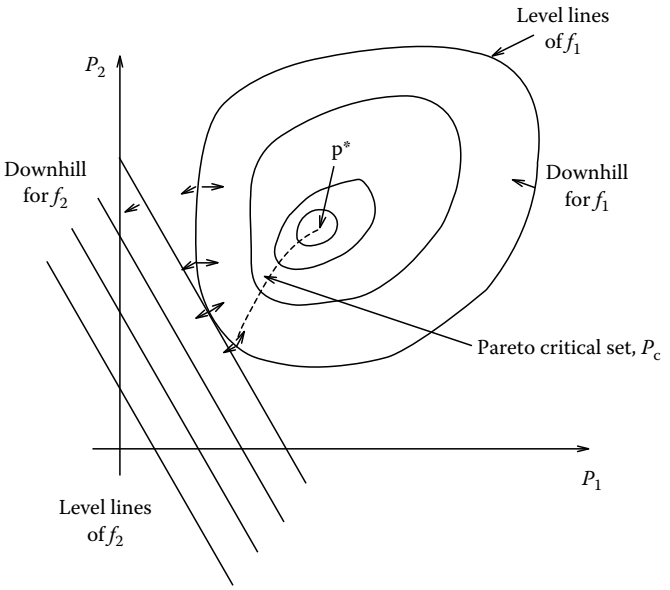


FIGURE 4.5 Exact conflict at a Pareto critical point. (From Brayton, R.K., Hachtel, G.D., and Sangiovanni-Vincentelli, A.L., *Proc. IEEE*, 69, 1334, 1981. With permission.)

an exhaustive search. If the set of all Pareto critical points is P_c, and the set of Pareto points is P, then clearly $P \subset P_c$. In general, there could be an infinite number of Pareto points, but the best circuit design must necessarily occur at a Pareto point $\mathbf{x} \in P$.

In Figure 4.5, the level curves of two objective functions are plotted in $\mathbf{R}^2$. f_1 is nonlinear and has a minimum at $\mathbf{x}^*$. f_2 is linear and decreases as both x_1 and x_2 decrease. The Pareto critical set, P_c, is given by the dashed curve. At few of the points, the unit normal to the level lines of f_1 and f_2, i.e., the negative gradients of f_1 and f_2, is shown. From the figure, it can be seen that if the unit normals at point $\mathbf{x}$ are not equal and opposite, then the unit normals will have a common downhill direction allowing a simultaneous decrease in f_1 and f_2, and hence, $\mathbf{x}$ would not be a Pareto critical point. Therefore, a Pareto critical point is one where the gradients of f_1 and f_2 are opposite in direction, i.e., $\lambda \nabla f_1 = -\nabla f_2$, where λ is some scale factor.

In higher dimensions, a Pareto critical point is characterized by the existence of a set of weights, $w_i > 0$ $\forall$ $1 \leq i \leq m$, such that

$$\sum_{i=1}^{m} w_i \nabla f_i = 0. \tag{4.22}$$

Some of the common methods that are used for multicriterion optimization are discussed below.

4.2.4.1 Weighted Sums Optimization

The multiple objectives $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ are combined as

$$F(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}) \tag{4.23}$$

where $w_i > 0$ $\forall$ $i = 1, \ldots, m$, and the function $F(\mathbf{x})$ is minimized.

At any local minimum point of $F(\mathbf{x})$, the relation in Equation 4.22 is seen to be valid, and hence, $\mathbf{x} \in P_c$. In general, $P \neq P_c$, but it can be shown that when each f_i is a convex function, then $P = P_c$; if so, it can also be shown that all Pareto points can be obtained by optimizing the function F in Equation 4.23. However, for nonconvex functions, there are points $\mathbf{x} \in P$ that cannot be obtained by the weighted sum optimization since Equation 4.22 is only a necessary condition for the minimum of F. A characterization of the Pareto points that can be obtained by this technique is provided in Ref. [2].

In practice, the w_i's must be chosen to reflect the magnitudes of the f_i's. For example, if one of the objectives is a voltage quantity whose typical value is a few volts, and another is a capacitor value that is typically a few picofarads, the weight corresponding to the capacitor value would be roughly 10^{12} times that for the voltage, in order to ensure that each objective has a reasonable contribution to the objective function value. The designer may further weigh the relative importance of each objective in choosing the w_i's. This objective may be combined with additional constraints to give a formulation of the type in Equation 4.1.

4.2.4.2 Minmax Optimization

The following objective function is used for Equation 4.1

$$\text{minimize } F(\mathbf{x}) = \max_{1 \leq i \leq m} w_i f_i(\mathbf{x}) \tag{4.24}$$

where the weights $w_i > 0$ are chosen as in the case of weighted sums optimization.

The above equation can equivalently be written as the following constrained optimization problem:

$$\begin{aligned} &\text{minimize } r \\ &\text{subject to } w_i f_i(\mathbf{x}) \leq r \end{aligned} \tag{4.25}$$

Minimizing the objective function described by Equation 4.24 with different sets of w_i values can be used to obtain all Pareto points [2].

Since this method can, unlike the weighted-sums optimization method, be used to find all Pareto critical points, it would seem to be a more natural setting for obtaining Pareto points than the weighted sum minimization. However, when the f_i's are convex, the weighted sums approach is preferred since it is an unconstrained minimization, and is computationally easier than a constrained optimization. It must be noted that when the f_i's are not all convex, the minmax objective function is nonconvex, and finding all local minima is a nontrivial process for any method.

4.3 Transistor Sizing Problem for CMOS Digital Circuits

4.3.1 Problem Description

Circuit delays in ICs often must be reduced to obtain faster response times. A typical CMOS digital IC consists of multiple stages of combinational logic blocks that lie between latches that are clocked by system clock signals. For such a circuit, delay reduction must ensure that valid signals are produced at each output latch of a combinational block, before any transition in the signal clocking the latch. In other words, the worst-case input–output delay of each combinational stage must be restricted to be below a certain specification.

Given the circuit topology, the delay of a combinational circuit can be controlled by varying the sizes of transistors in the circuit. Here, the size of a transistor is measured in terms of its channel width, since the channel lengths of MOS transistors in a digital circuit are generally uniform. In coarse terms, the circuit delay can usually be reduced by increasing the sizes of certain transistors in the circuit. Hence, making

the circuit faster usually entails the penalty of increased circuit area. The area-delay trade-off involved here is, in essence, the problem of transistor size optimization.

Three formal statements of the problem are stated below:

$$\text{Minimize } \textit{Area} \text{ subject to } \textit{Delay} \leq T_{\text{spec}} \tag{4.26}$$

$$\text{Minimize } \textit{Delay} \text{ subject to } \textit{Area} \leq A_{\text{spec}} \tag{4.27}$$

$$\text{Minimize } \textit{Area} \cdot [\textit{Delay}]^c \tag{4.28}$$

where c is a constant. Of all of these, the first form is perhaps the most useful practical form, since a designer's objective is typically to meet a timing constraint dictated by a system clock.

Depending on the type of analysis, the delay computation may be either deterministic or statistical [6]. In the former case, the problem reduces to a conventional nonlinear optimization problem. In the latter case, the optimization is statistical, though it can be transformed into a deterministic problem.

4.3.2 Delay Modeling

We examine the delay modeling procedure used in TILOS (Timed Logic Synthesizer) [7] at the transistor, gate, and circuit levels. Many transistor sizing algorithms use minor variations on this theme, although some use more exact models. The inherent trade-off here is between obtaining a provably optimal solution under inexact models, versus being uncertain about optimality under more correct models.

4.3.2.1 Transistor Level Model

A MOS transistor is modeled as a voltage-controlled switch with an on-resistance, R_{on}, between drain and source, and three grounded capacitances, C_{d}, C_{s}, and C_{g}, at the drain, source, and gate terminals, respectively, as shown earlier in Figure 4.6. The behaviors of the resistance and capacitances associated with a MOS transistor of channel width x are modeled as

$$R_{\text{on}} \propto 1/x \tag{4.29}$$

$$C_{\text{d}}, C_{\text{s}}, C_{\text{g}} \propto x. \tag{4.30}$$

4.3.2.2 Gate Level Model

At the gate level, delays are calculated in the following manner. For each transistor in a pull-up or pull-down network of a complex CMOS gate, the largest resistive path from the transistor to the gate output is computed, as well as the largest resistive path from the transistor to a supply rail. Thus, for each transistor, the network is transformed into an RC line, and its Elmore time constant [8] is computed and is taken to be the gate delay.

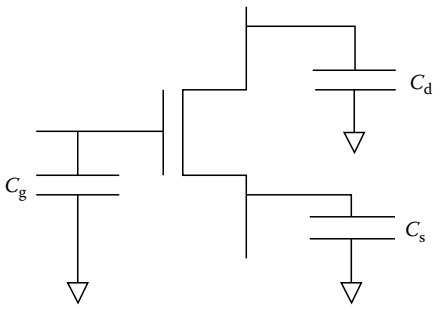


FIGURE 4.6 RC transistor model.

While finding the Elmore delay, the capacitances that lie between the switching transistor and the supply rail are assumed to be at the voltage level of the supply rail at the time of the switching transition, and do not contribute to the Elmore delay. For example, in Figure 4.7, the capacitance at node n_1 is ignored while computing the Elmore delay, the expression for which is

$$(R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3. \tag{4.31}$$

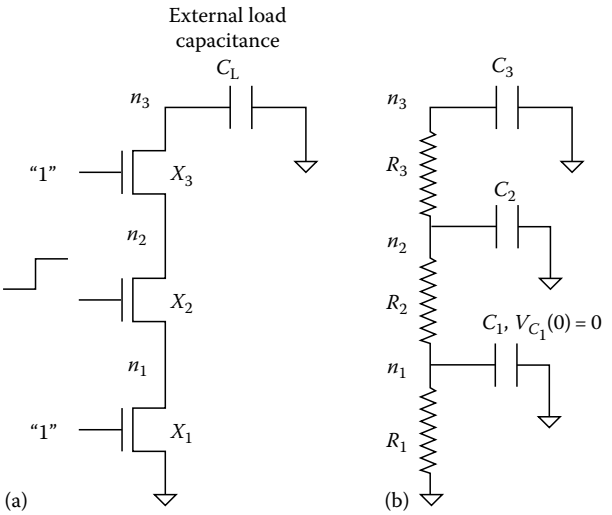


FIGURE 4.7 (a) Sample pull-down network and (b) its RC representation. (From Fishburn, J. and Dunlop, A., *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1985, pp. 326–328. With permission.)

Each R_i is inversely proportional to the corresponding transistor size, x_i, and each C_i is some constant (for wire capacitance) plus a term proportional to the width of each transistor whose gate, drain, or source is connected to node i. Thus, Equation 4.31 can be rewritten as

$$(A/x_1 + A/x_2)(Bx_2 + Cx_3 + D) + (A/x_1 + A/x_2 + A/x_3)(Bx_3 + E),$$

which is a posynomial function (see Section 4.2.2) of x_1, x_2, and x_3. More accurate modeling techniques, using generalized posynomials, are described in Ref. [4].

4.3.2.3 Circuit Level Model

At the circuit level, the program evaluation and review technique (PERT), which will be described shortly, is used to find the circuit delay. The procedure is best illustrated by means of a simple example. Consider the circuit in Figure 4.8, where each box represents a gate, and the number within the box

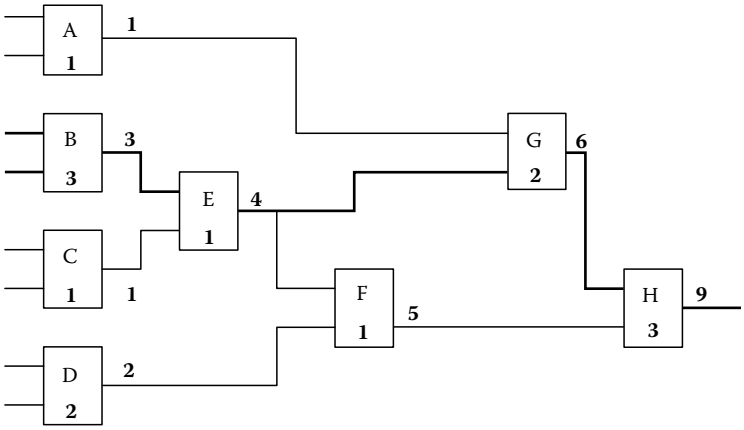


FIGURE 4.8 PERT.

represents its delay. We assume that the worst-case arrival time for a transition at any primary input, i.e., at the inputs to boxes A, B, C, and D, is 0.

A component is said to be ready for processing when the signal arrival time information is available for all of its inputs. Initially, since signal arrival times are known only at the primary inputs, only those components that are fed solely by primary inputs are ready for processing. These are placed in a queue and are scheduled for processing. In the iterative process, the component at the head of the queue is scheduled for processing. Each processing step consists of

- Finding the latest arriving input to the component, which triggers the output transition. This involves finding the maximum of all worst-case arrival times of inputs to the component.
- Adding the delay of the component to the latest arriving input time, to obtain the worst-case transition time at the output.
- Checking all of the components that the current component fans out to, to find out whether they are ready for processing. If so, the component is added to the tail of the queue.

The iterations end when the queue is empty. In the example, the algorithm is executed as follows:

Step 1. Initially, Queue = {A,B,C,D}.

Step 2. Schedule A; $Delay_A = 0 + 1 = 1$. Queue = {B,C,D}.

Step 3. Schedule B; $Delay_B = 0 + 3 = 3$. Queue = {C,D}.

Step 4. Schedule C; $Delay_C = 0 + 1 = 1$. Queue = {D,E}. (E is added to the queue.)

Step 5. Schedule D; $Delay_D = (0 + 2) = 2$. Queue = {E}.

Step 6. Schedule E; $Delay_E = (\max(3,1) + 1) = 4$. Queue = {F,G}.

Step 7. Schedule F; $Delay_F = (\max(4,2) + 1) = 5$. Queue = {G}.

Step 8. Schedule G; $Delay_G = (\max(4,1) + 2) = 6$. Queue = {H}.

Step 9. Schedule H; $Delay_H = (\max(6,5) + 3) = 9$. Queue = {}. The algorithm terminates.

The worst-case delays at the output of each component are shown in Figure 4.8. The critical path, defined as the path between an input and an output with the maximum delay, can now easily be found by successively tracing back, beginning from the primary output with the latest transition time, and walking back along the latest arriving fan-in of the current gate, until a primary input is reached. In the example, the critical path from the input to the output is B-E-G-H.

In the case of CMOS circuits, the rise and fall delay transitions are calculated separately. For inverting CMOS gates, the latest arriving input rise (fall) transition triggers off a fall (rise) transition at the output. This can easily be incorporated into the PERT method described above, by maintaining two numbers, t_r and t_f, for each gate, corresponding to the worst-case rise (high transition) and fall (low transition) delays from a primary input. To obtain the value of t_f at an output, the largest value of t_r at an input node is added to the worst-case fall transition time of the component; the computation of t_r is analogous. For noninverting gates, t_r and t_f are obtained by adding the rise (fall) transition time to the worst-case input rise (fall) transition time.

Since each gate delay is a posynomial, and the circuit delay found by the PERT is a sum of gate delays, the circuit delay is also a posynomial function of the transistor sizes. In general, the path delay can be written as

$$\sum_{i,j=1}^{n} a_{ij} \frac{x_i}{x_j} + \sum_{i=1}^{n} \frac{b_i}{x_i} + K. \tag{4.32}$$

In contrast to deterministic timing, statistical timing analysis techniques model gate delays as statistical functions of the underlying process parameter variations. Instead of propagating delays as fixed numbers, they propagate the probability density functions (PDFs) of delays. Techniques based on Gaussian or non-Gaussian delays, under linear or nonlinear models, with and without correlation, have been presented in the literature [9–12].

4.3.3 Area Model

The exact area of a circuit cannot easily be represented as a function of transistor sizes. This is unfortunate, since a closed functional form facilitates the application of optimization techniques. As an approximation, the following formula is used by many transistor sizing algorithms, to estimate the active circuit area:

$$Area = \sum_{i=1}^{n} x_i \tag{4.33}$$

where
x_i is the size of the ith transistor
n is the number of transistors in the circuit

In other words, the area is approximated as the sum of the sizes of transistors in the circuit that, from the definition Equation 4.8, is clearly a posynomial function of the x_i's.

4.3.4 Sensitivity-Based TILOS Algorithm

4.3.4.1 Steps in the Algorithm

The algorithm that was implemented in TILOS [7] was the first to recognize the fact that the area and delay can be represented as posynomial functions of the transistor sizes. The algorithm proceeds as follows.

An initial solution is assumed where all transistors are at the minimum allowable size. In each iteration, a static timing analysis is performed on the circuit, as explained in Section 4.3.2, to determine the critical path for the circuit.

Let N be the primary output node on the critical path. The algorithm then walks backward along the critical path, starting from N. Whenever an output node of a gate, $Gate_i$, is visited, TILOS examines the largest resistive path between V_{DD} (ground) and the output node if $Gate_i$'s t_r (t_f) causes the timing failure at N. This includes

- Critical transistor, i.e., the transistor whose gate terminal is on the critical path. In Figure 4.7, X_2 is the critical transistor.
- Supporting transistors, i.e., transistors along the largest resistive path from the critical transistor to the power supply (V_{DD} or ground). In Figure 4.7, X_1 is a supporting transistor.
- Blocking transistors, i.e., transistors along the highest resistance path from the critical transistor to the logic gate output. In Figure 4.7, X_3 is a blocking transistor.

TILOS finds the sensitivity, which is the reduction in circuit delay per increment of transistor size, for each critical, blocking, and supporting transistors. (The procedure of sensitivity computation is treated in greater detail shortly.) The size of the transistor with the greatest sensitivity is increased by multiplying it by a constant, BUMPSIZE, a user-settable parameter that defaults to 1.5. The above process is repeated until all constraints are met, implying that a solution is found, or the minimum delay state has been

passed, and any increase in transistor sizes would make it slower instead of faster, in which case TILOS cannot find a solution.

Note that since in each iteration, exactly one transistor size is changed, the timing analysis method can employ incremental simulation techniques to update delay information from the previous iteration. This substantially reduces the amount of time spent in critical path detection.

4.3.4.2 Sensitivity Computation

Figure 4.9 illustrates a configuration in which the critical path extends back along the gate of the upper transistor, which is the critical transistor. The sensitivity for this transistor is calculated as follows: set all transistor sizes, except x, to the size of the critical transistor. R is the total resistance of an RC chain driving the gate and C is the total capacitance of an RC chain being driven by the configuration. The total delay, $D(x)$, of the critical path is

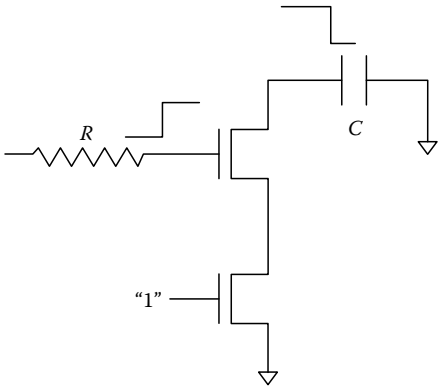


FIGURE 4.9 Sensitivity calculation in TILOS. (From Fishburn, J. and Dunlop, A., *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1985, pp. 326–328. With permission.)

$$D(x) = K + RC_{\mathrm{u}}x + \frac{R_{\mathrm{u}}C}{x} \tag{4.34}$$

where

R_{u} and C_{u} are the resistance and capacitance of a unit-sized transistor, respectively

K is a constant that depends on the resistance of the bottom transistor, the capacitance in the driving RC chain, and the resistance in the driven RC chain

The sensitivity, $D'(x)$, is then

$$D'(x) = RC_{\mathrm{u}} - \frac{R_{\mathrm{u}}C}{x^2}. \tag{4.35}$$

The sensitivities of supporting transistors and blocking transistors can similarly be calculated.

Note that increasing the size of a transistor with negative sensitivity only means that the delay along the current critical path can be reduced by changing the size of this transistor, and does not necessarily mean that the circuit delay can be reduced; the circuit delay is the maximum of all path delays in the circuit, and a change in the size of this transistor could increase the delay along some other path, making a new path critical. This is the rationale behind increasing the size of the most sensitive transistor by only a small factor.

From an optimization viewpoint, the procedure of bumping up the size of the most sensitive transistor could be looked upon in the following way. Let the ith transistor (out of n total transistors) be the one with the maximum sensitivity. Define $\mathbf{e_i} \in \mathbf{R}^n$ as $(\mathbf{e_i})_j = 0$ if $i \neq j$ and $(\mathbf{e_i})_j = 1$ if $i = j$. In each iteration, the TILOS optimization procedure works in the n-dimensional space of the transistor sizes, chooses $\mathbf{e_i}$ as the search direction, and attempts to find the solution to the problem by taking a small step along that direction.

4.3.5 Transistor Sizing Using the Method of Feasible Directions

Shyu et al. proposed a two-stage optimization approach to solve the transistor sizing problem. The delay estimation algorithm is identical to that used in TILOS.

The algorithm can be summarized in the following pseudocode:

```
Use TILOS to size the entire circuit;
While (TRUE) {
    Select G_1, . . . , G_k, the k most critical paths,
    and X = {x_i}, the set of design parameters
    Solve the optimization problem
```

$$\begin{aligned} &\text{minimize} && \sum_{x_i \in X} x_i \\ &\text{such that} && G_i(X) \leq T \ \forall\ i = 1, \ldots, k \\ &\text{and} && x_i \geq \text{minsize} \ \forall\ x_i \in X. \end{aligned}$$

```
    If all constraints are satisfied, exit  }
```

In the first stage, the TILOS heuristic described in Section 4.3.4 is used to generate an initial solution. The heuristic finds a solution that satisfies the constraints, and only the sized-up transistors are used as design parameters. Although TILOS is not guaranteed to find an optimal solution, it can serve as an initial guess solution for an iterative technique.

In the second stage of the optimization process, the problem is converted into a mathematical optimization problem, and is solved by a method of feasible directions (MFD) algorithm described in Section 4.2.3, using the feasible solution generated in the first stage as an initial guess. To reduce the computation, a sequence of problems with a smaller number of design parameters is solved.

At first, the transistors on the worst-delay paths (usually more than one) are selected as design parameters. If, with the selected transistors, the optimizer fails to meet the delay constraints, and some new paths become the worst-delay paths, the algorithm augments the design parameters with the transistors on those paths, and restarts the process. However, while this procedure reduces the run time of the algorithm, one faces the risk of finding a suboptimal solution since only a subset of the design parameters is used in each step.

The MFD optimization method proceeds by finding a search direction **d**, a vector in the n-dimensional space of the design parameters, based on the gradients of the cost function and some of the constraint functions. Once the search direction has been computed, a step along this direction is taken, so that the decrease in the cost and constraint functions is large enough. The computation stops when the length of this step is sufficiently small.

Since this algorithm has the feature that once the feasible region (the set of transistor sizes where all delay constraints are satisfied) is entered, all subsequent improvements will remain feasible, and the algorithm can be terminated at any time with a feasible solution.

4.3.5.1 Practical Implementational Aspects

4.3.5.1.1 Generalized Gradient

For convergence, the MFD requires that the objective and constraint functions be continuously differentiable. However, since the circuit delay is defined as the maximum of all path delays, the delay constraint functions are usually not differentiable. To illustrate that the maximum of two continuously differentiable functions, $g_1(x)$ and $g_2(x)$, need not be differentiable, consider the example in Figure 4.10. The maximum function, shown by the bold lines, is nondifferentiable at x_0.

To cope with the nondifferentiability of the constraint functions, a modification of the MFD is used, which employs the concept of the generalized gradient. The idea is to use a convex combination of the gradients of the active, or nearly active, constraints near a discontinuity.

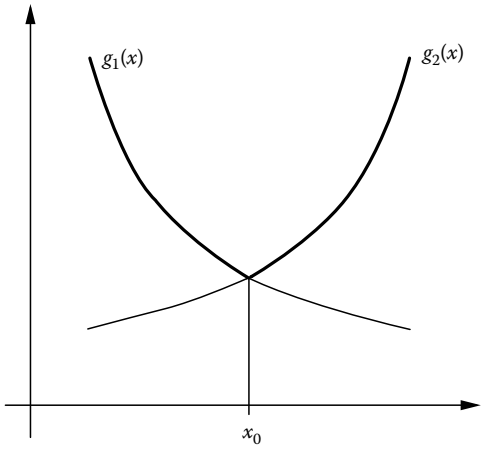


FIGURE 4.10 Nondifferentiability of the max function.

4.3.5.1.2 Scaling

It is important to scale the gradients of the cost and path delay functions since the use of their unscaled values may produce poor descent search directions due to the large difference in magnitude of the gradients of the constraint and objective functions. When a gradient has a magnitude that is much smaller than other gradients, it dominates the search direction. In such a case, the descent direction is unjustly biased away from the other constraints or the cost function.

4.3.6 Lagrangian Multiplier Approaches

As can be seen from the approaches studied so far, the problem of transistor sizing can be formulated as a constrained nonlinear programming problem. Hence, the method of Lagrangian multipliers, described in Section 4.2.3, is applicable.

An early approach by Marple [13] begins with a prespecified layout, and performs the optimization using an area model for that layout. While such an approach has the disadvantage that it may not result in the minimal area over all layouts, it still maintains the feature that the area and delay constraints are posynomials. Apart from the delay constraints, there also exist some area constraints, modeled by constraint graphs that are commonly used in layout compaction. These constraints maintain the minimum spacing between objects in the final layout, as specified by design rules.

The delay of the circuit is modeled by a delay graph, D (V, E) where V is the set of nodes (gates) in D, and E is the set of arcs (connections among gates) in D. This is the same graph on which the PERT analysis is to be carried out. Let m_i represent the worst-case delay at the output of gate i, from the primary inputs. Then for each gate, the delay constraint is expressed as

$$m_i + d_j \leq m_j, \tag{4.36}$$

where gate $i \in$ fan-in (gate j), and d_j is the delay of gate j. Thus, the number of delay constraints is reduced from a number that could, in the worst case, be exponential in $|V|$, to one that is linear in $|E|$, using $|V|$ additional variables.

More recent work [14] recognizes that many of the Lagrange multipliers can be removed to build a relaxed problem, and uses Lagrangian relaxation on the dual to obtain a very fast optimal solution.

4.3.7 Two-Step Optimization

Since the number of variables in the transistor sizing problem, which equals the number of transistors in a combinational segment, is typically too large for most optimization algorithms to handle efficiently, many algorithms choose a simpler route by performing the optimization in two steps. The most successful of these methods is given in Ref. [15], which goes alternately through a D-phase and a W-phase; one of these creates delay budgets for each gate by solving a flow problem, and the other assigns transistor sizes based on these budgets.

4.3.8 Convex Programming-Based Approach

The chief shortcoming of most of the approaches above is that the simplifying assumptions that are made to make the optimization problem more tractable may lead to a suboptimal solution. The algorithm in iCONTRAST solves the underlying optimization problem exactly.

The objective of the algorithm is to solve the transistor sizing problem in Equation 4.28, where both the area and the delay are posynomial functions of the vector **x** of transistor sizes. The procedure described below may easily be extended to solve the formulations in Equations 4.29 and 4.30 as well; however, these formulations are not as useful to the designer. The variable transformation, $(x_i) = (e^{z_i})$, maps the problem in Equation 4.28 to

$$\begin{aligned} &\text{minimize} \quad Area(\mathbf{z}) = \sum_{i=1}^{n} e^{z_i} \\ &\text{subject to} \quad D(\mathbf{z}) \leq T_{\text{spec}}. \end{aligned} \tag{4.37}$$

The delay of a circuit is defined to be the maximum of the delays of all paths in the circuit. Hence, it can be formulated as the maximum of posynomial functions of **x**. This is mapped by the above transformation onto a function $D(\mathbf{z})$ that is a maximum of convex functions; a maximum of convex functions is also a convex function. The area function is also a posynomial in **x**, and is transformed into a convex function by the same mapping. Therefore, the optimization problem defined in Equation 4.28 is mapped to a convex programming problem, i.e., a problem of minimizing a convex function over a convex constraint set. Due to the unimodal property of convex functions over convex sets, any local minimum of Equation 4.28 is also a global minimum.

Vaidya's convex programming method described in Section 4.2.3 is then used to find the unique global minimum of the optimization problem.

4.3.8.1 Gradient Calculations

In an iteration of Vaidya's convex programming algorithm, when the center $\mathbf{z}_c$ of a polytope lies within the feasible region *S*, the gradient of the area function is required to generate the new hyperplane passing through the center. The gradient of the area function (Equation 4.37) is given by

$$\nabla Area(\mathbf{z}) = [e^{z_1}, e^{z_2}, \ldots, e^{z_n}]. \tag{4.38}$$

In the case when the center $\mathbf{z}_c$ lies outside the feasible region *S*, the gradient of the critical path delay function $D_{\text{critpath}}(\mathbf{z}_c)$ is required to generate the new hyperplane that is to be added. Note that transistors in the circuit can contribute to the *k*th component of the gradient of the delay function in either of two ways:

1. If the *k*th transistor is a critical, supporting, or blocking transistor (as defined in Section 4.3.4)
2. If the *k*th transistor is a capacitive load for some critical transistor

Transistors that satisfy neither of these two requirements have no contribution to the gradient of the delay function.

4.3.9 Statistical Optimization

In the presence of statistical variations, it is important to perform timing optimization based on statistical static timing analysis. An obvious way to increase the timing yield of a digital circuit is to pad the specifications to make the circuit robust to variations, i.e., choose a delay specification of the circuit that is tighter than the required delay. This new specification must be appropriately selected to avoid large area or power overheads due to excessively conservative padding.

Early work in Ref. [9] proposes formulation of statistical objective and timing constraints, and solves the resulting nonlinear optimization formulation, but this is highly computational. Other approaches have extended TILOS heuristic to optimize $\mu + 3\sigma$, where μ is the mean of the delay PDF and σ its standard deviation, but while these methods work reasonably in practice, they have no theoretical backing.

More formal optimization approaches have also been used. The work in Ref. [16] presents approaches to optimize the statistical power of the circuit, subject to timing yield constraints under convex formulation of the problem as a second-order conic program. The work in Ref. [17] proposes a gate sizing technique based on robust optimization theory: robust constraints are added to the original constraints set by modeling the intrachip random process parameter variations as Gaussian variables, contained in a constant probability density uncertainty ellipsoid, centered at the nominal values.

4.4 Analog Design Centering Problem

4.4.1 Problem Description

While manufacturing an analog circuit, it is inevitable that process variations will cause design parameters, such as component values, to waver from their nominal values. As a result, the manufactured analog circuit may no longer meet some behavioral specifications, such as requirements on the delay, gain, and bandwidth, that it has been designed to satisfy. The procedure of design centering attempts to select the nominal values of design parameters to move the circuit to the center of the feasible region.

The values of n design parameters may be ordered as an n-tuple that represents a point in $\mathbf{R}^n$. A point is feasible if the corresponding values for the design parameters satisfy the behavioral specifications on the circuit. The feasible region (or the region of acceptability), $R_f \subset \mathbf{R}^n$, is defined as the set of all design points for which the circuit satisfies all behavioral specifications.

The random variations in the values of the design parameters are modeled by a PDF, $\Phi(\mathbf{z})$: $\mathbf{R}^n \rightarrow [0, 1]$, with a mean corresponding to the nominal value of the design parameters. The yield of the circuit, Y, as a function of the mean, $\mathbf{x}$, is given by

$$Y(\mathbf{x}) = \int_{R_f} \Phi_{\mathbf{x}}(\mathbf{z}) d\mathbf{z}. \tag{4.39}$$

The design center is the point $\mathbf{x}$ at which the yield, $Y(\mathbf{x})$, is maximized. There have traditionally been two approaches to solving this problem: one based on geometrical methods and another based on statistical sampling. In addition, several methods that hybridize these approaches also exist. We now provide an exposition of geometrical approaches to the optimization problem of design centering.

A common assumption made by geometrical design centering algorithms is that R_f is a convex-bounded body. Geometrical algorithms recognize that the evaluation of the integral (Equation 4.39) is computationally difficult, and generally proceed as follows: the feasible region in the space of design parameters, i.e., the region where the behavioral specifications are satisfied, is approximated by a known geometrical body, such as a polytope or an ellipsoid. The center of this body is then approximated, and is taken to be the design center.

4.4.2 Simplicial Approximation Method

4.4.2.1 Outline of the Method

The simplicial approximation method [18] is a method for approximating a feasible region by a polytope and finding its center. This method proceeds in the following steps:

1. Determine a set of $m \geq n + 1$ points on the boundary of R_f.
2. Find the convex hull (see Section 4.2.2) of these points and use this polyhedron as the initial approximation to R_f. In the two-dimensional example in Figure 4.11a, the points 1, 2, and 3 are chosen in Step 1, and their convex hull is the triangle with vertices 1, 2, and 3. Set $k = 0$.

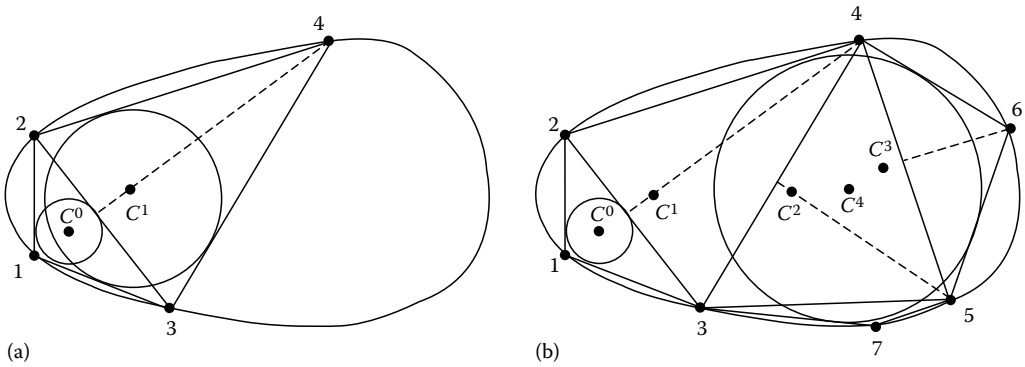


FIGURE 4.11 Simplicial approximation method. (From Director, S.W. and Hachtel, G.D., *IEEE Trans. Circuits Syst.*, CAS-24, 363, 1977. With permission.)

3. Inscribe the largest n-dimensional hypersphere in this approximating polyhedron and take its center as the first estimate of the design center. This process involves the solution of a linear program. In Figure 4.11a, this is the hypersphere C^0.
4. Find the midpoint of the largest face of the polyhedron, i.e., the face in which the largest $n-1$-dimensional hypersphere can be inscribed. In Figure 4.11a, the largest face is 2–3, the face in which the largest one-dimensional hypersphere can be inscribed.
5. Find a new boundary point on R_f by searching along the outward normal of the largest face found in Step 4 extending from the midpoint of this face. This is carried out by performing a line search. In Figure 4.11a, point 4 is thus identified.
6. Inflate the polyhedron by forming the convex hull of all previous points, plus the new point generated in Step 5. This corresponds to the quadrilateral 1, 2, 3, and 4 in Figure 4.11a.
7. Find the center of the largest hypersphere inscribed in the new polyhedron found in Step 6. This involves the solution of a linear program. Set $k = k + 1$, and go to Step 4. In Figure 4.11a, this is the circle C^1.

Further iterations are shown in Figure 4.11b. The process is terminated when the sequence of radii of the inscribed hypersphere converges.

4.4.2.2 Inscribing the Largest Hypersphere in a Polytope

Given a polytope specified by Equation 4.3, if the $\mathbf{a}_i$'s are chosen to be unit vectors, then the distance of a point $\mathbf{x}$ from each hyperplane of the polytope is given by $r = \mathbf{a}_i^{\mathrm{T}}\mathbf{x} - b_i$.

The center $\mathbf{x}$ and radius r of the largest hypersphere that can be inscribed within the polytope $\mathcal{P}$ are then given by the solution of the following linear program:

$$\begin{array}{ll} \text{minimize} & r \\ \text{subject to} & \mathbf{a}_i^{\mathrm{T}}\mathbf{x} - r \geq b_i \end{array} \tag{4.40}$$

Since the number of unknowns of this linear program is typically less than the number of constraints, it is more desirable to solve its dual [1]. A similar technique can be used to inscribe the largest hypersphere in a face of the polytope.

This method has also been generalized for the inscription of maximal norm bodies, to handle joint PDFs with (nearly) convex level contours.

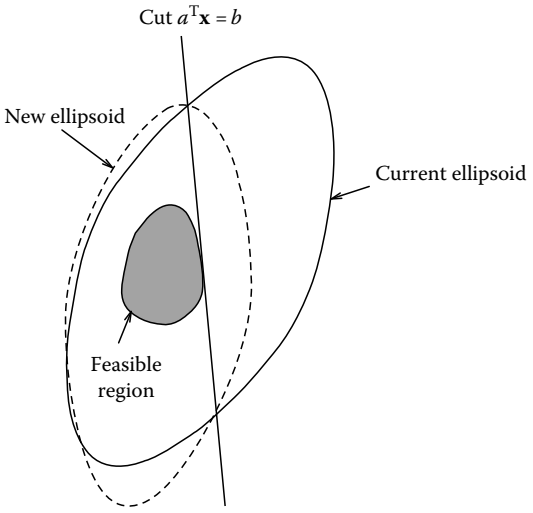


FIGURE 4.12 Ellipsoidal method. (From Abdel-Malek, H.L. and Hassan, A.-K.S.O., *IEEE Trans. Comput. Aided Des.*, 10, 1006, 1991. With permission.)

4.4.2.3 Elongated Feasible Regions

If the design centering procedure outlined earlier is applied to a rectangular feasible region, the best possible results may not be obtained by inscribing a hypersphere. For elongated feasible regions, it is more appropriate to determine the design center by inscribing an ellipsoid rather than a hypersphere. Simplicial approximation handles this problem by scaling the axes so that the lower and upper bounds for each parameter differ by the same magnitude, and one may inscribe the largest ellipsoid by inscribing the largest hypersphere in a transformed polytope. This procedure succeeds in factoring in reasonably the fact that feasible regions may be elongated; however, it considers only a limited set of ellipsoids, which have their axes aligned with the coordinate axis, as candidates for inscription within the polytope.

4.4.3 Ellipsoidal Method

This method [19] is based on principles similar to those used by the Shor–Khachiyan ellipsoidal algorithm for linear programming. This algorithm attempts to approximate the feasible region by an ellipsoid, and takes the center of the approximating ellipsoid as the design center. It proceeds by generating a sequence of ellipsoids, each smaller than the last, until the procedure converges. Like other methods, this procedure assumes that an initial feasible point is provided by the designer. The steps involved in the procedure are as follows (Figure 4.12):

1. Begin with an ellipsoid, E_0, that is large enough to contain the desired solution. Set $j = 0$.
2. From the center of the current ellipsoid, choose a search direction, and perform a binary search to identify a boundary point along that direction. One convenient set of search directions are the parameter directions, searching along the ith, $i = 1, 2, \ldots, n$ in a cycle, and repeating the cycle, provided the current ellipsoid center is feasible. If not, a linear search is conducted along a line from the current center to the given feasible point.
3. A supporting hyperplane [1] at the boundary point can be used to generate a smaller ellipsoid, E_{j+1}, that is guaranteed to contain the feasible region R_f, if R_f is convex. The equation of E_{j+1} is provided by an update procedure.
4. Increment j, and go to Step 1 unless the convergence criterion is met. The convergence criterion is triggered when the volume is reduced by a given factor, ε. Upon convergence, the center of the ellipsoid is taken to be the design center.

4.4.4 Convexity-Based Approaches

4.4.4.1 Introduction

This section describes the approach in Ref. [20]. Here, the feasible region is first approximated by a polytope in the first phase. Next, two geometrical approaches to find the design center are proposed. In the first, the properties of polytopes are utilized to inscribe the largest ellipsoid within the approximating polytope. The second method proceeds by formulating the design centering problem as a convex

programming problem, assuming that the variations in the design parameters are modeled by Gaussian probability distributions, and uses Vaidya's convex programming algorithm described in Section 4.2.3 to find the solution.

4.4.4.2 Feasible Region Approximation

The feasible region, $R_f \subset \mathbf{R}^n$, is approximated by a polytope given by Equation 4.3 in this step. The algorithm begins with an initial feasible point, $\mathbf{z}_0 \in R_f$. An n-dimensional box, namely, $\{\mathbf{z} \in \mathbf{R}^n \mid z_{min} \leq z_i \leq z_{max}\}$, containing R_f is chosen as the initial polytope $\mathcal{P}_0$. In each iteration, n orthogonal search directions, $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$ are chosen (possible search directions include the n coordinate directions). A binary search is conducted from $\mathbf{z}_0$ to identify a boundary point $\mathbf{z}_{\mathbf{b}i}$ of R_f, for each direction $\mathbf{d}_i$. If $\mathbf{z}_{\mathbf{b}i}$ is relatively deep in the interior of $\mathcal{P}$, then the tangent plane to R_f at $\mathbf{z}_{\mathbf{b}i}$ is added to the set of constraining hyperplanes in Equation 4.3. A similar procedure is carried out along the direction $-\mathbf{d}_i$. Once all of the hyperplanes have been generated, the approximate center of the new polytope is calculated. Then $\mathbf{z}_0$ is reset to be this center, and the above process is repeated.

Therefore, unlike simplicial approximation that tries to expand the polytope outward, this method starts with a large polytope and attempts to add constraints to shrink it inward. The result of polytope approximation on an ellipsoidal feasible region is illustrated in Figure 4.13.

4.4.4.3 Algorithm I: Inscribing the Largest Hessian Ellipsoid

For a polytope given by Equation 4.3, the log-barrier function is defined as

$$F(\mathbf{z}) = -\sum_{i=1}^{m} \log_e(\mathbf{a_i}^T\mathbf{z} - b_i). \tag{4.41}$$

The Hessian ellipsoid centered at a point $\mathbf{x}$ in the polytope $\mathcal{P}$ is defined as the ellipsoid E($\mathbf{x}$,$\mathcal{H}(\mathbf{x})$,1) (see Equation 4.2), where $\mathcal{H}(\mathbf{x}_c)$ is the Hessian [1] of the log-barrier function above, and is given by

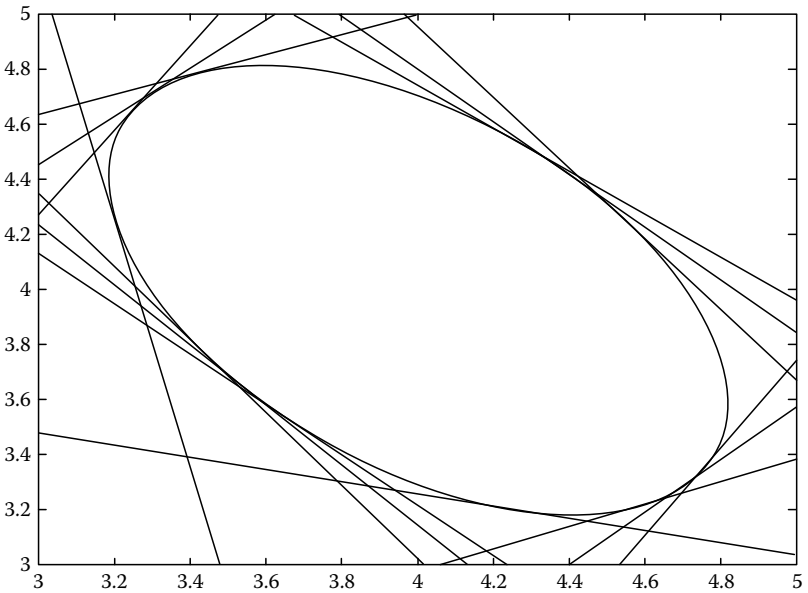


FIGURE 4.13 Polytope approximation for the convexity-based methods. (From Saptnekar, S.S., Vaidya, P.M. and Kang, S.M., *IEEE Trans. Comput. Aided Des.*, 13, 1536, 1994. With permission.)

$$\mathcal{H}(\mathbf{z}) = \nabla^2 F(\mathbf{z}) = \sum_{i=1}^{m} \frac{\mathbf{a_i}\mathbf{a_i}^{\mathrm{T}}}{(\mathbf{a_i}^{\mathrm{T}}\mathbf{z} - b_i)^2}. \tag{4.42}$$

This is known to be a good approximation to the polytope locally around $\mathbf{x}$.

Hence, the goal is to find the largest ellipsoid in the class $\mathbf{E}(\mathbf{x}, \mathcal{H}(\mathbf{x}),\mathrm{r})$ that can be inscribed in the polytope, and its center $\mathbf{x}_c$. The point $\mathbf{x}_c$ will be taken to be the computed design center. An iterative process is used to find this ellipsoid. In each iteration, the Hessian at the current point $\mathbf{x}_k$ is calculated, and the largest ellipsoid $\mathbf{E}(\mathbf{x}, \mathcal{H}(\mathbf{x}_k),\mathrm{r})$ is inscribed in the polytope. The inscription of this ellipsoid is equivalent to inscribing a hypersphere in a transformed polytope. The process of inscribing a hypersphere in a polytope is explained in Section 4.4.2.

4.4.4.4 Algorithm II: Convex Programming Approach

When the PDFs that represent variations in the design parameters are Gaussian in nature, the design centering problem can be posed as a convex programming problem.

The joint Gaussian PDF of n independent random variables $\mathbf{z} = (z_1, \ldots, z_n)$ with mean $\mathbf{x} = (x_1, \ldots, x_n)$ and variance $\sigma = (\sigma_1, \ldots, \sigma_n)$ is given by

$$\Phi_{\mathbf{x}}(\mathbf{z}) = \frac{1}{(2\pi)^{n/2}\sigma_1\sigma_2\cdots\sigma_n} \exp\left[\sum_{i=0}^{i=n} -\frac{(z_i - x_i)^2}{2\sigma_i^2}\right] \tag{4.43}$$

This is known to be a log-concave function of $\mathbf{x}$ and $\mathbf{z}$. Also, note that arbitrary covariance matrices can be handled, since a symmetric matrix may be converted into a diagonal form by a simple linear (orthogonal) transformation. The design centering problem is now formulated as

$$\begin{aligned} &\text{maximize} \quad \mathrm{Y}(\mathbf{x}) = \int_{\mathcal{P}} \Phi_{\mathbf{x}}(\mathbf{z})\mathrm{d}\mathbf{z} \\ &\text{such that} \quad \mathbf{x} \in \mathcal{P}. \end{aligned} \tag{4.44}$$

where $\mathcal{P}$ is the polytope approximation to the feasible region R_f. It is a known fact that the integral of a log-concave function over a convex region is also a log-concave function. Thus, the yield function $\mathrm{Y}(\mathbf{x})$ is log-concave, and the above problem reduces to a problem of maximizing a log-concave function over a convex set. Hence, this can be transformed into a convex programming problem. The convex programming algorithm in Section 4.2.3 is then applied to solve the optimization problem.

4.4.5 Concluding Remarks

The above list of algorithms is by no means exhaustive, but provides a general flavor for how optimization methods are used in geometrical design centering. The reader is referred to [21–24] for further information about statistical design. In conclusion, it is appropriate to list a few drawbacks associated with geometrical methods:

Limitations of the approximating bodies: In the case of ellipsoidal approximation, certain convex bodies cannot be approximated accurately, because an ellipsoid is symmetric about any hyperplane passing through its center, and is inherently incapable of producing a good approximation to a body that has a less symmetric structure. A polytope can provide a better approximation to a convex body than an ellipsoid since any convex body can be thought of as a polytope with an infinite number of faces. However, unlike the ellipsoidal case, calculating the exact center of a polytope is computationally difficult and one must resort to approximations.

Statistical effects are ignored: Methods such as simplicial approximation, ellipsoidal approximation, and Algorithm I of the convexity-based methods essentially approximate the feasible region by means of an ellipsoid, and take the center of that ellipsoid to be the design center, regardless of the probability distributions that define variations in the design parameters. However, the design center could be highly dependent on the exact probability distributions of the variables, and would change according to these distributions.

Nonconvexities: Real feasible regions are seldom convex. While in many cases, they are nearly convex, there are documented cases where the feasible region is not very well behaved. In a large number of cases of good designs, since the joint PDF of the statistical variables decays quite rapidly from the design center, a convex approximation does not adversely affect the result. However, if the nominal design has a very poor yield, a convex approximation will prove to be inadequate.

The curse of dimensionality: Geometrical methods suffer from the so-called curse of dimensionality, whereby the computational complexity of the algorithm increases greatly with the number of variables. However, as noted in Ref. [25], for local circuit blocks within a die, performance variations in digital MOS circuits depend on only four independent statistical variables. Moreover, for this class of circuits, the circuit performance can be modeled with reasonable accuracy by linear functions. For such circuits, the deterministic (i.e., nonstatistical) algorithm in Ref. [25] is of manageable complexity.

References

1. D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd edn. Reading, MA: Addison-Wesley, 1984.
2. R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, A survey of optimization techniques for integrated-circuit design, *Proceedings of the IEEE*, 69: 1334–1362, October 1981.
3. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.
4. K. Kasamsetty, M. Ketkar, and S. S. Sapatnekar, A new class of convex functions for delay modeling and their application to the transistor sizing problem, *IEEE Transactions on Computer-Aided Design*, 19: 779–788, July 2000.
5. S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, An exact solution to the transistor sizing problem for CMOS circuits using convex optimization, *IEEE Transactions on Computer-Aided Design*, 12: 1621–1634, November 1993.
6. S. S. Sapatnekar, *Timing*, Boston, MA: Springer, 2004.
7. J. Fishburn and A. Dunlop, TILOS: A posynomial programming approach to transistor sizing, in *Proceedings of the IEEE International Conference on Computer-Aided Design*, Santa Clara, CA, pp. 326–328, 1985.
8. J. Rubenstein, P. Penfield, and M. A. Horowitz, Signal delay in RC tree networks, *IEEE Transactions on Computer-Aided Design*, CAD-2: 202–211, July 1983.
9. E. Jacobs and M. Berkelaar, Gate sizing using a statistical delay model, in *Design, Automation, and Test in Europe*, Paris, France, pp. 283–290, 2000.
10. H. Chang and S. S. Sapatnekar, Statistical timing analysis considering spatial correlations using a single PERT-like traversal, in *Proceedings of the IEEE International Conference on Computer-Aided Design*, San Jose, CA, pp. 621–625, 2003.
11. Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma, Correlation-aware statistical timing analysis with non-Gaussian delay distributions, in *Proceedings of the ACM/IEEE Design Automation Conference*, Anaheim, CA, pp. 77–82, 2005.
12. J. Singh and S. S. Sapatnekar, Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis, in *Proceedings of the ACM/IEEE Design Automation Conference*, San Francisco, CA, pp. 155–160, 2006.

13. D. Marple, Performance optimization of digital VLSI circuits, Technical Report No. CSL-TR-86-308, Stanford University, Stansford, CA, October 1986.
14. C.-P. Chen, C. C.-N. Chu, and M. D.-F. Wong, Fast and exact simultaneous gate and wire sizing by Lagrangian relaxation, *IEEE Transactions on Computer-Aided Design*, 18: 1014–1025, July 1999.
15. V. Sundararajan, S. S. Sapatnekar, and K. K. Parhi, Fast and exact transistor sizing based on iterative relaxation, *IEEE Transactions on Computer-Aided Design*, 21: 568–581, May 2002.
16. M. Mani, A. Devgan, and M. Orshansky, An efficient algorithm for statistical power under timing yield constraints, in *Proceedings of the ACM/IEEE Design Automation Conference*, Anaheim, CA, pp. 309–314, 2005.
17. J. Singh, V. Nookala, Z.-Q. Luo, and S. Sapatnekar, Robust gate sizing by geometric programming, in *Proceedings of the ACM/IEEE Design Automation Conference*, Anaheim, CA, pp. 315–320, 2005.
18. S. W. Director and G. D. Hachtel, The simplicial approximation approach to design centering, *IEEE Transactions on Circuits and Systems*, CAS-24 (7): 363–372, 1977.
19. H. L. Abdel-Malek and A.-K. S. O. Hassan, The ellipsoidal technique for design centering and region approximation, *IEEE Transactions on Computer-Aided Design*, 10: 1006–1014, August 1991.
20. S. S. Sapatnekar, P. M. Vaidya, and S. M. Kang, Convexity-based algorithms for design centering, *IEEE Transactions on Computer-Aided Design*, 13: 1536–1549, December 1994.
21. S. W. Director, P. Feldmann, and K. Krishna, Statistical integrated circuit design, *IEEE Journal of Solid-State Circuits*, 28: 193–202, March 1993.
22. M. D. Meehan and J. Purviance, *Yield and Reliability in Microwave Circuit and System Design*, Boston, MA: Artech House, 1993.
23. S. W. Director, W. Maly, and A. J. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*, Boston, MA: Kluwer Academic, 1990.
24. R. Spence and R. S. Soin, *Tolerance Design of Integrated Circuits*, Reading, MA: Addison-Wesley, 1988.
25. D. E. Hocevar, P. F. Cox, and P. Yang, Parametric yield optimization for MOS circuit blocks, *IEEE Transactions on Computer-Aided Design*, 7: 645–658, June 1988.

5

Statistical Design Optimization

5.1 Introduction **5**-1
5.2 Problems and Methodologies of Statistical Circuit Design **5**-1
5.3 Underlying Concepts and Techniques **5**-2
Circuit Variables, Parameters, and Performances • Statistical Modeling of Circuit (Simulator) Variables • Acceptability Regions • Methods of Acceptability Region Approximation • Manufacturing (Parametric) Yield
5.4 Statistical Methods of Yield Optimization **5**-12
Large-Sample vs. Small-Sample Methods • Methods Using Standard Deterministic Optimization Algorithms • Large-Sample Heuristic Methods for Discrete Circuits • Large-Sample, Derivative-Based Methods for Discrete Circuits • Large-Sample, Derivative-Based Method for Integrated Circuits • Small-Sample Stochastic Approximation-Based Methods • Small-Sample Stochastic Approximation Methods for Integrated Circuits • Case Study: Process Optimization for Manufacturing Yield Enhancement • Generalized Formulation of Yield, Variability, and Taguchi Circuit Optimization Problems
5.5 Conclusion **5**-30
References **5**-31

Maciej A. Styblinski
Texas A&M University

5.1 Introduction

Manufacturing process variations and environmental effects (such as temperature) result in the variations of the values of circuit elements and parameters. Statistical methods of circuit design optimization take those variations into account and apply statistical (or statistical/deterministic) optimization techniques to obtain an ''optimal'' design. Statistical design optimization belongs to a general area of statistical circuit design.

5.2 Problems and Methodologies of Statistical Circuit Design

A broad class of problems exists in this area: statistical analysis involves studying the effects of element variations on circuit performance. It applies statistical techniques, such as Monte Carlo simulation [34] and the variance propagation method [39], to estimate variability of performances. Design centering attempts to find a center of the acceptability region [12] such that manufacturing yield is maximized.

Direct methods of yield optimization use yield as the objective function and utilize various statistical (or mixed statistical/deterministic) algorithms to find the yield maximum in the space of designable circuit/process parameters. Design centering and tolerance assignment (used mostly for discrete circuits) attempt to find the design center, with simultaneous optimal assignment of circuit element tolerances, minimizing some suitable cost function and providing 100% yield (worst-case design) [4,6]. To solve this problem, mostly deterministic algorithms of nonlinear programming are used. Worst-case design is often too pessimistic and too conservative, leading to substantial overdesign. This fact motivates the use of statistical techniques, which provide a much more realistic estimation of the actual performance variations and lead to superior designs. Stringent requirements of the contemporary very large scale integration (VLSI) design prompted a renewed interest in the practical application of these techniques. The most significant philosophy introduced recently in this area is statistical Design for Quality (DFQ). It was stimulated by the practical appeal of the DFQ methodologies introduced by Taguchi [30], oriented toward "on-target" design with performance variability minimization. In what follows, mostly the techniques of manufacturing yield optimization and their design for quality generalization are discussed.

5.3 Underlying Concepts and Techniques

5.3.1 Circuit Variables, Parameters, and Performances

5.3.1.1 Designable Parameters

Designable parameters, represented by the n-dimensional vector* $x = (x_1, \ldots, x_n)$, are used by circuit designers as "decision" variables during circuit design and optimization. Typical examples are nominal values of passive elements, nominal MOS transistor mask dimensions, process control parameters, etc.

5.3.1.2 Random Variables

The t-dimensional vector of random variables (or "noise" parameters in Taguchi's terminology [30]) is denoted as $\theta = (\theta_1, \ldots, \theta_t)$. It represents statistical R, L, C element variations, disturbances or variations of manufacturing process parameters, variations of device model parameters such as t_{ox} (oxide thickness), V_{TH} (threshold voltage), and environmental effects such as temperature, supply voltage, etc. Usually, θ represents principal random variables, selected to be statistically independent and such that all other random parameters can be related to them through some statistical models. Probability density function (p.d.f.) of θ parameters will be denoted as $f_\theta(\theta)$.

5.3.1.3 Circuit (Simulator) Variables

These variables represent parameters and variables used in circuit, process, or system simulators such as SPICE. They are represented as the c-dimensional vector $e = (e_1, \ldots, e_c)$. Specific examples of e variables are: R, L, C elements, gate widths W_j and lengths L_j of MOS transistors, device model parameters, or, if a process simulator is used, process-related control, and physical and random parameters available to the user. The e vector contains only those variables that are directly related to the x and θ vectors.† This relationship is, in general, expressed as

$$e = e(x, \theta) \tag{5.1}$$

The p.d.f. of θ is transformed into $f_e(e)$ the p.d.f. of e. This p.d.f. can be singular, i.e., defined in a certain subspace of the e-space (see examples below). Moreover, it can be very complicated, with highly nonlinear statistical dependencies between different parameters, so it is very difficult to represent it directly as

* Vectors are denoted by lowercase letters subscripts or superscripts.
† This means that, e.g., some SPICE parameters will always be *fixed*.

a p.d.f. of e. In the majority of cases, the analytic form of $f_e(e)$ is not known. For that reason, techniques of statistical modeling are used (see the next section).

5.3.1.4 Circuit Performances

The vector of circuit performance function (or simply performances) is defined as the m-dimensional vector $y = (y_1, \ldots, y_m)$. Its elements can be gain, bandwidth, slew rate, signal delay, and circuit response for a single frequency or time, etc. Each of the performances y_j is a function of the vector of circuit elements e: $y_j = y_j(e) = y_j(e(x, \theta))$. These transformations are most often not directly known in analytical form and a circuit simulator (such as SPICE) must be used to find the values of y_j's corresponding to the given values of x and θ. The overall simulator time required for the determination of all the performances y can be substantial for large circuits. This is the major limiting factor for the practical application of statistical circuit design techniques. To circumvent this problem, new statistical macromodeling techniques are being introduced [31] (see example in Section 5.4.9).

5.3.2 Statistical Modeling of Circuit (Simulator) Variables

Statistical modeling is the process of finding a suitable transformation $e = e(x, \theta)$, such that given the distribution of θ, the distribution of e can be generated. The transformation $e = e(x, \theta)$ can be described by closed form analytical formulas or by a computer algorithm.

For discrete active RLC circuits (e.g., such as a common emitter amplifier), vector e is composed of two parts: the first part contains the actual values of statistically perturbed RLC elements: i.e., $e_i = x_i + \theta_i$ for those elements. In this formula, θ represents absolute element spreads and its expected (average) value, $E\{\theta\} = 0$ and x_i is the nominal value of e_i, often selected as the expected value of e_i. This implies that the variance of e_i, $\text{var}\{e_i\} = \text{var}\{\theta_i\}$; $E\{e_i\} = x_i$ and the distribution of e_i is the same as that of θ_i, with the expected value shifted by x_i. Alternatively, if θ_i represents relative element spreads, $e_i = x_i (1 + \theta_i)$, where $E\{\theta_i\} = 0$. Therefore, $E\{e_i\} = x_i$; $\text{var}\{e_i\} = x_i^2\ \text{var}\{\theta_i\}$, i.e., the standard deviations σ_{ei} and $\sigma_{\theta i}$ are related: $\sigma_{ei} = x_i\ \sigma_{\theta i}$, or $\sigma_{ei}/E\{e_i\} = \sigma_{ei}/x_i = \sigma_{\theta i}$. This means that with fixed $\sigma_{\theta i}$, the relative standard deviation of e_i is constant, as it is often the case in practice, where standard deviations of RLC elements are described in percents of the element nominal values. Both forms of e_i indicate that each e_i is directly associated with its corresponding θ_i and x_i, and that there is one-to-one mapping between e_i and θ_i. These dependencies are important, since many of the yield optimization algorithms were developed assuming that $e_i = x_i + \theta_i$. A typical p.d.f. for discrete elements is shown in Figure 5.1, before "binning" into different categories (the whole curve) and after binning into ±1%, ±5%, and ±10% resistors (the shaded and white areas: e.g., the ±10% resistors will have the distribution characterized by the external shaded areas, with a ±5% "hole" in the middle).

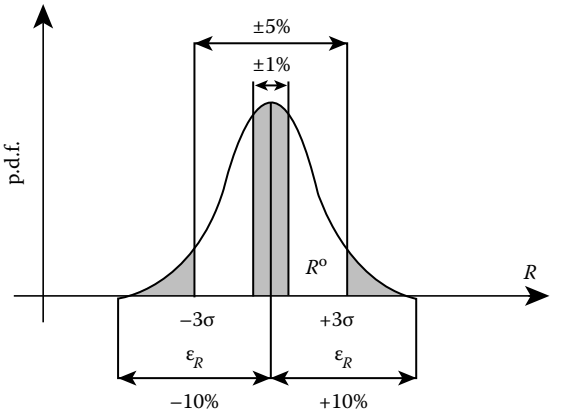


FIGURE 5.1 Typical probability density function (p.d.f.) of a discrete resistor before and after "binning."

Usually, passive discrete elements are statistically independent* as shown in Figure 5.2.

* But, for instance, if R_L (R_C) is a loss resistance of an inductor L (capacitor C) then L and R_L (C and R_C) are correlated.

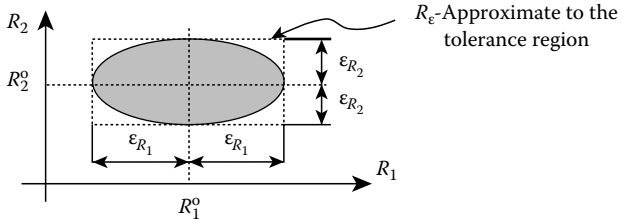


FIGURE 5.2 Level set (cross section) of a p.d.f. function for two discrete resistors after manufacturing.

The cross-section shown is often called a level set, a norm body [12], or a tolerance body. It is defined as a set of element values for which the element p.d.f. is larger than a prescribed value.

This value is selected such that the probability of the element values falling into the tolerance body is equal to e.g., 95%, (i.e., the tolerance body represents 95% of the entire element population). Alternatively, the tolerance body can be represented as a (hyper-) box shown in the figure, with the sides equal to $2\varepsilon_i$ ($2\varepsilon_{R_i}$ in the figure), called the tolerance region and denoted by R_ε. Figure 5.2 also shows that the dependence $e_i = x_i + \theta_i$ is equivalent in this case to $R_i = R_{\mathrm{NOM},i} + \Delta R_i$, where $x_i = R_{\mathrm{NOM},i}$; $\theta = \Delta R_i$).

The second part of e is composed of the parameters representing active device (e.g., BJT, MOS) model parameters. They are usually strongly correlated within each device model (same applies to ICs), but typically, no correlations occur between device model parameters of different devices. Each of the device model parameters e_d is related through a specific model $e_\mathrm{d} = e_\mathrm{d}(x, \theta)$ to the vector θ parameters,* representing principal random variables, which are themselves often some device model parameters (such as oxide thickness t_{ox} of MOS transistors), and/or are some dummy random variables. For example, in the BJT empirical statistical model introduced in [3], the base transient time T_n is modeled as follows:

$$e_\mathrm{d} = e_\mathrm{d}(x_\mathrm{d}, \theta_1, \theta_2) = T_n(\beta, X_{r5}) = \left(a + \frac{b}{\sqrt{\beta}}\right)(1 + cX_{r5}) \tag{5.2}$$

i.e., it is the function of the current gain $\theta_1 = \beta$ (the principal random variable, affecting the majority of the BJT model parameters) and $\theta_2 = X_{r5}$ (a dummy random variable, uniformly distributed in the interval $[-1, 1]$ independent of β and having no physical meaning); a, b, and c are empirically selected constants (Figure 5.3b).

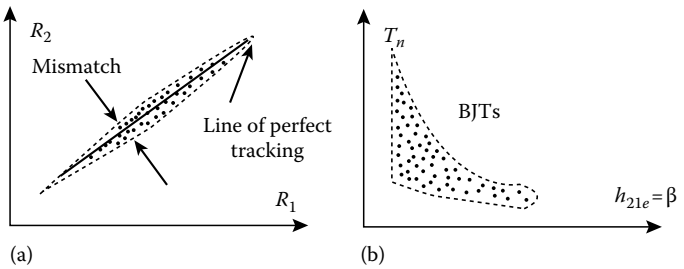


FIGURE 5.3 Dependencies between passive IC elements and between device model parameters: (a) linear, between two resistors R_1 and R_2 and (b) nonlinear, between the base transit time T_n and the current gain $h_{21e} = \beta$ for bipolar junction transistors.

* Observe that these models are parameterized by x: e.g., the MOS transistor model parameters e_i will also depend on the device length L and width W.

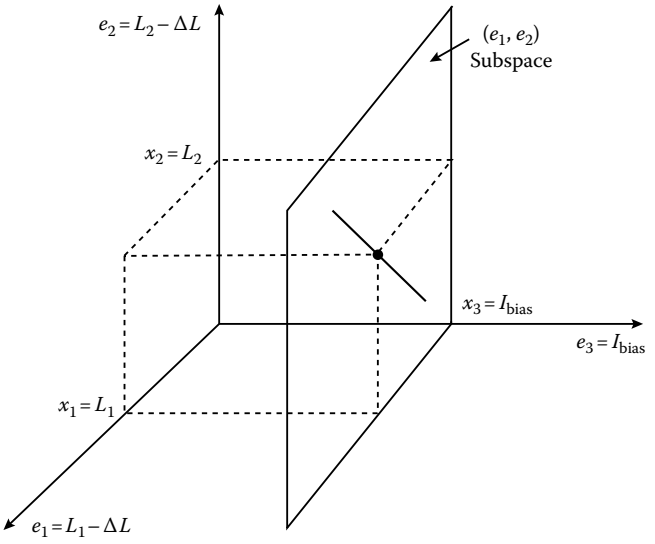


FIGURE 5.4 Singular distribution $f_e(e_1, e_2, e_3)$ in three-dimensional e-space, represented by the thick line in the (e_1, e_2)-subspace (plane). This is due to the perfect matching of $\Delta L \equiv \Delta L_1 \equiv \Delta L_2$ values.

In the discrete circuit case, each designable parameter x_i has a random variable θ_i added to it (i.e., x and θ in the same space in which the p.d.f. $f_e(e) = f_\theta(x + \theta)$ is defined.

For integrated circuits, the passive elements (R's, C's), active dimensions (such as Ws and Ls for MOS transistors), and other transistor model parameters are strongly correlated (see Figure 5.3a). Because of this, the distribution of θ parameters is limited to a certain subspace of the entire e-space, i.e., $f_e(e(x, 0))$ is singular, and the formula $e_i = x_i + \theta_i$ does not hold. As an example, consider a subset of all e parameters representing the gate lengths $x_1 = L_1$, $x_2 = L_2$ (designable parameters) of two MOS transistors T_1, T_2 and an independent designable parameter $x_3 + I_{bias}$, representing the transistor bias current. Assume also (as is frequently the case) that $\theta \equiv \Delta L$, the technological gate length reduction (a common random parameter) changes the same way for T_1 and T_2, i.e., there is ideal matching of $\Delta L \equiv \Delta L_1 = \Delta L_2$,* i.e., $e_1 + x_1 + \theta + L_1 - \Delta L$, $e_2 + x_2 + \theta + L_2 - \Delta L$, $e_3 + x_3 + I_{bias}$. The only random variable in this model is ΔL, as shown in Figure 5.4. The p.d.f. $f_e(e)$ is in this case defined only in the one-dimensional subspace of the e-space (i.e., all the realization of the vector e are located on the thick line in Figure 5.4). The major consequence of this difference is that, in general, many yield optimization algorithms developed for discrete circuits (i.e., using the explicit assumption that $(e_i = e_i + \Delta_i)$, cannot be used for IC design.

Statistical modeling for ICs is concerned with representing global variations and parameter mismatches† occurring between different devices on the same chip. In the last example, mismatch can be modeled by two additional independent local mismatch variables ε_{L_1}, ε_{L_2}, representing small random local deviations of ΔL_1 and ΔL_2 from the ideal value ΔL, i.e., $e_1 + x_1 + \theta + \varepsilon_i + L_1 - \Delta L - \varepsilon_{L_1}$, $e_2 + x_2 + \theta + \varepsilon_i + L_2 - \Delta L - \varepsilon_{L_2}$. With this model, ΔL represents a global or common factor, affecting both channel lengths, and ε_{L_1}, ε_{L_2}, represented *specific* (local) variations. The model just presented is a simple (simple, one-factor) case of the Factor Analysis (FA) [20] model of correlations between e_1 and e_2.

In [8], FA analysis, together with Principal Component Analysis (PCA)‡ and nonlinear regression were used for the determination of linear and nonlinear statistical models for CMOS transistor parameters. The following common factors $F_1, \ldots, F_8$ were identified: t_{ox} (oxide thickness common to n- and p-type

* Such a model is commonly used for digital ICs [9], since it is of sufficient accuracy for digital applications.

† Element mismatches are very important for analog ICs.

‡ Principal component analysis involves coordinate rotation and leads to *uncorrelated principal components*.

transistors) $N_{SUB,n}$, $N_{SUB,p}$ (n- and p-type substrate doping), ΔL_n, ΔL_p (length reduction), ΔW_n, ΔW_p (width reduction—for narrow transistors only), and XJ_p (p-type junction depth). These variables were causing about 96% of the total variability of all parameters. All the other CMOS transistor model parameters were related to the $F_1, \ldots, F_8$ factors through quadratic (or linear in simplified models) regression formulas. The resulting models were able to represent—with a high level of accuracy—the strong nonlinear statistical dependencies existing between some model parameters.

The major theoretical and empirical findings in mismatch modeling are the device area (or length) and distance dependencies of the mismatch level (see [23], where references to other authors can be found). For example, the variance of the difference $e_1 - e_2$ between two MOS transistor model parameters e_1, e_2 is modeled as [3,9]

$$\text{var}(e_1 - e_2) = \frac{a_p}{2W_1L_1} + \frac{a_p}{2W_2L_2} + s_p^2 d_{12}^2 \tag{5.3}$$

where

W_1, L_1, W_2, and L_2 are widths and lengths of the two transistors
d_{12} is the distance between the transistor centers
a_p and s_p are empirical coefficients adjusted individually for each model parameter

Using this model together with PCA, and introducing other concepts, two quite sophisticated linear statistical models in the form $e_i = e(x_i, \theta, W_i, L_i, d)$ were proposed in [23]. They include the transistor separation distance information, collected into the vector d in two different forms. The models, constructed from on-chip measured data, were used for practical yield optimization. θ parameters were divided into two groups: a group of correlated random variables responsible for the common part of each parameter variance and correlations between model parameters of each *individual* transistor, and the second group of local (mismatched related) random variables, responsible for mismatches between different transistors. Additional dependencies, related to transistor spacing and device area related coefficients, maintain proper mismatch dependencies.

5.3.3 Acceptability Regions

The acceptability region A_y is defined as a region in the space of performance parameters y (y-space), for which all inequality and equality constraints imposed on y are fulfilled. In the majority of cases, A_y is a hyperbox, i.e., all the constraints are of the form: $S_j^L \le y_j \le S_j^U$; $j = 1, \ldots, m$, where S_j^L and S_j^U are the (designer defined) lower and upper bounds imposed on y_j, called also designer's specifications. More complicated specifications, involving some relations between y_i parameters, or S_j^L and S_j^U bounds can also be defined.

For the simplest case of the y-space box-constraints, the acceptability region A in the e-space is defined as such a set of e vectors in the c-dimensional space, for which all inequalities $S_j^L \le y_j(e) \le S_j^U, j = 1, \ldots, m$, are fulfilled. Illustration of this definition is shown in Figure 5.5. It can be interpreted as the mapping of the acceptability region A_y from the y-space into the e-space. Acceptability regions A can be very complicated: they can be nonconvex and can contain internal infeasible regions (or "holes"), as shown in Figure 5.6 for a simple active RC filter [25].

For discrete circuits, A is normally represented in the e-space, due to the simple relationship $e_i = x_i + \theta_i$ (or $e_i = x_i(1 + \theta_i)$), between e_i, x_i, and θ_i. For integrated circuits, e is related to x and θ through the statistical model, x and θ are in different spaces (or subspaces), the dimension of θ is lower than the dimension e, and the p.d.f. $f_e(e)$ is singular and usually unknown in analytic form. For these reasons, it is more convenient to represent A in the joint (x, θ)-space, as shown in Figure 5.7. For a fixed x, A can be defined in the θ-space and labeled as $A\theta\ (x)$, since it is parameterized by the actual values of x, as shown in Figure 5.7. The shape and location of $A\theta\ (x)$ change with x, as shown. For a fixed x, $A\theta\ (x)$ is defined as

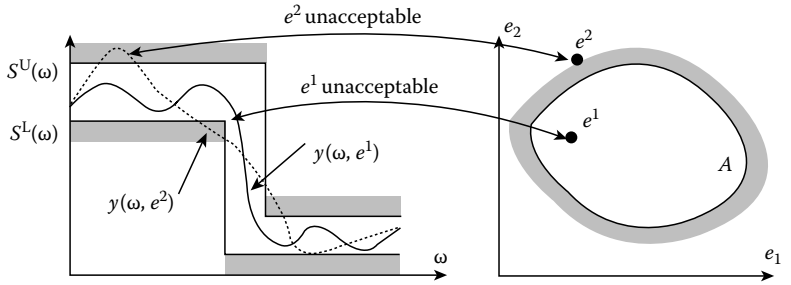


FIGURE 5.5 Illustration of the mapping of the $S^L(\omega)$, $S^U(\omega)$ constraints imposed on $y(\omega, e)$ into the e-space of circuit parameters.

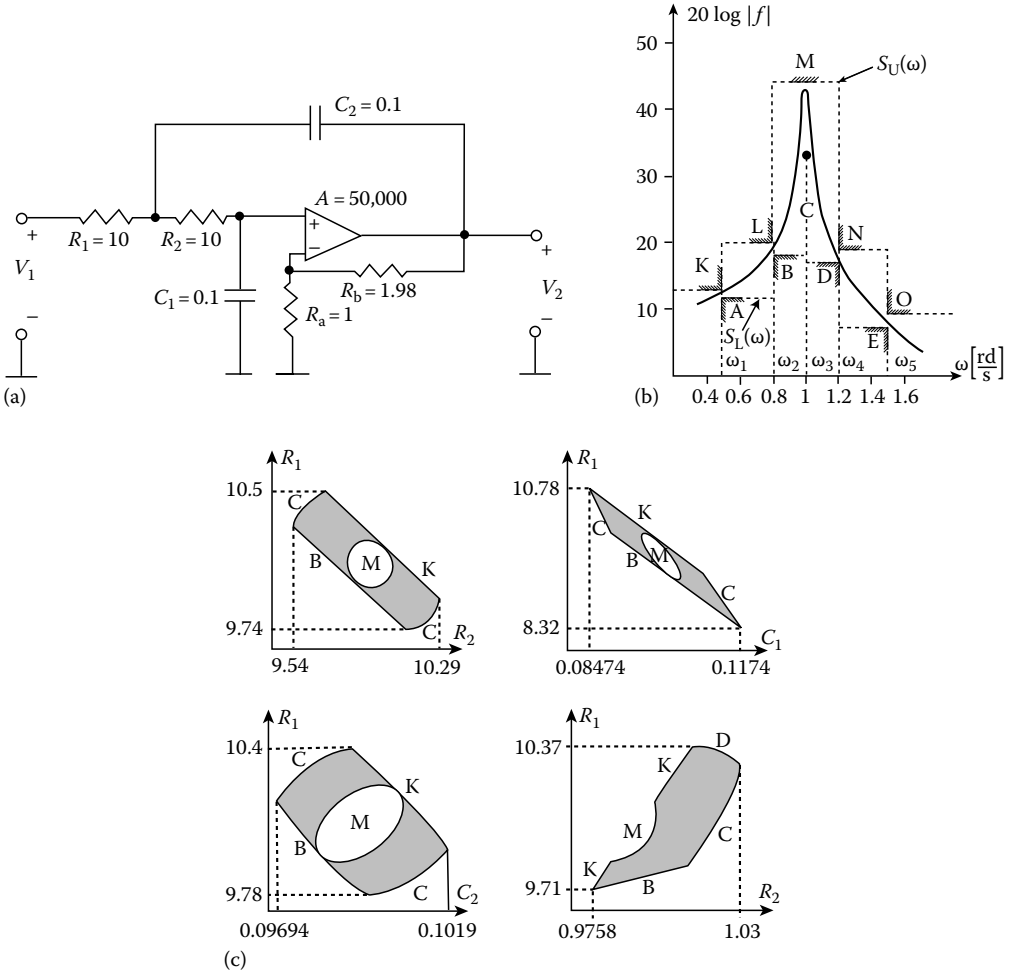


FIGURE 5.6 (a) Sallen–Key active filter, (b) lower and upper bounds imposed in the filter frequency response, and (c) two-dimensional cross sections of the acceptability region A. Capital letters in (b) and (c) indicate the correspondence of constraints to the boundaries of A.

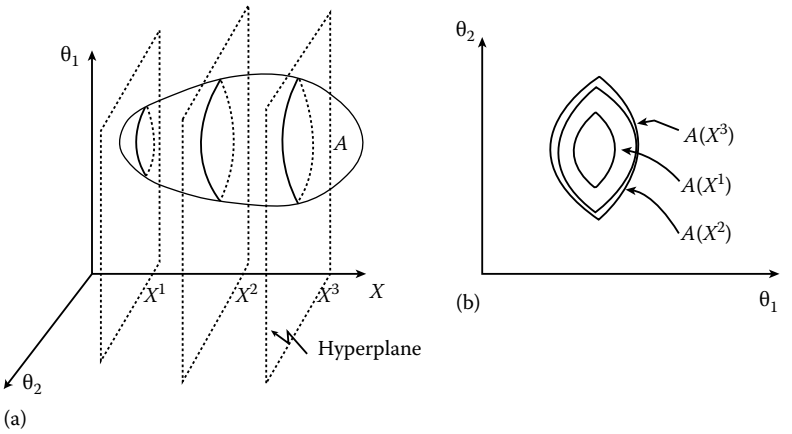


FIGURE 5.7 Acceptability region for integrate circuits: (a) in the joint (x, θ)-space and (b) in the θ-space, parameterized by different values of x. The hyperplanes shown represent t-dimensional subspaces of θ parameters.

such a region in the t-dimensional θ space, for which all the inequalities $S_j^L \leq y_j(e(x, \theta)) \leq S_j^U; j = 1, \ldots, m$, are fulfilled.

In order to recognize if a given point $e(x, \theta)$ in the circuit parameter space belongs to A [or $A\theta$ (x)], an indicator function $\phi(\cdot)$ is introduced:

$$\phi(e(x,\theta)) = \begin{cases} 1 & \text{if } e(x,\theta) \text{ belongs to } A \text{ (a successful, or ''pass point'')} \\ 0 & \text{otherwise (a ''fall point'')} \end{cases} \tag{5.4}$$

A complementary indicator function $\phi_F(e(x, \theta)) = \phi(e(x, \theta)) - 1$ is equal to 1 if $e(x, \theta)$ does not belong to A and 0 otherwise. Both indicator functions will be used in what follows.

5.3.4 Methods of Acceptability Region Approximation

Except for some simple cases, the acceptability region A in the e (or the joint (x, θ))-space is unknown and it is impossible to fully define it. For yield optimization and other statistical design tasks and implicit or explicit knowledge of A and/or its boundary is required. If only the points belonging to A are stored, this can be considered a point-based ''approximation'' to A. Some of the point-based methods are Monte Carlo-based design centering, centers of gravity method, point-based simplicial approximation and yield evaluation (see below), ''parametric sampling-based'' yield optimization [36], yield optimization with ''reusable'' points [39], and others.

The acceptability segment-based method of the A-region approximation was called in [27] a one-dimensional orthogonal search (ODOS) technique leading to several yield optimization methods [25,46]. Its basic principle is shown in Figure 5.8b, where line segments passing through the points e^i randomly sampled in the e-space and parallel to the coordinate axes, are used for the approximation of A. ODOS is very efficient for large linear circuits, since the intersections with A can be directly found from analytical formulas. The two-dimensional cross-sections of A, shown in Figure 5.6, were obtained using this approach. The surface integral-based yield and yield gradient estimation and optimization method proposed in [13] also use the segment approximation to the boundary of A. A variant of this method is segment approximation in one direction, as shown in Figure 5.8c. This method was then extended to plane and hyperplane approximation to A in [42] (Figure 5.8d). In another approach, called ''radial exploration of space'' in [54], the segments approximating A are in radial directions, as shown in Figure 5.8e.

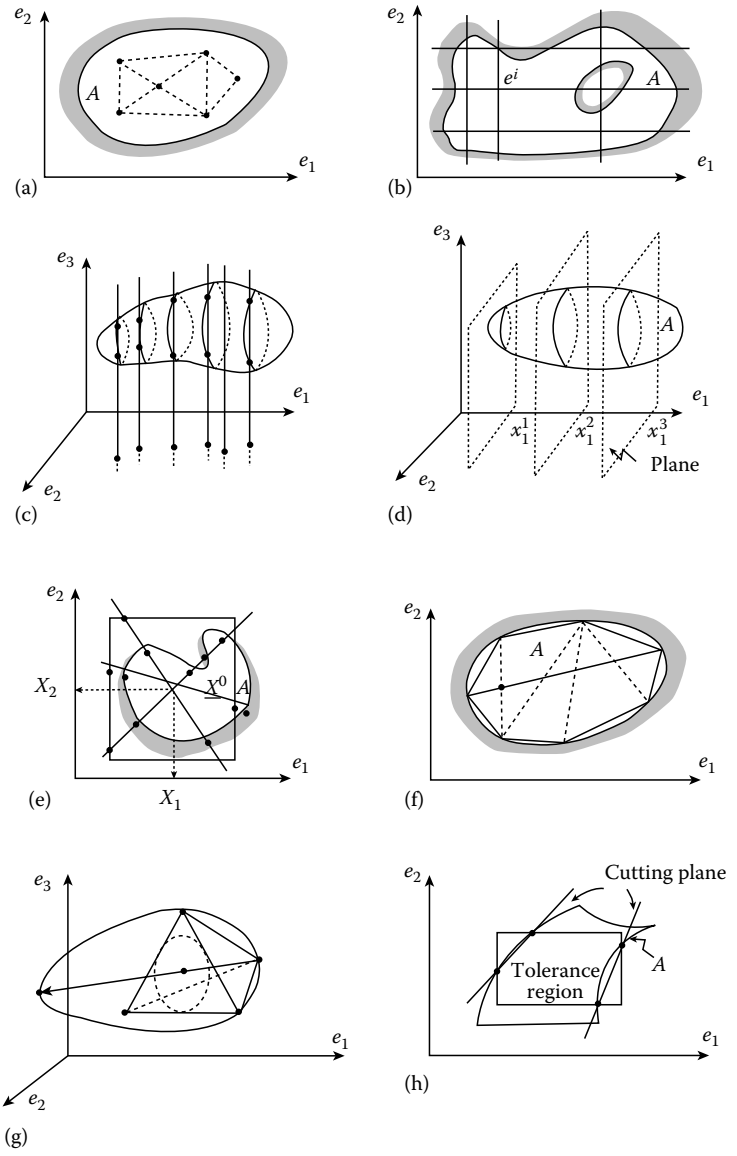


FIGURE 5.8 Various methods of acceptability region approximation: (a) ''point-based'' simplicial approximation to the A-region, (b) segment approximation to A in all directions along e_i axes, (c) segment approximation to A in one direction, (d) (hyper)-plane approximation A, (e) segment approximation to A in radial directions, (f) simplicial approximation to A in two dimensions, (g) simplicial approximation to A in three dimensions, and (h) cutting-plane approximation to A. *Note:* All A-regions are shown in e-space, for the discrete case, i.e., for $e = x + \theta$.

The techniques just described rely on the fact that ''segment'' yields (calculated in the e subspace) can be, for some special cases [27,54], calculated more efficiently than using a standard Monte Carlo method. This leads to higher efficiency and accuracy of yield estimation, in comparison to the point-based yield estimation.

The simplicial approximation proposed in [10] (described in more detail in Section 4.4), is based on approximating the boundary of A in the e-space, by a polyhedron, i.e., by the union of those partitions of a set of c-dimensional hyperplanes which lie inside of the boundary of A or on it. The boundary of

A-region is assumed to be convex (see Figure 5.8f and g). The approximating polyhedron is a convex hull of points. Simplicial approximation is obtained by locating points on the boundary of A, by a systematic expansion of the polyhedron. The search for next vertex is always performed in the direction passing through the center of the largest face of the polyhedron already existing. In the MC-oriented "point-based" version of the method [11] (see Figure 5.8a), subsequent simplicial approximations $\tilde{A}_i$ to A are not based on the points located on the boundary of A (which is computationally expensive) but on the points e^i belonging to A already generated during the MC simulation; after each new point is generated, it is checked if e^i belongs to $\tilde{A}^{i-1}$, (where $\tilde{A}^{i-1}$ is the latest simplicial approximation to A); if yes, the sampled point is considered successful *without* performing the circuit analysis; if e^i does not belong to $\tilde{A}^{i-1}$ and the circuit analysis reveals that e^i belongs to A, the polyhedron is expanded to include this point. Several versions of this general approach, leading to the substantial reduction of the computational effort of yield estimation, are described in the original article.

The method of "cuts" proposed in [5], creates a "cutting-plane" approximation to A in the corners of the tolerance region R_ε assumed to be a hypercube, as shown in Figure 5.8h. The method was combined with discretization of the p.d.f. $f_e(e)$ and multidimensional quadratic approximation to circuit constraints, leading to simplified yield and yield derivative formulas and yield optimization for arbitrary statistical distributors [1].

5.3.5 Manufacturing (Parametric) Yield

Manufacturing yield is defined as the percentage of the total number of products manufactured that fulfill both functional and parametric performance requirements.* Functional circuit performance is the circuit ability to perform desired functions. Catastrophic (or "hard") circuit failures (such as shorts or open circuit faults caused, e.g., by particular wafer contamination) will completely eliminate some of the circuit functions, thus decreasing the part of the overall yield called functional yield. Parametric circuit performance is a measure of circuit quality and is represented by measurable performance functions such as gain, delay, bandwidth, etc., constituting the y-parameter vector. The part of yield related to parametric circuit performance is called parametric yield and is the only type of yield considered in what follows. The actual manufacturing yield is smaller than the parametric yield, since it is equal to the product of the functional or parametric yield [12]. In general, parametric yield is used during circuit electrical design, while functional yield is used during circuit layout design.† Both are used to predict and optimize yield during circuit design.

Parametric yield, therefore, is equal to the percentage of circuits that fulfill all parametric requirements, i.e., it is equal to the probability that e belongs to the acceptability region A. So, it can be calculated as the integral of the p.d.f. of e, $f_e(e)$ over A, for a given vector of designable parameters x. Since $e = e(x, \theta)$ is a function of x, then e, $f_e(e) = f_e(e, x)$ (e.g., $E\{e\}$ and $\mathrm{var}\{e\}$ can be both functions of x). Therefore,‡

$$Y(x) = P\{e \in A\} = \int_A f_e(e, x)\mathrm{d}e = \int_{R^c} \phi(e) f_e(e, x)\mathrm{d}e = E_e\{\phi(e)\} \tag{5.5}$$

where

$P\{\cdot\}$ denotes probability

$\phi(e)$ is the indicator function (Equation 5.4)

$E_e\{\cdot\}$ is expectation with respect to the random variable e

---

* For more detailed yield definitions involving different types of yield, e.g., design yield, wafer yield, probe yield, processing yield, etc., see [12].

† Layout design (i.e., transistor spacing, location, size) has also influence on parameter variations and mismatch, as discussed in Section 5.3.

‡ Multiple integration performed below is over the acceptability region A, or over the entire c-dimensional space R^c of real numbers. ε means "belongs to."

The above formula is useful if $f_e(e, x)$ is a nonsingular p.d.f., which is usually the case for discrete circuits, for which $e_i = x_i + \theta_i$, (or $e_i = x_i\ (1 + \theta_i)$). In a general case, however (e.g., for integrated circuits), the p.d.f. $f_e(e, x)$ is not known, since it has to be obtained from a complicated transformation $e = e\ (x, \theta)$, given the p.d.f. $f_\theta(\theta)$ of θ. Therefore, it is more convenient to integrate directly in the θ-space. Since parametric yield is also the probability that θ belongs to $A\theta(x)$ (the acceptability region in the θ-space for any fixed x), yield becomes

$$\begin{aligned} Y(x) = P\{\theta \in A_\theta(x)\} &= \int_{A_{\theta(x)}} f_\theta(\theta)\mathrm{d}\theta \\ &= \int_{R^t} \phi(e(x,\theta))f_\theta(\theta)\mathrm{d}\theta = E_\theta\{\phi(e(x,\theta))\} \end{aligned} \tag{5.6}$$

Equation 5.6 is general, and is valid for both discrete and integrated circuits. An unbiased estimator of $E_\theta\{\rho(e(x, \theta))\} + E_\theta\{\rho(\theta)\}$ (for fixed x), is the arithmetic mean, based on N points θ^i, sample in θ-space with the p.d.f. $f_\theta(\theta)$, for which the function $\phi(\theta^i)$ is calculated (this involves circuit analyses). Thus, the yield estimator $\hat{Y}$ is expressed as

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} \phi\left(\theta^i\right) = \frac{N_s}{N} \tag{5.7}$$

where N_S is the number of successful trials, i.e., the number of circuits for which $\theta \in A\theta\ (x)$ (all circuit constraints are fulfilled). Integral of Equation 5.6 is normally calculated using Monte Carlo (MC) simulations [34] and Equation 5.7. The MC method is also used to determine statistical parameters of the p.d.f. $f_y(y)$ of $y = y(\mathrm{x}, \theta)$. In order to sample the θ parameters with p.d.f. $f_\theta(\theta)$, special numerical procedures, called random number generators, are used. The basic MC algorithm is as follows:

1. Set $i = 0$, $N_S = 0$ (i is the current index of a sampled point and N_S is the total number of successful trials).
2. Substitute $i = i + 1$, generate the ith realization of $\theta : \theta^i = (\theta_1^i, \ldots, \theta_t^i)$, with the p.d.f. $f_\theta(\theta)$.
3. Calculate the ith realization of $y^i = (y_1^i, \ldots, y_m^i) = y(x, \theta^i)$, with the aid of an appropriate circuit analysis program, and store the results.
4. Check if all circuit constraints are fulfilled, i.e., if $S^L \leq y^i \leq S^U$; if yes, set $N_S = N_S + 1$.
5. If $i \neq N$, go to (2); otherwise, find the yield estimator $\hat{Y} = N_S/N$. If needed, find also some statistical characteristics of y-parameters (e.g., create histograms of y, find statistical moments of y, etc.).

To generate θ^i's with the p.d.f. $f_\theta(\theta)$, the uniformly distributed random numbers are generated first and then transformed to $f_\theta(\theta)$. The most typical random number generator (r.n.g.), generating a sequence of pseudorandom, uniformly distributed integers θ_k in the interval $[0, M]$ (M is an integer), is a multiplicative r.n.g., using the formula [34]: $\theta_{k+1} = c\theta_k \pmod M$ where c is an integer constant and $\theta_k \pmod M$ denotes a remainder from dividing θ_k by M. The initial value θ_0 of θ_k is called the ''seed'' of the r.n.g., and, together with c, should usually be chosen very carefully, to provide good quality of the random sequence generated. Several other r.n.g.s are used in practice [34]. The r_k numbers in the [0, 1) interval are obtained from $r_k = \theta_k/M$. Distributions other than uniform are obtained through different transformations of the uniformly distributed random numbers, such as the inverse of the cumulative distribution function [34]. To generate correlated normal variables $\theta = (\theta_1, \ldots, \theta_n)$, with a given covariance matrix K^θ, the transformation $\theta = CZ$ is used, where $Z = (z_1, z_2, \ldots, z_n)$ is the vector of independent normal variables with $E\{z_i\} = 0$, $\mathrm{var}\{z_i\} = 1$, $i = 1, \ldots, n$, and C is a matrix obtained from the so-called

Cholesky decomposition of the covariance matrix K^{θ} such that $K^{\theta} = CC^{t}$ where t denotes transposition. C is usually lower or upper triangular and can be easily constructed from a given matrix K^{θ} [34].

The yield estimator $\hat{Y}$ is a random variable, since performing different, independent MC simulations, one can expect different values of $\hat{Y} = N_S/N$. As a measure of $\hat{Y}$ variations, variance or standard deviation of $\hat{Y}$ can be used. It can be shown [7] that the standard deviation of $\hat{Y}$, is equal to $\sigma_{\hat{Y}} = Y(1-Y)/N$, i.e., it is proportional to $1/\sqrt{N}$ Hence, to decrease the error of $\hat{Y}$ 10 times, the number of samples has to be increased 100 times. This is a major drawback of the MC method. However, the accuracy of the MC method (measured by $\sigma_{\hat{Y}}$) is independent of the dimensionality of the θ-space, which is usually a drawback of other methods of yield estimation. One of the methods of variance reduction of the $\hat{Y}$ estimator is importance sampling [7,34,39]. Assume that instead of sampling θ with the p.d.f. $f_\theta(\theta)$, some other p.d.f. $g_\theta(\theta)$ is used. Then,

$$Y = \int_{R'} \phi[e(\theta)] \frac{f_\theta(\theta)}{g_\theta(\theta)} g_\theta(\theta)\, d\theta \equiv E\left\{ \phi[e(\theta)] \frac{f_\theta(\theta)}{g_\theta(\theta)} \right\} \tag{5.8}$$

where $g_\theta(\theta) \neq 0$ if $\phi(\theta) = 1$. Yield Y can now be estimated as

$$\tilde{Y} = \frac{1}{N} \sum_{i=1}^{N} \phi\left[e(\theta^i)\right] \frac{f_\theta(\theta^i)}{g_\theta(\theta^i)} \tag{5.9}$$

sampling N points θ^i with the p.d.f. $g_\theta(\theta)$. The variance of this estimator is var $\{\tilde{Y}\} = E\{[\phi(\theta)/g_\theta(\theta) - Y]^2\}/N$. If it is possible to choose $g_\theta(\theta)$ such that it mimics (or is similar to) $\phi(\theta)f_\theta(\theta)/Y$, the variability of $[\phi(\theta)f_\theta(\theta)/g_\theta(\theta) - Y]$ is reduced, and thus the variance of $\hat{Y}$. This can be accomplished if some approximation to $\phi(\theta)$ i.e., to the acceptability region A is known. Some possibilities of using importance sampling techniques were studied, e.g., in [16]. One of such methods, called parametric sampling was used in [36], and other variants of important sampling were used in [2,40] for yield optimization. There are several other methods of variance reduction, such as the method of control variates, correlated sampling, stratified sampling, antithetic variates, and others [7,34,39]. Some of them have been used for statistical circuit design [7,39].

5.4 Statistical Methods of Yield Optimization

The objective of the yield optimization is to find a vector of designable parameters $x = x_{opt}$, such that $Y(x_{opt})$ is maximized. This is illustrated in Figure 5.9 for the case of discrete circuits where $e_1 = x_1 + \theta_1$, $e_2 = x_2 + \theta_2$.* Figure 5.9a corresponds to low initial yield proportional to the weighted (by the p.d.f. $f_e(e)$) area (hypervolume, in general), represented by the dark shaded part of the tolerance body shown. Figure 5.9b corresponds to optimized yield, obtained by shifting the nominal point (x_1, x_2) to the vicinity of the geometric center of the acceptability region A. Because of this geometric property, approximate yield maximization can often be accomplished using methods called deterministic or geometrical design centering. They solve the yield maximization problem indirectly (since yield is not the objective function optimized), using a geometrical concept of maximizing the distance from x_{opt} to the boundary of A. The best known method of this class [10,12] inscribes the largest hypersphere (or other norm-body) into the simplicial approximation to the boundary of A, shown in Figure 5.8g. This approach is described in more detail in Section 4.4, together with other geometrical design centering methods.

A class of methods known as "performance-space oriented design centering methods" is also available. These methods attempt to maximize the scaled distances of the performances y_j from the lower S_j^L

* Alternatively, the model $e_i = x_i(1 + \theta_i)$ can be used, in which case the size of the tolerance body (see Section 5.3) will increase proportionally to x_i.

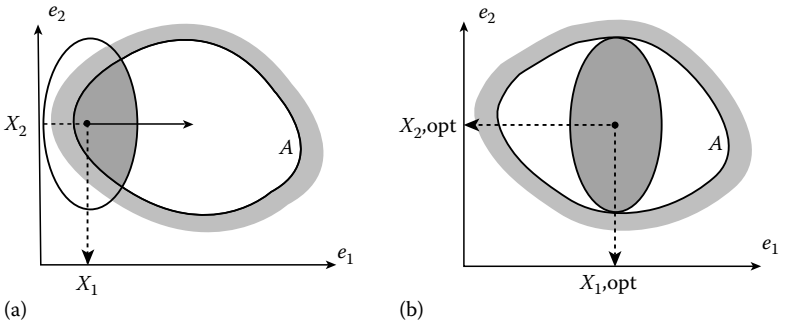


FIGURE 5.9 Interpretation of yield maximization for discrete circuits: (a) initial (low) yield and (b) optimized yield.

and/or upper S_j^U specifications, leading to approximate design centering. Some of these methods are also used for variability minimization and performance tuning, which are the most important tenets of design for quality [30], to be briefly discussed later in this section.

The major feature of statistical yield optimization methods, referred to as statistical design centering, is statistical sampling in either θ-space only or in both θ and x spaces. Sampling can be also combined with some geometrical approximation to the A-region, such as the segment, or radial-segment approximation.

In dealing with various statistical methods of yield optimization, the type of the transformation $e = e(x, \theta)$ from the θ-space to the circuit-parameter space e, has to be considered. The early, mostly heuristic yield optimization methods were based on the simple additive model $e_i = x_i + \theta_i$, valid for discrete circuits. Because of that, the majority of these methods cannot be used for IC or manufacturing process optimization, where θ and x in different spaces, or subspaces, and the distribution of θ is defined over some subspace of the e-space, i.e., it is singular.

The type of the statistical yield optimization algorithm to be used in practice strongly depends on the type of the circuit to be considered and the information available, namely: whether the transformation $e = e(x, \theta)$ is a simple one: $e_i = x_i + \theta_i$ (or $e_i = x_i(1 + \theta_i)$)(for discrete circuits) or general: $e = e(x, \theta)$ (for ICs); whether the values of $\gamma_j(x, \theta)$ for given x and θ only or also the derivatives of y_j with respect to x_k and/or θ_s are available from the circuit simulator; whether some approximation $\tilde{y}(x, \theta)$ to $y(x, \theta)$ is available—either with respect to (w.r.t.) θ only (for a fixed x), or w.r.t. both x and θ; whether the analytical form of $f_\theta(\theta)$ is known and $f_\theta(\theta)$ is differentiable w.r.t. θ, or only samples θ^i of θ are given (obtained from a numerical algorithm or from measurements), so, analytical forms of $f_\theta(\theta)$ and its derivatives w.r.t. θ are not known.

Different combinations of the cases listed above require different optimization algorithms. The more general a given algorithm is, the largest number of cases it is able to cover, but simultaneously it can be less efficient than specialized algorithms covering only selected cases. An ideal algorithm would be the one that is least restrictive and could use the minimum necessary information, i.e., it could handle the most difficult case characterized by the general transformation $e = e(x, \theta)$, with the values of $y = y(x, \theta)$ only available (generated from a circuit simulator) but without derivatives, no approximation to $y(x, \theta)$ available, and unknown analytic form of $f_\theta(\theta)$ (only the samples of θ given). Moreover, under these circumstances such an algorithm should be reasonably efficient even for large problems, and with the presence of some additional information, should become more efficient. The selected yield optimization algorithms discussed below fulfill the criteria of the algorithm ''optimality'' to a quite different level of satisfaction. It has to be stressed that due to different assumptions made during the development of different algorithms and the statistical nature of the results, an entirely fair evaluation of the actual algorithm efficiency is very difficult, and is limited to some specific cases only. Therefore, in what follows, no algorithm comparison is attempted.

5.4.1 Large-Sample vs. Small-Sample Methods

Yield optimization is concerned with the maximization of the regression function $Y(x) = E_\theta\{\phi(x, \theta)\}$ with respect to (w.r.t.) x (see Equation 5.6). In solving general problems of this type, $\phi(\cdot)$ is replaced by an arbitrary function $w(\cdot)$. Large-sample methods of optimizing $E_\theta\{w(x, \theta)\}$ calculate the expectation (average) of w (and/or its gradient) w.r.t. θ for each $x^0, x^1, x^2, \ldots$ from a large number of θ^i samples. Therefore, the averages used in a specific optimization procedure are relatively accurate, following to take relatively large steps $x^{k+1} - x^k$. On the other hand, small-sample methods use just a few (very often just one) samples of $w(x, \theta^i)$ for any given point x, and make relatively small steps in the x-space, but they utilize also a special averaging procedure, which calculates the average of w or its gradient over a certain number of steps. So, in this case, the averaging in θ-space and progression in the x-space are combined, while in the large-sample methods they are separated. Both techniques have proven convergence under certain (different) conditions. The majority of yield optimization methods belong to the large-sample category (but some can be modified to use a small number of samples per iteration). *A* class of small-sample yield optimization methods was proposed in [50] and is based on the well-known techniques of stochastic approximation [34], to be discussed later in this section.

5.4.2 Methods Using Standard Deterministic Optimization Algorithms

The most natural method of yield optimization would be to estimate the yields $Y(x^0), Y(x^1), \ldots$ from a large number of samples (as described in the previous section) for each $x^0, x^1, \ldots$ of the sequence $\{x^k\}$ generated by a standard, nonderivative deterministic search algorithm, such as the simplex method of Nelder and Mead, Powell, or other algorithms discussed in [14]. This is very appealing, since most of the conditions for algorithm's "optimality" are fulfilled: it would work for any $e = e(x, \theta)$ and only the values of $y = y(x, \theta)$ and the samples of θ would be required. However, if no approximation to $y = y(x, \theta)$ was available, the method would require tens of thousands of circuit analyses, which would be prohibitively expensive. Moreover, if the number of samples per iteration was reduced to increase efficiency, the optimizer would be receiving a highly noise-corrupted information leading to poor algorithm convergence or divergence, since standard optimization algorithms work poorly with noisy data (special algorithms, able to work under uncertainly—such as stochastic approximation algorithms—have to be used).

If some approximating functions $\hat{y} = \hat{y}(x^k, \theta)$ are available separately for each x^k, a large number of MC analyses can be cheaply performed, reducing the statistical error. In practice, such an approach is most often too expensive, due to the high cost of obtaining the approximating formulas, if the number of important θ parameters is large. The approximating functions $\hat{y}_{j(x,\theta)}$ for each y_j can be also created in the *joint* (x, θ) space [56,61]. In [45], an efficient new approximating methodology was created, highly accurate for a relatively large range of the x_i values. However, also in this case the dimension of the joint space (x, θ) cannot be too large, since the cost of obtaining the approximating functions $\hat{y}_{j(x,\theta)}$ for becomes itself prohibitively high. Because of these difficulties, several dedicated yield optimization methods have been developed, for which the use of function approximation is not required. Some of these methods are described in what follows.

5.4.3 Large-Sample Heuristic Methods for Discrete Circuits

These methods have been developed mostly for discrete circuits, for which $e_i = x_i + \theta_i$. Only $y^i = y(x + \theta^i)$ function values are required, for the sample θ^i obtained in an arbitrary way. Approximation in the e-space can be constructed to increase efficiency, but for discrete circuits the number of θ parameters can be large (proportional to the number of active devices, since no correlations between different devices exist), so the use of approximation is most often not practical. The most typical representative of this class is the centers of gravity method [39]. The method is based on a simple observation that if $\bar{x}_S$ is the center of gravity of "pass" points (as shown in Figure 5.10), defined as $\bar{x}_S = (x_A^1 + x_A^2 + \cdots + x_A^{N_A})/N_A$,

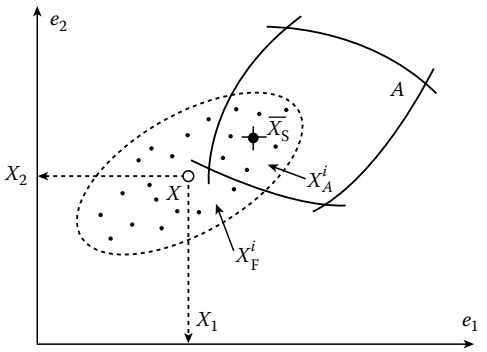


FIGURE 5.10 Interpretation of the original centers of gravity method.

where N_A is the number of points x_A^i falling into the A-region, then a step from x to $\bar{x}_S$ will improve yield. In [37], also the center of gravity of the "fail" points $\bar{x}_F = (x_F^1 + x_F^2 + \cdots + x_F^{N_F})/N_F$ was defined, and the direction of yield increase taken as going from $\bar{x}_F$ through $\bar{x}_A$, as shown in Figure 5.11. Moving in this direction with the step-size equal to $\mu(\bar{x}_S - \bar{x}_F)$, where $\mu \approx 0.2 - 2$ (often taken as $\mu = 1$) leads to a sequence of optimization steps, which is stopped if $\|\bar{x}_S - \bar{x}_F\|$ is less than a predefined small constant. This is based on the property (proved for a class of p.d.f.s in [43] that, under some conditions, at the yield maximum $\|\bar{x}_S - \bar{x}_F\| = 0$ (and $\bar{x}_S - \bar{x}_F = \hat{x}$ where $\hat{x}$ is the point of the yield maximum). It was also shown in [43] that for the normal p.d.f. $f_\theta(\theta)$ with zero correlations and all standard deviations $\sigma_{\theta_i} = \sigma_\theta$, $i = 1, \ldots, t$, equal, the "centers of gravity" direction coincides with the yield gradient direction. However, with correlations and σ_{θ_i}s not equal, the two directions can be quite different. Various schemes, aimed at the reduction of the total required number of analyses were developed based on the concepts of "reusable" points [39].

In [18,19] the original centers of gravity method was significantly improved, introducing a concept of "Gaussian adaptation" of the covariance matrix of the sampled points of θ^i, such that they (temporarily) adopt to the shape of the acceptability region, leading to higher (optimal) efficiently of the algorithm. The method was successfully used on large industrial design examples, involving as many as 130 designable parameters, not only for yield optimization, but also for standard function minimization.

The methods of "radial exploration of space" [54] (see Figure 5.8e) and one-dimensional orthogonal searches (ODOS) [26,27] (see Figure 5.8b and c) discussed in Section 5.3 in the context of Acceptability Region approximation, have also been developed into yield optimization methods: in the "radial exploration" case the asymmetry vectors were introduced, generating a direction of yield increase, and in the ODOS case using a Gauss–Seidel optimization method. Both techniques were especially efficient for linear circuits due to a high efficiency of performing circuit analyses in radial and orthogonal directions.

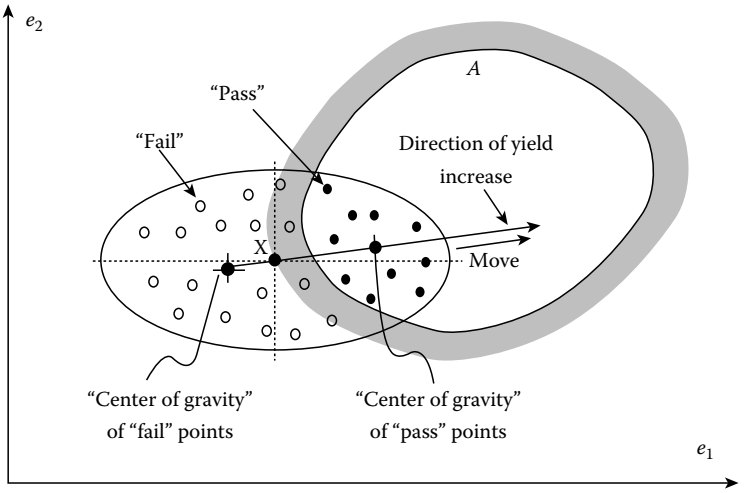


FIGURE 5.11 Interpretation of the modified centers of gravity method.

5.4.4 Large-Sample, Derivative-Based Methods for Discrete Circuits

For discrete circuits, the relation $e_i = x_i + \theta_i$ holds for a part of the entire e vector, so, x_i and θ_i are in the same space. Moreover, the p.d.f. $f_\theta(\theta)$ is most often known. Let x denote the vector of expectations $x \equiv E_\theta\{e(x,\theta)\}$, and $f_e(e, x)$ is the p.d.f. $f_\theta(\theta)$ transformed to the e-space (i.e., of the same shape as $f_\theta(\theta)$, but shifted by x). Then, from Equation 5.5, differentiating w.r.t. x_i, one obtains

$$\frac{\partial Y(x)}{\partial x_i} = \int_{R^n} \phi(e) \frac{\partial f_e(e,x)}{\partial x_i} \frac{f_e(e,x)}{f_e(e,x)} de = E_e\left\{\phi(e) \frac{\partial \ln f_e(e,x)}{\partial x_i}\right\} \tag{5.10}$$

where the equivalence $[\partial f_e(e,x)/f_e(e,x)]/\partial x_i \equiv \partial \ln f_e(e,x)/\partial x_i$ was used. Therefore, yield derivatives w.r.t. x_i can be calculated as the average of the expression in the braces of Equation 5.10, calculated from the same θ^i samples as those used for yield estimation, provided that the p.d.f. $f_e(e, x)$ is differentiable w.r.t x (e.g., the normal or log-normal p.d.f.s are differentiable, but the uniform p.d.f is not). Notice that instead of sampling with the p.d.f. $f_e(e, x)$, some other (better) p.d.f. $g_e(e, x)$ can be used as in the importance sampling yield estimation (see Equation 5.8). Then,

$$\frac{\partial Y(x)}{\partial x_i} = E_e\left\{\phi(e) \frac{\partial f_e(e,x)}{\partial x_i} \left[\frac{f_e(e,x)}{g_e(e,x)}\right]\right\} \tag{5.11}$$

where sampling is performed with the p.d.f. $g_e(e, x) \neq 0$. This technique was used in [2,36,51] (to be discussed next). Consider the multivariate normal p.d.f., with the positive definite covariance matrix K:

$$f_e(e) = \frac{1}{(2\pi)^{t/2}\sqrt{\det K}} \exp\left[-\frac{1}{2}(e-x)^{\mathrm{t}} K^{-1}(e-x)\right] \tag{5.12}$$

where $e - x \equiv \theta$ (discrete circuits), and det K is the determinant of K. Then, it can be shown that the yield gradient $\nabla_x Y(x)$ is expressed by

$$\nabla_x Y(x) = E\{\phi(e) K^{-1}(e-x)\} = Y(x) K^{-1}(\bar{x}_S - x) \tag{5.13}$$

where $\bar{x}_S$ is the center of gravity of "pass" points. If yield $Y(x)$ is a continuously differentiable function of x, then the necessary condition for the yield maximum is $\nabla_x y(\hat{x}) = 0$, which combined with Equation 5.13 means that the stationery point $\hat{x}$ for the yield function (the yield maximum if $Y(x)$ is also concave) is $\hat{x} = \bar{x}_S$, the center of gravity of the pass points. This result justifies (under the assumptions stated previously) the centers of gravity method of yield optimization (since its objective is to make $\hat{x} = \bar{x}_S \equiv \bar{x}_F$).

For $K = \mathrm{diag}\{\sigma_{e_1}^2, \ldots, \sigma_{e_t}^2\}$, i.e., with zero correlations, the yield gradient w.r.t. x is expressed as

$$\nabla_x Y(x) = E_e\left\{\phi(e)\left[\frac{\theta_1}{\sigma_{\theta_1}^2}, \cdots, \frac{\theta_t}{\sigma_{\theta_t}^2}\right]^{\mathrm{t}}\right\} \tag{5.14}$$

where $\sigma_{e_\theta} \equiv \sigma_{e_i}$ was used instead of σ_{e_i}. It can be readily shown that for all $\sigma_{\theta_i} = \cdots = \sigma_{\theta_t} = \sigma_\theta$ equal, the yield gradient direction coincides with the center-of-gravity direction [43] (Figure 5.12).

Since all higher-order derivatives of the normal p.d.f. exist, all higher order yield derivatives can also be estimated from the same sampled points θ^i, as those used for yield estimation. The yield gradient can also be calculated from the "fail" points simply using the $\phi_F(\cdot) = \phi(\cdot) - 1$ indicator function in all the expressions above. Then, the two resulting estimators can be combined as one joint average, as it was done in the $\bar{x}_S$, $\bar{x}_F$-based centers of gravity method. Actually, there exists an optimal weighted combination of the two estimators for any given problem, resulting in the minimum variability of the

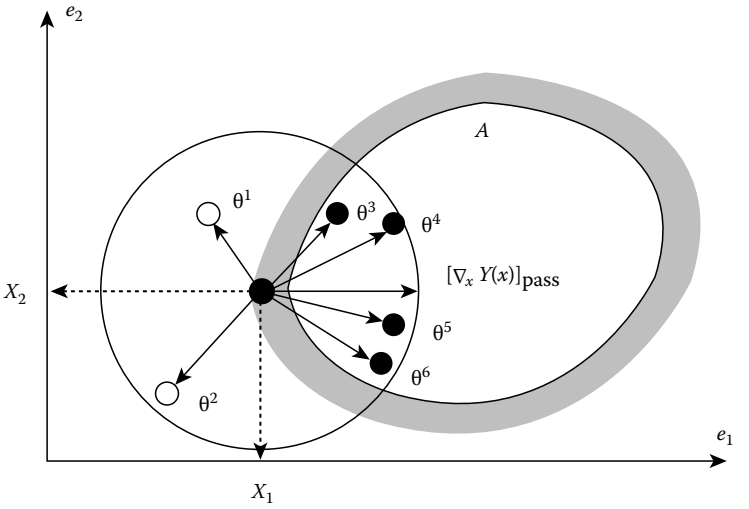


FIGURE 5.12 Interpretation of the yield gradient formula for normal p.d.f., with no correlations and $\sigma_{\theta 1} = \sigma_{\theta 2} = 1$. For the "pass" points (black dots) $[\nabla_x Y(x)]_{\text{pass}} \approx (\theta^3 + \theta^4 + \theta^5 + \theta^6)/4$; for the "fail" points (white dots) $[\nabla_x Y(x)]_{\text{fail}}$ $(-\theta^1 - \theta^2)/2$. The two estimators can be combined together (equal weighting assumed): $\nabla_x Y(x) = -\theta^1 - \theta^2 + \theta^3 + \theta^4 + \theta^5 + \theta^6)/6$. It is clearly seen that the two yield gradient estimators coincide with the center of gravity of the "pass" and "fail" points, respectively.

gradient, but it is difficult to precisely determine in practice. A general rule is that at the beginning of optimization, when x is far away from $\hat{x}$ (the optimal point), the yield gradient estimator based on the "pass" points should be more heavily weighted; the opposite is true at the end of optimization, when the "fail" points carry more precise gradient information. An interpretation of the yield gradient formula Equation 5.14 is shown in Figure 5.12, for the case where $\sigma_{\theta_1} = \sigma_{\theta_2} = 1$.

In the majority of practical applications of large-sample derivative methods, it was assumed that $f_e(e)$ was normal. A typical iteration step is made in the gradient direction:

$$x^{k+1} = x^k + \alpha_k \nabla_x Y(x^k) \tag{5.15}$$

where α_k is most often selected empirically, since yield maximization along the gradient direction is too expensive, unless some approximating functions $\hat{y} = \hat{y}(x + \theta)$ are used (normally, this is not the case for the class of methods discussed). Since the number of points θ^i sampled for each x^k, is large, the main difference between various published algorithms is how to most efficiently use the information already available. The three methods to be discussed introduced almost at the same time [2,36,51]), utilize for that purpose some form of importance sampling, discussed in Section 5.3.

In [36], a "parametric sampling" technique was proposed, in which the θ^i points were sampled with the p.d.f. $g_e(e, x)$ in a broader range than for the original p.d.f. (i.e., all the σ_{θ_i}'s were artificially increased). All points sampled were stored in a database, and the gradient-direction steps made according to Equation 5.15. The importance sampling-based gradient formula (Equation 5.11) was used in subsequent iterations within the currently available database. Then, a new set of points was generated and the whole process repeated.

The methods developed in [2,51] were also based on the importance sampling concept, but instead of using the gradient steps as in Equation 5.15, the yield gradient and Hessian* matrices were calculated

* Matrix of second derivatives.

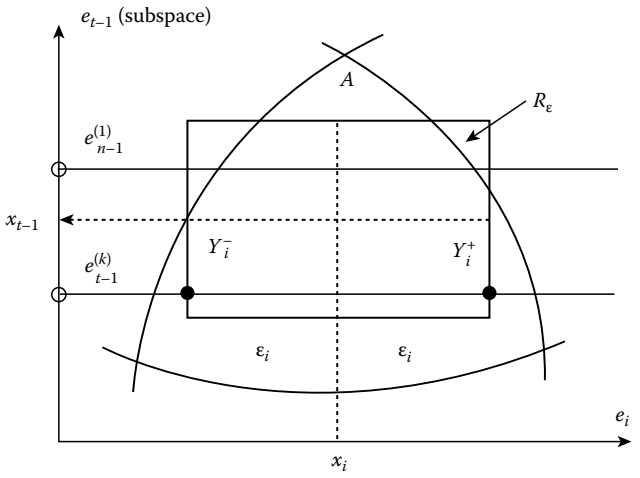


FIGURE 5.13 Yield and yield derivative estimation for uniform distribution. Y^+ and Y^- denote symbolically the "yields" calculated on the $(t-1)$-dimensional faces of the tolerance hypercube R_ε. The $e_{t-1}{}^{(k)}$ points are sampled in the $(t-1)$-dimensional subspaces of e_{t-1} parameters, with the uniform p.d.f. $f_{et-1}(e_{t-1})$.

and updated within a given database. Then, a more efficient Newton's direction was taken in [51] or a specially derived and efficient "yield prediction formula" used in [2]. In order to deal with the singularity or nonpositive definiteness of the Hessian matrix (which is quite possible due to the randomness of data and the behavior of $Y(x)$ itself), suitable Hessian corrections were implemented using different kinds of the Hessian matrix decomposition (Cholesky-type in [51] and eigenvalue decomposition in [2]).

As it was the case for the heuristic methods, the methods just discussed are relatively intensive to the dimensionality of x and θ spaces.

For the uniform p.d.f., centered at $e = x$ (Figure 5.13) and defined within a hyperbox $x_i - \varepsilon_i \le e_i \le x_i + \varepsilon$, $i = 1, \ldots, t$, where ε_i are element tolerances (see Figure 5.2), the yield gradient formula (Equation 5.10) cannot be used, since the uniform p.d.f. is nondifferentiable w.r.t. x_i. It can be shown [26,42,43] that yield can be calculated by sampling in the $(t-1)$-dimensional subspace of the e-space, represented by $e_{t-1} + (e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_t)$ and analytical integration in the one-dimensional subspace e_i as shown in Figure 5.13. Using this approach, it can be further proved [43] that the yield derivatives w.r.t. x_i are expressed by

$$\frac{\partial Y(x)}{\partial x_i} = \frac{1}{2\varepsilon_i}\left(Y_i^+ - Y_i^-\right) \tag{5.16}$$

where $Y_i{}^+$, $Y_i{}^-$ are "yields" calculated on the faces of the $t-1$ tolerance hyperbox R_ε, corresponding to $x_i + \varepsilon_i$ and $x_i - \varepsilon_i$, respectively. Calculation of these "yields" is very expensive, so in [43]* different algorithms improving efficiency were proposed. In [53], an approximate method using efficient three-level orthogonal array (OA) sampling on the faces of R_ε was proposed, in which (due to specific properties of OAs) the same sample points were utilized on different faces (actually one-third of all sampled points were available for a single face). This has lead to substantial computational savings and faster convergence.

* In [43], general formulas for gradient calculation for *truncated* p.d.f.s were derived.

5.4.5 Large-Sample, Derivative-Based Method for Integrated Circuits

In this case, which is typical of IC yield optimization, yield gradient calculations cannot be performed in the *e*-space, as was the case for discrete circuits. In general, yield gradient could be calculated by differentiating the $\phi(e(x,\theta)) \equiv \phi(x,\theta)$ term in the θ-space-based yield formula (Equation 5.6), derived for the general case, where the general transformation $e = e(x,\theta)$ is used. Differentiation of $\phi(x, \theta)$ is, however, not possible in the traditional sense, since $\phi(x,\theta)$ is a nondifferentiable unit step function determined over the acceptability region *A*. One possible solution was proposed in [15]. In what follows, a related, but more general method proposed in [13] is discussed.

It was first shown in [13] that yield can be evaluated as a surface-integral rather than the volume-integral (as it is normally done using the MC method). To understand this, observe that yield can be evaluated by sampling in the $t-1$ subspace of the *t*-dimensional θ-subspace, as shown in Figure 5.14, evaluating "local yields" along the lines parallel to the θ_i axis, and averaging all the local yields.* Each "local yield" can be evaluated using the values of the cumulative (conditional) distribution function† along each parallel, at the points of its intersection with the boundary of the acceptability region *A*. This process is equivalent to calculating the surface integral over the value of the cumulative density function calculated (with appropriate signs) on the boundary of the acceptability region.

The next step in [13] was to differentiate the "surface-integral" based yield formula w.r.t. x_i, leading to the following yield gradient formula:

$$\nabla Y(x) = E_{\theta_{t-1}}\left\{\sum_k f_{\theta_i}\left[\theta_i^{(k)}\right] \cdot \frac{\nabla_x y_a(x,\theta)}{\left|\partial y_a(x,\theta)/\partial\theta_i\right|}\bigg|_{x,\theta_{t-1},\theta_i^{(k)}}\right\} \tag{5.17}$$

Where summation is over all intersection points $\theta_i^{(k)}$ of the parallel shown in Figure 5.14 with the boundary of $A_\theta(x)$; y_a is that specific performance function $y_j(x,\theta)$ which, out of all the other performances, actually determines the boundary of the acceptability region at the $\theta_i^{(k)}$ intersection point.‡ The gradient $\nabla_x y_a(x,\theta)$, and the derivative $\partial y_a(x,\theta)/\partial\theta_i$ have to be calculated for every fixed sampled point

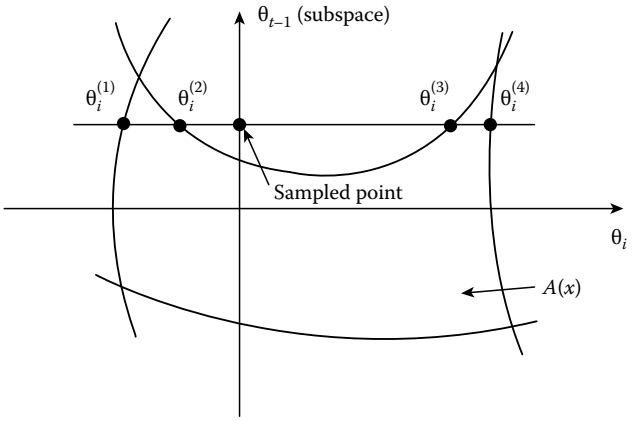


FIGURE 5.14 Interpretation of "local yield" calculation along a parallel.

* Identical technique was used in the previous section [42,43] to derive the yield derivative formula (Equation 5.16) (see Figure 5.13).

† If θ_i's are independent, it is the *marginal* p.d.f. of θ_i, as it was assumed in [13].

‡ At this point, a specific equation $y_a(x, \theta_{t-1}, 1_i^{(k)} = S_a$ (where $S_a + S_a^L$ or $S_a + S_a^U$ or are lower and upper bounds on y_a, respectively) must be *solved* to find $\theta_i^{(k)}$.

(x, θ_{t-1}), at each intersecting point $\theta_i^{(k)}$ shown in Figure 5.14. Observe that the derivative calculations in Equation 5.17 can (and often will have to) be performed in two steps: since $y_a(x, \theta) = y_a(e(x, \theta))$ so, $\partial y_a/\partial x_p = \Sigma_s(\partial y_a/\partial e_s)(\partial e_s/\partial x_p)$, and $\partial y_a/\partial \theta_i = \Sigma_s(\partial y_a/\partial e_s)(\partial e_s/\partial \theta_i)$, where the derivatives appearing in the first parentheses of both formulas are calculated using a circuit simulator and those in the second parentheses from a given statistical model $e = e(x, \theta)$.

In the practical implementation presented in [13], random points θ^r are sampled in the θ-space, in the same way as shown in Figure 5.8b but replacing e by θ, then searches for the intersections $\theta_i^{(k)}$ are performed in all axis directions, the formula in the braces of Equation 5.17 is calculated for each intersection, and averaged out over all outcomes. Searches along the parallel lines in all directions are performed to increase the accuracy of the yield gradient estimator. Observe that this technique requires tens of thousands of circuit analyses to iteratively find the intersection points $\theta_i^{(k)}$, plus additional analyses (if needed) for the calculation of the gradient and the derivatives in Equation 5.17. This problem has been circumvented in [13] by constructing approximating functions $\hat{y} = \hat{y}(x, \theta)$ w.r.t. θ for each x, together with approximating functions for all the derivatives.* Due to a high level of statistical accuracy obtained in evaluating both yield and its gradients, an efficient, gradient-based deterministic optimization algorithm, based on sequential quadratic programming was used, requiring a small number of iterations (from 5 to 11 for the examples discussed in [13]). The gradient $\nabla_x y_a(x, \theta)$ was either directly obtained from circuit simulator, or (if not available) using (improved) finite difference estimators. The method showed to be quit efficient for a moderate size of the θ-space (10–12 parameters).

The resulting yield optimization method is independent of the form of $e = e(x, \theta)$ the derivatives of y_j w.r.t. both x_k and θ_s are required and the analytical form of $f_\theta(\theta)$ and its cumulative function distribution must be known. The method cannot practically work without constructing the approximating functions $\tilde{y} = y(x, \theta)$ (approximating function in the joint space (x, θ) could also be used, if available).

5.4.6 Small-Sample Stochastic Approximation-Based Methods

Standard methods of nonlinear programming perform poorly in solving problems with statistical errors in calculating the objective functions (i.e., yield) and its derivatives.† One of the methods dedicated to the solution of such problems is the stochastic approximation (SA) approach [33] developed for solving the regression equations, and then adopted to the unconstrained and constrained optimization by several authors. These methods are aimed to the unconstrained and (maximum) of a function corrupted by noise (a regression function). The SA methods were first applied to yield optimization and statistical design centering in [49,50]. The theory of SA methods is well established, so its application to yield optimization offers the theoretical background missing, e.g., in the heuristic methods of Section 5.3. As compared to the large-sample methods, the SA algorithms to be discussed use a few (or just one) randomly sampled points per iteration, which is compensated for by a large number of iterations exhibiting a trend toward the solution. The method tends to bring large initial improvements with a small number of circuit analyses, efficiently utilizing the high content of the deterministic information present at the beginning of optimization.

In 1951, in their pioneering work, Robins and Monro [33] proposed a scheme for finding a root of a regression function, which they named the stochastic approximation procedure. The problem was to find a zero of a function, whose ''noise corrupted'' values could be observed only, namely $G(x) = g(x) + \theta(x)$ where $g(x)$ (unknown) function of x, θ is a random variable, such that $E\{\theta\} = 0$ and $\text{var}\{\theta\} \leq L < \infty$. Therefore, a zero of the regression function $g(x) + E\{G(x)\} = 0$ was to be found. The SA algorithm proposed in [33] works as follows: given a point $x^{(k)}$, set the next point as $x^{(k+1)} = x^{(k)} - a^k G[x^{(k)}]$.

* Low-degree polynomials were used with very few terms, generated from a *stepwise regression* algorithm.

† This was the reason a high level of accuracy was required while calculating both yield and its derivatives using the method described in the previous section (and *consistency* between the yield and gradient estimators), otherwise, the deterministic optimization algorithm would often diverge, as observed in [13].

The sequence of $\{x^{(k)}\}$ points converges to the solution $\hat{x}$ under some conditions (one of them is, for instance, that a_k can change according to the harmonic sequence $\left\{\mathbf{1}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\right\}$). Assuming that $G(x) \equiv \xi^k$ is a "noise corrupted" observation of the gradient of a regression function $f(x) = E_\theta\{w(x, \theta)\}$, the algorithm can be used to find a stationary point (e.g., a maximum) of $f(x)$, since this is equivalent to finding $\hat{x}$ such the $E\{G(\hat{x})\} = E\{\xi(\hat{x})\} = \nabla_x Y(\hat{x}) = 0$. For yield optimization $f(x) \equiv Y(x) = E\{\phi(x, \theta)\}$. This simplest scheme that can be used, if the yield gradient estimator is available is

$$x^{k+1} = x^k + \tau_k \xi^k \tag{5.18}$$

where $\xi^k + \widehat{\nabla_x Y}(x^k)$ is an estimate of the yield gradient, based on one (or more) point θ^i sampled with the p.d.f. $f_\theta(\theta)$ and $\tau_k > 0$ is the step length coefficient selected such that the sequence: $\{\tau_k\} \to 0$, and $\sum_{k=0}^{\infty} \tau_k = \infty, \sum_{k=0}^{\infty} \tau_k^2 < \infty$ (e.g., the harmonic series $\left\{\mathbf{1}, \frac{1}{2}, \frac{1}{3}, \ldots\right\}$ fulfills these conditions). For the convergence with probability 1, it is also required that the conditional expectation $E\{\xi^k | x^1, x^2, \ldots, x^k\} = \nabla Y(x^k)$. The algorithm of Equation 5.18 is similar to the steepest ascent algorithms of nonlinear programming, so, it will be slowly convergent for ill-conditioned problems. A faster algorithm was introduced in [35] and used for yield optimization in [49,50]. It is based on the following iterations:

$$x^{k+1} = x^k + \tau_k d^k \tag{5.19}$$

$$d^k = (1 - \rho_k) d^{k-1} + \rho_k \xi^k, \quad 0 < \rho_k < 1 \tag{5.20}$$

where ξ_k is a (one- or more-point) estimator of $\nabla_x Y(x^k)$ and $\{\tau_k\} \to 0$, $\{\rho_k\} \to 0$ are nonnegative coefficients. $d^{(k)}$ is a convex combination of the previous (old) direction d^{k-1} and the new gradient estimate ξ_k, so the algorithm is an analog of a more efficient, conjugate gradient method. Equation 5.2 provides gradient averaging. The ρ_k coefficient controls the "memory" or "inertia" of the search direction d^k, as shown in Figure 5.15. If ρ_k is small, the "inertia" of the algorithm is large, i.e., the algorithm tends to follow the previous gradient directions. For convergence with probability 1, the same conditions must hold as those for Equation 5.18.

The coefficients τ_k and ρ_k automatically determined based on some heuristic statistical algorithms proposed in [35]. Several other enhancements were used to speed up the convergence of the algorithm, especially in its much refined version proposed in [35]. For solving the optimization problems, the one-or two-point yield gradient estimator is found from Equation 5.10 in general, and from Equation 5.13 or 5.14 for the normal p.d.f.

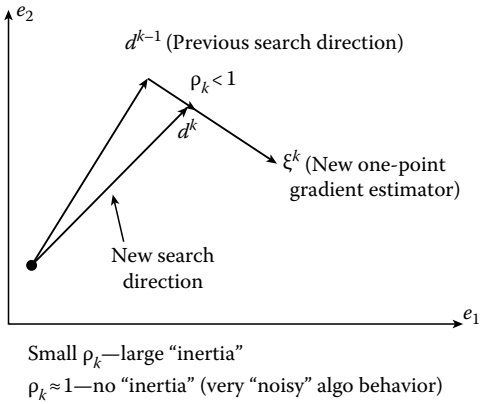


FIGURE 5.15 Illustration of the gradient averaging equation.

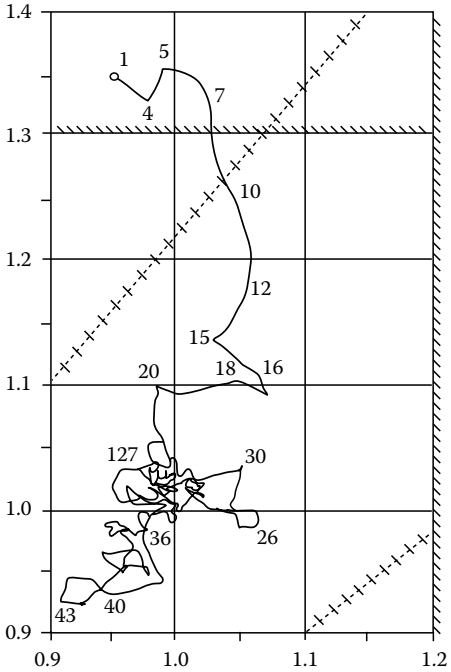


FIGURE 5.16 Typical trajectory of the stochastic approximation algorithm (Example 5.1).

Example 5.1

A simple two-dimensional case is considered in order to illustrate the algorithm properties. The acceptability region (for a voltage divider) [49] is defined in the two-dimensional space (e_1, e_2) by the inequalities:

$$0.45 \leq \frac{e_2}{e_1 + e_2} \leq 0.55 \quad 0 \leq e_1 \leq 1.2 \quad 0 \leq e_2 \leq 1.3 \tag{5.21}$$

where $e_i = x_i + \theta_i$ and the p.d.f. of θ_i is normal with $E\{\theta_1\} = E\{\theta_2\} = 0$ and $\text{Cov}\{\theta_1, \theta_2\} = 0$ (no correlations).

A typical algorithm trajectory is shown in Figure 5.16. The one-sample yield gradient formula (Equation 5.14) was used. The initial yield was low (7.4%). Very few (25–30) iterations (equal to the number of circuit analyses) were required to bring the minimal point close to the final solution. This is due to the fact that at the beginning of optimization, there is a high content of deterministic gradient information available even from a few sampled points, so the algorithm progress is fast. Close to the solution, however, the yield gradient estimator is very noisy, and the algorithm has to filter out the directional information out of noise, which takes the majority of the remaining iterations. After the total of 168 circuit analyses, the yield increases to 85.6%. Other starting points resulted in a similar algorithm behavior. It is also observed that the yield optimum is "flat" in the diagonal direction, reflected in random nominal point movement in this direction (iterations 30–43).

Example 5.2

Parametric yield for the Sallen–Key active filter of Figure 5.6a (often used in practice as a test circuit) with the specifications on its frequency response shown in Figure 5.6b was optimized (recall that the shape of the acceptability region for this case was very complicated, containing internal nonfeasible regions

["holes"]). All *R*, *C* variables were assumed designable. The relative standard deviations were assumed equal to 1% for each element, the p.d.f. was normal with 0.7 correlation coefficient between the like elements (*R*'s or *C*'s) and zero correlations between different elements. The initial yield was 6.61%. Using the SA algorithm as in the previous example, 50 iterations (equal to the number of circuit analyses) brought yield to 46.2%; the next 50 iterations to 60%, while the remaining 132 iterations increased yield to only 60.4% (again the initial convergence was very fast). The algorithm compared favorably with the Hessian matrix-based, large-sample method discussed previously [51], which required about 400 circuit analyses to obtain the same yield level, and whose initial convergence was also much slower. It has to be stressed, however, that the results obtained are statistical in nature and it is difficult to draw strong general conclusions. One observation (confirmed also by other authors) is that the SA-type algorithms provide, in general, fast initial convergence into the neighborhood of the optimal solution, as it was shown in the examples just investigated.

5.4.7 Small-Sample Stochastic Approximation Methods for Integrated Circuits

The methods of yield gradient estimation for discrete circuits cannot be used for ICs because of the form of the $e = e(x, \theta)$ transformation, as discussed at the beginning of Section 5.4. Previously in this section, a complicated algorithm was described for gradient estimation and yield optimization in such situations. In this section, a simple method based on random perturbations in x-space, proposed in [47,48], is described. It is useful in its own merit, but especially in those cases, where the conditions for the application of the yield gradient formula (Equation 5.17) are hard to meet, namely: the cost of constructing the approximating functions $\hat{y} = \hat{y}(x, \theta)$ for fixed x is high and calculating the gradient of y w.r.t. x is also expensive (which is the case, e.g., if the number of important x and θ parameters is large), and the analytical form of $f_\theta(\theta)$ is not available (as it is required in Equation 5.17).

The method applications go beyond yield optimization: a general problem is to find a minimum of a general regression function $f(x) = E_\theta\{w(x, \theta)\}$, using the SA method, in the case where the gradient estimator of $f(x)$ is not directly available. Several methods have been proposed to estimate the gradient indirectly, all based on adding some extra perturbations to x parameters. Depending on their nature, size, and the way the perturbations are changed during the optimization process, different interpretations result, and different problems can be solved. In the simplest case, some extra deterministic perturbations (usually double-sided) are added individually to each x_k (one-at-a-time), while random sampling is performed in the θ-space, and the estimator of the derivative of $f(x)$ w.r.t. x_k is estimated from the difference formula $\hat{\xi}_k = \widehat{\partial f(x)}/\partial x_k = (1/N)\sum_{i=1}^{N}\left\{\left[w\left(x + a_k e_k, \theta_1^i\right) - w\left(x - a_k e_k, \theta_2^i\right)\right]/(2a_k)\right\}$, where $a_k > 0$ is the size of the perturbation step and e_k is the unit vector along the e_k coordinates axis. Usually, $\theta_1^i \equiv \theta_2^1$ to reduce the variance of the estimator. Normally, $N = 1$. Other approaches use random direction derivative estimation by sampling points randomly on: a unit sphere of radius a [34], randomly at the vertices of a hypercube in the x-space [38], or at the points generated by orthogonal arrays [53], commonly used in the design of experiments. Yet another approach, dealing with nondifferentiable p.d.f.s was proposed in [55]. In what follows, the random perturbation approach resulting in convolution function smoothing is described for a more general case, where a *global* rather than a local minimum of $f(x)$ is to be found.

The multi-extremal regression function $f(x)$ defined previously, can be considered a superposition of an uni-extremal function (i.e., having just one minimum) and other multi-extremal functions that add some deterministic "noise" to the uni-extremal function (which itself has also some "statistical noise"—due to θ—superimposed on it). The objective of convolution smoothing can be visualized as "filtering out" but types of noise and performing minimization on the "smoothed" uni-extremal function (or on a family of these functions), in order to reach the global minimum. Since the minimum of the smoothed uni-extremal function does not, in general, coincide with the global function minimum, a sequence of minimization runs is required with the amount of smoothing eventually reduced to zero in the

neighborhood of the global minimum. The smoothing process is performed by averaging $f(x)$ over some region of the n-*dimensional* parameter space x using a proper weighting (or smoothing) function $\hat{h}(x)$, defined below. Let the n-dimensional vector η denote a vector of random perturbations; it is added to x to create the convolution function [34]:

$$\tilde{f}(x,\beta) = \int_{R^n} \hat{h}(\eta,\beta) f(x-\eta)\mathrm{d}\eta$$
$$= \int_{R^n} \hat{h}(x-\eta,\beta) f(\eta)\mathrm{d}\eta = E_\eta\{f(x-\eta)\} \tag{5.22}$$

where $\hat{f}(x,\beta)$ is the smoothed approximation to the original multi-extremal function $f(x)$, and the kernel function $\hat{h}(\eta,\beta)$ is the p.d.f. used to sample η. Note that $f(x,\beta)$ can be interpreted as an averaged version of $f(x)$ weighted by $\hat{h}(\eta,\beta)$. Parameter β controls the dispersion of $\hat{h}$, i.e., the degree of $f(x)$ smoothing (e.g., β can control the standard deviations of $\eta_1 \ldots \eta_n$). $E_\eta\{f(x-\eta)\}$ is the expectation with respect to the random variable η. Therefore, an unbiased estimator $\hat{f}(x,\beta)$ of $\tilde{f}(x,\beta)$ is the average: $\hat{f}(x,\beta) = (1/N)\sum_{i-1}^{N} f(x-\eta^i)$, where η is sampled with the p.d.f. $\hat{h}$ $(\eta-\beta)$. The kernel function $\hat{h}(\eta,\beta)$ should have certain properties discussed in [34], fulfilled by several p.d.f.s, e.g., the Gaussian and uniform. For the function $f(x) = x^4 - 16x^2 + 5x$, which has two distinct minima, smoothed functionals, obtained using Equation 5.22, are plotted in Figure 5.17 for different values of $\beta \to 0$, for (a) Gaussian and (b) uniform kernels. As seen, smoothing is able to eliminate the local minima of $\tilde{f}(x,\beta)$ if β is sufficiently large. If $\beta \to 0$, then $\tilde{f}(x,\beta) \to f(x)$.

The objective now is to solve the following optimization problem: minimize the smoothed functional $\tilde{f}(x,\beta)$ with $\beta \to 0$ as $x \to \hat{x}$, where $\hat{x}$ is the global minimum of the original function $f(x)$. The modified optimization problem can be written as minimize $\tilde{f}(x,\beta)$ w.r.t. x, with $\beta \to 0$ as $x \to \hat{x}$. Differentiating Equation 5.22 and using variable substitution, the gradient formula is obtained:

$$\nabla_x \tilde{f}(x,\beta) = \int_{R^n} \nabla_\eta \hat{h}(\eta,\beta) f(x-\eta)\mathrm{d}\eta = \frac{1}{\beta}\int_{R^n} \nabla_\eta h(\eta) f(x-\beta\eta)\mathrm{d}\eta, \tag{5.23}$$

where $\hat{h}(\cdot)$ is as defined previously, and $h(y)$ is a normalized version of $\hat{h}(y)$ (obtained if $\beta = 1$). For normalized multinormal p.d.f. with zero correlations, the gradient of $\tilde{f}(x,\beta)$ is

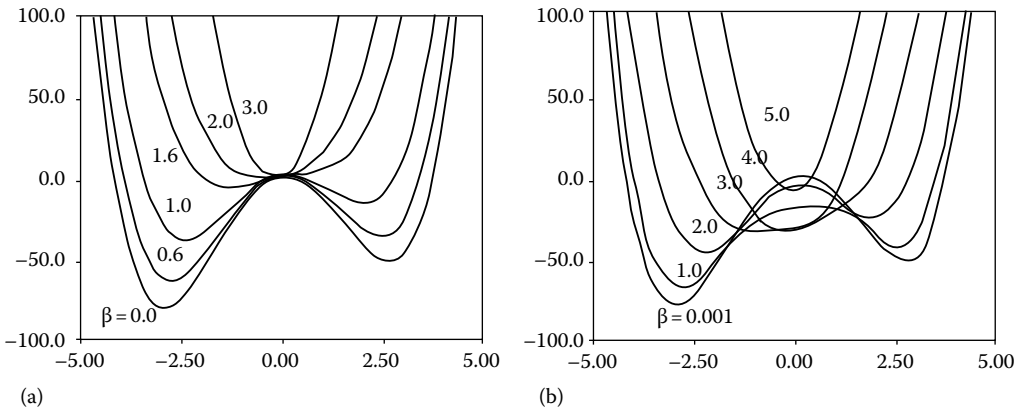


FIGURE 5.17 Smoothed functional $f(x,\beta)$ for different β's using (a) Gaussian kernel and (b) uniform kernel.

$$\nabla_x \tilde{f}(x,\beta) = \frac{-1}{\beta}\int_{R^n} \eta f(x-\beta\eta)h(\eta)d\eta = \frac{-1}{\beta}E_\eta\{\eta f(x-\beta\eta)\}$$
$$= \frac{-1}{\beta}E_{\eta,\theta}\{\eta w(x-\beta\eta,\theta)\} \tag{5.24}$$

where sampling is performed in x-space with the p.d.f. $h(\eta)$, and in θ-space with the p.d.f. $f_\theta(\theta)$; $E_{\eta,\theta}$ denotes expectation w.r.t. *both* η and θ, and it was taken into account for $f(x)$ is a noise corrupted version of $w(x, \theta)$ i.e., $f(x) = E_\theta\{w(x, \theta)\}$. The unbiased single-sided gradient estimator is therefore

$$\hat{\nabla}_x \tilde{f}(x,\beta) = \frac{-1}{\beta}\frac{1}{N}\sum_{i=1}^{N}\eta^i w(x-\beta\eta^i,\theta^i) \tag{5.25}$$

In practice, a double-sided estimator [34] of smaller variance is used

$$\hat{\nabla}_x \tilde{f}(x,\beta) = \frac{1}{2\beta}\frac{1}{N}\sum_{i=1}^{N}\eta^i\left[w(x+\beta\eta^i,\theta_2^i)\right]. \tag{5.26}$$

Normally, $N = 1$ for best overall efficiency. Statistical properties of these two estimators (such as their variability) were studied in [52]. To reduce variability, the same $\theta_1^i = \theta_2^i$ are usually used in Equation 5.26 for positive and negative $\beta\eta^i$ perturbations. For yield optimization, $w(\cdot)$ is simply replaced by the indicator function $\phi(\cdot)$. For multi-extremal problems, β values should be originally relatively large and then systematically reduced to some small number rather than to zero. For single-extremal problems (this might be the case for yield optimization) it is often sufficient to perform just a *single* optimization with a relatively small value of β, as it was done in the examples to follow.

5.4.8 Case Study: Process Optimization for Manufacturing Yield Enhancement

The object of the work presented in [48] was to investigate how to modify the MOS control process parameters together with a simultaneous adjustment of transistor widths and lengths to maximize parametric yield.* To make it possible, both process/device simulators (such as FABRICS, SUPREM, PISCES, etc.) and a circuit simulator must be used. In what follows FABRICS [12,22,24] is used as a process/device simulator. IC technological process parameters are statistical in nature, but variations of some of the parameters (e.g., times of operations, implant doses, etc.) might have small relative variations, and some parameters are common to several transistors on the chip. Because of that, the transformation $e = e(x, \theta)$ (where θ are now process related random variables), is such that standard methods of yield optimization developed for discrete circuits cannot be used. Let $Z_1 = z_1 + \xi_1$ be the process control parameters (doses, times, temperatures), where ξ_1 are random parameter variations, and z_1 are deterministic designable parameters. Let $Z_2 = z_2 + \xi_2$ be the designable layout dimensions, where z_2 are designable and ξ_2 are random variations (common to several transistors on a chip). $P = p + \psi$ are process physical parameters (random, nondesignable) such as diffusivities, impurity concentrations, etc. (as above, p is the nominal value and ψ is random). All random perturbations are collected into the vector of random parameters θ, called also the vector of process disturbances: $\theta = (\xi_1, \xi_2, \psi)$. The vector of designable parameters $x = (z_1, z_2)$ is composed of the process z_1 and layout z_2 designable parameters. Therefore, x and θ are in different spaces (subspaces). There are also other difficulties: the analytical form of the p.d.f. of θ is most often not known, since θ parameters are hierarchically generated from a

* This might be important, e.g., for the process refinement and IC cell *redesign*.

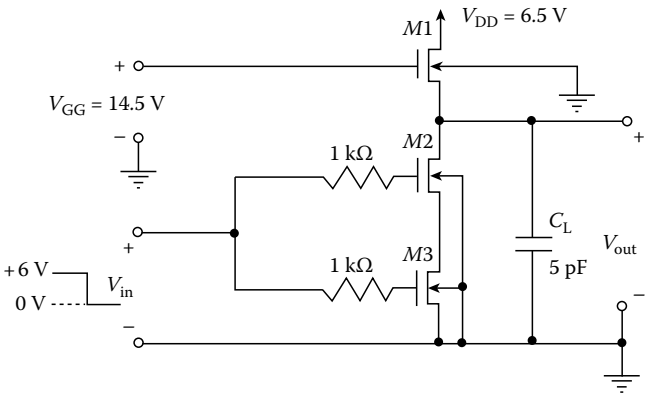


FIGURE 5.18 NMOS NAND gate of Example 5.3.

numerical procedure [12], the derivatives of the performances y w.r.t. x and θ are not known from FABRICS and can only be estimated by finite differences, and the θ and x spaces are very large (see below). So, creating approximation and/or finding derivatives using finite differences is expensive. Because of these difficulties, the smoothed-functional approach discussed in this section will be used, as shown in the following example.

Example 5.3

The objective is to maximize parametric yield for the NMOS NAND gate shown in Figure 5.18 [48], by automated adjustment of process and layout parameters. Specifications are: $V_0 = V_{out}\ (t = 0) \leq 0.7$ V, $V_{out}\ (t_1 = 50$ ns$) > 6.14$ V, circuit area $\leq 2500\ \mu m^2$. There are 45 designable parameters: all 39 technological process parameters, 6 transistor dimensions, and about 40 noise parameters, so it is a large problem suitable for the use of the SA-based random perturbation method described above.

The initial yield was $Y = 20\%$. After the first optimization using the method of random perturbations with 2% relative perturbations of each of the designable parameters involving the total of 110 FABRICS/SPICE analyses, yield increased to 100% and the nominal area decreased to 2138 μm^2. Then, specs were tightened to $V_{out}\ (t_2 = 28$ ns$) > 6.14$ V with the same constraints on V_0 and area, causing yield to drop to 10.1%. After 60 FABRICS/SPICE analyses, using the perturbation method, yield increased to: $Y = 92\%$ and area $= 2188\ \mu m^2$. These much-improved results produced the nominal circuit responses shown in Figure 5.19. Several technological process parameters were changed during optimization in the range between 0.1% and 17%: times of oxidation, annealing, drive-in, partial pressure of oxygen, and others, while the transistor dimensions changed in the range between 0.8% and 6.3%. The cost of obtaining these results was quite reasonable: the total of 170 FABRICS/SPICE analyses, despite the large number of optimized and noise parameters. Other examples are discussed in [48].

5.4.9 Generalized Formulation of Yield, Variability, and Taguchi Circuit Optimization Problems

Parametric yield is not the only criterion that should be considered during statistical circuit design. Equally, and often more important is minimization of performance variability caused by various manufacturing and environmental disturbances. Variability minimization has been an important issue in circuit design for many years [17] (see [17] for earlier reference). It leads (indirectly) to parametric yield improvement. Circuits characterized by low performance variability are regarded as high quality products. Most recently, variability minimization has been reintroduced into practical industrial design

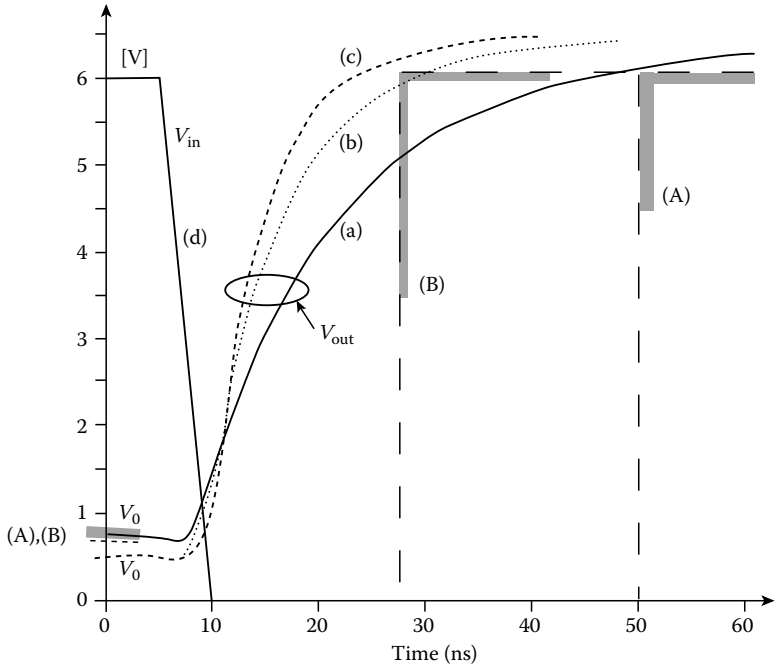


FIGURE 5.19 Nominal transient responses for Example 5.3: (a) initial, (b) after first optimization, and (c) after tightening the specs and second optimization.

due to the work of Taguchi [30]. He has successfully popularized the notion of ''Off-Line Quality Control'' through an intensive practical implementation of his strategy of designing products with low performance variability, tuned to the ''target'' values S_j^T of the performance functions y_j.

In what follows, a generalized approach proposed in [44] is discussed, in which a broad range of various problems, including yield optimization and Taguchi's variability minimization as special cases can be solved, using the SA approach and the general gradient formulas developed previously in this section.

It is convenient to introduce the M-dimensional vector $g(e)$ of scaled constraints, composed of the vectors $g^L(e)$, and $g^U(e)$, defined as

$$g_k^L(e) = \frac{S_k^T - y_k(e)}{S_k^T - S_k^L}, \quad k = 1, \ldots, M_L \tag{5.27}$$

$$g_t^U(e) = \frac{y_t(e) - S_t^T}{S_t^U - S_t^T}, \quad t = 1, \ldots, M_U \tag{5.28}$$

where $e = e(x, \theta)$, $M = M_L + M_U \leq 2$ m (note that in general $M \leq 2$ m, since some of the lower or upper specs might not be defined). These constraints are linear functions of $y_i(\cdot)$; they have important properties: $g^L = g^U = 0$ if $y = S^T$, $g^L = 1$ if $y = S^L$, and $g^U = 1$ if $y = S^U$. For $S^L < y < S^U$ and $y \neq S^T$, either g^L or g^U is greater than zero, but never both.

For any ''good'' design, we would like to make each of the normalized constraints introduced previously equal to zero, which might not be possible due to various existing design trade-offs. Therefore, this is a typical example of a multiobjective optimization problem. The proposed scaling helps to compare various trade-off situations. The Taguchi ''on-target'' design with variability minimization is formulated as follows (for a single performance function $y(e)$)

$$\begin{aligned}\text{minimize}\{M_{\text{TAG}}(x) &= E_\theta\left\{\left[y(e(x,\theta)) - S^{\text{T}}\right]^2\right\} \\ &= var\{y(x)\} + \left[\bar{y}(x) - S^{\text{T}}\right]^2\right\}\end{aligned} \tag{5.29}$$

where

$\bar{y}(x)$ is the mean (average) value of y (w.r.t θ)

M_{TAG} is called the "loss" function (in actual implementations Taguchi uses related but different performance statistics)

Generalization of Equation 5.29 to several objectives can be done in many different ways. One possible approach is to introduce $u(e) = \max_{s=1,\ldots,M}\{g_s(e)\}$ and define the generalized loss function (GLF) as

$$M_{\text{GLF}}(x) = E_\theta\left\{[u(e(x,\theta))]^2\right\} \tag{5.30}$$

This generalization is meaningful, since $u(e) = \max_{s=1,\text{K},M}\{g_s(\text{e})\}$ is a scalar, and for the proposed scaling either $u(\cdot) > 0$, or it is equal to zero if all the performances assume their target values S_i^{T}. Since Equations 5.27 and 5.28 are less or equal to 1 any time $S_i^{\text{T}} \le y_i\ (\text{e}) \le S_i^{\text{U}}$, then $e = e(x,\ \theta)$ belongs to A (the acceptability region in the e-space), if $u(e) < 1$. Let us use the complementary indicator function $\phi_{\text{F}}[u\ (\text{e})] = 1$, if $u(e) \ge 1$, i.e., $e \notin A$ (or $\theta \notin A_\theta(\text{x})$) (failure) and equal to zero otherwise. Maximization of the parametric yield $Y(x)$ is equivalent to the minimization of the probability of failures $F(x) = 1 - Y(x)$, which can be formulated as the following minimization process, w.r.t. x:

$$\left.\begin{aligned}\text{minimize}\{F(x) &= P\{\theta \notin A_\theta(x)\} \\ &= \int_{R^t} \phi_{\text{F}}(u(x,\theta)) f_\theta(\theta)\text{d}\theta = E_\theta[\phi_{\text{F}}(u(x,\theta))]\end{aligned}\right\} \tag{5.31}$$

The scalar "step" function $\phi_{\text{F}}(\cdot)$ (where the argument is also scalar function $u(e) = \max_{s=1,\text{K},M}\{g_s(\text{e})\}$) can now be generalized into a scalar weight function $w(\cdot)$, in the same spirit as in the Zadeh fuzzy set theory [62,63]. For further generalization, the original p.d.f. $f_\theta(\theta)$ used in Equation 5.31 is parameterized, multiplying θ by the smoothing parameters β to control the dispersion of e, which leads to the following optimization problem, utilizing a generalized measure $M_w(\text{x}, \beta)$

$$\text{minimize}\{M_w(x,\beta) = E_\theta\{w[u(e(x,\beta\theta))]\}\} \tag{5.32}$$

where $u(e(x,\ \beta\theta)) = \max_{s=1,\ldots,M}\{g_s(e(x,\ \beta\theta))\}$. If $0 \le w(\cdot) \le 1$, $M_w(\cdot)$ corresponds to the probability measure of fuzzy event introduced by Zadeh [63]. The choice of $w(\cdot)$ and β leads to different optimization problems and different algorithms for yield/variability optimization. The standard yield optimization problem results if $w(\alpha) = \phi_{\text{F}}(\alpha)$ and $\beta = 1$ [Equation 5.31] (Figure 5.20a). Yield can be also approximately optimized indirectly, using a "smoothed" (e.g., sigmoidal) membership function $w(\alpha)$ (Figure 5.20e). For variability minimization, we have a whole family of possible approaches: (1) The generalized Taguchi approach with $w(\alpha) = \alpha^2$ (see Equation 5.30) and $\beta = 1$ (see Figure 5.20c). (2) If $w(\alpha) = \alpha$ and $\beta = 1$ is kept constant, we obtain a statistical mini–max problem, since the expected value of the max function is used; this formulation will also lead to performance variability reduction. (3) If $w(\alpha)$ is piecewise constant (Figure 5.20d), the approach is equivalent to the income index maximization with separate quality classes, introduced in [28], and successfully used there for the increase of the percentage of circuits belonging to the best classes (i.e., those characterized by small values of $u(e)$); this approach also reduces variability of circuit performance functions. For all the cases, optimization is performed using the proposed SA approach. The smoothed functional gradient formulas (Equation 5.25 and 5.26)

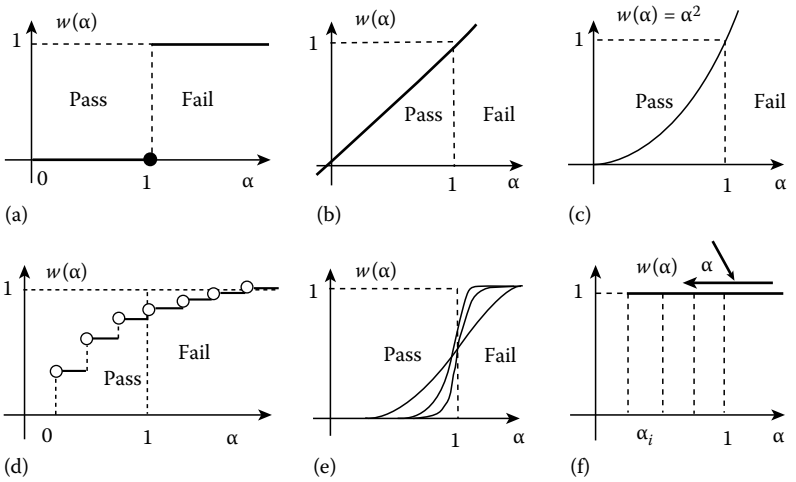


FIGURE 5.20 Various cases of the weight function $w(\alpha)$.

have to be used, in general, since the ''discrete-circuit'' type gradient formulas are valid only for a limited number of situations.

Example 5.4: Performance Variability Minimization for a MOSFET-C Filter

The objective of this example [31] was to reduce performance variability and to tune to target values the performances of the MOSFET-C filter in Figure 5.21. The MOSFETS are used in groups composed of four

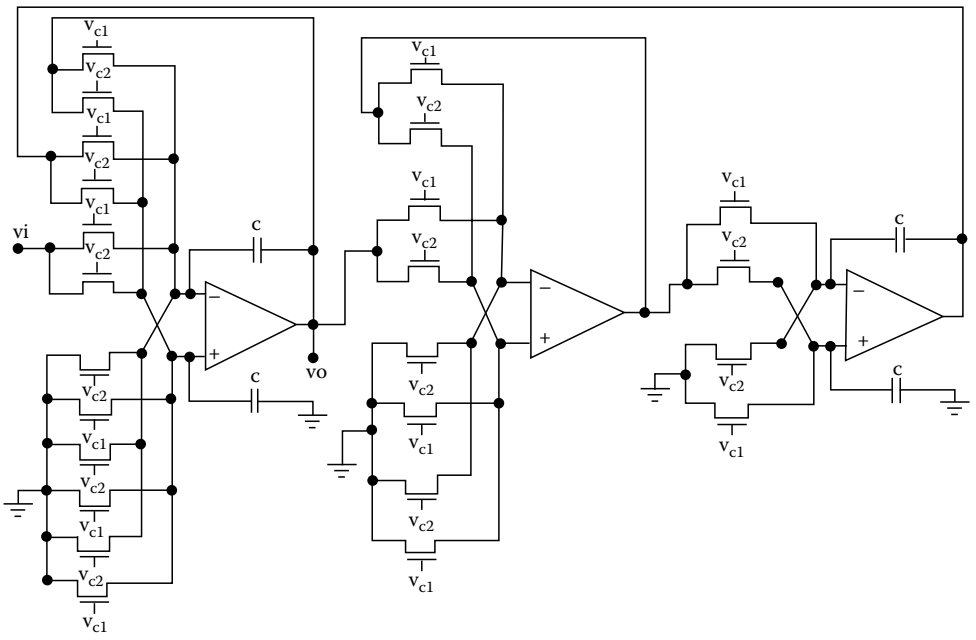


FIGURE 5.21 MOSFET-C bandpass filter optimized in Example 5.4.

transistors implementing equivalent resistors with improved linearity. Design specifications are f_0, the center frequency, with the S^L, S^T, S^U values specified as {45, 50, 55} kHz; H_0, the voltage gain at f_0, with the specs {15, 20, 25} dB; and Q, the pole quality factor, with the specs {8, 10, 12}. The Taguchi-like "on-target" design and tuning was performed by minimizing the generalized measure Equation 5.32 with $w(x) = x^2$, using the SA approach with convolution smoothing, since the designable and random parameters are in different (sub)spaces. The circuit has the total number of 90 transistors so its direct optimization using transistor-level simulation would be too costly. Therefore, in [31] the operational amplifiers (op-amps) shown in Figure 5.21 were modeled by a statistical macromodel, representing the most relevant op-amp characteristics: DC gain A_0, output resistance R_0, and the −20 dB/dec frequency roll-off. Transistor model parameters were characterized by a statistical model developed in [8], based on six common factors (see Section 5.3): t_{ox} (oxide thickness), ΔL_n, ΔL_p (length reduction), $N_{SUB,n}$, $N_{SUB,p}$ (substrate doping), and x_{jp} (junction depth). All other transistor model parameters were calculated from the six common factors using second-order regression formulas developed in [8]. The major difficulty was to create statistical models of the op-amp macromodel parameters A_0, R_0, and f_{3dB} as the functions of the six common factors listed previously. A special extraction procedure (similar in principle to the model of factor analysis [20]) was developed and the relevant models created. Perfect matching between the transistor model parameter of the 22 transistors of each of the op-amps was assumed for simplicity* (correlational dependencies between individual parameters for individual transistors were maintained using the statistical mode of [8], described earlier). Moreover, perfect matching between the three macromodels (for the three op-amps) was also assumed. Mismatches between the threshold voltages and K_p (gain) coefficients of all the transistors in Figure 5.21 were taken into account, introducing 48 additional noise parameters. So, including the 6 global noise parameters discussed previously, the overall number of $\boldsymbol{\theta}$ parameters was 54. Moreover, it was found that even if a large number of noise parameters had small individual effect, their total contributions was significant, so, no meaningful reduction of the dimension of the $\boldsymbol{\theta}$-space was justified. Because of that, it was difficult to use approximating models in the $\boldsymbol{\theta}$-space, and even more difficult in the joint $(x, \boldsymbol{\theta})$ space of $7 + 54 = 61$ parameters. Therefore, no approximation was used.

The Monte Carlo studies showed that the proposed statistical macromodel provided quite reasonably statistical accuracy for f_0 (<1% errors for both the mean and the standard deviation), and <9.5% errors for both, H_0 and Q. The macromodel-based analysis was about 30 times faster than using the full device-level SPICE analysis. Twenty-five designable parameters were selected, including transistor channel length of the "resistor-simulating transistors," four capacitors, and one of the gate voltages. Due to the designable parameter tracking for transistor quadruples, the actual vector of optimized parameters was reduced to 7: $x = (V_{Gs}, C_1, C_2, L_2, L_3, L_4, L_5)$, where V_{Gs} denotes the gate voltage, and C_i, L_j are capacitor and channel length values, respectively. The original yield was 43.5% and relative standard deviations for H_0, f_0, and Q were reduced to 3.98%, 7.24%, and 22.06%, respectively, and all nominal values were close to their target values. After SA optimization with convolution smoothing involving 5% perturbations for each of the x parameters, and the total of about 240 circuit analyses, yield increased to 85.0%, and H_0, f_0, and Q relative standard deviations were reduced to 3.87%, 6.35%, and 13.91%, respectively. Therefore, the largest variability reduction (for Q) was about 37%, with simultaneous significant yield increase.

The preceding example demonstrates a typical approach that has to be taken in the case of large analog circuits: the circuit has to be (hierarchically) macromodeled first† and suitable *statistical* macromodels have to be created, including mismatch modeling.

5.5 Conclusion

Several different techniques of statistical yield optimization were presented. Other approaches to yield estimation and optimization can be found, e.g., in [21] (process optimization), and in [15,57,59,60,61]. It was also shown that yield, variability minimization, Toguchi design, and other approaches can be generalized into one methodology called design for quality‡ (DFQ). DFQ is a quickly growing area [30],

* A more sophisticated model, taking the mismatches into account, was later proposed in [32].

† Behavioral models can also be used.

‡ Other approaches to yield generalization/Design for Quality were presented in [13,28,58].

where the major objective is not to maximize the parametric yield as the only design criterion (yield is often only vaguely defined), but to minimize the performance variability around the designer specified target performance values. This has been a subject of research for many years using a sensitivity-based approach, and most recently, using the Taguchi methodology, based on some of the techniques of Design of Experiments rather than on the use of sensitivities. To increase the efficiency of the (mostly manual) Taguchi techniques [30], some automated methods started to appear, such as the generalized methodology described previously, or the automated approach based on capability indices C_p/C_{pk} (used extensively in process quality control) proposed in [41].

Statistical design optimization is still an active research area, but several mature techniques have been already developed and practically applied to sophisticated industrial IC design. This is of great importance to the overall manufacturing cost reduction, circuit quality improvement, and shortening of the overall IC design cycle.

References

1. H. L. Abdel-Malek and J. W. Bandler, Yield optimization for arbitrary statistical distributions: Part I—Theory, *IEEE Trans. Circuits Syst.*, CAS-27(4): 245–253, April 1980.
2. K. J. Antreich and R. K. Koblitz, Design centering by yield prediction, *IEEE Trans. Circuits Syst.*, CAS-29: 88–95, February 1982.
3. P. Balaban and J. J. Golembeski, Statistical analysis for practical circuit design, *IEEE Trans. Circuits Syst.*, CAS-22(2): 100–108, February 1975.
4. J. W. Bandler, Optimization of design tolerance using nonlinear programming, *J. Optimization Theory Appl.*, 14: 99, 1974; also in *Proceedings of the Princeton Conference on Information Science and Systems*, p. 655, Princeton, NJ, February 1972.
5. J. W. Bandler and H. L. Abdel-Malek, Optical centering, tolerancing and yield determination via updated approximations and cuts, *IEEE Trans. Circuits Syst.*, CAS-25: 853–871, 1978.
6. J. W. Bandler, P. C. Liu, and H. Tromp, A nonlinear programming approach to optimal design centering, tolerancing and tuning, *IEEE Trans. Circuits Syst.*, CAS-23: 155, March 1976.
7. P. W. Becker and F. Jensen, *Design of Systems and Circuits for Maximum Reliability or Maximum Production Yield*, New York: McGraw-Hill, 1977.
8. J. Chen and M. A. Styblinski, A systematic approach of statistical modeling and its application to CMOS circuits, in *Proceedings of the IEEE International Symposium on Circuits and Systems '93*, pp. 1805–1808, Chicago, IL, May 1993.
9. P. Cox, P. Yang, S. S. Mahant-Shetti, and P. Chatterjee, Statistical modeling for efficient parametric yield estimation of MOS VLSI circuits, *IEEE Trans. Electron Devices*, ED-32: 471–478, February 1985.
10. S. W. Director and G. D. Hatchel, The simplicial approximation to design centering, *IEEE Trans. Circuits Syst.*, CAS-24(7): 363–372, July 1977.
11. S. W. Director and G. D. Hatchel, A point basis for statistical design, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, New York, 1978.
12. S. W. Director, W. Maly, and A. J. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement,* Boston, MA: Kluwer Academic, 1990.
13. P. Feldman and S. W. Director, Integrated circuit quality optimization using surface integrals, *IEEE Trans. Comput. Aided Des.*, 12(12): 1868–1879, December 1993.
14. P. H. Gill, W. Murray, and M. H. Wright, *Practical Optimizations*, San Diego, CA: Academic Press, 1981.
15. D. E. Hocevar, P. F. Cox, and P. Yang, Parametric yield optimization for MOS circuit blocks, *IEEE Trans. Comput. Aided Des.*, 7(6): 645–658, June 1988.
16. D. E. Hocevar, M. R. Lightner, and T. N. Trick, A study of variance reduction techniques of estimating circuit yields, *IEEE Trans. Comput. Aided Des.*, CAD-2(3): 180–192, July 1983.

17. A. Ilumoka, N. Maratos, and R. Spence, Variability reduction: Statistically based algorithms for reduction of performance variability of electrical circuits, *IEEE Proc.*, 129, Part G(4): 169–180, August 1982.
18. G. Kjellstrom and L. Taxen, Stochastic optimization in system design, *IEEE Trans. Circuits Syst.*, CAS-28: 702–715, July 1981.
19. G. Kjellstrom, L. Taxen, and P. O. Lindberg, Discrete optimization of digital filter using Gaussian adaptation and quadratic function minimization, *IEEE Trans. Circuits Syst.*, CAS-34(10): 1238–1242, October 1987.
20. D. N. Lawley and A. E. Maxwell, *Factor Analysis as a Statistical Method*, New York: Elsevier, 1971.
21. K. K. Low and S. W. Director, An efficient methodology for building macromodels of IC fabrication processes, *IEEE Trans. Comput. Aided Des.*, 8(12): 1299–1313, December 1989.
22. W. Maly and A. J. Strojwas, Statistical simulation of IC manufacturing process, *IEEE Trans. Comput. Aided Des.*, Vol. CAD-1: 120–131, July 1982.
23. C. Michael and M. Ismail, *Statistical Modeling for Computer-Aided Design of MOS VLSI Circuits*, Boston, MA: Kluwer Academic, 1993.
24. S. R. Nassif, A. J. Strojwas, and S.W. Director, FABRICS II, *IEEE Trans. Comput. Aided Des.*, CAD-3: 40–46, January 1984.
25. J. Ogrodzki, L. Opalski, and M. A. Styblinksi, Acceptability regions for a class of linear networks, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Houston, TX, pp. 187–190, May 1980.
26. J. Ogrodzki and M. A. Styblinski, Optimal tolerancing, centering and yield optimization by one-dimensional orthogonal search (ODOS) technique, in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD)*, Vol. 2, pp. 480–485, Warsaw, Poland, September 1980.
27. L. Opalski, M. A. Styblinski, and J. Ogrodzki, An orthogonal search approximation to acceptability regions and its application to tolerance problems, in *Proceedings of the Conference on SPACECAD*, Bologna, Italy, September 1979.
28. L. J. Opalski and M. A. Styblinski, Generalization of yield optimization problem: Maximum income approach, *IEEE Trans. Comput. Aided Des. ICAS*, CAD-5(2): 346–360, April 1986.
29. M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, Matching properties of MOS transistors, *IEEE J. Solid State Circuits*, 24: 1334–1362, October 1989.
30. M. S. Phadke, *Quality Engineering Using Robust Design*, Englewood Cliff, NJ: Prentice Hall, 1989.
31. M. Qu and M. A. Styblinski, Hierarchical approach to statistical performance improvement of CMOS analog circuits, in *SRC TECHCON'93*, Atlanta, GA, pp. 1809–1812, September 1993.
32. M. Qu and M. A. Styblinski, Statistical characterization and modeling of analog functional blocks, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, May–June 1994.
33. H. Robins and S. Monro, A stochastic approximation method, *Annal. Math. Stat.*, 22: 400–407, 1951.
34. R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, New York: John Wiley & Sons, 1981.
35. A. Ruszczynski and W. Syski, Stochastic approximation algorithm with gradient averaging for constrained problems, *IEEE Trans. Autom. Control*, AC28: 1097–1105, December 1983.
36. K. Singhal and J. F. Pinel, Statistical design centering and tolerancing using parametric sampling, *IEEE Trans. Circuits Syst.*, CAS-28: 692–702, July 1981.
37. R. S. Soin and R. Spence, Statistical exploration approach to design centering, in *Proc. Inst. Elect. Eng.*, 127, Part G: 260–263, 1980.
38. J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Trans. Autom. Control*, 37(3): 332–341, 1992.
39. R. Spence and R. S. Soin, *Tolerance Design of Electronic Circuits*, Electronic Systems Engineering Series, Reading, MA: Addison-Wesley, 1988.
40. W. Strasz and M. A. Styblinski, A second derivative Monte Carlo optimization of the production yield, in *Proceedings of the European Conference on Circuit Theory and Design*, Vol. 2, pp. 121–131, Warsaw, Poland, September 1980.

41. M. A. Styblinski and S. A. Aftab, IC variability minimization using a new C_p and C_{pk} based variability/performance measure, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, London, May–June 149–152, 1994.
42. M. A. Styblinski, Estimation of yield and its derivatives by Monte Carlo sampling and numerical integration in orthogonal subspaces, in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD)*, Vol. 2, pp. 474–479, Warsaw, Poland, September 1980.
43. M. A. Styblinski, Problems of yield gradient estimation for truncated probability density functions, *IEEE Trans. Comput. Aided Des. ICAS*, CAD-5(1): 30–38, January 1986 (special issue on statistical design of VLSI circuits).
44. M. A. Styblinski, Generalized formulation of yield, variability, minimax and Taguchi circuit optimization problems, *Microelectron Reliability*, 34(1): 31–37, 1994.
45. M. A. Styblinski and S. A. Aftab, Combination of interpolation and self organizing approximation techniques—A new approach to circuit performance modeling, *IEEE Trans. Comput. Aided Des.*, 12(11): 1775–1785, November 1993.
46. M. A. Styblinski, J. Ogrodzki, L. Opalski, and W. Strasz, New methods of yield estimation and optimization and their application to practical problems (invited paper), in *Proceedings of the International Symposium on Circuits and Systems*, Chicago, IL, 1981.
47. M. A. Styblinski and L. J. Opalski, A random perturbation method for IC yield optimization with deterministic process parameters, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 977–980, Montreal, Canada, May 7–10, 1984.
48. M. A. Styblinski and L. J. Opalski, Algorithms and software tools for IC yield optimization based on fundamental fabrication parameters, *IEEE Trans. Comput. Aided Des. ICAS*, CAD-5(1): 79–89, January 1986 (special issue on statistical design of VLSI circuits).
49. M. A. Styblinski and A. Ruszczynski, Stochastic approximation approach to production yield optimizaion, in *Proceedings of the 25th Midwest Symposium on Circuits and Systems*, Houghton, MI, August 30–31, 1982.
50. M. A. Styblinski and A. Ruszczynski, Stochastic approximation approach to statistical circuit design, *Electron Lett.*, 19(8): 300–302, April 14, 1983.
51. M. A. Styblinski and W. Strasz, A second derivative Monte Carlo optimization of the production yield, in *Proceedings of the European Conference on Circuit Theory and Design (ECCTD'80)*, Vol. 2, pp. 121–131, Warsaw, Poland, September 1980.
52. M. A. Styblinski and T.-S. Tang, Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing, *Neural Networks J.*, 3(4): 467–483, 1990.
53. M. A. Styblinski and J. C. Zhang, Orthogonal array approach to gradient based yield optimization, in *Proceedings of the International Symposium on Circuits and Systems*, pp. 424–427, New Orleans, LA, May 1990.
54. K. S. Tahim and R. Spence, A radial exploration algorithm for the statistical analysis of linear circuits, *IEEE Trans. Circuits Syst.*, CAS-27(5): 421–425, May 1980.
55. T.-S. Tang and M. A. Styblinski, Yield optimization for non-differentiable density functions using convolution techniques, *IEEE Trans. Comput. Aided Des. IC Syst.*, 7(10): 1053–1067, 1988.
56. W. J. Welch, T.-K. Yu, S. M. Kang, and J. Sacks, Computer experiments for quality control by parameter design, J. *Qual. Technol.*, 22(1): 15–22, January 1990.
57. P. Yang, D. E. Hocevar, P. F. Cox, C. Machala, and P. K. Chatterjee, An integrated and efficient approach for MOS VLSI statistical circuit design, *IEEE Trans. Comput. Aided Des. VLSI Circuits Syst.*, CAD-5: 5–14, January 1986.
58. D. L. Young, J. Teplik, H. D. Weed, N. T. Tracht, and A. R. Alvarez, Application of statistical design and response surface methods to computer-aided VLSI device design II: Desirability functions and Taguchi methods, *IEEE Trans. Comput. Aided Des.*, 10(1): 103–115, January 1991.

59. T. K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick, Statistical performance modeling and parametric yield estimation of MOS VLSI, *IEEE Trans. Comput. Aided Des. VLSI Circuits Syst*., CAD-6(6): 1013–1022, November 1987.
60. T. K. Yu, S. M. Kang, I. N. Hajj, and T. N. Trick, iEDISON: An interactive statistical design tool for MOS VLSI circuits, in *IEEE International Conference on Computer-Aided Design (ICCAD-88)*, pp. 20–23, Santa Clara, CA, November 7–10, 1988.
61. T. K. Yu, S. M. Kang, J. Sacks, and W. J. Welch, Parametric yield optimization of MOS integrated circuits by statistical modeling of circuit performances, Technical Report No. 27, Department of Statistics, University of Illinois, Champaign, IL, July 1989.
62. L. A. Zadeh, Fuzzy sets, *Info. Control*, 8: 338–353, 1965.
63. L. A. Zadeh, Probability measures of fuzzy events, *J. Math. Anal. Appl*., 23: 421–427, 1968.

6

Physical Design Automation*

6.1 Introduction **6**-1
6.2 Very Large-Scale Integration Design Cycle **6**-2
6.3 Physical Design Cycle **6**-4
6.4 Design Styles **6**-6
6.5 Partitioning **6**-9
Classification of Partitioning Algorithms • Kernighan–Lin Partitioning Algorithm
6.6 Other Partitioning Algorithms **6**-12
6.7 Placement **6**-12
Classification of Placement Algorithms • Simulated Annealing Placement Algorithm • Other Placement Algorithms
6.8 Routing **6**-15
6.9 Classification of Global Routing Algorithms **6**-16
6.10 Classification of Detailed Routing Algorithms **6**-16
Lee's Algorithm for Global Routing • Greedy Channel Router for Detailed Routing • Other Routing Algorithms
6.11 Compaction **6**-21
Classification of Compaction Algorithms • Shadow-Propagation Algorithm for Compaction • Other Compaction Algorithms
6.12 Summary **6**-23
References **6**-24

Naveed A. Sherwani
Intel Corporation

6.1 Introduction

In the last three decades integrated circuit (IC) fabrication technology has evolved from integration of a few transistors in small-scale integration (SSI) to integration of several million transistors in very large-scale integration (VLSI). This phenomenal progress has been made possible by automating the process of design and fabrication of VLSI chips.

Integrated circuits consist of a number of electronic components, built by layering several different materials in a well-defined fashion on a silicon base called a wafer. The designer of an IC transforms a circuit description into a geometric description, which is known as a layout. A layout consists of a set of planar geometric shapes in several layers. The layout is then checked to ensure that it meets all the design

* The material presented in this chapter was adapted from the author's book *Algorithms for VLSI Physical Design Automation*, Boston: Kluwer Academic, 1993, with editorial changes for clarity and continuity. Copyright permission has been obtained from Kluwer Academic Publishers.

requirements. The result is a set of design files in a particular unambiguous representation, known as an intermediate form, which describes the layout. The design files are then converted into pattern generator files, which are used to produce patterns called masks by an optical pattern generator. During fabrication, these masks are used to pattern a silicon wafer using a sequence of photolithographic steps. The component formation requires very exacting details about geometric patterns and separation between them. The process of converting the specifications of an electrical circuit into a layout is called the physical design. It is an extremely tedious and error-prone process because of the tight tolerance requirements and the minuteness of the individual components. Currently, the smallest geometric feature of a component can be as small as 0.35 μm (1 $\mu m = 1.0 \times 10^{-6}$ m). However, it is expected that the feature size can be reduced to 0.1 μm within 5 years. This small feature size allows fabrication of as many as 4.5 million transistors on a 25×25 mm maximum size chip. Due to the large number of components and the exacting details required by the fabrication process, the physical design is not practical without the help of computers. As a result, almost all phases of physical design extensively use computer-aided design (CAD) tools and many phases have already been partially or fully automated. This automation of the physical design process has increased the level of integration, reduced the turnaround time, and enhanced chip performance.

VLSI physical design automation is essentially the study of algorithms related to the physical design process. The objective is to study optimal arrangements of devices on a plane (or in a three-dimensional space) and efficient interconnection schemes between these devices to obtain the desired functionality. Because space on a wafer is very expensive real estate, algorithms must use the space very efficiently to lower costs and improve yield. In addition, the arrangement of devices plays a key role in determining the performance of a chip. Algorithms for physical design must also ensure that all the rules required by the fabrication are observed and that the layout is within the tolerance limits of the fabrication process. Finally, algorithms must be efficient and should be able to handle very large designs. Efficient algorithms not only lead to fast turnaround time, but also permit designers to iteratively improve the layouts.

With the reduction in the smallest feature size and increase in the clock frequency, the effect of electrical parameters on physical design will play a more dominant role in the design and development of new algorithms.

In this section, we present an overview of the fundamental concepts of VLSI physical design automation. Different design styles are discussed in Section 6.4. Section 6.2 discusses the design cycle of a VLSI circuit. In Section 6.3 different steps of the physical design cycle are discussed. The rest of the sections discuss each step of the physical design cycle.

6.2 Very Large-Scale Integration Design Cycle

Starting with a formal specification, the VLSI design cycle follows a series of steps and eventually produces a packaged chip. A flow chart representing a typical design cycle is shown in Figure 6.1.

1. *System specification*: The specifications of the system to be designed are exactly specified here. This necessitates creating a high-level representation of the system. The factors to be considered in this process include performance, functionality, and the physical dimensions. The choice of fabrication technology and design techniques are also considered. The end results are specifications for the size, speed, power, and functionality of the VLSI system to be designed.
2. *Functional design*: In this step, behavioral aspects of the system are considered. The outcome is usually a timing diagram or other relationships between subunits. This information is used to improve the overall design process and to reduce the complexity of the subsequent phases.
3. *Logic design*: In this step, the functional design is converted into a logical design, typically represented by Boolean expressions. These expressions are minimized to achieve the smallest logic design which conforms to the functional design. This logic design of the system is simulated and tested to verify its correctness.

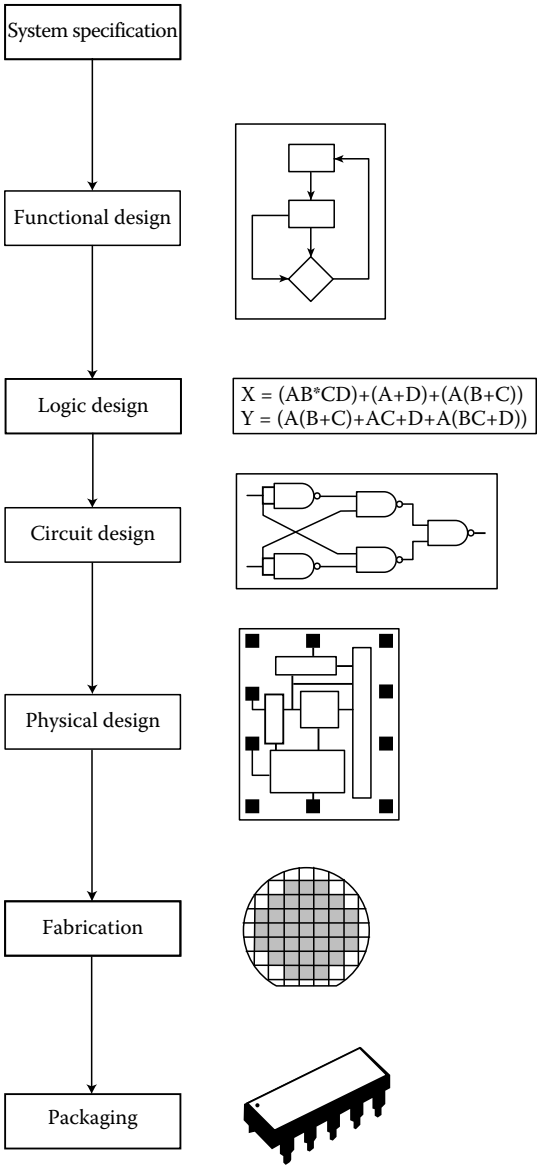


FIGURE 6.1 Design process steps.

4. *Circuit design*: Here, the Boolean expressions are converted into a circuit representation by taking into consideration the speed and power requirements of the original design. The electrical behavior of the various components are also considered in this phase. The circuit design is usually expressed in a detailed circuit diagram.
5. *Physical design*: In this step, the circuit representation of each component is converted into a geometric representation. This representation is in fact a set of geometric patterns which perform the intended logic function of the corresponding component. Connections between different components are also expressed as geometric patterns. As stated earlier, this geometric representation of a circuit is called a layout. The exact details of the layout also depend on design rules, which are guidelines based on the limitations of the fabrication process and the

electrical properties of the fabrication materials. Physical design is a very complex process; therefore, it is usually broken down into various substeps in order to handle the complexity of the problem. In fact, physical design is arguably the most time-consuming step in the VLSI design cycle.

6. *Design verification*: In this step, the layout is verified to ensure that the layout meets the system specifications and the fabrication requirements. Design verification consists of design rule checking (DRC) and circuit extraction. DRC is a process which verifies that all geometric patterns meet the design rules imposed by the fabrication process. After checking the layout for design rule violations and removing them, the functionality of the layout is verified by circuit extraction. This is a reverse engineering process and generates the circuit representation from the layout. This reverse engineered circuit representation can then be compared to the original circuit representation to verify the correctness of the layout.
7. *Fabrication*: After verification, the layout is ready for fabrication. The fabrication process consists of several steps: preparation of wafer, deposition, and diffusion of various materials on the wafer according to the layout description. A typical wafer is 10 cm in diameter and can be used to produce between 12 and 30 chips. Before the chip is mass produced, a prototype is made and tested.
8. *Packaging, testing, and debugging*: In this step, the chip is fabricated and diced in a fabrication facility. Each chip is then packaged and tested to ensure that it meets all the design specifications and that it functions properly. Chips used in printed circuit boards (PCBs) are packaged in a dual-in-line package (DIP) or pin grid array (PGA). Chips which are to be used in a multichip module (MCM) are not packaged because MCMs use bare or naked chips.

The VLSI design cycle is iterative in nature, both within a step and between steps. The representation is iteratively improved to meet system specifications. For example, a layout is iteratively improved so that it meets the timing specifications of the system. Another example may be detection of design rule violations during design verification. If such violations are detected, the physical design step needs to be repeated to correct the error. The objective of VLSI CAD tools is to minimize the number of iterations and thus reduce the time-to-market.

6.3 Physical Design Cycle

Physical design cycle converts a circuit diagram into a layout. This is accomplished in several steps such as partitioning, floorplanning, placement, routing, and compaction, as shown in Figure 6.2.

1. *Partitioning*: The complex task of chip layout is divided into several smaller tasks. A chip may contain several million transistors. Layout of the entire circuit cannot be handled due to the limitation of memory space as well as computation power available. Therefore, it is normally partitioned by grouping the components into blocks (subcircuits/modules). The actual partitioning process considers many factors such as size of the blocks, number of blocks, and number of interconnections between the blocks. The output of partitioning is a set of blocks along with the interconnections required between blocks. The set of interconnections required is referred to as a netlist. Figure 6.2a shows that the input circuit has been partitioned into three blocks. In large circuits the partitioning process is hierarchical and at the topmost level a chip may have between 5 and 25 blocks. Each module is then partitioned recursively into smaller blocks.
2. *Placement*: In this step, good layout alternatives are selected for each block, as well as the entire chip. The area of each block can be calculated after partitioning and is based approximately on the number and the type of components in that block. The actual rectangular shape of the block, which is determined by the aspect ratio may, however, be varied within a prespecified range. Floorplanning is a critical step, as it sets up the groundwork for a good layout. However, it is

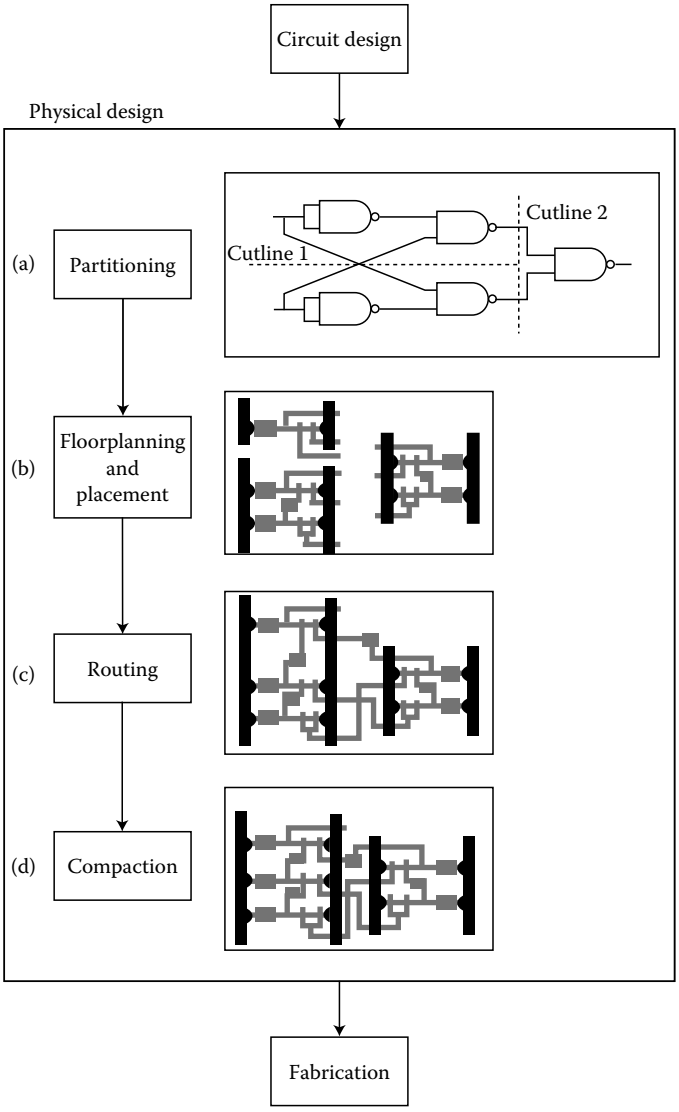


FIGURE 6.2 Physical design cycle.

computationally quite hard. Very often the task of floorplan layout is done by a design engineer, rather than by a CAD tool. This is sometimes necessary as the major components of an IC are often intended for specific locations on the chip. The placement process determines the exact positions of the blocks on the chip, so as to find a minimum area arrangement for the blocks that allows completion of interconnections between the blocks. Placement is typically done in two phases. In the first phase an initial placement is created. In the second phase the initial placement is evaluated and iterative improvements are made until the layout has minimum area and conforms to design specifications. Figure 6.2b shows that three blocks have been placed. It should be noted that some space between the blocks is intentionally left empty to allow interconnections between blocks. Placement may lead to unroutable design, i.e., routing may not be possible in the space provided. Thus, another iteration of placement is necessary. To limit the number of iterations of the placement algorithm, an estimate of the required routing space is used during the placement

phase. A good routing and circuit performance heavily depend on a good placement algorithm. This is due to the fact that once the position of each block is fixed, very little can be done to improve the routing and the overall circuit performance.

3. *Routing*: In this step, the objective is to complete the interconnections between blocks according to the specified netlist. First, the space not occupied by the blocks (called the routing space) is partitioned into rectangular regions called channels and switchboxes. The goal of a router is to complete all circuit connections using the shortest possible wire length and using only the channels and switchboxes. This is usually done in two phases, referred to as the global routing and detailed routing phases. In global routing, connections are completed between the proper blocks of the circuit disregarding the exact geometric details of each wire and pin. For each wire, the global router finds a list of channels which are to be used as a passageway for that wire. In other words, global routing specifies the ''loose route'' of a wire through different regions in the routing space. Global routing is followed by detailed routing, which completes point-to-point connections between pins on the blocks. Loose routing is converted into exact routing by specifying geometric information such as width of wires and their layer assignments. Detailed routing includes channel routing and switchbox routing. Routing is a very well-studied problem and several hundred articles have been published about all its aspects. Because almost all problems in routing are computationally hard, the researchers have focused on heuristic algorithms. As a result, experimental evaluation has become an integral part of all algorithms and several benchmarks have been standardized. Due to the nature of the routing algorithms, complete routing of all the connections cannot be guaranteed in many cases. As a result, a technique called rip-up and reroute is used, which basically removes troublesome connections and reroutes them in a different order. The routing phase of Figure 6.2c shows that all the interconnections between three blocks have been routed.
4. *Compaction*: In this step, the layout is compressed in all directions such that the total area is reduced. By making the chip smaller, wire lengths are reduced, which in turn reduces the signal delay between components of the circuit. At the same time, a smaller area may imply more chips can be produced on a wafer, which in turn reduces the cost of manufacturing. However, the expense of computing time mandates the extensive compaction is used only when large quantities of ICs are produced. Compaction must ensure that no rules regarding the design and fabrication process are violated during the process. The final diagram in Figure 6.2d shows the compacted layout.

Physical design, like VLSI design, is iterative in nature, and many steps such as global routing and channel routing are repeated several times to obtain a better layout. In addition, the quality of results obtained in a step depends on the quality of solutions obtained in earlier steps. For example, a poor quality placement cannot be ''cured'' by high quality routing. As a result, earlier steps have more influence on the overall quality of the solution. In this sense partitioning, floorplanning, and placement problems play a more important role in determining the area and chip performance, as compared to routing and compaction. Because placement may produce an ''unroutable'' layout, the chip might need to be replaced or repartitioned before another routing is attempted. In general, the whole design cycle may be repeated several times to accomplish the design objectives. The complexity of each step varies depending on the design constraints as well as the design style used.

6.4 Design Styles

Even after decomposing physical design into several conceptually easier steps, it has been shown that each step is computationally very hard. Market requirements demand a quick time-to-market and high yield. As a result, restricted models and design styles are used in order to reduce the complexity of

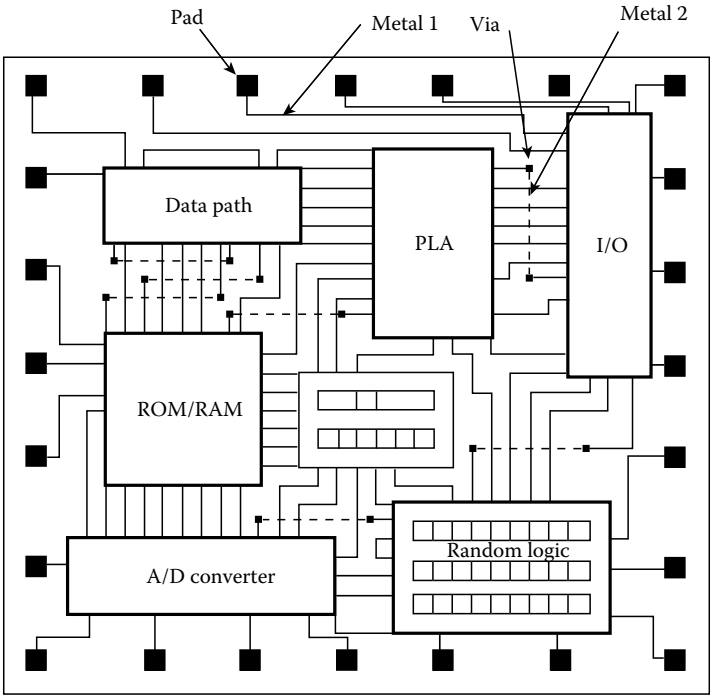


FIGURE 6.3 Full-custom structure.

physical design. This practice began in the late 1960s and led to the development of several restricted design styles [8].

The most general form of layout is called the full-custom design style, in which the circuit is partitioned into a collection of subcircuits according to some criteria such as functionality of each subcircuit. In this design style, each subcircuit is called a functional block or simply a block. The full-custom design style allows functional blocks to be of any size. Figure 6.3 shows an example of very simple circuit with few blocks. Internal routing in each block is not shown for the sake of clarity. Blocks can be placed at any location on the chip surface without restriction. In other words, this style is characterized by the absence of any constraints on the physical design process. This design style allows for very compact designs. However, the process of automating a full-custom design style has a much higher complexity than other restricted models. For this reason, it is used only when final design must have a minimum area and designing time is less of a factor. The automation process for a full-custom layout is still a topic of intensive research. Some phases of physical design of a full-custom chip may be done manually to optimize the layout. Layout compaction is a very important aspect in full-custom. The rectangular solid boxes around the boundary of the circuit are called I–O pads. Pads are used to complete interconnections between chips or interconnections between chip and the board. The space not occupied by blocks is used for routing of interconnecting wires. Initially all the blocks are placed within the chip area, with the objective of minimizing the total area. However, enough space must be left between the blocks to complete the routing. Usually several metal layers are used for routing interconnections. Currently, two metal layers are common for routing and the three-metal layer process is gaining acceptance, as the fabrication costs become more feasible. The routing area needed between the blocks becomes increasingly smaller as more routing layers are used. This is because some routing is done on top of the transistors in the additional metal layers. If all the routing can be done on top of the transistors, the total chip area is determined by the area of the transistors.

In a hierarchical design of circuit each block in full-custom design may be very complex and may consist of several subblocks, which in turn may be designed using the full-custom design style or other design styles. It is easy to see that because any block is allowed to be placed anywhere on the chip, the problem of optimizing area and interconnection of wires becomes difficult. Full-custom design is very time consuming; thus, the method is inappropriate for very large circuits, unless performance is of utmost importance. Full-custom is usually used for the layout of microprocessors.

A more restricted design style is called the standard cell design style. The design process in standard cell design style is simpler than a full-custom design style. Standard cell methodology considers the layout to consist of rectangular cells of the same height. Initially, a circuit is partitioned into several smaller blocks, each of which is equivalent to some predefined subcircuit (cell). The functionality and the electrical characteristics of each predefined cell are tested, analyzed, and specified. A collection of these cells is called a cell library, usually consisting of 200–400 cells. Terminals on cells may be located either on the boundary or in the center of the cells. Cells are placed in rows and the space between two rows is called a channel. These channels are used to perform interconnections between cells. If two cells to be interconnected lie in the same row or in adjacent rows, then the channel between the rows is used for interconnection. However, if two cells to be connected lie in two nonadjacent rows, then their interconnection wire passes through the empty space between any two cells, or feedthrough.

Standard cell design is well suited for moderate-size circuits and medium production volumes. Physical design using standard cells is simpler as compared to full-custom and efficient using modern design tools. The standard cell design style is also widely used to implement the ''random logic'' of the full-custom design, as shown in Figure 6.3. While standard cell designs are developed more quickly, a substantial initial investment is needed in the development of the cell library, which may consist of several hundred cells. Each cell in the cell library is ''handcrafted'' and requires a highly skilled design engineer. Each type of cell must be created with several transistor sizes. Each cell must then be tested by simulation and its performance must be characterized. A standard cell design usually takes more area than a full-custom or a handcrafted design. However, as more metal layers become available for routing, the difference in area between the two design styles will gradually be reduced.

The gate array design style is a simplified version of the standard cell design style. Unlike the cells in standard cell designs, all the cells in gate array are identical. The entire wafer is prefabricated with an array of identical gates or cells. These cells are separated by both vertical and horizontal spaces called vertical and horizontal channels. The circuit design is modified such that it can be partitioned into a number of identical blocks. Each block must be logically equivalent to a cell on the gate array. The name ''gate array'' signifies the fact that each cell may simply be a gate, such as a three-input NAND gate. Each block in the design is mapped or placed onto a prefabricated cell on the wafer during the partitioning/placement phase, which is reduced to a block-to-cell assignment problem. The number of partitioned blocks must be less than or equal to that of the total number of cells on the wafer. Once the circuit is partitioned into identical blocks, the task is to make the interconnections between the prefabricated cells on the wafer using horizontal and vertical channels to form the actual circuit. The uncommitted gate array is taken into the fabrication facility and routing layers are fabricated on top of the wafer. The completed wafer is also called a customized wafer.

This simplicity of gate array design is gained at the cost of rigidity imposed upon the circuit both by the technology and the prefabricated wafers. The advantage of gate arrays is that the steps involved for creating any prefabricated wafer are the same, and only the last few steps in the fabrication process actually depend on the application for which the design will be used. Hence, gate arrays are cheaper and easier to produce than full-custom or standard cell. Similar to standard cell design, gate array is also a nonhierarchical structure. The gate array architecture is the most restricted form of layout. This also means it is the simplest for algorithms to work with. For example, the task of routing in gate array is to determine if a given placement is routable. The routability problem is conceptually simpler as compared to the routing problem in standard cell and full-custom design styles.

The choice of design style for a particular circuit depends on many factors such as functionality of the chip, time-to-market, and the volume of chips to be manufactured. Full-custom is typically reserved for high-performance, high-volume chips, while standard cells are used for moderate performance, where the cost of full-custom cannot be justified. Gate arrays are typically used for low-performance, low-cost applications. A design style may be applicable to the entire chip or a block of the chip.

Irrespective of the choice of the design style, all steps of the physical design cycle need to be carried out. However, the complexity, the effectiveness, and the algorithms used differ considerably depending on the design style. The following sections discuss algorithms for different phases of the physical design cycle.

6.5 Partitioning

As stated earlier, the basic purpose of partitioning is to simplify the overall design process. The circuit is decomposed into several subcircuits to make the design process manageable. This section considers the partitioning phase of the physical design cycle, study constraints, and objective functions for this problem and presents efficient algorithms.

Given a circuit, the partitioning problem is to decompose it into several subcircuits subject to constraints while optimizing a given objective function. The constraints for the partitioning problem include area constraints and terminal constraints. Area constraints are user specified for design optimization and terminal count depends on the area and aspect ratio of the block. In particular, the terminal count for a partition is given by the ratio of the perimeter of the partition to the terminal pitch. The minimum spacing between two adjacent terminals is called terminal pitch and is determined by the design rules. The objective functions for a partitioning problem include the minimization of the number of nets that cross the partition boundaries, and the minimization of the maximum number of times a path crosses the partition boundaries. The constraints and the objective functions used in the partitioning problem vary depending upon the partitioning level and the design style used. The actual objective function and constraints chosen for the partitioning problem may also depend on the specific problem. The number of nets that connect a partition to other partitions cannot be greater than the terminal count of the partition. In addition, the number of nets cut by partitioning should be minimized to simplify the routing task. The minimization of the number of nets cut by partitioning is one of the most important objectives in partitioning.

A disadvantage of the partitioning process is that it may degrade the performance of the final design. During partitioning, critical components should be assigned to the same partition. If such an assignment is not possible, then appropriate timing constraints must be generated to keep the two critical components close together. Usually, several components, forming a critical path, determine the chip performance. If each component is assigned to a different partition, the critical path may be too long. Minimizing the length of critical paths improves system performance.

After a chip has been partitioned, each of the subcircuits must be placed on a fixed plane and the nets between all the partitions must be interconnected. The placement of the subcircuits is done by the placement algorithms and the nets are routed by using routing algorithms.

6.5.1 Classification of Partitioning Algorithms

The partitioning algorithms also may be classified based on the nature of the algorithms, of which two types exist: deterministic algorithms and probabilistic algorithms. Deterministic algorithms produce repeatable or deterministic solutions. For example, an algorithm which makes use of deterministic functions will always generate the same solution for a given problem. On the other hand, the probabilistic algorithms are capable for producing a different solution for the same problem each time they are used, as they make use of some random functions.

The partitioning algorithms also may be classified on the basis of the process used for partitioning. Thus, we have group migration algorithms, simulated annealing and evolution-based algorithms, and other partitioning algorithms.

The group migration algorithms [9,17] start with some partitions, usually generated randomly, and then move components between partitions to improve the partitioning. The group migration algorithms are quite efficient. However, the number of partitions must be specified, which is usually not known when the partitioning process starts. In addition, the partitioning of an entire system is a multilevel operation, and the evaluation of the partitions obtained by the partitioning depends on the final integration of partitions at all levels, from the basic subcircuits to the whole system. An algorithm used to find a minimum cut at one level may sacrifice the quality of cuts for the following levels. The group migration method is a deterministic method that is often trapped at a local optimum and cannot proceed further.

The simulated annealing/evolution algorithms [3,10,18,26] carry out the partitioning process by using a cost function, which classifies any feasible solution, and a set of moves, which allows movement from solution to solution. Unlike deterministic algorithms, these algorithms accept moves which may adversely affect the solution. The algorithm starts with a random solution and as it progresses, the proportion of adverse moves decreases. These degenerate moves act as a safeguard against entrapment in local minima. These algorithms are computationally intensive as compared to group migration and other methods.

Among all the partitioning algorithms, the group migration and simulated annealing/evolution have been the most successful heuristics for partitioning problems. The group migration and simulated annealing/evolution methods are most widely used, and extensive research has been carried out in these two types of algorithms.

6.5.2 Kernighan–Lin Partitioning Algorithm

The Kernighan–Lin (K–L) algorithm is a bisectioning algorithm. It starts by initially partitioning the graph $G = (V, E)$ into two subsets of equal size. Vertex pairs are exchanged across the bisection if the exchange improves the cutsize. The preceding procedure is carried out iteratively until no further improvement can be achieved.

The basic idea of the K–L algorithm is illustrated with the help of an example before presenting the algorithm formally. Consider the example given in Figure 6.4a. The initial partitions are

$$A = \{1,2,3,4\}$$
$$B = \{5,6,7,8\}$$

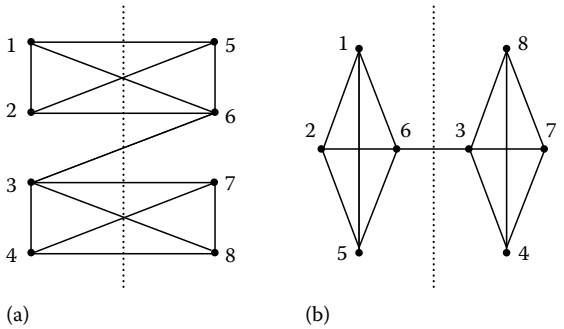


FIGURE 6.4 Graph bisected by K–L algorithm: (a) initial bisections and (b) final bisections.

TABLE 6.1 Log of the Vertex Exchanges

i	Vertex Pair	$g(i)$	$\sum_{j=1}^{i} g(i)$	Cutsize
0	—	—	—	9
1	(3, 5)	3	3	6
2	(4, 6)	5	8	1
3	(1, 7)	−6	2	7
4	(2, 8)	−2	0	9

Notice that the initial cutsize is 9. The next step of the K–L algorithm is to choose a pair of vertices whose exchange results in the largest decrease of the cutsize or results in the smallest increase, if no decrease is possible. The decrease of the cutsize is computed using gain values $D(i)$ of vertices v_i. The gain of a vertex v_i is defined as

$$D(i) = \text{inedge}(i) - \text{outedge}(i)$$

where

inedge(i) is the number of edges of vertex i that do not cross the bisection boundary
outedge(i) is the number of edges that cross the boundary

The amount by which the cutsize decreases, if vertex v_i changes over to the other partition, is represented by $D(i)$. If v_i and v_j are exchanged, the decrease of the cutsize is $D(i) + D(j)$. In the example given in Figure 6.4, a suitable vertex pair is (3, 5), which decreases the cutsize by 3. A tentative exchange of this pair is made. These two vertices are then locked. This lock on the vertices prohibits them from taking part in any further tentative exchanges. The preceding procedure is applied to the new partitions, which gives a second vertex pair of (4, 6). This procedure is continued until all the vertices are locked. During this process, a log of all tentative exchanges and the resulting cutsizes is stored. Table 6.1 shows the log of vertex exchanges for the given example. Note that the partial sum of cutsize decrease $g(i)$ over the exchanges of first i vertex pairs is given in the table; e.g., $g(1) = 3$ and $g(2) = 8$. The value of k for which $g(k)$ gives the maximum value of all $g(i)$ is determined from the table. In this example $k = 2$ and $g(2) = 8$ is the maximum partial sum. The first k pairs of vertices are actually exchanged. In the example the first two vertex pairs (3, 5) and (4, 6) are actually exchanged, resulting in the bisection shown in Figure 6.4b. This completes an iteration and a new iteration starts. However, if no decrease of cutsize is possible during an iteration, the algorithm stops. Figure 6.5 presents the formal description of the K–L algorithm.

The procedure INITIALIZE finds initial bisections, and initializes the parameters in the algorithm. The procedure IMPROVE tests if any improvement has been made during the last iteration, while the procedure UNLOCK checks if any vertex is unlocked. Each vertex has a state of either *locked* or *unlocked*. Only those vertices whose status is *unlocked* are candidates for the next tentative exchanges. The procedure TENT-EXCHGE tentatively exchanges a pair of vertices. The procedure LOCK locks the vertex pair, while the procedure LOG stores the log table. The procedure ACTUAL-EXCHGE determines the maximum partial sum of $g(i)$, selects the vertex pairs to be exchanged, and fulfills the actual exchange of these vertex pairs.

The time complexity of the K–L algorithm is $O(n^3)$. The K–L algorithm is, however, quite robust. It can accommodate additional constraints, such as a group of vertices requiring to be in a specified partition. This feature is very important in layout because some blocks of the circuit are to be kept together due to the functionality. For example, it is important to keep all components of an adder together. However, the K–L algorithm has several disadvantages. For example, the algorithm is not applicable for hypergraphs, it cannot handle arbitrarily weighted graphs and the partition sizes must be specified before partitioning. Finally, the complexity of the algorithm is considered too high even for moderate-size problems.

Algorithm *KL*

```
Begin
    INITIALIZE();
    while (IMPROVE (table) = TRUE) do
    (* if an improvement has been made during last iteration,
    the process is carried out again.*)
        while (UNLOCK(A) = TRUE) do
        (* if there exists any unlocked vertex in A,
        more tentative exchanges are carried out.*)
            for (each a∈A) do
                if (a = unlocked) then
                    for (each b ∈ B) do
                        if (b = unlocked) then
                            if (D_max < D(a) + D(b)) then
                                D_max = D(a) + D(b);
                                a_max = a;
                                b_max = b;
            TENT-EXCHGE(a_max, b_max);
            LOCK(a_max, b_max);
            LOG(table);
            D_max = −∞;
        ACTUAL-EXCHGE(table);
End;
```

FIGURE 6.5 Algorithm K–L.

6.6 Other Partitioning Algorithms

In order to overcome the disadvantages of the K–L algorithm, several extensions of the K–L algorithm such as the Fiduccia–Mattheyses algorithm, the Goldberg and Burstein algorithm, the component replication algorithm, and the ratio cut algorithm were developed. In the class of probabilistic and iterative algorithms simulated annealing and evolution-based algorithms have been developed for partitioning. For details on these partitioning algorithms, refer to Chapter 4 of [29].

6.7 Placement

The placement phase follows the partitioning phase of the physical design cycle. After the circuit has been partitioned, the area occupied by each block (subcircuit) can be calculated and the number of terminals (pins) required by each block is known. Partitioning also generates the netlist which specifies the connections between the blocks. The layout is completed by arranging the blocks on the layout surface and interconnecting their pins according to the netlist. The arrangement of blocks is done in the placement phase, while interconnection is completed in the routing phase. In the placement phase, blocks are assigned a specific shape and are positioned on a layout surface in such a fashion that no two blocks overlap and enough space is left on the layout surface to complete the interconnections between the blocks. The blocks are positioned so as to minimize the total area of the layout. In addition, the locations of pins on each block are also determined.

The input to the placement phase is a set of blocks, the number of terminals for each block, and the netlist. If the layout of the circuit within a block has been completed, then the dimensions of the block are also known. The blocks for which the dimensions are known are called fixed blocks and the blocks for which dimensions are yet to be determined are called flexible blocks. Thus, during the placement phase, we need to determine an appropriate shape for each block (if shape is not known), the location of each block on the layout surface, and determine the locations of pins on the boundary of the blocks. The

problem of assigning locations to the fixed blocks on a layout surface is called the placement problem. If some or all of the blocks are flexible, the problem is called the floorplanning problem. Hence, the placement problem is a restricted version of the floorplanning problem. The terminology is slightly confusing as floorplanning problems are placement problems as well, but these terminologies have been widely used and accepted. It is desirable that the pin locations are identified at the same time the block locations are fixed. However, due to the complexity of the placement problem, the problem of identifying the pin locations for the blocks is solved after the locations of all the blocks are known. This process of identifying pin locations is called pin assignment.

The placement phase is crucial in the overall physical design cycle because an ill-placed layout cannot be improved by high-quality routing. In other words, the overall quality of the layout in terms of area and performance is mainly determined in the placement phase.

6.7.1 Classification of Placement Algorithms

The placement algorithms can be classified on the basis of the input to the algorithms, the nature of output generated by the algorithms, and the process used by the algorithms.

Depending on the input, the placement algorithms can be classified into two major groups: constructive placement and iterative improvement methods. The input to the constructive placement algorithms consists of a set of blocks along with the netlist. The algorithm finds the locations of blocks. On the other hand, iterative improvement algorithms start with an initial placement. These algorithms modify the initial placement in search of a better placement. These algorithms are typically used in an iterative manner until no improvement is possible.

The nature of output produced by an algorithm is another way of classifying the placement algorithms. Some algorithms generate the same solution when presented with the same problem, i.e., the solution produced is repeatable. These algorithms are called deterministic placement algorithms. Algorithms that function on the basis of fixed connectivity rules (or formulas) or determine the placement by solving simultaneous equations are deterministic and always produce the same result for a particular placement problem. Some algorithms, on the other hand, work by randomly examining configurations and may produce a different result each time they are presented with the same problem. Such algorithms are called probabilistic placement algorithms.

The classification based on the process used by the placement algorithms is perhaps the best way of classifying these algorithms. Two important classes of algorithms come under this classification: simulation-based algorithms and partitioning-based algorithms. Simulation-based algorithms simulate some natural phenomenon while partitioning-based algorithms use partitioning for generating the placement. The algorithms which use clustering and other approaches are classified under ''other'' placement algorithms.

6.7.2 Simulated Annealing Placement Algorithm

Simulated annealing is one of the most well-developed methods available [2,10–12,16,19,23,25–28]. The simulated annealing technique has been successfully used in many phases of VLSI physical design, e.g., circuit partitioning. A detailed description of the application of this method to partitioning may be found in Chapter 4 of [29]. Simulated annealing is used in placement as an iterative improvement algorithm. Given a placement configuration, a change to that configuration is made by moving a component or interchanging locations of two components. In the case of the simple pairwise interchange algorithm, it is possible that an achieved configuration has a cost higher than that of the optimum, but no interchange can cause a further cost reduction. In such a situation, the algorithm is trapped at a local optimum, and cannot proceed further. Actually, this happens quite often when this algorithm is used in real-life examples. Simulated annealing avoids getting stuck at a local optimum by occasionally accepting moves that result in a cost increase.

Algorithm *Simulated Annealing*

```
Begin
    temp = INIT-TEMP;
    place = INIT-PLACEMENT;
    while (temp > FINAL-TEMP) do
      while (inner_loop_criterion = FALSE) do
        new_place = PERTURB (place);
        ΔC = COST(new_place) - COST (place);
        if (ΔC < 0) then
          place = new_place;
        else if (RANDOM(0,1) > e^(Δc/T)) then
          place = new_place;
      temp = SCHEDULE(temp);
End;
```

FIGURE 6.6 Algorithm simulated annealing.

In simulated annealing, all moves that result in a decrease in cost are accepted. Moves that result in an increase in cost are accepted with a probability that decreases over the iterations. The analogy to the actual annealing process is heightened with the use of a parameter called temperature T. This parameter controls the probability of accepting moves that result in an increased cost. Additional moves are accepted at higher values of temperature than at lower values. The acceptance probability can be given by $e^{\Delta C/T}$, where ΔC is the increase in cost. The algorithm starts with a very high value of temperature, which gradually decreases so that moves that increase cost have a lower probability of being accepted. Finally, the temperature reduces to a very low value which causes only moves that reduce cost to be accepted. In this way the algorithm converges to an optimal or near-optimal configuration.

In each stage, the configuration is shuffled randomly to obtain a new configuration. This random shuffling could be achieved by displacing a block to a random location, an interchange of two blocks, or any other move that can change the wire length. After the shuffle, the change in cost is evaluated. If a decrease in cost occurs, the configuration is accepted; otherwise, the new configuration is accepted with a probability that depends on the temperature. The temperature is then lowered using some function which, for example, could be exponential in nature. The process is stopped when the temperature has dropped to a certain level. The outline of the simulated annealing algorithm is shown in Figure 6.6.

The parameters and functions used in a simulated annealing algorithm determine the quality of the placement produced. These parameters and functions include the cooling schedule consisting of initial temperature (*init_temp*), final temperature (*final_temp*), and the function used for changing the temperature (SCHEDULE), *inner_loop_criterion*, which is the number of trials at each temperature, the process used for shuffling a configuration (PERTURB), acceptance probability (F), and the cost function (COST). A good choice of these parameters and functions can result in a good placement in a relatively short time.

6.7.3 Other Placement Algorithms

Several other algorithms which simulate naturally occurring processes have been developed for routing. The simulated evolution algorithm is analogous to the natural process of mutation of species as they evolve to better adapt to their environment. The force directed placement explores the similarity between the placement problem and the classical mechanics problem of a system of bodies attached to springs. The partitioning-based placement techniques include Breuer's algorithm and the terminal propagation algorithm. Several other algorithms, such as the cluster growth algorithm, the quadratic assignment algorithm, the resistive network optimization algorithm, and the branch and bound algorithm, also exist. For more details on these algorithms, refer to Chapter 5 of [29].

6.8 Routing

The exact locations of circuit blocks and pins are determined in the placement phase. A netlist is also generated which specifies the required interconnections. Space not occupied by the blocks can be viewed as a collection of regions. These regions are used for routing and are called routing regions. The process of finding the geometric layouts of all the nets is called routing. Each routing region has a capacity, which is the maximum number of nets that can pass through that region. The capacity of a region is a function of the design rules and dimensions of the routing regions and wires. Nets must be routed within the routing regions and must not violate the capacity of any routing region. In addition, nets must not short circuit; that is, nets must not intersect each other. The objective of routing is dependent on the nature of the chip. For general-purpose chips, it is sufficient to minimize the total wire length. For high-performance chips, total wire length may not be a major concern. Instead, we may want to minimize the longest wire to minimize the delay in the wire and therefore maximize its performance. Usually routing involves special treatment of such nets as clock nets, power, and ground nets. In fact, these nets are routed separately by special routers.

Channels and switchboxes are the two types of routing regions. A switchbox is a rectangular area bounded on all sides. A channel is a rectangular area bounded on two opposite sides by the blocks. The capacity of a channel is a function of the number of layers (l), height (h) of the channel, wire width (w), and wire separation (s); i.e., $capacity = (l \times h)/(w + s)$. For example, if for channel C (Figure 6.7), $l = 2$, $h = 18\lambda$, $w = 3\lambda$, $s = 3\lambda$, then the capacity is $(2 \times 18)/(3 + 3) = 6$.

A VLSI chip may contain several million transistors. As a result, tens of thousands of nets must be routed to complete the layout. In addition, several hundred routes are possible for each net. This makes the routing problem computationally hard. In fact, even when the routing problem is restricted to channels, it cannot be solved in polynomial time; i.e., the channel routing problem is NP-complete [30]. Therefore, routing traditionally is divided into two phases. The first phase is called global routing and generates a ''loose'' route for each net. In fact, it assigns a list of routing regions to each net without specifying the actual geometric layout of wires. The second phase, which is called detailed routing, finds the actual geometric layout of each net within the assigned routing regions (Figure 6.8b). Unlike global routing, which considers the entire layout, a detailed router considers just one region at a time. The exact layout is produced for each wire segment assigned to a region, and vias are inserted to complete the layout. Detailed routing includes channel routing and switchbox routing. Another approach to routing is called area routing, which is a single-phase routing technique. However, this technique is computationally infeasible for general VLSI circuits and is typically used for specialized problems.

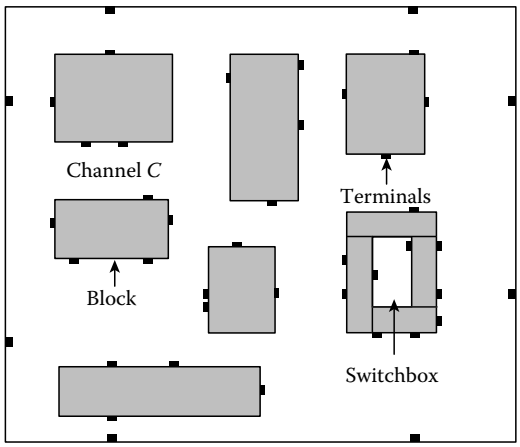


FIGURE 6.7 Layout of circuit blocks and pins after placement.

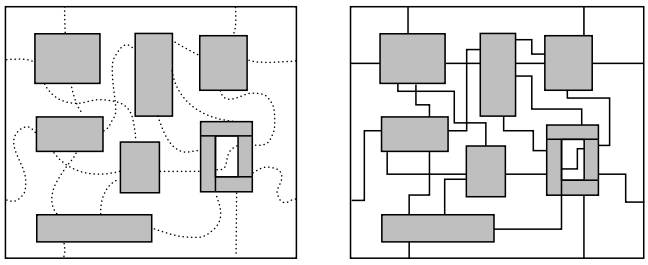


FIGURE 6.8 (a) Global routing and (b) detailed routing.

6.9 Classification of Global Routing Algorithms

Two approaches solve the global routing problem: the sequential and the concurrent:

1. *Sequential approach*: As the name suggests, nets are routed one by one. However, once a net has been routed it may block other nets which are yet to be routed. As a result, this approach is very sensitive to the order in which the nets are considered for routing. Usually, the nets are sequenced according to their criticality, perimeter of the bounding rectangle, and the number of terminals. The criticality of a net is determined by the importance of the net.
2. *Concurrent approach*: This approach avoids the ordering problem by considering routing of all the nets simultaneously. The concurrent approach is computationally hard and no efficient polynomial algorithms are known, even for two-terminal nets. As a result, integer programming methods have been suggested. The corresponding integer program is usually too large to be employed efficiently. Hence, hierarchical methods that work from the top down are employed to partition the problem into smaller subproblems, which can be solved by integer programming.

6.10 Classification of Detailed Routing Algorithms

Many ways are possible for classifying the detailed routing algorithms. The algorithms could be classified on the basis of the routing models used. Some routing algorithms use grid-based models, while some other algorithms use the gridless model. The gridless model is more practical as all the wires in a design do not have the same width. Another possible classification scheme could be to classify the algorithms based on the strategy they use. Thus, we could have greedy routers or hierarchical routers to name two. We classify the algorithms based on the number of layers used for routing. Single-layer routing problems frequently appear as subproblems in other routing problems which deal with more than one layer. Two-layer routing problems have been thoroughly investigated because until recently, due to limitations of the fabrications process, only two metal layers were allowed for routing. A third metal layer is now allowed, thanks to improvements in the fabrication process, but it is expensive compared to the two-layer metal process. Several multilayer routing algorithms also were developed recently, which can be used for routing MCMs, which have up to 32 layers.

6.10.1 Lee's Algorithm for Global Routing

This algorithm, which was developed by Lee in 1961 [20], is the most widely used algorithm for finding a path between any two vertices on a planar rectangular grid. The key to the popularity of Lee's maze router is its simplicity and its guarantee of finding on optimal solution, if one exists.

The exploration phase of Lee's algorithm is an improved version of the breadth-first search. The search can be visualized as a wave propagating from the source. The source is labeled ''0'' and the wavefront

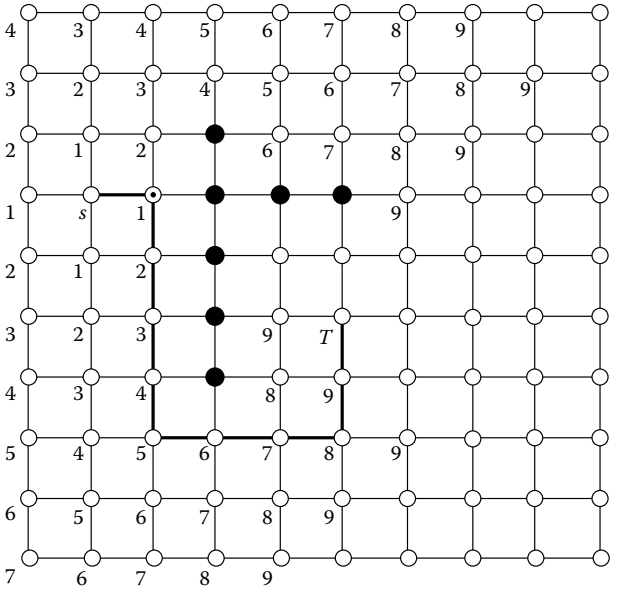


FIGURE 6.9 Net routed by Lee's algorithm.

propagates to all the unblocked vertices adjacent to the source. Every unblocked vertex adjacent to the source is marked with a label "1." Then, every unblocked vertex adjacent to vertices with a label "1" is marked with a label "2," and so on. This process continues until the target vertex is reached or no further expansion of the wave can be carried out. An example of the algorithm is shown in Figure 6.9. Due to the breadth-first nature of the search, Lee's maze router is guaranteed to find a path between the source and the target, if one exists. In addition, it is guaranteed to be the shortest path between the vertices.

The input to Lee's algorithm is an array B, the source, s, and target, t, vertex. $B[v]$ denotes if a vertex v is blocked or unblocked. The algorithm uses an array, L, where $L[v]$ denotes the distance from the source to the vertex v. This array will be used in the procedure RETRACE that retraces the vertices to form a path, P, which is the output of Lee's algorithm. Two linked lists, *plist* (propagation list) and *nlist* (neighbor list), are used to keep track of the vertices on the wavefront and their neighbor vertices, respectively. These two lists are always retrieved from tail to head. We also assume that the neighbors of a vertex are visited in counterclockwise order, that is, top, left, bottom, and then right.

The formal description of Lee's algorithm appears in Figure 6.10. The time and space complexity of Lee's algorithm is $O(h \times w)$ for a grid of dimension $h \times w$.

Lee's routing algorithm requires a large amount of storage space and its performance degrades rapidly when the size of the grid increases. Numerous attempts have been made to modify the algorithm to improve its performance and reduce its memory requirements.

Lee's algorithm requires up to $k+1$ bits per vertex, where k bits are used to label the vertex during the exploration phase and an additional bit is needed to indicate whether the vertex is blocked. For an $h \times w$ grid, $k = \log_2(h \times w)$. Aker [1] noticed that in the retrace phase of Lee's algorithm, only two types of neighbors of a vertex need to be distinguished; vertices toward the target and vertices toward the source. This information can be coded in a single bit for each vertex. The vertices in wavefront, L, are always adjacent to the vertices in wavefront $L-1$ and $L+1$. Thus, during wave propagation, instead of using a sequence 1, 2, 3, . . . , the wavefronts are labeled by a sequence such as 0, 0, 1, 1, 0, 0, ... The predecessor of any wavefront is labeled differently from its successor. Thus, each scanned vertex is either labeled "0" or "1." Besides these two states, additional states ("block" and "unblocked") are needed for each vertex. These four states of each vertex can be represented by using exactly 2 bits, regardless of the problem size.

```
Algorithm Lee-Router (B, s, t, P)

Input: B, s, t
Output: P

Begin
    plist = s;
    nlist = ϕ;
    temp = 1;
    path_exists = FALSE;
    while plist ≠ ϕ do
        for each vertex v_i in plist do
            for each vertex v_j neighboring v_i do
            if B[v_j] = UNBLOCKED then
                L[v_j] = temp;
                INSERT (v_j, nlist);
                if v_j = t then
                    path_exists = TRUE;
                    exit while;
        temp = temp + 1;
        plist = nlist;
        nlist = ϕ;
    if path_exists = TRUE then RETRACE (L, P);
    else path does not exist;
End;
```

FIGURE 6.10 Lee's routing algorithm.

Compared with Acker's scheme, Lee's algorithm requires at least 12 bits per vertex for a grid size of 2000×2000.

It is important to note that Acker's coding scheme only reduces the memory requirement per vertex. It inherits the search space of Lee's original routing algorithm, which is $O(h \times w)$ in the worst case.

6.10.2 Greedy Channel Router for Detailed Routing

Assigning the complete trunk of a net or a two-terminal net segment of a multiterminal net severely restricts LEA and dogleg routers. Optimal channel routing can be obtained if for each column it can be guaranteed that only one horizontal track per net exists. Based on this observation, one approach to reduce channel height could be to route nets column by column trying to join split horizontal tracks (if any) that belong to the same net as much as possible.

Based on the preceding observation and approach, Rivest and Fiduccia [24] developed the greedy channel router. This makes fewer assumptions than LEA and dogleg routers. The algorithm starts from the leftmost column and places all the net segments of a column before proceeding to the next right column. In each column the router assigns net segments to tracks in a greedy manner. However, unlike the dogleg router, the greedy router allows the doglegs in any column of the channel, not necessarily where the terminal of the doglegged net occurs.

Given a channel routing problem with m columns, the algorithm uses several steps while routing a column. In the first step the algorithm connects any terminal to the trunk segment of the corresponding net. This connection is completed by using the first empty track, or the track that already contains the net. In other words, minimum vertical segment is used to connect a trunk to a terminal. For example, net 1 in Figure 6.11a in column 3 is connected to the same net. The second step attempts to collapse any split nets (horizontal segments of the same net present on two different tracks) using a vertical segment as shown in Figure 6.11b. A split net will occur when two terminals of the same net are located on different

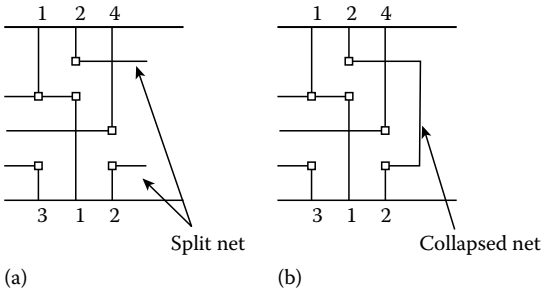


FIGURE 6.11 (a) Split net and (b) collapsed split net.

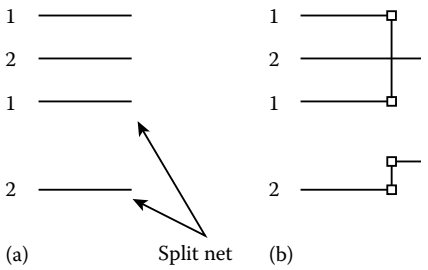


FIGURE 6.12 (a) Reducing the distance between split nets and (b) its result.

sides of the channel and cannot be connected immediately because of existing vertical constraints. This step also brings a terminal connection to the correct track if it has stopped on an earlier track. If two sets of split nets overlap, the second step will only be able to collapse one of them.

In the third step, the algorithm tries to reduce the range or the distance between two tracks of the same net. This direction is accomplished by using a dogleg, as shown in Figure 6.12a and b. The fourth step attempts to move the nets closer to the boundary which contains the next terminal of that net. If the next terminal of a net being considered is on the top (bottom) boundary of the channel, then the algorithm tries to move the net to the upper (lower) track. In case no track is available, the algorithm adds extra tracks and the terminal is connected to this new track. After all five steps have been completed, the trunks of each net are extended to the next column and the steps are repeated. A detailed description of the greedy channel routing algorithm is shown in Figure 6.13.

The greedy router sometimes gives solutions which contain an excessive number of vias and doglegs. It does, however, have the capability of providing a solution even in the presence of cyclic vertical constraints. The greedy router is more flexible in the placement of doglegs due to fewer assumptions about the topology of the connections. An example routed by the greedy channel router is shown in Figure 6.14.

6.10.3 Other Routing Algorithms

Soukup proposed an algorithm that basically cuts down the search time of Lee's algorithm by exploring in the direction toward the target, without changing the direction until it reaches the target or an obstacle. An alternative approach to improve upon the speed was suggested by Hadlock. One class of algorithm is called the line-probe algorithms. The basic idea of the line-probe algorithms is to reduce the memory requirement by using line segments instead of grid notes in the search. Several algorithms based on Steiner trees have been developed. The main advantage of these algorithms is that they can be used for routing multiterminal nets. For further details on these global routing algorithms, refer to Chapter 6 of [29].

Extensive research has been carried out in the area of detailed routing and several algorithms exist for channel and switchbox routing. There are LEA-based algorithms which use a reserved layer model and do not allow any vertical constraints or doglegs. The YACR2 algorithm can handle vertical constraint

```
Algorithm Greedy-Channel-Router (N)

Input: N

Begin
    d = DENSITY (N);
  (* calculate the lower bound of channel density *)
    insert d tracks to channel;
    for i = 1 to m do
      T1 = GET-EMPTY-TRACK;
      if T1 = 0 then
        ADD-TRACK(T1);
        ADD-TRACK(T2);
      else
        T2 = GET-EMPTY-TRACK;
        if T2 = 0 then
          ADD-TRACK(T2);
      CONNECT(T_i,T1);
      CONNECT(B_i,T2);
      join split nets as much as possible;
      bring split nets closer by jogging;
      bring nets closer to either top or bottom boundary;
    while split nets exists do
      increase number of column by 1;
      join split nets as much as possible;
End;
```

FIGURE 6.13 Algorithm greedy channel router.

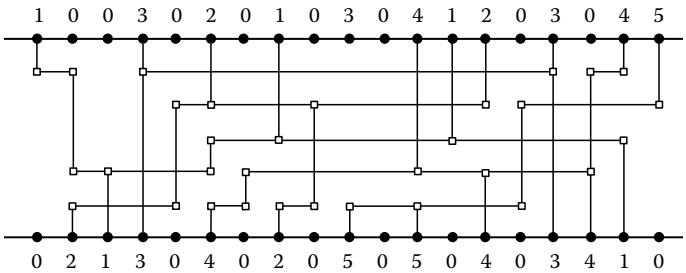


FIGURE 6.14 Channel routed using a greedy router.

violations. The net merge channel router works for two-layer channel routing problems and it exploits the graph theoretic structure of channel routing problems. Glitter is an algorithm for gridless channel routing and can handle nets of varying widths. The hierarchical channel router divides the channel routing problem into smaller problems, each of which is solved in order to generate the complete routing for the nets in the channel. Several algorithms such as the extended net merge channel routing algorithm, HVH routing from the HV solution, and the hybrid HVH–VHV channel routing algorithm exist for the three-layer channel routing problem. For further details on these detailed routing algorithms, refer to Chapter 7 of [29].

6.11 Compaction

After completion of detailed routing, the layout is functionally complete. At this stage, the layout is ready to be used to fabricate a chip. However, due to nonoptimality of placement and routing algorithms, some vacant space is present in the layout. In order to minimize the cost, improve performance, and yield, layouts are reduced in size by removing the vacant space without altering the functionality of the layout. This operation of layout area minimization is called layout compaction.

The compaction problem is simplified by using symbols to represent primitive circuit features, such as transistors and wires. The representation of layout using symbols is called a symbolic layout. Special languages [4,21,22] and special graphic editors [13,14] are available to describe symbolic layouts. To produce the actual masks, the symbolic layouts are translated into actual geometric features. Although a feature can have any geometric shape, in practice only rectangular shapes are considered.

The goal of compaction is to minimize the total layout area without violating any design rules. The area can be minimized in three ways:

1. *By reducing the space between features*: This can be performed by bringing the features as close to each other as possible. However, the spacing design rules must be met while moving features closer to each other.
2. *By reducing the size of each feature*: The size rule must be met while resizing the features.
3. *By reshaping the features*: Electrical characteristics must be preserved while reshaping the feature.

Compaction is a very complex phase in the physical design cycle. It requires understanding many details of the fabrication process such as the design rules. Compaction is critical for full-custom layouts, especially for high-performance designs.

6.11.1 Classification of Compaction Algorithms

Compaction algorithms can be classified in two ways. The first classification scheme is based on the direction of movements of the components (features): one-dimensional (1-D) and two-dimensional (2-D). In 1-D compaction components are moved only in x- or the y-direction. As a result, either the x- or the y-coordinate of the components is changed due to the compaction. If the compaction is done along the x-direction, then it is called x-compaction. Similarly, if the compaction is done along the y-direction, then it is called y-compaction. In 2-D compaction the components can be moved in both the x- and y-directions simultaneously. As a result, in 2-D compaction both x- and y-coordinates of the components are changed at the same time in order to minimize the layout area.

The second approach to classify the compaction algorithms is based on the technique for computing the minimum distance between features. In this approach, we have two methods: constraint graph-based compaction and virtual grid-based compaction. In the constraint graph method, the connections and separations rules are described using linear inequalities which can be modeled using a weighted directed graph (constraint graph). This constraint graph is used to compute the new positions for the components. On the other hand, the virtual grid method assumes the layout is to be drawn on a grid. Each component is considered attached to a grid line. The compaction operation compresses the grid along with all components placed on it, keeping the grid lines straight along the way. The minimum distance between two adjacent grid lines depends on the components on these grid lines. The advantage of the virtual grid method is that the algorithms are simple and can be easily implemented. However, the virtual grid method does not produce compact layouts as does the constraint graph method.

In addition, compaction algorithms can be classified on the basis of the hierarchy of the circuit. If compaction is applied to different levels of the layout, it is called hierarchical compaction. Any of the above-mentioned methods can be extended to hierarchical compaction. A variety of hierarchical compaction algorithms have been proposed for both constraint graph and virtual grid methods. Some compaction algorithms actually ''flatten the layout'' by removing all hierarchy and then

perform compaction. In this case it may not be possible to reconstruct the hierarchy, which may be undesirable.

6.11.2 Shadow-Propagation Algorithm for Compaction

A widely used and one of the best known techniques for generating a constraint graph is the shadow-propagation used in the CABBAGE system [15]. The ''shadow'' of a feature is propagated along the direction of compaction. The shadow is caused by shining an imaginary light from behind the feature under consideration (see Figure 6.15). Usually the shadow of the feature is extended on both sides of the features in order to account for diagonal constraints. This leads to greater than minimal Euclidean spacings because an enlarged rectangle is used to account for corner interactions. (See shadow of feature in Figure 6.15.)

When the shadow is obstructed by another feature, an edge is added to the graph between the vertices corresponding to the propagating feature and the obstructing feature. The obstructed part of the shadow is then removed from the front and no longer propagated. The process is continued until all of the shadow has been obstructed. This process is repeated for each feature in the layout. The algorithm shadow-propagation, given in Figure 6.16, presents an overview of the algorithm for *x*-compaction of a single feature from left to right.

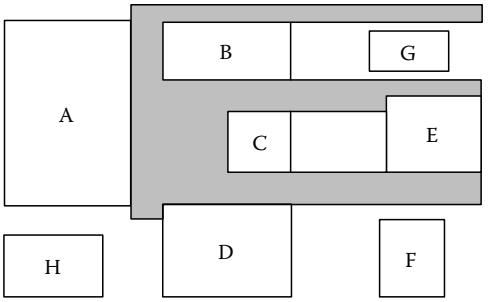


FIGURE 6.15 Example of shadow propagation.

The shadow-propagation routine accepts the list of components (*Comp_list*), which is sorted on the *x*-coordinates of the left corner of the components and the component (*component*) for which the constraints are to be generated. The procedure, INITIALIZE-SCANLINE, computes the total length of the interval in which the shadow is to be generated. This length includes the design rule separation distance. The *y*-coordinate of the top and the bottom of this interval are stored in the global variables, *top* and *bottom*, respectively. The procedure, GET-NXT-COMP, returns the next component (*curr_comp*) from *Comp_list*. This component is then removed

Algorithm *Shadow-Propagation* (*Comp_list, component*)

Input: *Comp_list, component*

```
Begin
    INITIALIZE-SCANLINE (component);
  I = ϕ;
   while ( (LENGTH-SCANLINE(I) < (top – bottom))
         and (Comp_list ≠ ϕ))
      curr_comp = GET-NXT-COMP (Comp_list);
      I_i = LEFT-EDGE (curr_comp);
      if (IN-RANGE (I_i, top, bottom))
         I' = UNION (I_i, I);
         if(I' ≠ I)
            ADD-CONSTRAINT (component, curr_comp);
            I = UNION(I_i, I);
End;
```

FIGURE 6.16 Shadow-propagation algorithm.

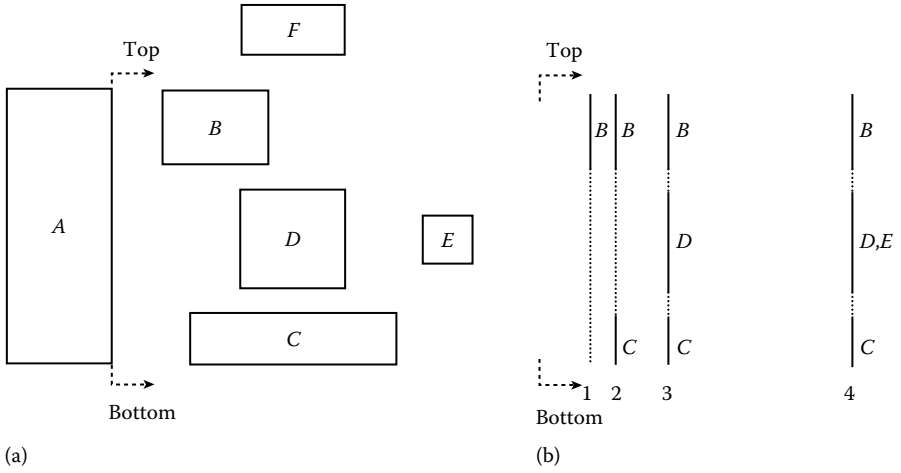


FIGURE 6.17 Interval generation for shadow propagation. (a) The layout of components and (b) the intervals in the interval set.

from the list. Procedure LEFT-EDGE returns the vertical interval of component, *curr_comp*. If this interval is within the *top* and *bottom* then *curr_comp* may have a constraint with *component*. This check is performed by the procedure IN-RANGE. If the interval for *curr_component* lies within *top* and *bottom* and if this interval is not already contained within one of the intervals in the interval set, $\mathcal{I}$, then the component lies in the shadow of *component*, and hence a constraint must be generated. Each interval represents the edge at which the shadow is blocked by a component. The constraint is added to the constraint graph by the procedure ADD-CONSTRAINT. The procedure UNION inserts the interval corresponding to *curr_comp* in the interval set at the appropriate position. This process is carried out until the interval set completely covers the interval from *top* to *bottom* or no more components are in *Comp_list*. Figure 6.17a shows the layout of components. The constraint for component *A* with other components is being generated. Figure 6.17b shows the intervals in the interval set as the shadow is propagated. From Figure 6.17b it is clear that the constraints will be generated between components *A* and components *B*, *C*, and *D* in that order. As component *F* lies outside the interval defined by *top* and *bottom* it is not considered for constraint generation. The interval generated by component *E* lies within one of the intervals in the interval set. Hence, no constraint is generated between components *A* and *E*.

6.11.3 Other Compaction Algorithms

Several algorithms such as constraint graph-based compaction algorithms, scanline algorithm, and grid-based compaction algorithms exist for the 1-D compaction problem. An algorithm based on a simulation of the zone-refining process also was developed. This compactor is considered a $1\frac{1}{2}$-D compactor, as the key idea is to provide enough lateral movements to blocks during compaction to resolve interferences. For further details on these algorithms, refer to Chapter 10 of [29].

6.12 Summary

The sheer size of the VLSI circuit, the complexity of the overall design process, the desired performance of the circuit, and the cost of designing a chip dictate that the whole design process must be automated. Also, the design process must be divided into different stages because of the complexity of the entire process. Physical design is one of the steps in the VLSI design cycle. In this step each component of a circuit is converted into a set of geometric patterns which achieves the functionality of the component.

The physical design step can be divided further into several substeps. All the substeps of the physical design step are interrelated. Efficient and effective algorithms are required to solve different problems in each of the substeps. Despite significant research efforts in this field, the CAD tools still lag behind the technological advances in fabrication. This calls for the development of efficient algorithms for physical design automation.

References

1. S. B. Aker, A modification of Lee's path connection algorithm, *IEEE Trans. Electron. Comput.*, EC-16(1): 97–98, February 1967.
2. P. Bannerjee and M. Jones, A parallel simulated annealing algorithm for standard cell placement on a hypercube computer, *Proceedings of the IEEE International Conference on Computer Design*, IEEE Computer Society Press, Los Alamitos, CA, 1986, p. 34.
3. A. Chatterjee and R. Hartley, A new simultaneous circuit partitioning and chip placement approach based on simulated annealing, *Proceedings of the 27th ACM/IEEE Design Automation Conference*, IEEE Computer Society Press, Orlando, FL, June 24–28, 1990, pp. 36–39.
4. P. A. Eichenberger, Fast symbolic layout translation for custom VLSI integrated circuits, PhD thesis, Stanford University, Stanford, CA, 1986.
5. A. El Gamal et al., An architecture for electrically configurable gate arrays, *IEEE J. Solid-State Circuits*, 24(2): 394–398, April 1989.
6. H. Hseih et al., A 9000-gate user-programmable gate array, *Proceedings of the IEEE 1988 Custom Integrated Circuits Conference*, May 1988, pp. 15.3.1–15.3.7.
7. S. C. Wong et al., A 5000-gate CMOS EPLD with multiple logic and interconnect arrays, *Proceedings of the IEEE 1988 Custom Integrated Circuits Conference*, May 1989, pp. 5.8.1–5.8.4.
8. M. Feuer, VLSI design automation: An introduction, *Proc. IEEE*, 71(1): 1–9, January 1983.
9. C. M. Fiduccia and R. M. Mattheyses, A linear-time heuristics for improving network partitions, *Proceedings of the 19th Design Automation Conference*, Las Vegas, NV, 1982, pp. 175–181.
10. J. Greene and K. Supowit, Simulated annealing without rejected moves, *Proceedings of the International Conference on Computer Design*, IEEE Computer Society Press, Los Alamitos, CA, October 1984, pp. 658–663.
11. L. K. Grover, Standard cell placement using simulated sintering, *Proceedings of the 24th ACM/IEEE Design Automation Conference*, IEEE Computer Society Press, Miami Beach, FL, June 28–July 1, 1987, pp. 56–59.
12. B. Hajek, Cooling schedules for optimal annealing, Math. *Oper. Res.*, 13(2): 311–329, May 1988.
13. D. D. Hill, ICON: A toll for design at schematic, virtual-grid and layout levels, *IEEE Des. Test*, 1(4): 53–61, 1984.
14. M. Y. Hsueh, Symbolic layout and compaction of integrated circuits, Technical Report No. UCB/ERL M79/80, Electronics Research Laboratory, University of California, Berkeley, CA, 1979.
15. M. Y. Hsueh and D. O. Pederson, Computer-aided layout of LSI circuit building-blocks, PhD thesis, University of California, Berkeley, CA, December 1979.
16. M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli, An efficient general cooling schedule for simulated annealing, *Proceedings of the IEEE International Conference on Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, CA, 1986, pp. 381–384.
17. W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs, *Bell Syst. Tech. J.*, 49: 291–307, 1970.
18. S. Kirkpatrick, G. D. Gellat, and M. P. Vecchi, Optimization by simulated annealing, *Science*, 220: 671–680, May 1983.
19. J. Lam and J. Delosme, Performance of a new annealing schedule, *Proceedings of the 25th ACM/IEEE Conference on Design Automation*, DAC'88, Anaheim, CA, June 12–15, 1988, pp. 306–311.

20. C. Y. Lee, An algorithm for path connections and its applications, *IRE Trans. Electron. Comput*., EC-10: 346–365, 1961.
21. T. M. Lin and C. A. Mead, Signal delay in general RC networks, *IEEE Trans. CAD*, CAD-3(4): 331–349, October 1984.
22. J. M. Da Mata, Allenda: A procedural language for the hierarchical specification of VLSI layout, *Proceedings of the 22nd ACM/IEEE Conference on Design Automation*, DAC'85, ACM Press, Las Vegas, NV, 1985, pp. 183–189.
23. T. Ohtsuke, *Partitioning, Assignment and Placement*, Amsterdam, the Netherlands: North-Holland, 1986.
24. R. Rivest and C. Fiduccia, A greedy channel router, *Proceedings of the 19th ACM/IEEE Design Automation Conference*, Las Vegas, NV, 1982, pp. 418–424.
25. F. Romeo and A. Sangiovanni-Vincentelli, Convergence and finite time behavior of simulated annealing, *Proceedings of the 24th Conference on Decision Control*, Ft. Lauderdale, FL, December 1985, pp. 761–767.
26. F. Romeo, A. S. Vincentelli, and C. Sechen, Research on simulated annealing at Berkeley, *Proceedings of the IEEE International Conference on Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, CA, 1984, pp. 652–657.
27. C. Sechen and K. W. Lee, An improved simulated annealing algorithm for row-based placement, *Proceedings of the IEEE International Conference on Computer-Aided Design*, IEEE Computer Society Press, Los Alamitos, CA, 1987, pp. 478–481.
28. C. Sechen and A. Sagiovanni-Vincentelli, The timber wolf placement and routing package, *IEEE J. Solid-State Circuits*, SC-20: 510–522, 1985.
29. N. Sherwani, *Algorithms for VLSI Physical Design Automation*, Boston: Kluwer Academic, 1993.
30. T. G. Szymanski, Dogleg channel routing is NP-Complete, *IEEE Trans. Comput. Aided Des*., CAD-4: 31–41, January 1985.

7

Design Automation Technology

7.1 Introduction 7-1
7.2 Design Entry 7-3
7.3 Conceptual Design 7-6
Design Planning • Decision Assistance • Exploring Alternative Designs • Applications
7.4 Synthesis 7-14
7.5 Verification 7-16
Timing Analysis • Simulation • Emulation
7.6 Test 7-21
Fault Modeling • Fault Testing
7.7 Frameworks 7-27
7.8 Summary 7-29
References 7-29

Allen M. Dewey*
Duke University

7.1 Introduction

The field of *design automation* technology, also commonly called computer-aided design (CAD) or computer-aided engineering (CAE), involves developing computer programs to conduct portions of product design and manufacturing on behalf of the designer. Competitive pressures to produce in less time and use fewer resources, new generations of products having improved function and performance are motivating the growing importance of design automation. The increasing complexities of microelectronic technology, shown in Figure 7.1, illustrate the importance of relegating portions of product development to computer automation [1,3]. Advances in microelectronic technology enable over 1 million devices to be manufactured on an integrated circuit substrate smaller in size than a postage stamp, yet the ability to exploit this capability remains a challenge. Manual design techniques are unable to keep pace with product design cycle demands and are being replaced by automated design techniques [2,4].

Figure 7.2 summarizes the historical development of design automation technology and computer programs. Design automation programs are also called ''applications'' or ''tools.'' Design automation efforts started in the early 1960s as academic research projects and captive industrial programs, focusing on individual tools for physical and logical design. Later developments extended logic simulation to more detailed *circuit* and *device* simulation and more abstract *functional* simulation. Starting in the mid to the

* Allen M. Dewey passed away before the publication of this edition.

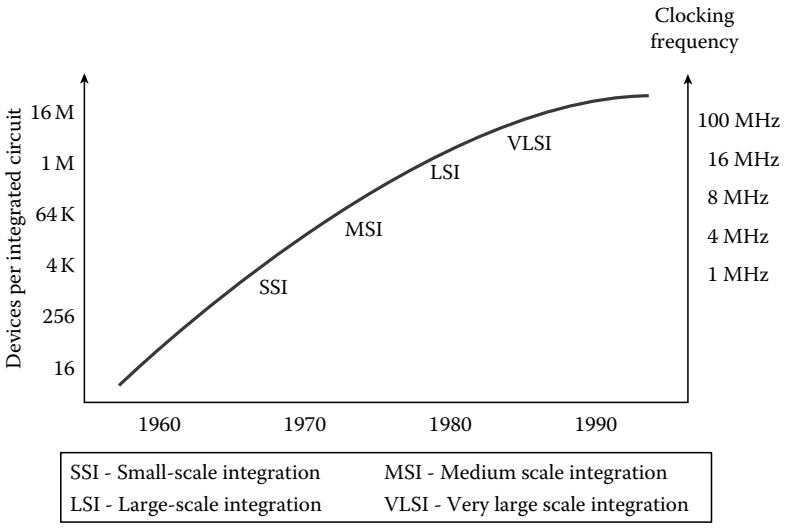


FIGURE 7.1 Complexity of microelectronic technology.

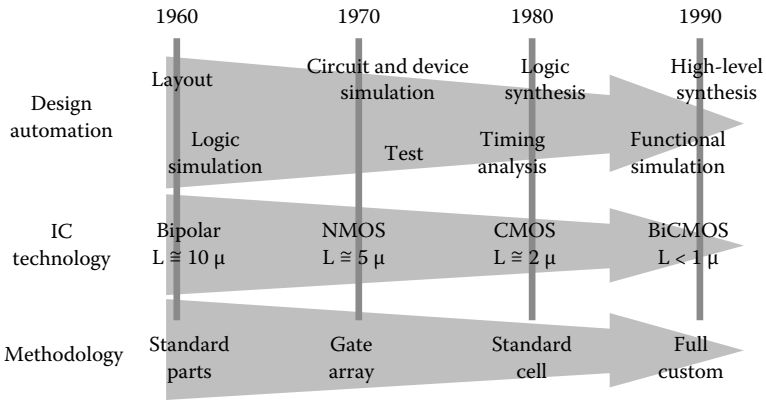


FIGURE 7.2 Development of design automation technology.

late 1970s, new areas of test and synthesis emerged and commercial design automation products appeared. Today, the electronic design automation industry is an international business with a well-established and expanding technical base [5]. The electronic design automation technology base will be examined by presenting an overview of the following topical areas:

- Design entry
- Conceptual design
- Synthesis
- Verification
- Testing
- Frameworks

7.2 Design Entry

Design entry, also called design capture, is the process of communicating with a design automation system. Design entry involves describing a system design to a design automation system, invoking applications to analyze and/or modify the system design, and querying the results. In short, design entry is how an engineer ''talks'' to a design automation application and/or system.

Any sort of communication is composed of two elements: language and mechanism. Language provides common semantics; mechanism provides a means by which to convey the common semantics. For example, two people communicate via a language such as English, French, or German, and a mechanism, such as a telephone, electronic mail, or facsimile transmission. For design, a digital system can be described in many ways, involving different perspectives or abstractions. An abstraction is a model for defining the behavior or semantics of a digital system, i.e., how the outputs respond to the inputs. Figure 7.3 illustrates several popular levels of abstractions and the trends toward higher levels of design entry abstraction to address greater levels of complexity [10,12].

The physical level of abstraction involves geometric information that defines electrical devices and their interconnection. Geometric information includes the shaped objects and where objects are placed relative to each other. For example, Figure 7.4 shows the geometric shapes defining a simple complementary metal–oxide semiconductor (CMOS) inverter. The shapes denote different materials, such as aluminum and polysilicon, and connections, called *contacts* or *vias*.

The design entry mechanism for physical information involves textual and graphic techniques. With textual techniques, geometric shape and placement are described via an artwork description language, such as Caltech Intermediate Form or Electronic Design Intermediate Form. With graphical techniques, geometric shape and placement are described by drawing the objects on a display terminal [13].

The electrical level abstracts geometric information into corresponding electrical devices, such as capacitors, transistors, and resistors. For example, Figure 7.5 shows the electrical symbols denoting a CMOS inverter. Electrical information includes the device behavior in terms of terminal or pin current and voltage relationships. Device behavior also may be defined in terms of manufacturing parameters, such as resistances or chemical compositions.

The logical level abstracts electrical information into corresponding logical elements, such as `and` gates, `or` gates, and inverters. Logical information includes truth table and/or characteristic switching algebra equations and active level designations. For example, Figure 7.6 shows the logical symbol for a CMOS inverter. Notice how the amount of information decreases as the level of abstraction increases.

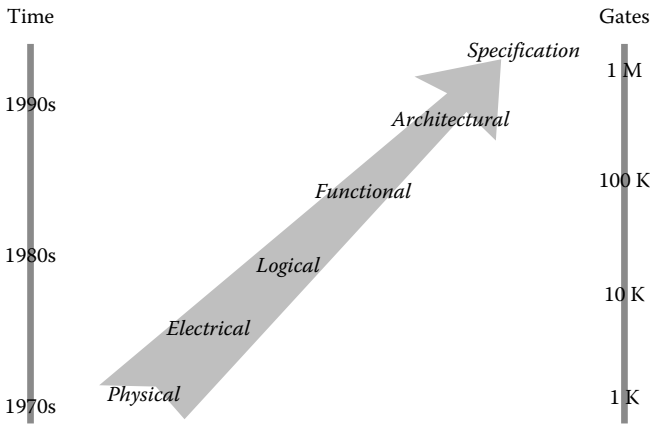


FIGURE 7.3 Design automation abstractions.

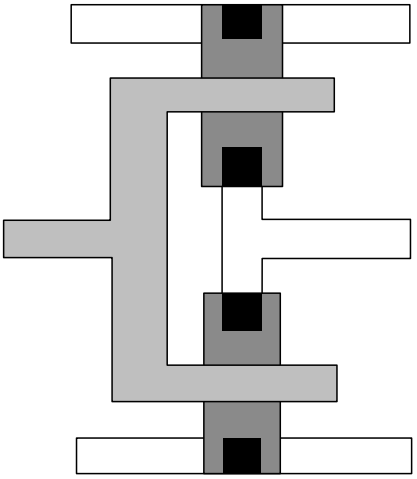


FIGURE 7.4 Physical information.

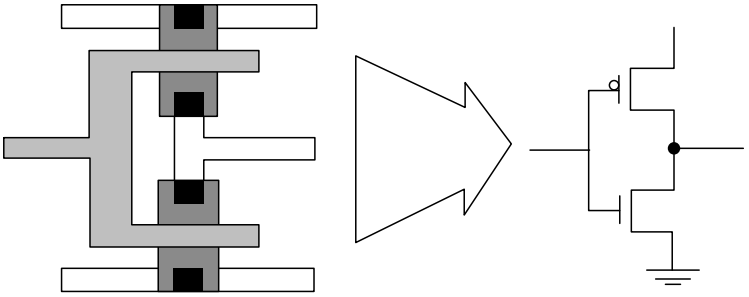


FIGURE 7.5 Electrical information.

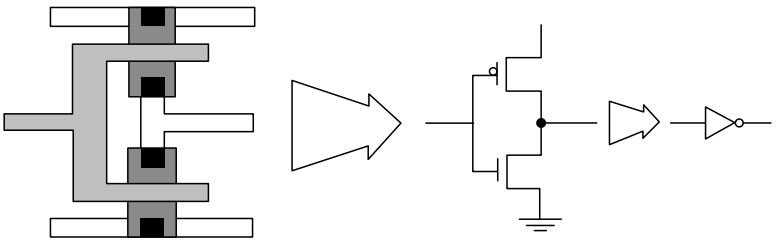


FIGURE 7.6 Logical information.

Design entry mechanisms for electrical and logical abstractions are commonly collectively called *schematic* capture techniques. Schematic capture defines hierarchical structures, commonly called *netlists*, of components. A designer creates instances of components supplied from a library of predefined components and connects component pins or ports via wires [9,11].

The functional level abstracts logical elements into corresponding computational units, such as registers, multiplexers, and arithmetic logic units (ALUs). The architectural level abstracts functional

information into computational algorithms or paradigms. Some examples of common computational paradigms are

- State diagrams
- Petri nets
- Control/data flow graphs
- Function tables
- Spreadsheets
- Binary decision diagrams

These higher levels of abstraction support a more expressive ''higher bandwidth'' communication interface between an engineer and design automation programs. An engineer can focus his or her creative, cognitive skills on concept and behavior, rather than the complexities of detailed implementation. Associated design entry mechanisms typically use hardware description languages with a combination of textual and graphic techniques [6].

Figure 7.7 shows a simple state diagram. The state diagram defines three states, denoted by circles. State-to-state transitions are denoted by labeled arcs; state transitions depend on the present state and the input *X*. The output, *Z*, per state is given within each state. Because the output is dependent on only the present state, the digital system is classified as a *Moore finite state machine*. If the output is dependent on the present state and input, then the digital system is classified as a *Mealy finite state machine*. The state table corresponding to the state diagram in Figure 7.7 is given in Table 7.1. A hardware description language model written in VHDL of the Moore finite state machine is given in Figure 7.8. The VHDL model, called a *design entity*, uses a ''date flow'' description style to describe the state machine [7,8]. The

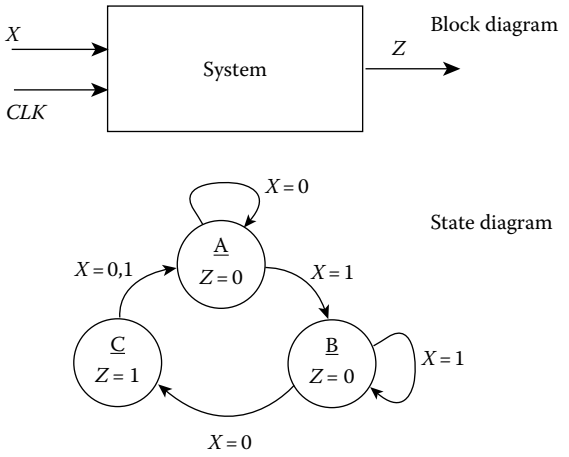


FIGURE 7.7 State diagram.

TABLE 7.1 State Table

	Next State		
Present State	$X = 0$	$X = 1$	Output
A	A	B	0
B	C	B	0
C	A	A	1

```
entity MOORE_MACHINE is
  port (X, CLK: in BIT; Z: out BIT);
end MOORE_MACINE;
architecture FSM of MOORE_MACHINE is
  type STATE_TYPE is (A, B, C);
  signal STATE: STATE_TYPE: = A;
begin
  NEXT_STATE:
  block (CLK = '1' and not CLK'STABLE)
  begin
     STATE <= guarded B when (STATE = A and X = '1') else
                          C when (STATE = B and X = '0') else
                          A when (STATE = C) else
                          STATE;
  end block NEXT_STATE;

  with STATE select
     Z <= '0' when A,
          '0' when B,
          '1' when C;
end FSM;
```

FIGURE 7.8 VHDL model.

entity statement defines the interface, i.e., the ports. The ports include two input signals, *X* and *CLK*, and an output signal, *Z*. The ports are of type BIT, which specifies that the signals may only carry the values "0" or "1." The architecture statement defines the I–O transform via two concurrent signal assignment statements. The internal signal STATE holds the finite state information and is driven by a guarded, conditional concurrent signal assignment statement that executes when the associated block expression (*CLK* = "1" and not CLK'STABLE) is true, which is only on the rising edge of the signal *CLK*. STABLE is a predefined attribute of the signal *CLK*; CLK'STABLE is true if *CLK* has not changed value. Thus, if not CLK'STABLE is true, meaning that *CLK* has just changed value, and *CLK* = "1", then a rising transition has occurred on *CLK*. The output signal *Z* is driven by a nonguarded, selected concurrent signal assignment statement that executes anytime STATE changes value.

7.3 Conceptual Design

Figure 7.9 shows that the conceptual design task generally follows the design entry task. Conceptual design addresses the initial decision-making process by assisting the designer to conduct initial feasibility studies, determine promising design alternatives, and obtain preliminary indication of estimated performance [16].

During early stages of the electronic design process, one typically makes a number of high-level decisions that can have a significant impact on the outcome of a design. For example, consider the early decisions usually made when designing a memory subsystem. The designer must choose a configuration (e.g., ×1, ×4, ×8, etc.), memory technology (e.g., static RAM, dynamic RAM, mask programmable ROM, etc.), package style (e.g., dual-in-line, chip carrier, pin grid array, etc.), and so on. Taken collectively, these decisions may be referred to as the design plan. The design plan then guides the ensuing, more detailed design process, acting as a high-level road map.

In general, many different plans may result in designs that satisfy the same set of functional specifications, but have varying trade-offs in terms of performances, such as area or power dissipation. For example, Figure 7.10 shows relative area and speed performance trade-offs for six combinational multiplier designs. Figure 7.10 shows the typical trend that increasing speed implies increasing area.

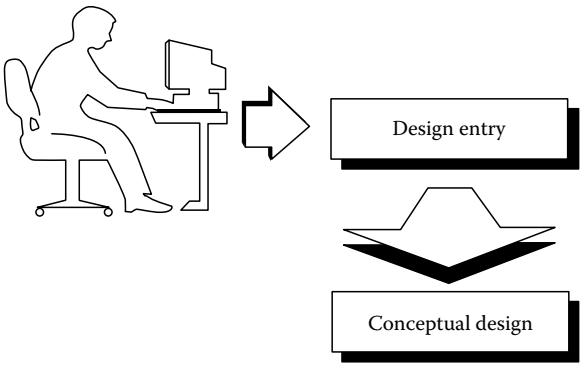


FIGURE 7.9 Design process.

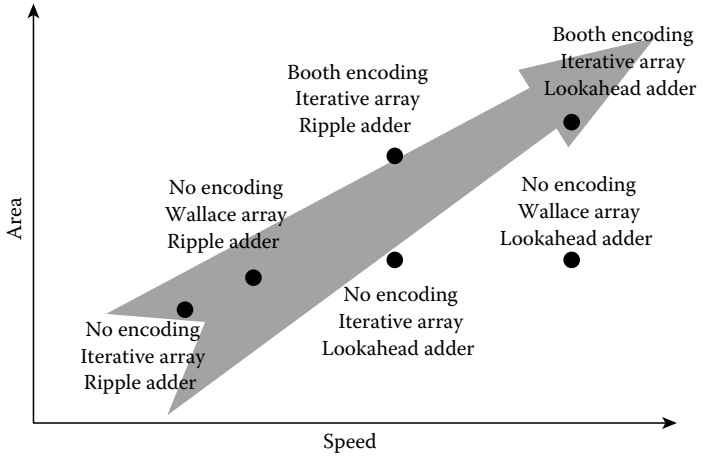


FIGURE 7.10 Performance trade-offs.

More importantly, the chart indicates how various multiplier designs relate to each other and which designs present the best area/speed trade-offs for a particular application.

Given this large set of possible design plans to consider, it is easy to imagine getting hopelessly lost in analyzing the various design plans and keeping track of the trade-offs and decision interdependencies. Choosing the most appropriate design plan is important because it is usually not economically feasible to choose an alternative plan and repeat the detailed design process if the final design does not meet the desired specifications. To aid the designer during the early stages of the design process in maximizing his or her chances that certain high-level decisions will result in a design that meets the desired specifications, conceptual design tools assist the designer in analyzing possible ramifications of high-level design decisions before actually undertaking the detailed design and fabrication process.

During conceptual design, the engineer has a global view of the design process in considering the key decisions associated with each lower level of abstraction. Thus, the engineer can undertake a breadth-first design methodology. Costly errors can be avoided by investing time at the outset in thoroughly thinking through a particular plan, the result being a more complete and credible specification that better guides the ensuing actions. In addition, the unnecessary use of detailed design CAD tools is minimized because conceptual design acts as a ''coarse filter'' eliminating infeasible design plans.

7.3.1 Design Planning

To discuss conceptual design, it is convenient to introduce the following terms:

- *Design issue*—Pertinent aspect of a design for which a decision must be made.
- *Design option*—Possible choice that can be made for a design issue.
- *Design decision*—Particular option chosen for a design issue.
- *Design plan*—Set of design decisions that identifies a high-level approach for implementing the design.

Conceptual design begins with the identification of the set of relevant design issues that must be considered before detailed design begins. For example, for the design of an analog amplifier, the design issues include selection of a type of amplifier (e.g., current, voltage, etc.), selection of a class of operation (e.g., A, B, AB, etc.), selection of a biasing scheme, etc. For the design of a digital signal processing filter, the design issues include selection of an algorithm (e.g., infinite impulse response or finite impulse response), selection of an architecture (e.g., direct, cascade, parallel, etc.), selection of a multiplier logic design, etc. Having identified the relevant design issues, conceptual design proceeds to develop a design plan in the step-by-step fashion outlined in Figure 7.11. For each design issue, the various associated options are examined, and the most appropriate option is selected. To make this selection, the designer chooses some method of discrimination and may need expert advice about the possible options or may desire a prediction about the effect a particular option has on performance. Such a conceptual design approach implies several important features, which are considered in more detail next.

Designers often structure design issues in a hierarchical fashion to control the complexity of the conceptual design process. Complex design issues are decomposed into sets of design subissues or subplans [22]. For example, an expanded, hierarchical listing of candidate design issues for digital signal processing filters is given next [14,20]:

1. Selection of a filter algorithm
2. Selection of a polynominal approximation technique
3. Selection of a filter architecture
   a. Selection of a filter structure
   b. Selection of a pipelining strategy
4. Selection of a logic design
   a. Selection of a multiplier logic design
   b. Selection of an encoding technique

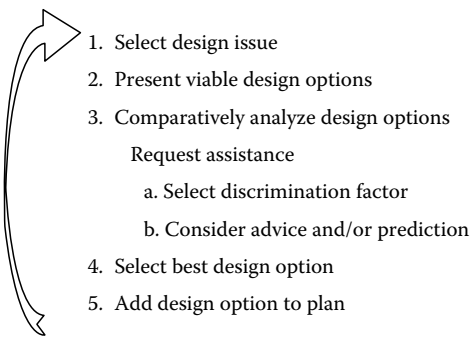


FIGURE 7.11 Constructing a design plan.

   c. Selection of an array design
   d. Selection of a final adder design
   e. Selection of an adder design style
5. Selection of a fabrication technology
6. Selection of a layout design style

The selection of a multiplier logic design can be decomposed into three design subissues: selection of an encoding technique, selection of a combinational array design, and selection of a final adder design. Notice that many details about digital filter design are not listed because conceptual design is typically not concerned with every design issue required to eventually fully design and fabricate the initial specification. Hierarchical design issues provide for a more intuitive representation of the conceptual design process, enabling the designer to better comprehend the overall magnitude and composition of the design domain.

Each design issue has associated commonly used options or selections. For example, the possible digital filter polynomial approximation techniques are

1. Equiripple
2. Frequency sampling
3. Windowing
4. Beta
5. Elliptic
6. Butterworth
7. Chebyshev

During conceptual design, the order in which the designer examines the various issues may not be completely arbitrary because the options available for a design issue may depend on which options were chosen for other design issues. Hence, a partial ordering may be imposed on addressing design issues. As an example, equiripple, frequency sampling, windowing, and beta digital filter approximations apply only to the finite impulse response algorithm, while elliptic, Butterworth, and Chebyshev digital filter approximations apply only to the infinite impulse response algorithm. Hence, the polynomial approximation technique depends on the choice of the filter algorithm and thus, the choice of a filter algorithm must precede the choice of a polynomial approximation technique. Similarly, if the designer chooses a bipolar logic family, then the Manchester carry adder logic design is not a viable option because this option requires pass transistor technology that exists only in MOS-related logic families. Hence, the adder's logic design depends on the choice of the logic family and thus, the choice of a logic family must precede the choice of a logic design style.

Conceptual design supports such ordering of design issues via ordering constraints, enforcing that the more pervasive design issues be addressed first [23]. In specifying the interdependencies between design issues, as many or as few ordering constraints may be imposed as is warranted by the particular design domain. At one extreme, a strict sequential ordering of design issues may be required. At the other extreme, no ordering may occur, signifying that the design issues have no interdependencies and that each issue may be examined independently of the others. Partial orderings may also occur in which some design issues typically arise that may be examined and some design issues that may not be examined, owing to interdependencies. Hence, ordering constraints provide a general mechanism within the conceptual design methodology for encoding various design strategies, such as top-down, bottom-up, or meet-in-the-middle.

In addition to interdependencies between design issues (ordering constraints), there may be interdependencies between options. For the digital adder design example, in which the designer investigates the logic design style issue, only the options that are consistent with the selected logic family option are viable. Interdependencies between design options are supported via consistency constraints. Using

consistency constraints, a system planner helps to determine, for a particular design issue, which options are consistent with earlier decisions. The designer is not presented with an option (step 2 in Figure 7.11) that is inconsistent with his or her earlier decisions. As such, the designer is prevented from developing an incorrect or inconsistent plan.

7.3.2 Decision Assistance

It may be the situation for complex conceptual designs that the designer consults an expert on a particular issue. The expert offers assistance in understanding the relative advantages and disadvantages of the available options in order to select the most appropriate option(s). Two general forms of assistance, advice and prediction, are available.

Advice describes the qualitative advantages and/or disadvantages that should be considered before accepting or rejecting a design option. On the other hand, prediction describes the quantitative advantages and/or disadvantages. Prediction is a valuable tool for reaching down through several levels of the design hierarchy to obtain quantitative insights into the implications of high-level design decisions. Such advice is given in lieu of the more expensive route of actually executing the detailed design process in order to determine the effects of a decision on design performance. While the predicted performance is only an approximation, it is often sufficient to allow the designer to choose a particular option, or at least to rule out an option.

Different types of prediction techniques are summarized in the taxonomy presented in Figure 7.12. Heuristic models use general guidelines or empirical studies to model the design process [15]. Analytical models employ a more rigorous, mathematical approach to model the design process. Some such models may be based on probability theory, e.g., wirability theory, and capture the particulars of a design activity in a set of random variables [17,18], while other analytical models may be based on graph and information theory, e.g., very large scale integration (VLSI) complexity theory [19,24,25]. Finally, simulation models include the judicious partial execution of a design activity on critical portions of the design. The use of previous designs serves to capture the experience or ''corporate history'' factor used by an expert designer in attempting to predict the performance of a new function based on previous similar designs.

As an example, Figure 7.13 shows how performance estimates for digital filters can be obtained in a hierarchical, incremental fashion, employing a variety of prediction models that cooperate to yield the

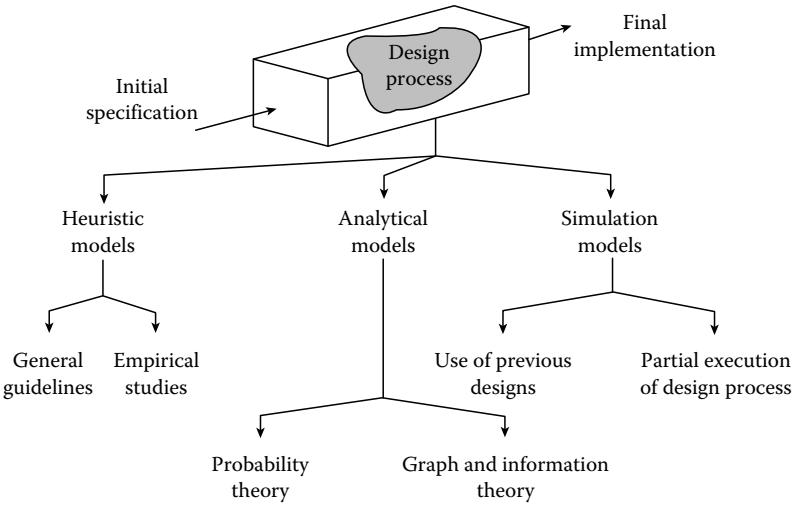


FIGURE 7.12 Types of prediction techniques.

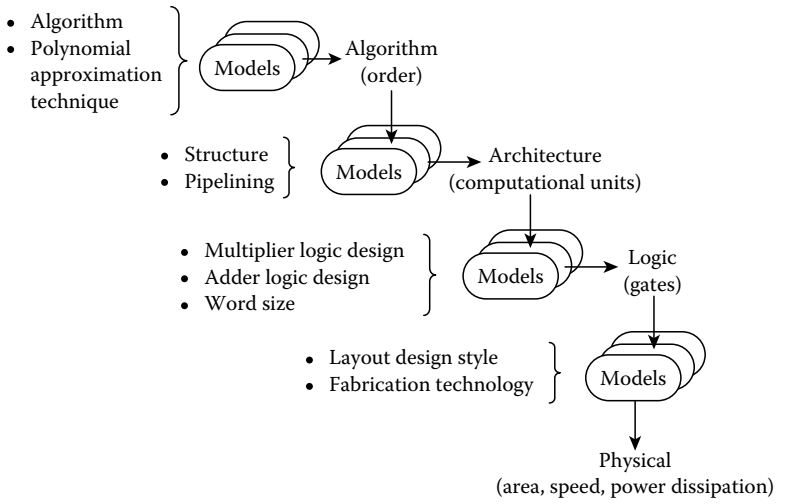


FIGURE 7.13 Hierarchical structure of filter performance prediction.

desired results. The prediction task is performed by incrementally translating the initial specifications into a series of complexity measures. Each succeeding complexity measure is a refinement of the performance estimate, starting with a performance estimate in terms of the filter order and ending with the performance estimate in terms of the physical units of area, speed, and power dissipation. Each prediction model draws its required input information from higher-level prediction models, the initial functional specifications, and the applicable decisions from the conceptual plan. Incrementally generating a prediction allows the flexibility to use different types of prediction techniques as the situation warrants to model the implications of different parts of the design plan.

The starting point in the prediction methodology is the initial functional specifications. For digital filters, functional specifications are usually given in terms of the desired frequency response, involving the following:

- Frequency bands of importance
- Widths of the transition bands between the desired frequency bands
- Levels of amplification or attenuation per frequency band
- Allowable tolerance or deviation from the specified levels of amplification and/or attenuation

An extensive set of prediction models has been developed for predicting the performance of linear, shift-invariant digital filters having the general I–O behavior given by the difference equation

$$y(n) = \sum_{k=1}^{N} a_k(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{7.1}$$

The variables $x(n)$ and $y(n)$ denote the values of signals $x(t)$ and $y(t)$ at time t_n. As an example of an algorithm-level prediction model, the order of a low-pass filter using the equiripple polynomial approximation technique is given by the empirical model shown in Equation 7.2 [21].

$$N \approx \frac{\left\{ \begin{array}{c} \left[5.309 \times 10^{-3}(\log_{10}\delta_1)^2 + 0.07114\log_{10}\delta_1 - 0.4761\right]\log_{10}\delta_2 \\ + \left[-2.66 \times 10^{-3}\left(\log_{10}\delta_1^2\right) - 0.5941\log_{10}\delta_1 - 0.4278\right] \end{array} \right\}}{\Delta F} - \left[0.51244(\log_{10}\delta_1 - \log_{10}\delta_2) + 11.01217\right]\Delta F + 1 \tag{7.2}$$

where N is the filter order, δ_1 is the passband tolerance, δ_2 is the stopband tolerance, and ΔF is the width of the transition band. As another example of an algorithm prediction model, the order of a low-pass filter using the elliptic polynomial approximation technique is given by

$$N = \frac{\Psi(\alpha)\Psi\left(\sqrt{1-\beta^2}\right)}{\Psi(\beta)\Psi\left(\sqrt{1-\alpha^2}\right)}$$
$$\alpha = \frac{\omega_p}{\omega_s} \tag{7.3}$$
$$\beta = \frac{\sqrt{\delta_1}\sqrt{2-\delta_1\delta_2}}{\sqrt{1-\delta_2^2}(1-\delta_1)}$$

where ω_p is the cutoff frequency or the high end of the passband, ω_s is the start or low end of the stopband, and $\Psi(\cdot)$ denotes the complete elliptic integral of the first kind.

As an example of a logic level prediction model, filter word size can be estimated to be the maximum of the required data word size and coefficient word size. Equation 7.4 estimates the data word size ($data_{ws}$), i.e., number of bits required to represent the filter input data samples to ensure an adequate signal:noise ratio:

$$data_{ws} = \left\lceil \frac{SNR_q - 4}{6} \right\rceil \tag{7.4}$$

where $[\cdot]$ denotes the ''smallest integer not less than'' and SNR_q denotes the root mean square signal: quantization noise ratio in decibels. Equation 7.5 estimates the coefficient word size ($coeff_{ws}$), i.e., number of bits required to represent the filter coefficients to ensure that finite precision coefficient quantization and arithmetic round-off errors do not appreciably deteriorate filter frequency response:

$$coeff_{ws} = \left\lceil \log_2\left(\frac{\omega_s + \omega_p}{\omega_s - \omega_p}\right) + \log_2\left(\frac{1}{\min(\delta_1, \delta_2)}\right) - \frac{1}{2}\log_2\left(\frac{\omega_S}{\omega_s - \omega_p}\right) + 3 \right\rceil \tag{7.5}$$

where ω_S denotes the sampling frequency.

7.3.3 Exploring Alternative Designs

After analyzing the viable options, the final steps in Figure 7.11 involve making the design decision. Sometimes designers want to delay making a decision during a conceptual design session until other aspects of the design can be explored. If more than one option for a particular design issue looks equally viable, then the designer may elect to consider all of the viable options. For example, returning to digital filters, the designer may wish to investigate the implications of using both the lookahead carry and block lookahead carry logic design styles for the adder. Later in the conceptual design session, when the competing adder design plans have been more fully developed and their relative merits more clearly understood, the designer may have the information necessary to select one logic design style over the other. Such an approach is often referred to as ''nonlinear planning'' or adopting a ''least-commitment strategy.''

Multiple, alternative plans support delaying design decisions. New plans are created each time more than one option is chosen for a design issue. For example, if the designer wishes to investigate both adder logic designs, then the planner records the decision by generating two plans. One design plan uses the lookahead carry adder logic design, and the other plan uses the block lookahead carry logic design. Having generated alternative plans, the designer can then further develop each plan independently by requesting that the planner switch the focus of the conceptual design session between the competing

plans. It should be noted that a consequence of creating alternative plans is that the conceptual design process may not always yield one plan. Rather, the conceptual design process may yield multiple plans that appear roughly equally promising. The conceptual design methodology is not intended to always possess the ability to definitively differentiate between alternative plans due to the general nature of the plans or the approximate nature of the predictions. In cases in which the conceptual design process yields more than one candidate plan, the designer could invoke more detailed synthesis and analysis design automation tools to further investigate and resolve the plans.

In a lengthy conceptual design session, a designer may want to reconsider and possibly change an earlier decision; this situation is supported via backtracking. When the designer requests a reconsideration of an earlier decision for a particular design issue, the planner must back up the state of the conceptual design session and reactivate the associated viable options. The designer may then reexamine the options and possibly select a different option. The backtracking process also involves checking the rest of the plan to see if any of the designer's other, earlier decisions might be affected by the new decision. If any other design decisions depend on the decision that the designer wants to change, then these decisions must be similarly undone. In this manner only the appropriate part of the conceptual design session is backed up; the decisions that are not dependent on the desired change are left unaffected.

7.3.4 Applications

Conceptual design involves well-defined, algorithmic-based knowledge and ill-defined, heuristic-based knowledge. For instance, maintaining alternative plans involves primarily algorithmic-based knowledge. For each plan, the available design issues are determined in accordance with the ordering constraints. For each available design issue, the viable options are determined in accordance with the decisions made thus far and the consistency constraints. As the conceptual design session progresses, new design decisions are recorded, or possibly, previous design decisions are changed.

In contrast, providing advice involves primarily heuristic-based knowledge. Based on the state of the plan, the appropriate piece of advice or prediction model is identified, guidance is formulated, and the results are supplied to the designer. Advice and predictions serve as knowledge sources working to solve the conceptual design ''problem.'' The knowledge sources opportunistically invoke themselves to address an applicable portion of the advice or prediction request, taking their input from the system plan(s), and posting their results back to the system plan(s) in an attempt to refine and develop a conceptual design.

Domain-dependent information concerning design issues, options, trade-offs, and interdependencies is typically specified as part of the knowledge acquisition process involved in initially constructing a system planner, and is separate from domain-independent information to facilitate creating planners for different design tasks and moving existing planners to new design tasks. If alternative plans are developed during conceptual design, multiple versions of the domain-dependent information are correspondingly created. Each version of the domain-dependent information, also called contexts, worlds, or belief spaces, contains a plan's state information, such as which design issues have been addressed, which options have been chosen, and which advice has been given. When a designer changes the focus of the conceptual design session between alternative plans, a context switch is performed between the associated versions of the domain-dependent information. An analogous situation is an operating system that allows multiple jobs. Each time the computer switches to a new job, the state of the current job must be saved, and the state of the new job must be restored.

Clearly, for complex design domains, the amount of domain-dependent information can be sizable. To limit the search time required to find applicable options or appropriate advice, the domain-dependent information can be segmented or partitioned by design issue. Such an organization allows for dynamically limiting the portion of information searched. For instance, in searching for appropriate advice, only the advice that is associated with the current design issue is made accessible. Thus, any search is confined to the advice associated with the current design issue; a search is never executed over the entire domain-dependent information.

Conceptual design is an emerging area of design automation that demonstrates the potential for significantly improving our design capability. Conceptual design involves studying a design process, modeling the process as a hierarchy of design issues, delineating the options, and identifying the associated interdependencies and discriminating characteristics. A conceptual design plan realizes the initial specifications by delineating the important design decisions. Such a methodology emphasizes spending more time at the initial stage of the design process in obtaining a more credible system specification. In addition to beginning an expensive design process with a careful definition of what is required of the design, conceptual design emphasizes the need for a general plan for how to realize the design.

Associated applications play a supporting role of managing and presenting the large and complex amount of information typically involved in conceptual design to assist the designer in making the decisions that constitute the conceptual design plan. Plans are developed incrementally and interactively. In a breadth-first fashion, conceptual design guides the designer through a myriad of alternative plans, allowing the designer to explore differences and seek optimal solutions. The advantage of conceptual design is that expensive design resources are not wasted in trying plans that eventually prove unsatisfactory. Applications also provide performance estimation, which can be useful in coordinating between groups involved with a design. For example, an early estimate of chip area can be used to check with the manufacturing group that the design will be within acceptable fabrication yield margins or will adhere to any packaging or printed circuit board size restrictions. An early estimate of chip performance also can be used to coordinate with marketing and management in projecting potential applications, probable costs, and future business.

7.4 Synthesis

Figure 7.14 shows that the synthesis task generally follows the conceptual design task. The designer describes the desired system via design entry, generates a conceptual plan of major design decisions, and then invokes synthesis design automation programs to assist in generating the required implementation [32].

Synthesis translates or transforms a design at one level of abstraction to another, more detailed level of abstraction. The more detailed level of abstraction may be only an intermediate step in the entire design process or it may be the final implementation. Synthesis programs that yield a final implementation are sometimes called ''silicon compliers'' because the programs generate sufficient detail to proceed directly to silicon fabrication [26,29].

Similar to design abstractions, synthesis techniques can be hierarchically categorized, as shown in Figure 7.15. The higher levels of synthesis offer the advantage of less complexity, but also the disadvantage of less control over the final implementation.

Algorithmic synthesis also called ''behavioral synthesis,'' addresses ''multicycle'' behavior, which means behavior that spans more than one control step. A control step equates to a clock cycle of a synchronous, sequential digital system, i.e., a state in a finite state machine controller or a microprogram step in a microprogrammed controller. Algorithmic synthesis typically accepts sequential design descriptions using a procedural or imperative modeling style. Such descriptions define the I–O transform, but provide little information about the parallelism of the implementation [28,30].

Partitioning decomposes the design description into smaller behaviors. Partitioning is an example of a

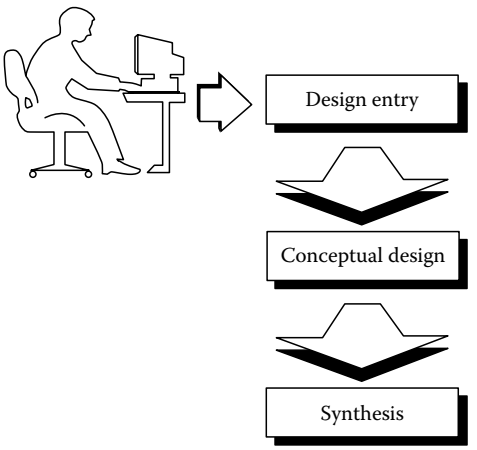


FIGURE 7.14 The design process.

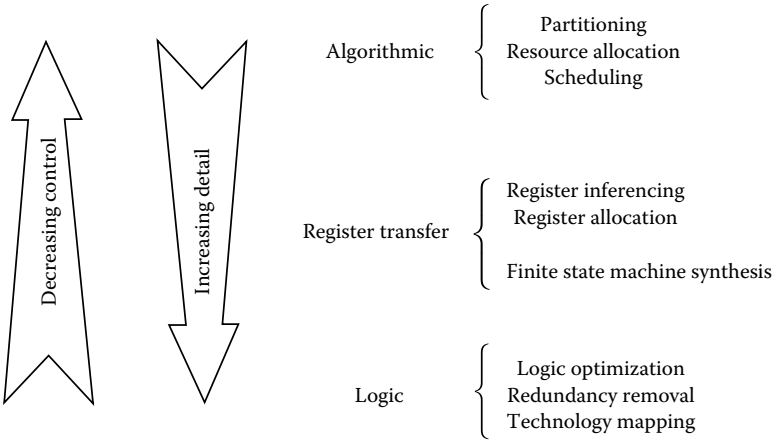


FIGURE 7.15 Taxonomy of synthesis techniques.

high-level transformation that modifies the initial sequential design description to optimize a hardware implementation. High-level transformations include several common software programming complier optimizations, such as loop unrolling, subprogram in-line expansion, constant propagation, and common subexpression elimination.

Resource allocation associates behaviors with hardware computational units and scheduling determines the order in which behaviors execute. Behaviors that are mutually exclusive can potentially share computational resources. Allocation is performed using a variety of graphic clique covering or node coloring algorithms. Allocation and scheduling are interdependent and different synthesis strategies perform allocation and scheduling in different ways. Sometimes scheduling is performed first, followed by allocation; sometimes allocation is performed first, followed by scheduling; and sometimes allocation and scheduling are interleaved.

Scheduling assigns computational units to control steps, thereby determining which behaviors execute in which clock cycles. At one extreme, all computational units can be assigned to a single control step, exploiting maximum concurrency. At the other extreme, computational units can be assigned to individual control steps, exploiting maximum sequentiality. Figure 7.16 illustrates two possible schedules

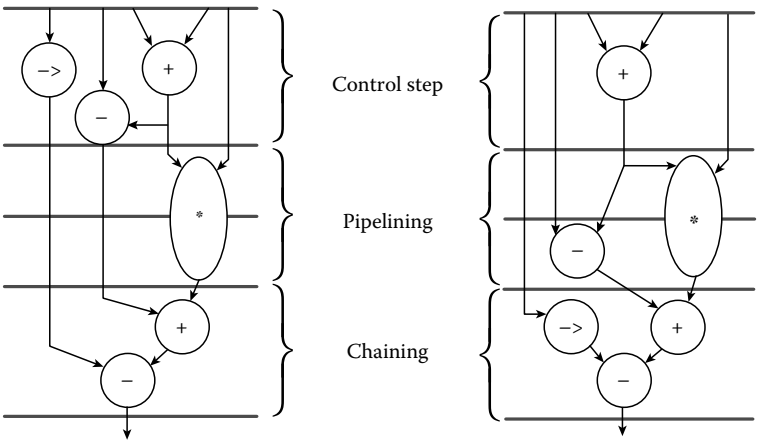


FIGURE 7.16 Scheduling.

of the same behavior. *Pipelining* schedules an operation across multiple control steps, whereas *chaining* combines multiple operations within a single control step.

Several popular scheduling algorithms are

- As soon as possible (ASAP)
- As late as possible (ALAP)
- List scheduling
- Force directed scheduling
- Control step splitting/merging

ASAP and ALAP scheduling algorithms order the computational units based on data dependencies. List scheduling is based on ASAP and ALAP scheduling, but can consider additional global constraints, such as a maximum number of control steps. Force-directed scheduling computes the probabilities of computational units being assigned to control steps and attempts to evenly distribute computation activity among all control steps. Control step splitting scheduling starts with all computational units assigned to one control step and generates a schedule by splitting the computational units into multiple control steps. Control step merging scheduling starts with all computational units assigned to individual control steps and generates a schedule by merging or combining units and steps [28,31].

Register transfer synthesis takes as input the results of algorithmic behavior and addresses ''per-cycle'' behavior, which means the behavior during one clock cycle. Register transfer synthesis selects logic to realize the hardware computational units generated during algorithmic synthesis, such as realizing an addition operation with a carry-save adder or realizing addition and subtraction operations with an ALU. Data that must be retained across multiple clock cycles are identified and registers are allocated to hold these data. Finally, state machine synthesis involves classical state minimization and state assignment techniques. State minimization seeks to eliminate redundant or equivalent states and state assignment assigns binary encodings for states to minimize combinational logic [27,33].

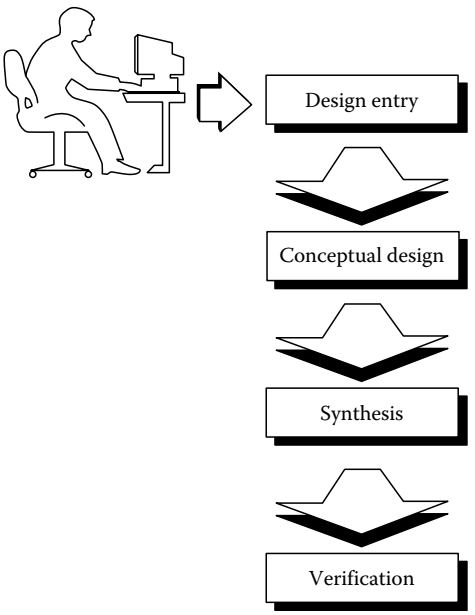


FIGURE 7.17 Design process.

Logic synthesis optimizes the logic generated by register transfer synthesis and maps the minimized logic operations onto physical gates supported by the target fabrication technology. Technology mapping considers the foundry cell library and associated electrical restrictions, such as fanin/fanout limitations.

7.5 Verification

Figure 7.17 shows that the verification task generally follows the synthesis task. The verification task checks the correctness of the function and performance of a design to ensure that an intermediate or final implementation faithfully realizes the initial, desired specification.

Several common types of verification are

- Timing analysis
- Simulation
- Emulation
- Formal verification

These types of verification are examined in more detail in the following sections [42].

7.5.1 Timing Analysis

As the name implies, timing analysis checks that the overall design satisfies operating speed requirements and individual signals within design satisfy transition requirements. Common signal transition requirements, also called timing hazards, include rise and fall times, propagation delays, clock times, race conditions, glitch detection, and setup and hold times. Set and hold timing checks are illustrated in Figure 7.18. For synchronous sequential digital systems, memory devices (level-sensitive latches or edge-sensitive flip-flops) require that the data and control signals obey setup and hold timings to ensure correct operation, i.e., that the memory device correctly and reliably stores the desired data. The control signal is typically a clock signal and Figure 7.18 assumes that a control signal transition, rising or falling, triggers activation of the memory device. The data signal carrying the information to be stored in the memory device must be stable for a period equal to the setup time prior to the control signal transition to ensure the correct value is sensed by the memory device. Also, the data signal must be stable for a period equal to the hold time after the control signal transition to ensure the memory device has enough time to store the sensed value.

Another class of timing transition requirements, commonly called ''signal integrity checks,'' include reflections, cross talk, ground bounce, and electromagnetic interference. Signal integrity checks are typically required for high-speed designs operating at clock frequencies above 75 MHz. At such high frequencies, the transmission line behavior of wires must be analyzed. A wire must be properly terminated, i.e., connected, to a port having an impedance matching the wire characteristic impedance to prevent signal reflections. Signal reflections are portions of an emanating signal that ''bound back'' from the destination to the source. Signal reflections reduce the power of the emanating signal and can damage the source. Cross talk refers to unwanted reactive coupling between physically adjacent signals, providing a connection between signals that are supposed to be electrically isolated. Cross talk causes information carried on a signal to interfere or corrupt information carried on a neighboring signal. Ground bounce is another signal integrity problem. Because all conductive material has finite impedance, a ground signal network does not in practice offer the same electrical potential throughout an entire design. These potential differences are usually negligible because the distributive impedance of the ground signal network is small compared to other finite component impedances. However, when many signals switch value simultaneously, a substantial current can flow through the ground signal network. High intermittent currents yield proportionately high intermittent potential drops, i.e., ground bounces, which can cause unwanted circuit behavior. Finally, electromagnetic interference refers to signal harmonics radiating from design components and interconnects. This harmonic radiation may interfere with other electronic equipment or may exceed applicable environmental safety regulatory limits [38].

Timing analysis can be done dynamically or statically. Dynamic timing analysis exercises the design via simulation or emulation for a period of time with a set of input stimuli and records the timing

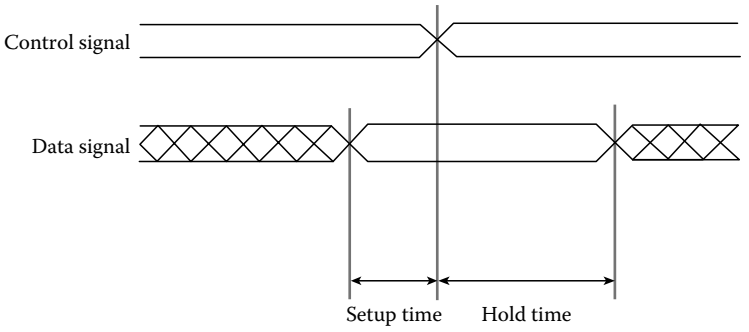


FIGURE 7.18 Signal transition requirements.

behavior. Dynamic timing analysis generally requires simulating many input test vectors to extensively exercise the design to ensure that signal paths are sufficiently identified and characterized. Static timing analysis does not exercise the design via simulation or emulation. Rather, static analysis records timing behavior based on the timing behavior (e.g., propagation delay) of the design components and their interconnection. With static timing analysis, the complexity of the verification task is partitioned into separate functional and performance checks.

Static timing analysis techniques are primarily *block oriented* or *path oriented*. Block-oriented timing analysis generates design input, also called primary input, to design output, also called primary output, propagation delays by analyzing the design, ''stage-by-stage'' and summing up the individual stage delays. All devices driven by primary inputs constitute stage 1, all devices driven by the outputs of stage 1 constitute stage 2, and so on. Starting with the first stage, all devices associated with a stage are annotated with worst-case delays. A worst-case delay is the propagation delay of the device plus the delay of the last input to arrive at the device, i.e., the signal path with the longest delay leading up to the device inputs. For example, the device labeled ''H'' in stage 3 in Figure 7.19 is annotated with the worst-case delay of 13, representing the device propagation delay of 4 and the delay of the last input to arrive through devices ''B'' and ''C'' of 9 [39]. When the devices associated with the last stage, i.e., the devices driving the primary outputs, are processed, the accumulated worst-case delays record the longest delay from primary inputs to primary outputs, also called the critical paths. The critical path for each primary output is highlighted in Figure 7.19.

Path-oriented timing analysis generates primary input to primary output propagation delays by traversing all possible signal paths one at a time. Thus, finding the critical path via path-oriented timing analysis is equivalent to finding the longest path through a directed acyclic graph, where devices are graph vertices and interconnections are graph edges [41].

A limitation of static timing analysis concerns detecting *false violations* or *false paths*. False violations are signal timing conditions that may occur due to the static structure of the design, but do not occur due to the dynamic nature of the design's response to actual input stimuli.

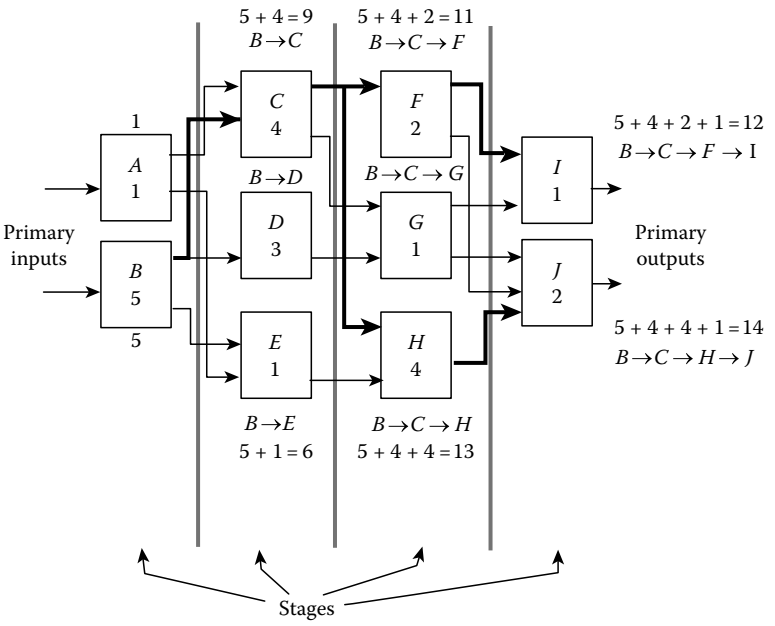


FIGURE 7.19 Block-oriented static timing analysis.

To account for realistic variances in component timing due to manufacturing tolerances, aging, or environmental effects, timing analysis often provides stochastic or statistical checking capabilities. Statistical timing analysis uses random number generators based on empirically observed probabilistic distributions to determine component timing behavior. Thus, statistical timing analysis describes design performance and the likelihood of the design performance.

7.5.2 Simulation

Simulation exercises a design over a period of time by applying a series of input stimuli and generating the associated output responses. The general event-driven, or schedule-driven, stimulation algorithm is diagrammed in Figure 7.20. An event is a change in signal value. Simulation starts by initializing the design; initial values are assigned to all signals. Initial values include starting values and pending values which constitute future events. Simulation time is advanced to the next pending event(s), signals are updated and sensitized models are evaluated. Sensitized models refer to models having outputs dependent on the updated signals [35,40]. The process of evaluating the sensitized model yields new, potentially different values for signals, i.e., a new set of pending events. These new events are added to the list of pending events, time is advanced to the next pending event(s), and the simulation algorithm repeats. Each pass through the loop of evaluating sensitized models at a particular time step is called a simulation cycle (see Figure 7.20). Simulation ends when the design yields no further activity, i.e., no more pending events exist to process.

Logic simulation is a computational intensive task for large, complex designs. Prior to committing to manufacturing, processor designers often simulate bringing up or "booting" the operating system. Such simulation tasks require sizable simulation computational resources. As an example, consider simulating 1s of a 200K gate, 20 MHz processor design. Assuming that on average only 10% of the total 200K gates are active or sensitized on each processor clock cycle, Equation 7.6 shows that simulating 1 s of actual processor time equates to 400 billion events.

$$\begin{aligned} 400\text{ B events} &= (20\text{ M clock cycles})(200\text{K gates})(10\%\text{ activity}) \\ 140\text{ h} &= (400\text{ B events})\left(\frac{50\text{ instructions}}{\text{event}}\right)\left(\frac{50\text{ M instruction}}{\text{s}}\right) \end{aligned} \tag{7.6}$$

Assuming that on average a simulation program executes 50 computer instructions per event on a computer capable of processing 50 million instructions per second, Equation 7.6 also shows that

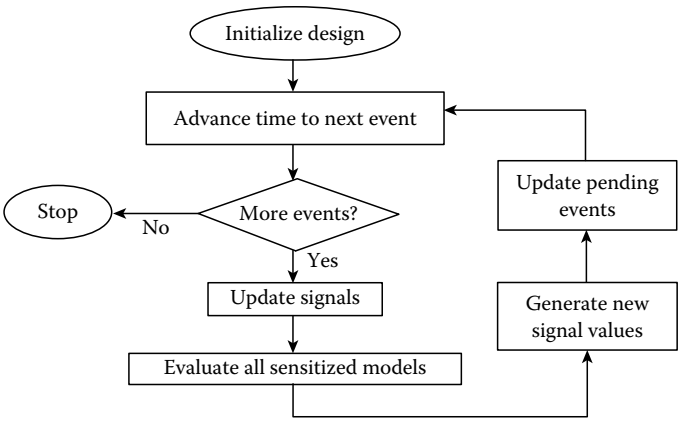


FIGURE 7.20 General event-driven simulation algorithm.

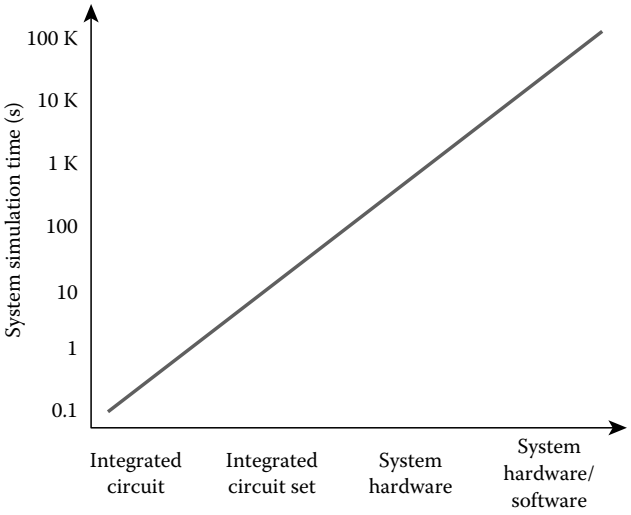


FIGURE 7.21 Simulation requirements.

processing 400 billion events requires 140 h or just short of 6 days. This simple example demonstrates the sizable computational properties of simulation. Figure 7.21 shows how simulation computation scales with design complexity.

To address the growing computational demands of simulation, several simulation acceleration techniques have been introduced. Schedule or event-driven simulation (explained previously) can be accelerated by removing layers of interpretation and running simulation as a native executable image; such an approach is called complied, scheduled-driven simulation. Schedule-driven simulation can be accelerated also by using more efficient event management schemes. In a conventional, central event management scheme all events are logged into a time-ordered list. As simulation time advances, pending events become actual events and the corresponding sensitized devices are executed to compute the response events. One the other hand, in a dataflow event management scheme events are ''self-empowered,'' active agents that flow through networks and trigger device evaluations without registering with a central time-order list and dispatcher [37].

Instead of evaluating a device in a stimulus–response manner, *cycle-driven* simulation avoids the overhead of event queue processing by evaluating all devices at regular intervals of time. Cycle-driven simulation is efficient when a design exhibits a high degree of concurrency, i.e., a large percentage of the devices are active per simulation cycle. Based on the staging of devices, the devices are *ranked-ordered* to determine the order in which they are evaluated at each time step to ensure the correct causal behavior yielding the proper ordering of events. For functional verification, logic devices are often assigned zero-delay and memory devices are assigned unit-delay. Thus, any number of stages of logic devices may execute between system clock periods.

Another simulation acceleration technique is *message-driven* simulation, also called parallel or distributed simulation. Device execution is divided among several processors, and the device simulations communicate event activity via messages. Messages are communicated using a conservative or an optimistic strategy. Optimistic message passing strategies, such as *time warp* and *lazy cancellation*, make assumptions about future event activity to advance local device simulation. If the assumptions are correct, the processors operate more independently and better exploit parallel computation. However, if the assumptions are incorrect, then local device simulations may be forced to ''roll back'' to synchronize local device simulations [34,36].

Schedule-driven, cycle-driven, and message-driven simulation are software-based simulation acceleration techniques. Simulation also can be accelerated by relegating certain simulation activities to dedicated hardware. For example, *hardware modelers* can be attached to simulators to accelerate the activity of device evaluation. As the name implies, hardware modeling uses actual hardware devices instead of software models to obtain stimulus–response information. In a typical scenario, the hardware modeler receives input stimuli from the software simulator. The hardware modeler then exercises the device and sends the output response back to the software simulator. Using actual hardware devices reduces the expense of generating and maintaining software models and provides an environment to support application software development. However, the hardware device must exist, which means hardware modeling has limited use in the initial stages of design in which hardware implementations are not available. Also, it is sometimes difficult for a slave hardware modeler to preserve accurate real-time device operating response characteristics within a master non-real-time software simulation environment. For example, some hardware devices may not be able to retain state information between invocations, so the hardware modeler must save the history of previous inputs and reapply them to bring the hardware device to the correct state in order to apply a new input.

Another technique for addressing the growing computational demands of simulation is via *simulation engines*. A simulation engine can be viewed as an extension of the simulation acceleration techniques of hardware modeling. With a hardware modeler, the simulation algorithm executes in software, and component evaluation executes in dedicated hardware. With a simulation engine, the simulation algorithm and component evaluation execute in dedicated hardware. Simulation engines are typically two to three orders of magnitude faster than software simulation [43].

7.5.3 Emulation

Emulation, also called ''computer-aided prototyping,'' verifies a design by realizing the design in ''preproduction'' hardware and exercising the hardware. The term preproduction hardware means nonoptimized hardware providing the correct functional behavior, but not necessarily the correct performance. That is, emulation hardware may be slower, require more area, or dissipate more power than production hardware. Presently, preproduction hardware commonly involves some form of *programmable logic devices*, typically *field-programmable gate arrays*. Programmable logic devices provide generic combinational and sequential digital system logic that can be programmed to realize a wide variety of designs [44].

Emulation begins by partitioning the design; each design segment is realized by a programmable logic device. The design segments are then interconnected to realize a preproduction implementation and exercised with a series of input test vectors.

Emulation offers the advantage of providing prototype hardware early in the design cycle to check for errors or inconsistencies in initial functional specifications. Problems can be isolated and design modifications can be accommodated easily by reprogramming the logic devices. Emulation can support functional verification at a computational rate much greater than conventional simulation. However, emulation does not generally support performance verification because, as explained previously, prototype hardware typically does not operate at production clock rates.

7.6 Test

Figure 7.22 shows that the test task generally follows the verification task. Although the verification and test tasks both seek to check for correct function and performance, verification focuses on a model of the design before manufacturing, whereas test focuses on the actual hardware after manufacturing. Thus, the primary objective of test is to detect a faulty device by applying input test stimuli and observing expected results [47,53].

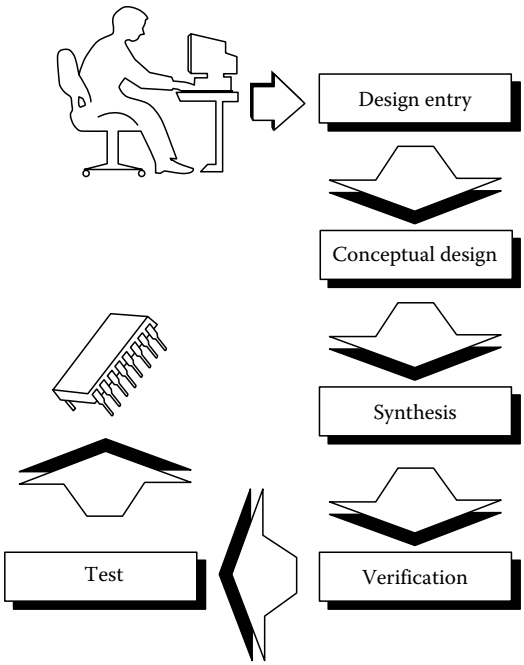


FIGURE 7.22 Design process.

The test task is difficult because designs are growing in complexity; more components provide more opportunity for manufacturing defects. Test is also challenged by new microelectronic fabrication processes having new failure modes which again provides more opportunity for manufacturing defects. New microelectronic fabrication processes also offer higher levels of integration with fewer access points to probe internal electrical nodes. To illustrate the growing demands of test, Table 7.2 shows the general proportional relationship between manufacturing defects and required fault coverage, i.e., quality of testing. Figure 7.23 shows the escalating costs of testing equipment.

Testing involves three general testing techniques or strategies: functional, parametric, and fault. Functional testing checks that the hardware device realizes the correct I–O digital system behavior. Parametric testing checks that the hardware device realizes the correct performance specifications, such as speed or power dissipation, and electrical specifications, such as voltage polarities and current sinking/sourcing limitation. Finally, fault testing checks for manufacturing defects or ''faults.''

TABLE 7.2 Fault Coverage and Defect Rate

	Defect Rate		
Microelectronic Fabrication Process	70% Fault Coverage	90% Fault Coverage	99% Fault Coverage
2 μm			
90% yield	3%	1%	0.1%
1.5 μm			
50% yield	19%	7%	0.7%
1 μm			
10% yield	50%	21%	2%

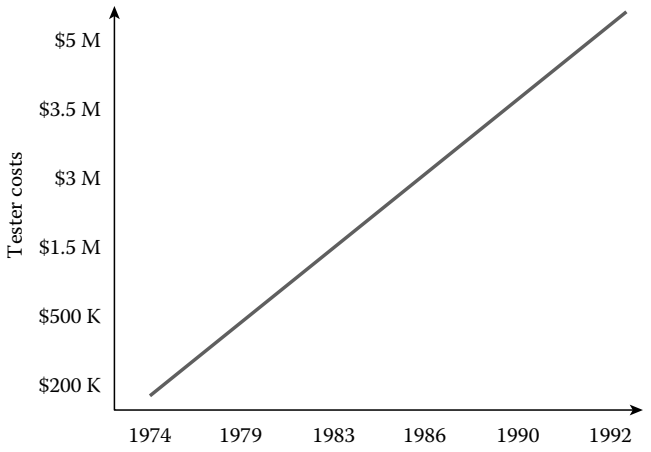


FIGURE 7.23 Integrated circuit tester equipment cost.

7.6.1 Fault Modeling

A fault is a manufacturing or aging defect that causes a device to operative incorrectly or to fail. A sample listing of common integrated circuit physical faults is given next:

- Wiring faults
- Dielectric faults
- Threshold faults
- Soft faults

Wiring faults are unwanted opens and shorts. Two wires or networks that should be electrically connected but are not constitute an open. Two wires or networks that should not be electrically connected but are constitute a short. Wiring faults can be caused by manufacturing defects such as metallization or etching problems, or aging defects, such as corrosion or electromigration. Dielectric faults are electrical isolation defects that can be caused by masking defects, material impurities or imperfections, and electrostatic discharge. Threshold faults occur when the turn-on and turn-off voltage potentials of electrical devices exceed allowed ranges. The faulty devices cannot be properly operated, which results in component failure. Soft faults occur when radiation exposure temporarily changes electrical charge distributions. Such changes can alter circuit voltage potentials, which can, in turn, change logical values, also called ''dropping bits.'' Radiation effects are called ''soft'' faults because the hardware is not permanently damaged [54].

To simplify the task of fault testing, the physical faults described above are translated into logical faults. Typically, a single logical fault covers several physical faults. A popular logical fault model is the *single stuck line* fault model. The single stuck line fault model considers faults where any single signal line or wire is permanently set to a logic 0, ''stuck-at-0,'' or a logic 1, ''stuck-at-1.'' These signal or interconnect faults are assumed to be time invariant.

Building on the single stuck line fault model, the *multiple stuck line* fault model considers multiple signal wire stuck-at-0/stuck-at-1 faults. The multiple stuck line fault model is more expressive and can cover more physical faults than the single stuck line model. However, fault testing for the multiple stuck line fault model is more difficult because of the exponential growth in the possible combinations of multiple signal faults.

Stuck fault models do not address all physical faults because not all physical faults result in signal lines permanently set to low or high voltages, i.e., stuck-at-0 or stuck-at-1 logic faults. Thus, other fault models

have been developed to address specific failure mechanisms. For example, the *bridging* fault model addresses electrical shorts that cause unwanted coupling or spurious feedback loops. As another example, the *pattern-sensitive* fault model addresses wiring and dielectric faults that yield unwanted interference between physically adjacent signals. Pattern-sensitive faults are generally most prevalent in high-density memories incorporating low signal:noise ratios and can be difficult to detect because they are often data-pattern and data-rate dependent. In other words, the part fails only under certain combinations of input stimuli and only under certain operating conditions.

7.6.2 Fault Testing

Having identified and categorized the physical faults that may cause device malfunction or failure and determined how the physical faults relate to logical faults, the next task is to develop tests to detect these faults. When the tests are generated by a computer program, this activity is called ''automatic test program generation.'' Examples of fault testing techniques are listed next:

- Stuck-at techniques
- Scan techniques
- Signature techniques
- Ad hoc techniques
- Coding techniques
- Electrical monitoring techniques

The following paragraphs review these testing strategies.

Basic stuck-at techniques generate input stimuli for fault testing combinational digital systems. Three of the most popular stuck-at fault testing techniques are the D algorithm, the path-oriented decision-making (Podem) algorithm, and the fan algorithm. These algorithms first identify a circuit fault (e.g., stuck-at-0 or stuck-at-1) and then try to generate an input stimulus that detects the fault and makes the fault visible as an output. Detecting a fault is often called ''fault sensitization'' and making a fault visible is often called ''fault propagation.'' To illustrate this process, consider the simple combinational design in Figure 7.24 [46,49,50]. The design is defective because a manufacturing defect has caused the output of the `and` gate to be permanently tied to ground, i.e., stuck-at-0, using a positive logic convention. To sensitize the fault, the inputs A and B should both be set to 1, which should force the `and` gate output to a 1 for a good circuit. To propagate the fault, the inputs C and D should both be set to 0, which will force the `xor` gate output to 1, again for a good circuit. Thus, if $A = 1$, $B = 1$, $C = 0$, and $D = 0$ in Figure 7.24, then a good circuit would yield a 1, but the defective circuit yields a 0 which detects the stuck-at-0 fault at the `and` gate output.

Sequential automatic test program generation is a more difficult task than combinational automatic test program generation because exercising or sensitizing a particular circuit path to detect the presence of a possible manufacturing fault may require a sequence of input test vectors. One technique for testing sequential digital systems is called scan fault testing. Scan fault testing is called a ''design-for-testability'' technique because it modifies or constrains the design in a manner that facilitates fault testing. Scan techniques impose a logic design discipline that all state registers be connected in one or more chains to form ''scan rings,'' as shown in Figure 7.25 [48]. During normal device operation, the scan rings are disabled and the registers serve as conventional memory (state) storage elements. During test operation, the scan rings are enabled and

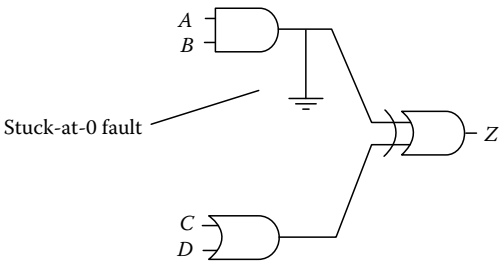


FIGURE 7.24 Combinational logic stuck-at fault testing.

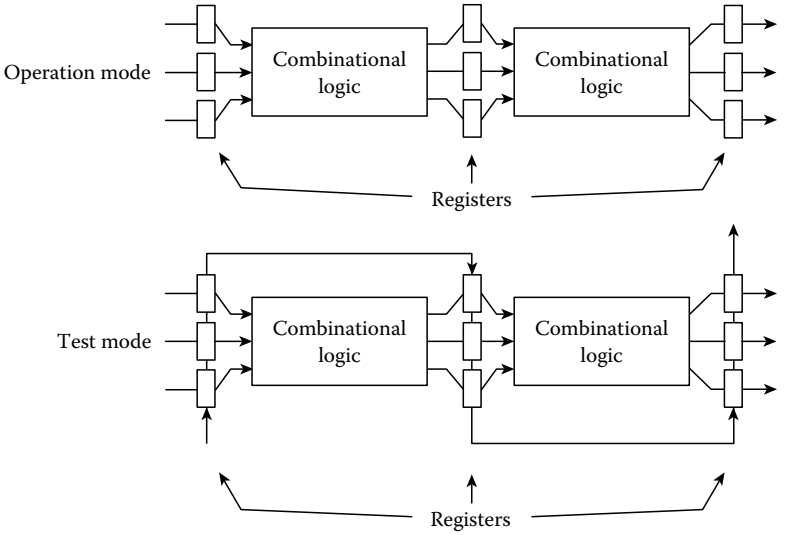


FIGURE 7.25 Scan-based design-for-testability.

stimulus test vectors are shifted into the memory elements to set the state of the digital system. The digital system is exercised for one clock cycle and then the results are shifted out of the scan ring to record the response.

The principal advantage of scan design-for-testability is that the scan ring decouples stages of combinational logic, thereby transforming a sequential design into effectively a combinational design and correspondingly a difficult sequential test task into a simpler combinational test task. However, fault tests must still be generated for the combinational logic, and shifting in stimuli and shifting out response requires time. Also, scan paths require additional hardware resources that can impose a performance (speed) penalty. To address these limitations, ''partial'' scan techniques have been developed that offer a compromise between testability and performance. Instead of connecting every register into scan chains, Figure 7.26 shows that a partial scan selectively connects a subset of registers into scan chains. Registers in critical performance circuit paths are typically excluded and registers providing control and observability to portions of the design are typically included. Similar to full scan test operation, stimulus test vectors are shifted into the memory elements to set the state of the digital system. Then, a partial scan test exercises the digital system for multiple clock cycles (two clock cycles for the partial scan shown in Figure 7.26), and then the results are shifted out of the scan ring to record the response.

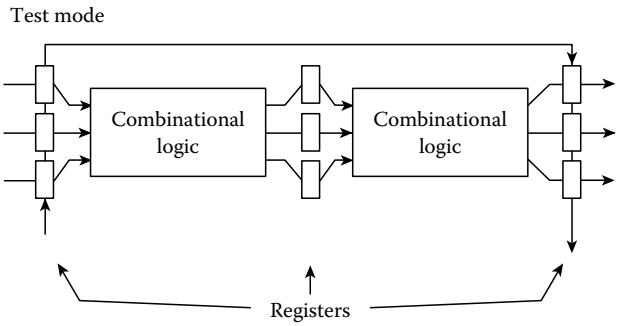


FIGURE 7.26 Partial scan-based design-for-testability.

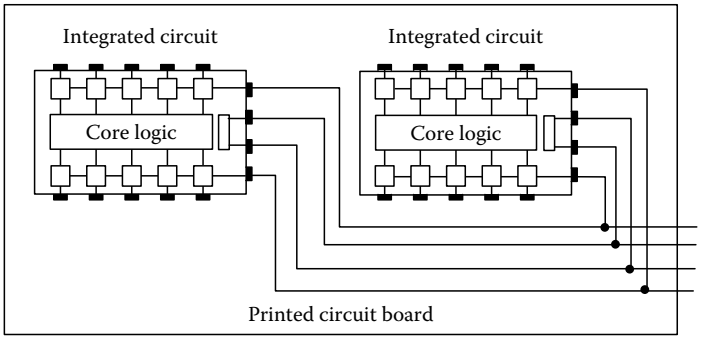


FIGURE 7.27 Boundary scan.

A variation of partial scan design-for-testability, called ''boundary scan,'' has been defined for testing integrated circuits on printed circuit boards. Printed circuit board manufacturing developments, such as fine-lead components, surface mount assembly, and multichip modules, yield high-density boards with fewer access points to probe individual pins. Such printed circuit boards are difficult to test. As the name implies, boundary scan imposes a design discipline on printed circuit board components, typically integrated circuits, such that the I–O pins on the integrated circuits can be connected into scan chains. Figure 7.27 shows that each integrated circuit configured for boundary scan contains scan registers between the I–O pins and the core logic to enable the printed circuit board test bus to control and observe the behavior of individual integrated circuits [51].

Another design-for-testability technique is signature analysis, also called ''built-in-self-test.'' Signature testing techniques use additional logic, typically linear feedback shift registers, to automatically generate pseudorandom test vectors. The output responses are compressed into a single vector and compared to a known good vector. If the output response vector does not exactly match the known good vector, then the design is considered faulty. Matching the output response vector and a known good vector does not guarantee correct hardware; however, if enough pseudorandom test vectors are exercised, then the chances are acceptably small of obtaining a false positive result. Signature analysis is often used to test memories [45].

Ad hoc testing techniques selectively insert test hardware and access points into a design to improve observability and controllability. Typical candidates for additional access points include storage elements (set and reset controls), major system communication buses, and feedback loops. Ad hoc testing techniques, also called behavioral testing techniques, can avoid the performance penalty of more structured, logical fault testing techniques, such as scan testing, and more closely mimics the actions of an expert test engineer. However, the number of access points is often restricted by integrated circuit or printed circuit board I–O pin limitations.

Coding test techniques encode signal information so that errors can be detected and possibly corrected. Although often implemented in software, coding techniques can also be implemented in hardware. For example, a simple coding technique called ''parity checking'' is often implemented in hardware. Parity checking adds an extra bit to multibit data. The parity bit is set such that the total number of logic 1's in the multibit data and parity bit is either an even number (even parity) or an odd number (odd parity). An error has occurred if an even parity encoded signal contains an odd number of logic 1's or an odd parity encoded signal contains an even number of logic 1's. Coding techniques are used extensively to detect and correct transmission errors on system buses and networks, storage errors in system memory, and computational errors in processors [52].

Finally, the electrical monitoring testing technique, also called current/voltage testing, relies on the simple observation that an out-of-range current or voltage often indicates a defective or bad part. Possibly a short or open is present, causing a particular I–O signal to have the wrong voltage or current.

Current testing (I_{ddq} testing) is particularly useful for digital systems using CMOS integrated circuit technology. Normally, CMOS circuits yield very low static or quiescent currents. However, physical faults, such as gate oxide defects, can increase static current by several orders of magnitude. Such a substantial change in static current is straightforward to detect. The principal advantages of current testing are that the tests are simple and the fault models address detailed transistor-level defects. However, current testing requires that enough time be allotted between input stimuli to allow the circuit to reach a static state, which slows down testing and causes problems with circuits that cannot be tested at scaled clock rates.

7.7 Frameworks

The previous sections discussed various types of design automation programs ranging from initial design entry tasks to final manufacturing test tasks. As the number and sophistication of design automation programs increases, the systems aspects of how the tools should be integrated with each other and underlying computing platforms become increasingly important. The first commercial design automation system product offerings in the 1970s consisted primarily of design automation programs and an associated computing platform ''bundled'' together in turnkey systems. These initial product offerings offered little flexibility to ''mix-and-match'' design automation programs and computing products from different source to take advantage of state-of-the-art capabilities and construct a design automation system tailored to particular end-user requirements. In response to these limitations, vendor offerings in the 1980s started to address *open systems*. Vendors ''unbundled'' software and hardware and introduced interoperability mechanisms to enable a design automation program to execute on different vendor systems. The interoperability mechanisms are collectively called ''frameworks'' [56].

Frameworks support design automation systems potentially involving many users, application programs, and host computing environments or platforms. Frameworks manage the complexities of a complete design methodology by coordinating and conducting the logistics of design automation programs and design data. Supporting the entire design cycle, also called ''concurrent engineering,'' improves productivity by globally optimizing the utilization of resources and the minimization of design errors and associated costs.

A general definition of a framework is given next:

> A CAD framework is a software infrastructure that provides a common operating environment for CAD tools. A framework should enable users to launch and manage tools; create, organize, and manage data; graphically view the entire design process; and perform design management tasks such as configuration and version management. Among the key elements of a CAD framework are platform-independent graphics and user interfaces, inter-tool communications, and design data and process management services. (CAD Framework Initiative, CFI)*

Figure 7.28 illustrates that a framework is essentially a domain-specific (i.e., electronic systems) layer of operating system software that facilitates ''plug-compatible'' design automation programs. Framework services are specific to electronic systems design and are mapped into the more generic services provided by general-purpose computing platforms [55].

User interface services provide a common and consistent ''look-and-feel'' to application programs. User interface services include support for consistent information display styles using menus and/or windows. These services also include support for consistent command styles using programming function keys and/or mouse buttons. Application program services provide support for program interactive/batch invocation and normal/abnormal termination. Application services also provide support for

* CAD Framework Initiative is a consortium of companies established to define and promote industry standards for design automation system interoperability.

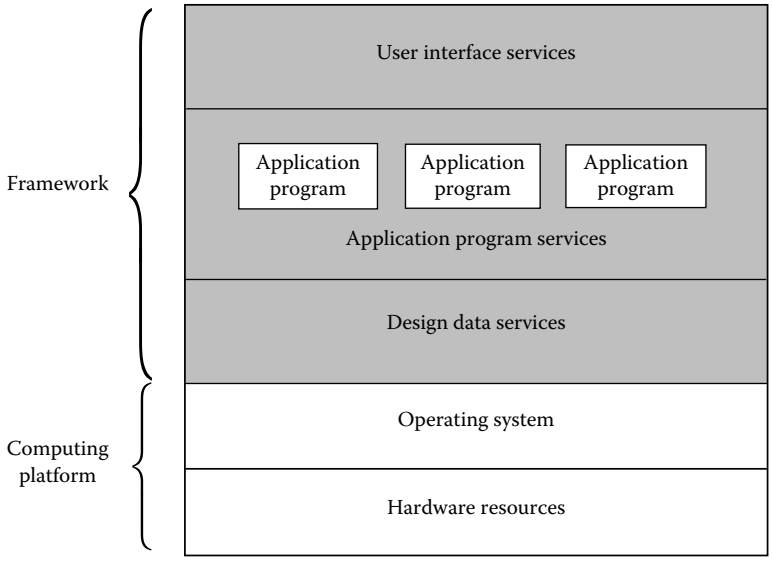


FIGURE 7.28 Framework architecture.

program-to-program communication and process management. Process management implements a design methodology that defines application-to-application and application-to-data dependencies. Process management describes a general sequencing of design automation programs and provisions for iterations. In other words, process management ensures a designer is working with the right tool at the right time with the right data.

Finally, a framework provides design data services to support access, storage, configuration, and integrity operations. Figure 7.29 shows relationships among relational, network, and object-oriented data management schemes. Due to the comparatively large size of design data and the length of design operations, framework data services are evolving toward object-oriented paradigms [57–59].

Object-oriented paradigms match data structures and operations with design automation objects and tasks, respectively. Figure 7.30 shows that hardware devices, components, and products become natural choices for software ''objects.''

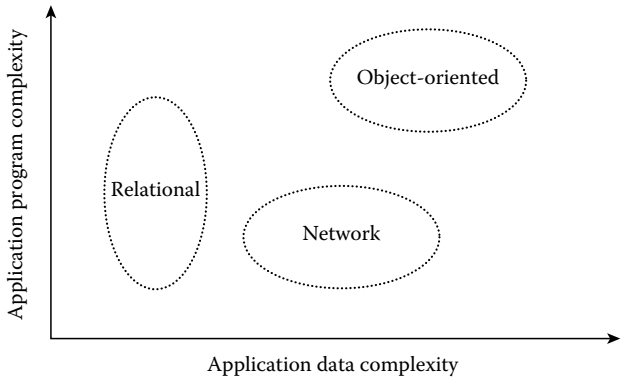


FIGURE 7.29 Data management technology.

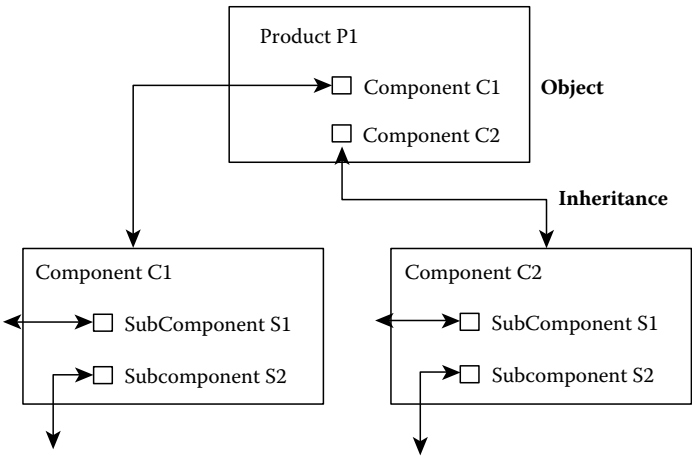


FIGURE 7.30 Object-oriented modeling.

The hierarchical relationships between hardware units become natural choices for software ''inheritance.'' Finally, design automation tasks and subtask that operate on hardware units, such as display or simulate, become natural choices for software ''methods.''

7.8 Summary

Design automation technology offers the potential of serving as a powerful fulcrum in leveraging the skills of a designer against the growing demands of electronic system design and manufacturing. Design automation programs help to relieve the designer of the burden of tedious, repetitive tasks that can be labor intensive and error prone.

Design automation technology can be broken down into several topical areas, such as design entry, conceptual design, synthesis, verification, testing, and frameworks. Each topical area has developed an extensive body of knowledge and experience.

Design entry defines a desire specification. Conceptual design refines the specification into a design plan. Synthesis refines the design plan into an implementation. Verification checks that the implementation faithfully realizes the desired specification. Testing checks that the manufactured part performs functionally and parametrically correctly. Finally, frameworks enable individual design automation programs to operate collectively and cohesively within a larger computing system environment.

References

1. D. Barbe, Ed. *Very Large Scale Integration* (*VLSI*)—*Fundamentals and Applications*, New York: Springer-Verlag, 1980.
2. T. Dillinger, *VLSI Engineering*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
3. E. Hollis, *Design of VLSI Gate Array Integrated Circuits*, Englewood Cliffs, NJ: Prentice Hall, 1987.
4. S. Sapiro, *Handbook of Design Automation*, Englewood Cliffs, NJ: Prentice Hall, 1986.
5. S. Trimberger, *An Introduction to CAD for VLSI*, San Jose, CA: Domancloud Publishers, 1990.
6. G. Birtwistle and P. Subrahmanyan, *VLSI Specification, Verification, and Synthesis*, Boston, MA: Kluwer Academic, 1988.
7. A. Dewey, VHSIC hardware description language development program, *Proc. Design Autom. Conf.*, June 1983.
8. A. Dewey, VHDL: Towards a unified view of design, *IEEE Design Test Comput.*, June 1992.

9. J. Douglas-Young, *Complete Guide to Reading Schematic Diagrams*, Englewood Cliffs, NJ: Prentice Hall, 1988.
10. C. Liu, *Elements of Discrete Mathematics*, New York: McGraw-Hill, 1985.
11. M. Pechet, Ed., *Handbook of Electrical Package Design*, New York: Marcel Dekker, 1991.
12. J. Peterson, *Petri Net Theory and Modeling of Systems*, Englewood Cliffs, NJ: Prentice Hall, 1981.
13. B. Spinks, *Introduction to Integrated Circuit Layout*, Englewood Cliffs, NJ: Prentice Hall, 1985.
14. N. Bose, *Digital Filters: Theory and Applications*, New York: North-Holland, 1985.
15. X. Chen and M. Bushnell, A module area estimator for VLSI layout, *Proc. Design Autom. Conf.*, June 1988.
16. A. Dewey, *Principles of VLSI Systems Planning: A Framework for Conceptual Design*, Boston: Kluwer Academic, 1990.
17. W. Donath, Wiring space estimation for rectangular gate arrays, *Proc. Int. Conf. VLSI*, 1981.
18. W. Heller, Wirability—Designing wiring space for chips and chip packages, *IEEE Design Test Comput.*, Aug. 1984.
19. C. Leiserson, Area-efficient VLSI computation, PhD dissertation, Pittsburgh, Carnegie Mellon University, 1981.
20. A. Oppenheim and R. Schafer, *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1975.
21. L. Rabiner, O. Hermann, and D. Chan, Practical design rules for optimum finite impulse response low-pass digital filters, *Bell Syst. Tech. J.*, July–Aug. 1973.
22. E. Sacerdoti, Planning in a hierarchy of abstraction spaces, *Artif. Intell.*, Sept. 1974.
23. M. Stefik, Planning with constraints, *Artif. Intell.*, Feb. 1980.
24. C. Thompson, Area-time complexity for VLSI, *Proc. Caltech Conf. VLSI*, Jan. 1979.
25. L. Valiant, Universality considerations in VLSI circuits, *IEEE Trans. Comput.*, Feb. 1981.
26. R. Ayres, *VLSI: Silicon Compilation and the Art of Automatic Microchip Design*, Englewood Cliffs, NJ: Prentice Hall, 1983.
27. R. Brayton et al., *Logic Minimization Algorithms for VLSI Synthesis*, Boston, MA: Kluwer Academic, 1992.
28. R. Camposano and W. Wolfe, *High-Level VLSI Synthesis*, Boston, MA: Kluwer Academic, 1991.
29. D. Gajski, Ed., *Silicon Compilation*, Reading, MA: Addison-Wesley, 1988.
30. D. Gajski et al., *High-Level Synthesis—Introduction to Chip and System Design*, Boston, MA: Kluwer Academic, 1992.
31. P. Paulin and J. Knight, Force-directed scheduling for the behavioral synthesis of ASIC's, *IEEE Design Test Comput.*, Oct. 1989.
32. B. Preas, M. Lorenzetti, and B. Ackland, Eds., *Physical Design Automation of VLSI Systems*, New York: Benjamin Cummings, 1988.
33. T. Sasao, Ed., *Logic Synthesis and Optimization*, Boston, MA: Kluwer Academic, 1993.
34. R. Bryant, Simulation on distributed systems, *Proc. Int. Conf. Distributed Systems*, 1979.
35. J. Butler, Ed., *Multiple-Valued Logic in VLSI Design*, New York: IEEE Computer Society Press, 1991.
36. K. Chandy and J. Misra, Asynchronous distributed simulation via a sequence of parallel computations, *Commun. ACM*, April 1981.
37. W. Hahn and K. Fischer, High performance computing for digital design simulation, *VLSI*85, Amsterdam: Elsevier, 1985.
38. R. McHaney, *Computer Simulation: A Practical Perspective*, New York: Academic Press, 1991.
39. T. McWilliams and L. Widdoes, SCALD—Structured computer aided logic design, *Proc. Design Autom. Conf.*, June 1978.
40. U. Pooch, *Discrete Event Simulation: A Practical Approach*, Boca Raton, FL: CRC Press, 1993.
41. T. Sasiki, et al., Hierarchical design and verification for large digital systems, *Proc. Design Autom. Conf.*, June 1978.
42. J. Sifakis, Ed., *Automatic Verification Methods for Finite State Systems*, New York: Springer-Verlag, 1990.

43. S. Takasaki, F. Hirose, and A. Yamada, Logic simulation engines in Japan, *IEEE Design Test Comput.*, Oct. 1989.
44. S. Walters, Computer-aided prototyping for ASIC-based synthesis, *IEEE Design Test Comput.*, June 1991.
45. V. Agrawal, C. Kime, and S. Saluja, A tutorial on built-in self test, *IEEE Design Test Comput.*, June 1993.
46. M. Breuer and A. Friedman, *Diagnosis and Reliable Design of Digital Systems*, New York: Computer Science Press, 1976.
47. A. Buckroyd, *Computer Integrated Testing*, New York: John Wiley & Sons, 1989.
48. E. Eichelberger and T. Williams, A logic design structure for LSI testability, *Proc. Design Autom. Conf.*, June 1977.
49. H. Fujiwara and T. Shimono, On the acceleration of test generation algorithms, *IEEE Trans. Comput.*, Dec. 1983.
50. P. Goel, An implicit enumeration algorithm to generalize tests for combinational logic circuits, *IEEE Trans. Comput.*, March 1981.
51. K. Parker, The impact of boundary scan on board test, *IEEE Design Test Comput.*, Aug. 1989.
52. W. Peterson and E. Weldon, *Error-Correcting Codes*, Cambridge, MA: MIT Press, 1972.
53. M. Weyerer and G. Goldemund, *Testability of Electronic Circuits*, Englewood Cliffs, NJ: Prentice Hall, 1992.
54. G. Zobrist, *VLSI Fault Modeling and Testing Technologies*, New York: Ablex Publishing, 1993.
55. T. Barnes, *Electronic CAD Frameworks*, Boston, MA: Kluwer Academic, 1992.
56. D. Bedworth, M. Henderson, and P. Wolfe, *Computer-Integrated Design and Manufacturing*, New York: McGraw-Hill, 1991.
57. R. Gupta and E. Horowitz, Eds., *Object-Oriented Databases with Applications to CASE Networks, and VLSI CAD*, Englewood Cliffs, NJ: Prentice Hall, 1990.
58. W. Kim, *Object-Oriented Concepts, Databases, and Applications*, Reading, MA: Addison-Wesley, 1989.
59. E. Nahouranii and F. Petry, Eds., *Object-Oriented Databases*, Los Alamitos, CA: IEEE Computer Society Press, 1991.

8

Computer-Aided Analysis

8.1 Circuit Simulation Using SPICE and SUPREM **8**-1
Introduction • DC (Steady-State) Analysis • AC Analysis • Transient Analysis • Process and Device Simulation • Process Simulation • Device Simulation
Appendix A **8**-20
Circuit Analysis Software • Process and Device Simulators
References **8**-20
8.2 Parameter Extraction for Analog Circuit Simulation **8**-21
Introduction • MOS DC Models • BSIM Extraction Strategy in Detail • Bipolar DC Model • MOS and Bipolar Capacitance Models • Bipolar High-Frequency Model • Miscellaneous Topics
References **8**-45

J. Gregory Rollins
Technology Modeling Associates, Inc.

Peter Bendix
Technology Modeling Associates, Inc.

8.1 Circuit Simulation Using SPICE and SUPREM

8.1.1 Introduction

Computer-aided simulation is a powerful aid during the design or analysis of electronic circuits and semiconductor devices. The first part of this chapter focuses on analog circuit simulation. The second part covers simulations of semiconductor processing and devices. While the main emphasis is on analog circuits, the same simulation techniques may, of course, be applied to digital circuits (which are, after all, composed of analog circuits). The main limitation will be the size of these circuits because the techniques presented here provide a very detailed analysis of the circuit in question and, therefore, would be too costly in terms of computer resources to analyze a large digital system.

The most widely known and used circuit simulation program is SPICE (simulation program with integrated circuit emphasis). This program was first written at the University of California at Berkeley by Laurence Nagel in 1975. Research in the area of circuit simulation is California at many universities and industrial sites. Commercial versions of SPICE or related programs are available on a wide variety of computing platforms, from small personal computers to large mainframes. A list of some commercial simulator vendors can be found in the Appendix A.

It is possible to simulate virtually any type of circuit using a program like SPICE. The programs have built-in elements for resistors, capacitors, inductors, dependent and independent voltage and current sources, diodes, MOSFETs, JFETs, bipolar junction transistors (BJTs), transmission lines, transformers, and even transformers with saturating cores in some versions. Found in commercial versions are libraries of standard components which have all necessary parameters prefitted to typical specifications.

These libraries include items such as discrete transistors, op-amps, phase-locked loops, voltage regulators, logic integrated circuits (ICs) and saturating transformer cores.

Computer-aided circuit simulation is now considered an essential step in the design of ICs, because without simulation the number of ''trial runs'' necessary to produce a working IC would greatly increase the cost of the IC. Simulation provides other advantages, however:

- Ability to measure ''inaccessible'' voltages and currents. Because a mathematical model is used all voltages and currents are available. No loading problems are associated with placing a voltmeter or oscilloscope in the middle of the circuit, with measuring difficult one-shot wave forms, or probing a microscopic die.
- Mathematically ideal elements are available. Creating an ideal voltage or current source is trivial with a simulator, but impossible in the laboratory. In addition, all component values are exact and no parasitic elements exist.
- It is easy to change the values of components or the configuration of the circuit. Unsoldering leads or redesigning IC masks are unnecessary.

Unfortunately, computer-aided simulation has its own problems:

- Real circuits are distributed systems, not the ''lumped elements models'' which are assumed by simulators. Real circuits, therefore, have resistive, capacitive, and inductive parasitic elements present besides the intended components. In high-speed circuits these parasitic elements are often the dominant performance-limiting elements in the circuit, and must be painstakingly modeled.
- Suitable predefined numerical modes have not yet been developed for certain types of devices or electrical phenomena. The software user may be required, therefore, to create his or her own models which are available in the simulator. (An example is the solid-state thyristor which may be created from a NPN and PNP bipolar transistor.)
- Numerical methods used may place constraints on the form of the model equations used.

The following sections consider the three primary simulation modes: DC, AC, and transient analysis. In each section an overview is given of the numerical techniques used. Some examples are then given, followed by a brief discussion of common pitfalls.

8.1.2 DC (Steady-State) Analysis

DC analysis calculates the state of a circuit with fixed (nontime varying) inputs after an infinite period of time. DC analysis is useful to determine the operating point (*Q*-point) of a circuit, power consumption, regulation and output voltage of power supplies, transfer functions, noise margin and fan-out in logic gates, and many other types of analysis. In addition, DC analysis is used to find the starting point for AC and transient analysis. To perform the analysis the simulator performs the following steps:

1. All capacitors are removed from the circuit (replaced with opens).
2. All inductors are replaced with shorts.
3. Modified nodal analysis is used to construct the nonlinear circuit equations. This results in one equation for each circuit node plus one equation for each voltage source. Modified nodal analysis is used rather than standard nodal analysis because an ideal voltage source or inductance cannot be represented using normal nodal analysis. To represent the voltage sources, loop equations (one for each voltage source or inductor), are included as well as the standard node equations. The node voltages and voltage source currents, then, represent the quantities which are solved for. These form a vector $\mathbf{x}$. The circuit equations can also be represented as a vector $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.
4. Because the equations are nonlinear, Newton's method (or a variant thereof) is used to solve the equations. Newton's method is given by the following equations:

$$J = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}^i}$$
$$\mathbf{x}^{i+1} = \mathbf{x}^i - J^{-1} \cdot \mathbf{F}(\mathbf{x}^i)$$

Here, if $\mathbf{x}^i$ is an estimate of the solution, $\mathbf{x}^{i+1}$ is a better estimate. The equations are used iteratively, and hopefully the vector **x** converges to the correct solution. The square matrix *J* of partial derivatives is called the Jacobian of the system. Most of the work in calculating the solution is involved in calculating *J* and its inverse J^{-1}. It may take as many a 100 iterations for the process to converge to a solution. Parameters control this process in most simulation programs (see the .OPTIONS statement in SPICE). For example, the maximum number of iterations allowed and error limits which must be satisfied before the process is considered to be converged, but normally the default limits are appropriate.

Example 8.1: Simulation Voltage Regulator

We shall now consider simulation of the type 723 voltage regulator IC, shown in Figure 8.1. We wish to simulate the IC and calculate the sensitivity of the output *IV* characteristic and verify that the output current follows a "fold-back" type characteristic under overload conditions.

The IC itself contains a voltage reference source and operational amplifier. Simple models for these elements are used here rather than representing them in their full form, using transistors to illustrate model development. The use of simplified models can also greatly reduce the simulation effort. (For example, the simple op-amp used here requires only eight nodes and ten components, yet realizes many advanced features.)

Note in Figure 8.1 that the numbers next to the wires represent the circuit nodes. These numbers are used to describe the circuit to the simulator. In most SPICE-type simulators the nodes are represented by numbers, with the ground node being node zero. Referring to Figure 8.2, the 723 regulator and its internal op-amp are represented by subcircuits. Each subcircuit has its own set of nodes and components. Subcircuits are useful for encapsulating sections of a circuit or when a certain section needs to be used repeatedly.

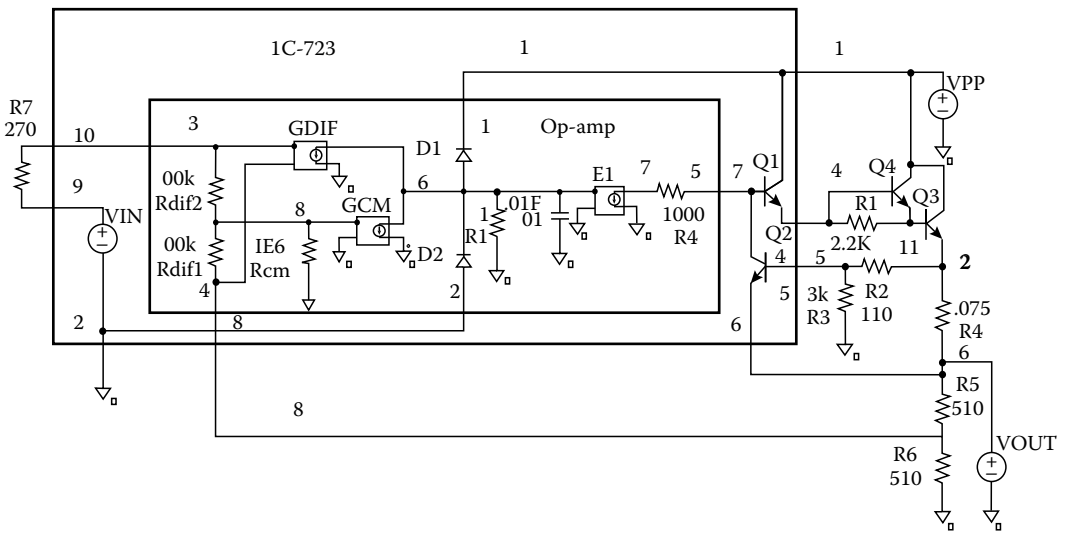


FIGURE 8.1 Regulator circuit to be used for DC analysis, created using PSPICE.

```
Regulator circuit.
*Complete circuit*
*Load source*
vout 6 0
*Power input*
vpp 1 0 11
x1 1 0 4 5 6 7 8 9 10 ic723
*Series Pass transistors*
q3 1 4 11 mq3
q4 1 11 2 mq4
r1 4 11 2.2k
r2 5 2 110
r3 5 0 3k
r4 2 6 0.075
r5 6 8 510
r6 8 0 510
r7 9 10 270
*Control cards*
.op
.model mq3 npn(is = 1e–9 bf = 30
+ br = 5 ikf = 50m)
model mq4 npn(is = 1e–6 bf = 30
+ br = 5 ikf = 10)
.dc vout 1 5.5 .01
.plot dc i(vout)
.probe

.subckt ic723 1 2 4 5 6 7 8 9 10
*Type 723 voltage regulator*
x1 1 2 10 8 7 opamp
*Internal voltage reference*
vr 9 2 2.5
q1 3 7 4 mm
q2 7 5 6 mm
.model mm npn (is = 1e–12 bf = 100)
+br = 5
.ends ic723
*Ideal opamp with limiting
.subckt opamp 1 2 3 4 5
*                vcc vee +in –in out
rdif1 3 8 le5
rdif2 4 8 le5
rcm 8 0 le6
*Common mode gain*
gcm 6 0 8 0 le–1
*Differential mode gain*
gdif 6 0 4 3 1 00
r1 6 0 1
*Single pole response*
c1 6 0 .01
d1 6 1 ideal
d2 2 6 ideal
e1 7 0 6 0 1
rout 5 7 le3
.model ideal d (is = le–6 n = .01)
.ends opamp
```

FIGURE 8.2 SPICE input listing of regulator circuit shown in Figure 8.1.

The following properties are modeled in the op-amp:

1. Common mode gain
2. Differential mode gain
3. Input impedance
4. Output impedance
5. Dominant pole
6. Output voltage clipping

The input terminals of the op-amp connect to a "T" resistance network, which sets the common and differential mode input resistance. Therefore, the common mode resistance is RCM + RDIF = 1.1E6 and the differential mode resistance is RDIF1 + RDIF2 = 2.0E5.

Dependent current sources are used to create the main gain elements. Because these sources force current into a $1-\Omega$ resistor, the voltage gain is Gm*R at low frequency. In the differential mode, this gives (GDIF*R1 = 100). In the common mode, this gives (GCM*R1*(RCM/(RDIF1 + RCM = 0.0909). The two diodes D1 and D2 implement clipping by preventing the voltage at node 6 from exceeding VCC or going below VEE. The diodes are made "ideal" by reducing the ideality factor n. Note that the diode current is $I_d = I_s[\exp(V_d/(nV_t)) - 1]$, where V_t is the thermal voltage (0.026 V). Thus, reducing n makes the diode turn on at a lower voltage.

A single pole is created by placing a capacitor (C1) in parallel with resistor R1. The pole frequency is therefore given by 1.0/2 * π * R1 * C1). Finally, the output is driven by the voltage-controlled voltage source E1 (which has a voltage gain of unity), through the output resistor R4. The output resistance of the op-amp is therefore equal to R4.

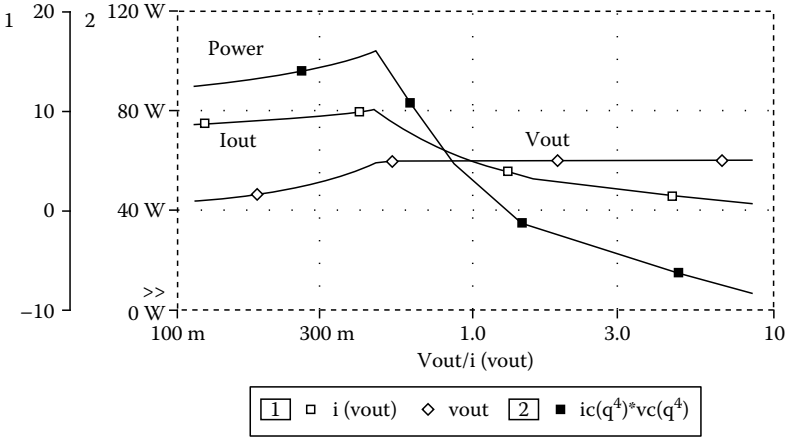


FIGURE 8.3 Output characteristics of regulator circuit using PSPICE.

To observe the output voltage as a function of resistance, the regulator is loaded with a voltage source (VOUT) and the voltage source is swept from 0.05 to 6.0 V. A plot of output voltage vs. resistance can then be obtained by plotting VOUT vs. VOUT/I(VOUT) (using PROBE in this case; see Figure 8.3). Note that for this circuit, even though a current source would seem a more natural choice, a voltage source must be used as a load rather than a current source because the output characteristic curve is multi-valued in current. If a current source were used it would not be possible to easily simulate the entire curve. Of course, many other interesting quantities can be plotted; for example, the power dissipated in the pass transistor can be approximated by plotting IC(Q3)*VC(Q3).

Several restrictions exist as to what constitutes a valid circuit, and in most cases the simulators will complain if the restrictions are violated:

1. All nodes must have a DC path to ground.
2. Voltage sources must not be connected in a loop.
3. Current sources may not be connected in series.
4. Each node must be connected to at least two elements.

For these simulations, PSPICE was used running on an IBM PC. The simulation took <1 min of CPU time.

Pitfalls. Many SPICE users forget that the first line in the input file is used as the title. Therefore, if the first line of the file is a circuit component, the component name will be used as the title and the component will not be included in the circuit. Convergence problems are sometimes experience if "difficult" bias conditions are created. An example of such a condition is if a diode is placed in the circuit backwards, resulting in a large forward bias voltage, SPICE will have trouble resolving the current. Another difficult case is if a current source were used instead of a voltage to bias the output in the previous example. If the user then tried to increase the output current above 10 A, SPICE would not be able to converge because the regulator will not allow such a large current.

8.1.3 AC Analysis

AC analysis uses phasor analysis to calculate the frequency response of a circuit. The analysis is useful for calculating the gain, 3 dB frequency input and output impedance, and noise of a circuit as a function of frequency, bias conditions, temperature, etc.

8.1.3.1 Numerical Method

1. DC solution is performed to calculate the Q-point for the circuit.
2. Linearized circuit is constructed at the Q-point. To do this, all nonlinear elements are replaced by their linearized equivalents. For example, a nonlinear current source $I = aV_1^2 + bV_2^3$ would be replaced by a linear voltage controlled current source $I = V_1(2aV_{1q}) + V_2(3bV_{2q}^2)$.
3. All inductors and capacitors are replaced by complex impedances, and conductances evaluated at the frequency of interest.
4. Nodal analysis is now used to reduce the circuit to a linear algebraic complex matrix. The AC node voltages may now be found by applying an excitation vector (which represents the independent voltage and current sources) and using Gaussian elimination (with complex arithmetic) to calculate the node voltages.

AC analysis does have limitations and the following types of nonlinear or large signal problems cannot be modeled:

1. Distortion due to nonlinearities such as clipping, etc.
2. Slew rate-limiting effects
3. Analog mixers
4. Oscillators

Noise analysis is performed by including noise sources in the models. Typical noise sources include thermal noise in resistors $I_n^2 = 4kT\Delta f/R$, and shot $I_n^2 = 2qI_d\Delta f$, and flicker noise in semiconductor devices. Here, T is temperature in Kelvin, k is Boltzmann's constant, and Δf is the bandwidth of the circuit. These noise sources are inserted as independent current sources, $\mathrm{In}_j(f)$ into the AC model. The resulting current due to the noise source is then calculated at a user-specified summation node(s) by multiplying by the gain function between the noise source and the summation node $A_{js}(f)$. This procedure is repeated for each noise source, and then the contributions at the reference node are root mean squared (RMS) summed to give the total noise at the reference node. The equivalent input noise is then easily calculated from the transfer function between the circuit input and the reference node $A_{is}(f)$. The equation describing the input noise is therefore:

$$I_i = \frac{1}{A_{is}(f)} \sqrt{\sum_j [A_{js}(f)\mathrm{In}_j(f)]^2}$$

Example 8.2: Cascode Amplifier with Macro Models

Here, we find the gain, bandwidth, input impedance, and output noise of a cascode amplifier. The circuit for the amplifier is shown in Figure 8.5. The circuit is assumed to be fabricated in a monolithic IC process, so it will be necessary to consider some of the parasitics of the IC process. A cross-section of a typical IC bipolar transistor is shown in Figure 8.4 along with some of the parasitic elements. These parasitic elements are easily included in the amplifier by creating a "macro model" for each transistor. The macro model is then implemented in SPICE form using subcircuits.

The PSPICE circuit simulator allows the user to define parameters which are passed into the subcircuits. This capability is very useful in this case because the resistor model will vary, depending on the value of the resistor. It if is assumed for a certain resistor type that the width w (measured perpendicular to current flow) of the resistor is fixed, e.g., to the minimum line width of the process, then the resistance must be proportional to the length (l) of the resistor ($R \propto l/w$). The parasitic capacitance of the resistor, on the other hand, is proportional to the junction area of the resistor, and therefore to the value of the resistance as well ($C \propto lw \propto R$). Using parameterized subcircuits these relations are easily implemented, and one subcircuit can be used to represent many different resistors (see Figure 8.6). Here,

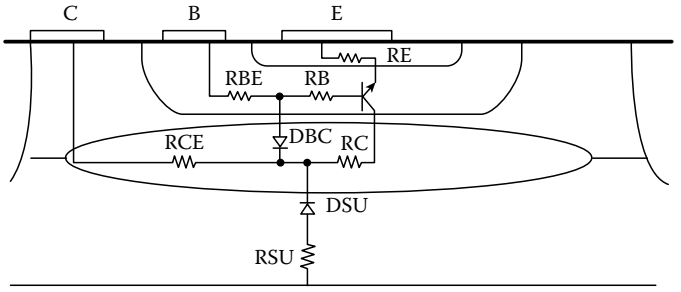


FIGURE 8.4 BJT cross-section with macro model elements.

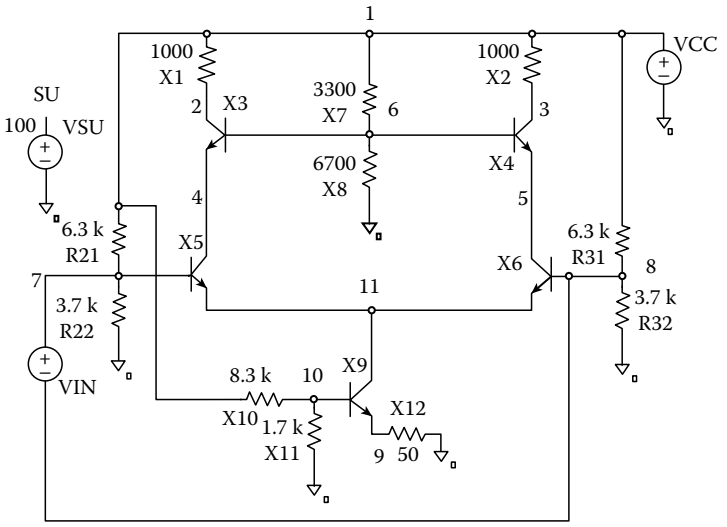


FIGURE 8.5 Cascode amplifier for AC analysis, created using PSPICE.

we represent the capacitance using two diodes one at each end of the resistor. This was done because the resistor junction capacitance is voltage dependent.

The input to the circuit is a voltage source (VIN), applied differentially to the amplifier. The output will be taken differentially across the collectors of the two upper transistors at nodes 2 and 3. The input impedance of the amplifier can be calculated as VIN/I(VIN) or because VIN = 1.0 just as 1/I(VIN). These quantities are shown plotted using PROBE in Figure 8.7. It can be seen that the gain of the amplifier falls off at high frequency as expected. The input impedance also drops because parasitic capacitances shunt the input.

It is also requested in Figure 8.6 that noise analysis be performed at every 20th frequency point. A portion of the noise printout is shown in Figure 8.8. It can be seen that the simulator calculates the noise contributions of each component in the circuit at the specified frequencies and displays them. The total noise at the specified summing node (differentially across nodes 2 and 3) in this case is also calculated as well as the equivalent noise referenced back to the input. This example took <1 min on an IBM PC.

Pitfalls. Many novice users will forget that AC analysis is a linear analysis. They will, for example, apply a 1 V signal to an amplifier with 5 V power supplies and a gain of 1000 and be surprised when SPICE tells them that the output voltage is 1000 V. Of course, the voltage generated in a simple amplifier must be less

```
Cascode amp with macro models.
*P type substrate is node 100
vcc 1 0 10
vsu 100 0 0
vin 7 8 ac 1
x1 1 2 100 icr PARAMS: val = 1k
x2 1 3 100 icr PARAMS: val = 1k
x3 2 6 4 100 tran
x4 3 6 5 100 tran
x5 4 7 11 100 tran
x6 5 8 11 100 tran
*cascode base bias divider*
x7 1 6 100 icr PARAMS: val = 3.3k
x8 6 0 100 icr PARAMS: val = 6.7k
*input bias dividers*
r21 1 7 6.3k
r22 7 0 3.7k
r31 1 8 6.3k
r32 8 0 3.7k
*Current Source*
x9 11 10 9 100 tran
x10 9 0 100 icr PARAMS: val = 100
x11 9 0 100 icr PARAMS: val = 100
x12 1 10 100 icr PARAMS: val = 8.3k
x13 10 0 100 icr PARAMS: val = 1.7k
.op
.noise v(2,3) vin 8
.ac dec 8 le6 le10
.probe

.subckt tran 1 2 3 4 PARAMS: val = 1
q1 5 2 3 mq{val}
rc 1 5 20/{val}
dbc 2 5 mdbc {val}
dsc 4 5 mdcs {val}
.model mq npn (is = le–15 bf = 100 br = 5
+ vaf =  30 var = 10 ikf = 5e–3 ikr =  le–4
+ re = 5 rb = 100 rc = 50 cje = .1p cjc = .05p
+ tf = 100p tr = 1n)
.model mdbc d (is = 1e–16 cjo =  .05p)
.model mdcs d (is = 1e–16 cjo =  .01p)
+ rs = 1000)
.ends tran

.subckt icr 1 2 3 PARAMS: val = 1000
rr 1 2 {val}
d1 3 1 md {val/1000}
d2 3 1 md {val/1000}
.model md d (is = le–16 cjo = 10f rs = ik)
.ends icr
```

FIGURE 8.6 SPICE input listing for cascode amplifier of Figure 8.5.

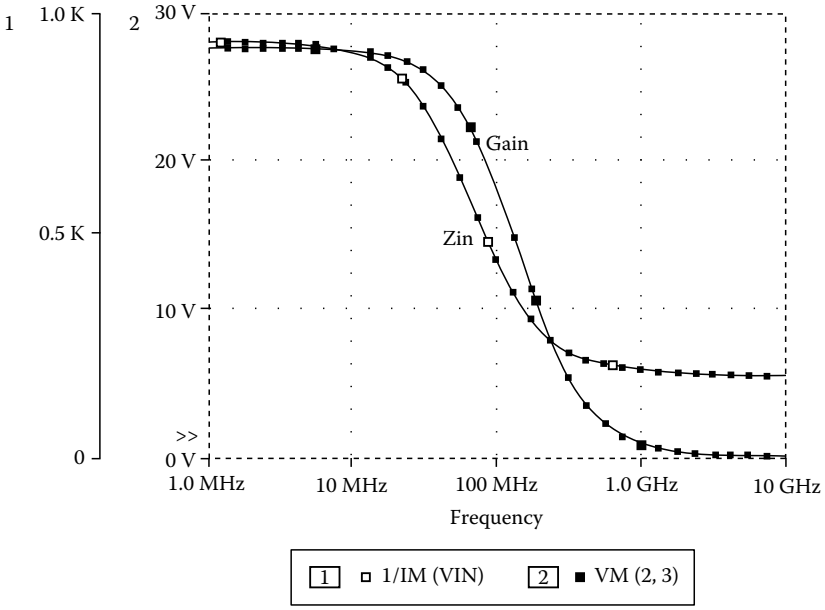


FIGURE 8.7 Gain and input impedance of cascode amplifier.

```
****     NOISE ANALYSIS                    TEMPERATURE = 27.000 DEG C
    FREQUENCY = 1.000E+09 HZ

****  DIODE SQUARED NOISE VOLTAGES (SQ V/HZ)

          x1.d1       x1.d2       x2.d1       x2.d2       x3.dsc      x4.dsc
RS        0.000E+00   0.000E+00   0.000E+00   0.000E+00   6.098E−21   6.098E−21
ID        0.000E+00   0.000E+00   0.000E+00   0.000E+00   2.746E−24   2.746E−24
FN        0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
TOTAL     0.000E+00   0.000E+00   0.000E+00   0.000E+00   6.1000E−21  6.100E−21

****  TRANSISTOR SQUARED NOISE VOLTAGES (SQ V/HZ)

          x3.q1       x4.q1       x5.q1       x6.q1       x7.q1
RB        1.519E−19   1.519E−19   1.522E−16   1.522E−16   0.000E+00
RC        3.185E−20   3.185E−20   7.277E−20   7.277E−20   0.000E+00
RE        8.056E−21   8.056E−21   7.418E−18   7.418E−18   0.000E+00
IB        2.190E−18   2.190E−18   3.390E−18   3.390E−18   0.000E+00
IC        8.543E−17   8.543E−17   1.611E−16   1.611E−16   0.000E+00
FN        0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
TOTAL     8.782E−17   8.782E−17   3.242E−16   3.242E−19   0.000E+00


****TOTAL OUTPUT NOISE VOLTAGE                    = 8.554E−16 SQ V/HZ
                                                  = 2.925E−08 V/RT HZ
    TRANSFER FUNCTION VALUE:
      V(2,3)/vin                    = 9.582E+00
    EQUIVALENT INPUT NOISE AT vin   = 3.052E−09 V/RT HZ
```

FIGURE 8.8 Noise analysis results for cascode amplifier.

than the power supply voltage, but to examine such clipping effects, transient analysis must be used. Likewise, selection of a proper *Q*-point is important. If the amplifier is biased in a saturated portion of its response and AC analysis is performed, the gain reported will be much smaller than the actual large signal gain.

8.1.4 Transient Analysis

Transient analysis is the most powerful analysis capability of a simulator because the transient response is so hard to calculate analytically. Transient analysis can be used for many types of analysis, such as switching speed, distortion, basic operation of certain circuits like switching power supplies. Transient analysis is also the most CPU intensive and can require 100 or 1000 times the CPU time as a DC or AC analysis.

8.1.4.1 Numerical Method

In a transient analysis time is discretized into intervals called time steps. Typically, the time steps are of unequal length, with the smallest steps being taken during portions of the analysis when the circuit voltages and currents are changing most rapidly. The capacitors and inductors in the circuit are then replaced by voltage and current sources based on the following procedure.

The current in a capacitor is given by $I_c = C\mathrm{d}V_c/\mathrm{d}t$. The time derivative can be approximated by a difference equation.

$$I_c^k + I_c^{k-1} = 2C\frac{V_c^k - V_c^{k-1}}{t^k - t^{k-1}}$$

In this equation, the superscript k represents the number of the time step. Here, k is the time step we are presently solving for and $(k-1)$ is the previous time step. This equation can be solved to give the capacitor current at the present time step.

$$I_c^k = V_c^k(2C/\Delta t) - V_c^{k-1}(2C/\Delta t) - I_c^{k-1}$$

Here $\Delta t = t^k - t^{k-1}$, or the length of the time step. As time steps are advanced, $V_c^{k-1} \rightarrow V_c^k; I_c^{k-1} \rightarrow I_c^k$. Note that the second two terms on the right hand side of the above equation are dependent only on the capacitor voltage and current from the previous time step, and are therefore fixed constants as far as the present step is concerned. The first term is effectively a conductance ($g = 2C/\Delta t$) multiplied by the capacitor voltage, and the second two terms could be represented by an independent current source. The entire transient model for the capacitor therefore consists of a conductance in parallel with two current sources (the numerical values of these are, of course, different at each time step). Once the capacitors and inductors have been replaced as indicated, the normal method of DC analysis is used. One complete DC analysis must be performed for each time point. This is the reason that transient analysis is so CPU intensive. The method outlined here is the trapezoidal time integration method and is used as the default in SPICE.

Example 8.3: Phase-Locked Loop Circuit

Figure 8.9 shows the phase-locked loop (PLL) circuit. The first analysis considers only the analog multiplier portion (also called a phase detector). The second analysis demonstrates the operation of the entire PLL. For the first analyses we wish to show that the analog multiplier does indeed multiply its input voltages. To do this, sinusoidal signals of 0.5 V amplitude are applied to the inputs, one at 2 MHz and the second at 3 MHz. The analysis is performed in the time domain and a Fourier analysis is used to analyze the output. Because the circuit functions as a multiplier, the output should be

$$V_{out} = AV_a \sin(2e6\pi t)V_b \sin(3e6\pi t) = AV_aV_b[\cos(1e6\pi t) - \cos(5e6\pi t)]$$

The 1 and 5 MHz components are of primary interest. Feedthrough of the original signal will also occur, which will give 2 and 3 MHz components in the output. Other forms of nonlinear distortion will

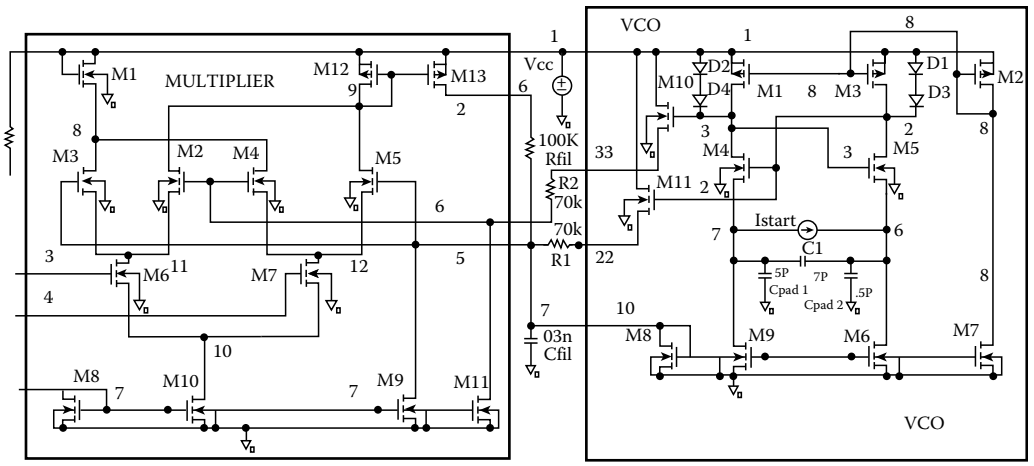


FIGURE 8.9 Phase-locked loop circuit for transient analysis, created with PSPICE.

```
Analog Multiplier Circuit.

Vcc 1 0 10.0
Rbias 1 2 800k
x1 1 9 3 4 5 6 2 10 mult
rdum 9 0 1meg

rin1 4 7 10k
vbias 7 0 2
vin 3 4 sin(0.5 2meg 0 0)

rin2 5 8 10k
vbias2 8 0 5.0
vin2 5 6 sin(0.5 3meg 0 0)

.tran 5n 3u 0 5n
.four 1meg 20 v(10)
.op
.probe
.options defl = 4u defas = 200p
defad = 200p

.subckt mult      1  2   3  4   5   6   7   8
* Pwr Iout In1 In2 Pin1 Pin2 Ibias
Vout
*load resistor*
m1 1 1 8 0 nmos w = 30u 1 = 4u
*Upper diff pairs*
m2 9 5 11 0 nmos w = 60u 1= 4u
m3 8 6 11 0 nmos w = 60u 1= 4u
m4 8 5 12 0 nmos w = 60u 1= 4u
m5 9 6 12 0 nmos w = 60u 1= 4u
*Lower diff pairs*
m6 11 3 10 0 nmos w = 60u 1= 4u
m7 12 4 10 0 nmos w = 60u 1= 4u
*Drive Current Mirror*
m8 7 7 0 0 nmos w = 60u 1= 4u
m9 5 7 0 0 nmos w = 60u 1= 4u
m10 10 7 0 0 nmos w = 60u 1= 4u
m11 6 7 0 0 nmos w = 60u 1= 4u
*Output current Mirror*
m12 9 9 1 1 pmos w = 60u 1= 4u
m13 2 9 1 1 pmos w = 60u 1= 4u

.model nmos nmos (level = 2 tox = 5e–8
+ nsub = 2e 15 tpg = 1 vto = .9 uo = 450
+ ucrit = 8e4 uexp = .15 cgso = 5.2e–10
+ cgdo = 5.2e–10)
.model pmos pmos (level = 2 tox = 5e–8
+ nsub = 2e 15 tpg = –1 vto = –.9 uo = 200
+ ucrit = 8e4 uexp = .15 cgso = 5.2e–10
+ cgdo = 5.2e–10)
.ends mult
```

FIGURE 8.10 SPICE input file for analog multiplier portion of PLL.

give components at higher frequencies. The SPICE input deck is shown in Figure 8.10. The output of the Fourier analysis is shown in Figure 8.11. It can be seen that the components at 1 and 5 MHz dominate and are a factor of 3 larger than the next largest component. The phase of the 1 and 5 MHz component is also offset by approximately 90° from the input (at 0°), as expected.

Simulation of the entire PLL will now be performed. The SPICE input deck is own in Figure 8.12. The phase detector and voltage-controlled oscillator are modeled in separate subcircuits. The phase

FOURIER COMPONENTS OF TRANSIENT RESPONSE V(10)

DC COMPONENT = 8.125452E+00

NO	FREQ HZ	FOURIER COMP.	NORMALIZED COMP.	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	1.000E+06	9.230E–02	1.000E+00	8.973E+01	0.000E+00
2	2.000E+06	6.462E–03	7.002E–02	–1.339E+01	–1.031E+02
3	3.000E+06	3.207E–02	3.474E–01	6.515E+01	–2.458E+01
4	4.000E+06	1.097E–02	1.189E–01	–1.604E+02	–2.501E+02
5	5.000E+06	8.229E–02	8.916E–01	–1.074E+02	–1.971E+02
6	6.000E+06	1.550E–02	1.680E–01	–1.683E+02	–2.580E+02
7	7.000E+06	3.695E–02	4.004E–01	7.984E+01	–9.886E+00
8	8.000E+06	7.943E–03	8.606E–02	1.302E+02	4.044E+00
9	9.000E+06	1.962E–02	2.126E–01	–1.149E+02	–2.046E+02

FIGURE 8.11 Results of transient and Fourier analyses of analog multiplier.

```
Phase locked loop circuit.
Vcc 1 0 10.0
Rbias 1 2 800k
rin 1 4 5 100k
vbias 5 0 2
Rfil 6 7 100k
Cfil 6 0 .03n
x1 1 6 3 4 18 19 2 10 mult
r1 8 18 70k
r2 9 19 70k
vsens 7 17 0
x2 1 8 9 17 vco
vin 3 4 sffm(0 1 600k 2 60k)
.tran .05u 60u
.probe
.options acct defl = 4u defas = 200p
+ defad = 200p

.subckt vco 1 22 33 10
*          Pwr out1 out2 Ict1

* P current mirror
m1 2 8 1 1 pmos w = 10u 1 = 4u
m2 8 8 1 1 pmos w = 10u 1 = 4u
m3 3 8 1 1 pmos w = 10u 1 = 4u

*Oscillator
m4 3 2 7 0 nmos w = 10u 1 = 4u
m5 2 3 6 0 nmos w = 10u 1 = 4u

*N current mirror
m6 6 10 0 0 nmos w = 20u 1 = 4u
m7 8 10 0 0 nmos w = 4u 1 = 4u
m8 10 10 0 0 nmos w = 20u 1 = 4u
m9 7 10 0 0 nmos w = 20u 1 = 4u

*source follower buffers
m10 13 33 0 nmos w = 80u 1 = 4u
m11 1 2 22 0 nmos w = 80u 1 = 4u

*Frequency setting capacitor*
c1 6 7 7pf
cpad6 6 0.5pf
cpad7 7 0.5pf

*Diode swing limiters*
d1 1 5 md
d2 1 4 md
d3 5 2 md
d4 4 3 md
*Pulse to start VCO*
istart 6 0 pulse (0 400u. 1us.01us
+.01us .3us 100)
```

FIGURE 8.12 SPICE input listing for phase-locked loop circuit.

detector, or multiplier, subcircuit is the same as that of Figure 8.10 and is omitted from Figure 8.12 for brevity. Examine the VCO subcircuit and note the PULSE-type current source ISTART connected across the capacitor. The source gives a current pulse 0.3E-6 s wide at the start of the simulation to start the VCO running. To start a transient simulation SPICE first computes a DC operating point (to find the initial voltages V_c^{k-1} on the capacitors). As this DC point is a valid, although not necessarily stable, solution, an oscillator will remain at this point indefinitely unless some perturbation is applied to start the oscillations. Remember, this is an ideal mathematical model and no noise sources or asymmetries exist that would start a real oscillator—it must be done manually. The capacitor C1 would have to be placed off-chip, and bond pad capacitance (CPAD1 and CPAD2) have been included at the capacitor nodes. Including the pad capacitances is very important if a small capacitor C1 is used for high-frequency operation.

In this example, the PLL is to be used as a FM detector circuit and the FM signal is applied to the input using a single frequency FM voltage source. The carrier frequency is 600 kHz and the modulation frequency is 60 kHz. Figure 8.13 shows the input voltage and the output voltage of the PLL at the VCO output and at the phase detector output. It can be seen that after a brief starting transient, the PLL locks onto the input signal and that the phase detector output has a strong 60 kHz component. This example took 251 s on a Sun SPARC-2 workstation (3046 time steps, with an average of 5 N iterations per time step).

Pitfalls. Occasionally, SPICE will fail and give the message ''Timestep too small in transient analysis,'' which means that the process of Newton iterations at certain time steps could not be made to converge. One of the most common causes of this is the specification of a capacitor with a value that is much too large, for example, specifying a 1 F capacitor instead of a 1 pF capacitor (an easy mistake to make by not adding the ''p'' in the value specification). Unfortunately, we usually have no way to tell which capacitor is at fault from the type of failure generated other than to manually search the input deck.

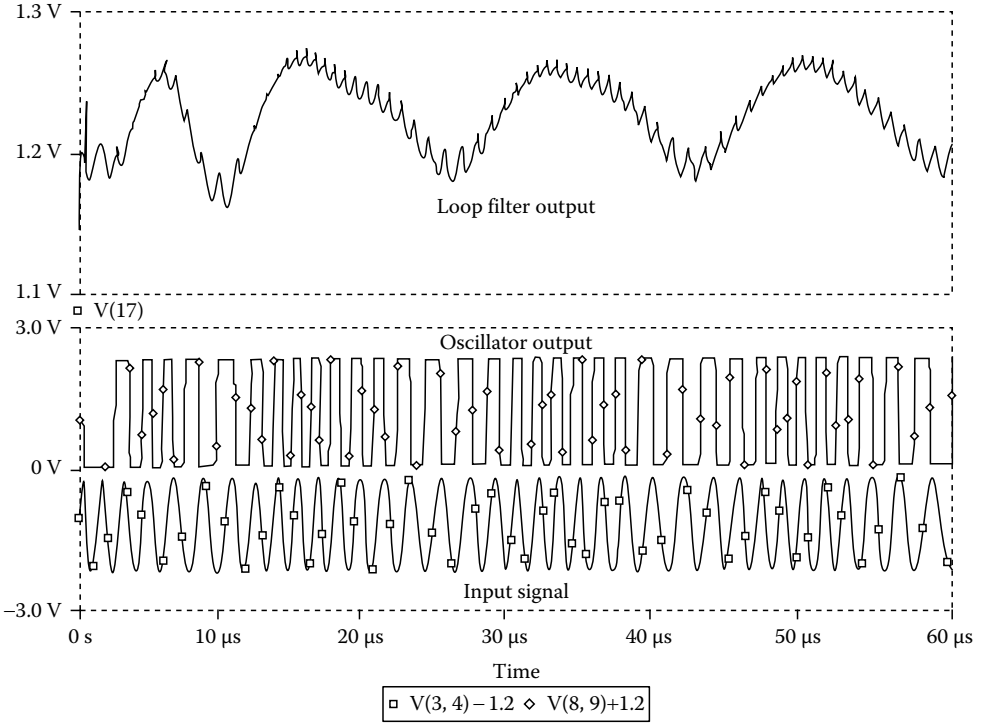


FIGURE 8.13 Transient analysis results of PLL circuit, created using PSPICE.

Other transient failures are caused by MOSFET models. Some models contain discontinuous capacitances (with respect to voltage) and others do not conserve charge. These models can vary from version to version so it is best check the user's guide.

8.1.5 Process and Device Simulation

Process and device simulation are the steps that precede analog circuit simulation in the overall simulation flow (see Figure 8.14). The simulators are also different in that they are not measurement driven as are analog circuit simulators. The input to a process simulator is the sequence of process steps performed (times, temperatures, gas concentrations) as well as the mask dimensions. The output from the process simulator is a detailed description of the solid-state device (doping profiles, oxide thickness, junction depths, etc.). This input to the device simulator is the detailed description generated by the process simulator (or via measurement). The output of the device simulator is the electrical characteristics of the device (I–V curves, capacitances, switching transient curves).

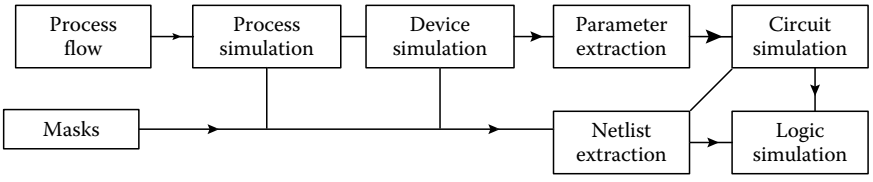


FIGURE 8.14 Data flow for complete process-device-circuit modeling.

Process and device simulation are becoming increasingly important and widely used during the IC design process. A number of reasons exist for this:

- As device dimensions shrink, second-order effects can become dominant. Modeling of these effects is difficult using analytical models.
- Computers have greatly improved, allowing time-consuming calculations to be performed in a reasonable amount of time.
- Simulation allows access to impossible to measure physical characteristics.
- Analytic models are not available for certain devices, for example, thyristors, heterojunction devices, and IGBTS.
- Analytic models have not been developed for certain physical phenomena, for example, single event upset, hot electron aging effects, latchup, and snap-back.
- Simulation runs can be used to replace split lot runs. As the cost of fabricate test devices increases, this advantage becomes more important.
- Simulation can be used to help device, process, and circuit designers understand how their devices and process work.

Clearly, process and device simulation is a topic which can be been the topic of entire texts. The following sections attempt to provide an introduction to this type of simulation, give several examples showing what the simulations can accomplish, and provide references to additional sources of information.

8.1.6 Process Simulation

IC processing involves a number of steps which are designed to deposit (deposition, ion implantation), remove (etching), redistribute (diffusion), or transform (oxidation) the material of which the IC is made. Most process simulation work has been in the areas of diffusion, oxidation, and ion implantation; however, programs are available that can simulate the exposure and development of photo-resist, the associated optical systems, as well as gas and liquid phase deposition and etch.

A number of programs are available (either from universities or commercial vendors) which can model silicon processing. The best known program is SUPREM-IV, which was developed at Stanford University (Stanford, CA). SUPREM-IV is capable of simulating oxidation, diffusion, and ion implantation, and has simple models for deposition and etch. In the following section a very brief discussion of the governing equations used in SUPREM will be given along with the results of an example simulation showing the power of the simulator.

8.1.6.1 Diffusion

The main equation governing the movement of electrically charged impurities (acceptors in this case) in the crystal is the diffusion equation:

$$\frac{\partial C}{\partial t} = \nabla \cdot \left(D\nabla C - \frac{DqC_a}{kT}\mathbf{E} \right)$$

Here

C is the concentration (#/cm^3) of impurities
C_a is the number of electrically active impurities (#/cm^3)
q is the electron charge
k is Boltzmann's constant
T is temperature in Kelvin
D is the diffusion constant
$\mathbf{E}$ is the built-in electric field.

The built-in electric field E in (V/cm) can be found from

$$\mathbf{E} = -\frac{kT}{q}\frac{1}{n}\nabla n$$

In this equation, n is the electron concentration (#/cm^3), which in turn can be calculated from the number of electrically active impurities (C_a). The diffusion constant (D) is dependent on many factors. In silicon the following expression is commonly used:

$$D = F_{IV}\left(D_x + D_+\frac{n_i}{n} + D_-\frac{n}{n_i} + D_=\left(\frac{n}{n_i}\right)^2\right)$$

The four D components represent the different possible charges states for the impurity: (x) neutral, ($+$) positive, ($-$) negative, ($=$) doubly negatively charged. n_i is the intrinsic carrier concentration, which depends only on temperature. Each D component is in turn given by an expression of the type

$$D = A\exp\left(-\frac{B}{kT}\right)$$

Here, A and B are experimentally determined constants, different for each type of impurity (x, +g, $-$, $=$). B is the activation energy for the process. This expression derives from the Maxwellian distribution of particle energies and will be seen many times in process simulation. It is easily seen that the diffusion process is strongly influenced by temperature. The term F_{IV} is an enhancement factor which is dependent on the concentration of interstitials and vacancies within the crystal lattice (an interstitial is an extra silicon atom which is not located on a regular lattice site; a vacancy is a missing silicon atom which results in an empty lattice site) $F_{IV} \propto C_I + C_V$. The concentration of vacancies, C_V, and interstitials, C_I, are in turn determined by their own diffusion equation:

$$\frac{\partial C_v}{\partial t} = +\nabla \cdot D_V \cdot \nabla C_V - R + G$$

In this equation, D_V is another diffusion constant of the form $A\exp(-B/kT)$. R and G represent the recombination and generation of vacancies and interstitials. Note that an interstitial and a vacancy may recombine and in the process destroy each other, or an interstitial and a vacancy pair may be simultaneously generated by knocking a silicon atom off its lattice site. Recombination can occur anywhere in the device via a bulk recombination process $R = A(C_V C_1)\exp(-b/kT)$. Generation occurs where there is damage to the crystal structure, in particular at interfaces where oxide is being grown or in regions where ion implantation has occurred, as the high-energy ions can knock silicon atoms off their lattice sites.

8.1.6.2 Oxidation

Oxidation is a process whereby silicon reacts with oxygen (or with water) to form new silicon dioxide. Conservation of the oxidant requires the following equation:

$$\frac{dy}{dt} = \frac{\mathbf{F}}{N}$$

Here

F is the flux of oxidant (#/cm^2/s)
N is the number of oxidant atoms required to make up a cubic centimeter of oxide
dy/dt is the velocity with which the Si–SiO_2, interface moves into the silicon.

In general, the greater the concentration of oxidant (C_0), the faster the growth of the oxide and the greater the flux of oxidant needed at the Si–SiO_2, interface. Thus, $F = k_s C_0$.

The flux of oxidant into the oxide from the gaseous environment is given by

$$F = h(HP_{ox} - C_0)$$

Here

H is a constant
P is the partial pressure of oxygen in the gas
C_0 is the concentration of oxidant in the oxide at the surface
h is of the form $A \exp(-B/kT)$.

Finally, the moment of the oxidant within the already existing oxide is governed by diffusion: $\mathbf{F} = D_0 \nabla C$ When all these equations are combined, it is found that (in the one-dimensional case) oxides grow linearly $dy/dt \propto t$ when the oxide is thin and the oxidant can move easily through the existing oxide. As the oxide grows thicker $dy/dt \propto \sqrt{t}$ because the movement of the oxidant through the existing oxide becomes the rate-limiting step.

Modeling two-dimensional oxidation is a challenging task. The newly created oxide must "flow" away from the interface where it is being generated. This flow of oxide is similar to the flow of a very thick or viscous liquid and can be modeled by a creeping flow equation:

$$\nabla^2 V \propto \nabla P$$
$$\nabla \cdot \mathrm{V} = 0$$

V is the velocity at which the oxide is moving and P is the hydrostatic pressure. The second equation results from the incompressibility of the oxide. The varying pressure P within the oxide leads to mechanical stress, and the oxidant diffusion constant D_0 and the oxide growth rate constant k_s are both dependent on this stress. The oxidant flow and the oxide flow are therefore coupled because the oxide flow depends on the rate at which oxide is generated at the interface and the rate at which the new oxide is generated depends on the availability of oxidant, which is controlled by the mechanical stress.

8.1.6.3 Ion Implantation

Ion implantation is normally modeled in one of two ways. The first involves tables of moments of the final distribution of the ions that are typically generated by experiment. These tables are dependent on the energy and the type of ion being implanted. The second method involves Monte Carlo simulation of the implantation process. In Monte Carlo simulation, the trajectories of individual ions are followed as they interact with (bounce off) the silicon atoms in the lattice. The trajectories of the ions, and the recoiling Si atoms (which can strike more Si atoms) are followed until all come to rest within the lattice. Typically several thousand trajectories are simulated (each will be different due to the random probabilities used in the Monte Carlo method) to build up the final distribution of implanted ions. Monte Carlo has advantages in that the damage to the lattice can be calculated during the implant process. This damage creates interstitials and vacancies that affect impurity diffusion, as was seen earlier. Monte Carlo can also model channeling, which is a process whereby the trajectories of the ions align with the crystal planes, resulting in greater penetration of the ion than in a simple amorphous target. Monte Carlo has the disadvantage of being CPU intensive.

Process simulation is always done in the transient mode using time steps as was done with transient circuit simulation. Because partial differential equations are involved, rather than ordinary differential equations, spatial discretization is needed as well. To numerically solve the problem, the differential equations are discretized on a grid. Either rectangular or triangular grids in one, two, or

three dimensions are commonly used. This discretization process results in the conversion of the partial differential equations into a set of nonlinear algebraic equations. The nonlinear equations are then solved using a Newton method in a way very similar to the method used for the circuit equations in SPICE.

Example 8.4: NMOS Transistor

In this example, the process steps used to fabricate a typical NMOS transistor will be simulated using SUPREM-4. These steps are

1. Grow initial oxide (30 min at 1000 K).
2. Deposit nitride layer (a nitride layer will prevent oxidation of the underlying silicon).
3. Etch holes in nitride layer.
4. Implant P + channel stop (boron dose = 5e12, energy = 50 keV).
5. Grow the field oxide (180 min at 1000 K wet O_2).
6. Remove all nitride.
7. Perform P channel implant (boron dose = 1e11, energy = 40 keV).
8. Deposit and etch polysilicon for gate.
9. Oxidize the polysilicon (30 min at 1000 K, dry O_2).
10. Implant the light doped drain (arsenic dose = 5e13 energy = 50 keV).
11. Deposit sidewall space oxide.
12. Implant source and drain (arsenic, dose = 1e15, energy = 200 keV).
13. Deposit oxide layer and etch contact holes.
14. Deposit and etch metal.

The top 4 μm of the completed structure, as generated by SUPREM-4, is shown in Figure 8.15. The actual simulation structure used is 200 μm deep to allow correct modeling of the diffusion of the vacancies and interstitials. The gate is at the center of the device. Notice how the edges of the gate have lifted up due to the diffusion of oxidant under the edges of the polysilicon (the polysilicon, as deposited in step 8, is flat). The dashed contours show the concentration of dopants in both the oxide and silicon layers. The short dashes indicate N-type material, while the longer dashes indicate P-type material. This entire simulation requires about 30 min on a Sun SPARC-2 workstation.

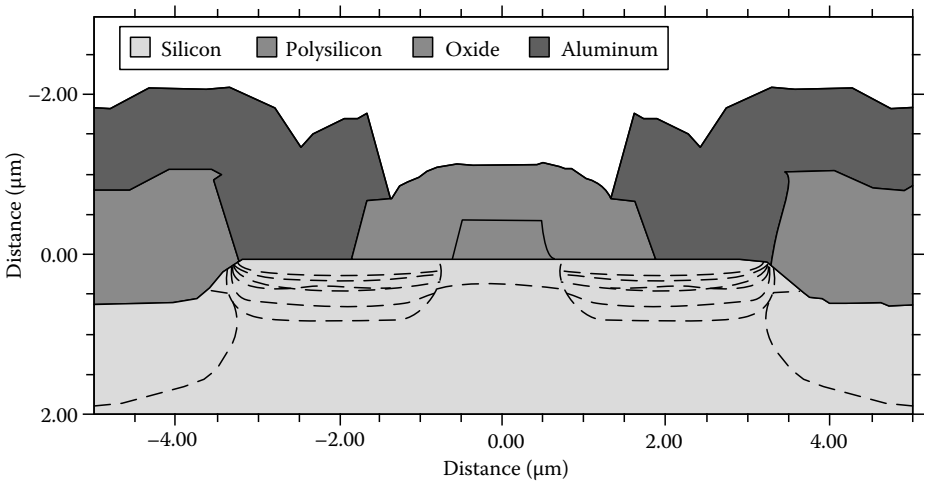


FIGURE 8.15 Complete NMOS transistor cross section generated by process simulation, created with TMA SUPREM-4.

8.1.7 Device Simulation

Device simulation uses a different approach from that of conventional lumped circuit models to determine the electrical device characteristics. Whereas with analytic or empirical models all characteristics are determined by fitting a set of adjustable parameters to measured data, device simulators determine the electrical behavior by numerically solving the underlying set of differential equations. The first of these equations is the Poisson equation, which describes the electrostatic potential within the device.

$$\nabla \cdot \varepsilon \cdot \nabla \Psi = q\left(N_{\mathrm{a}}^{-} - N_{\mathrm{d}}^{+} - p + n - Q_{\mathrm{f}}\right)$$

N_{d} and N_{a} are the concentration of donors and acceptors, i.e., the N- and P-type dopants. Q_{f} is the concentration of fixed charge due, for example, to traps or interface charge. The electron and hole concentrations are given by n and p, respectively, and Ψ is the electrostatic potential.

A set of continuity equations describes the conservation of electrons and holes:

$$\frac{\partial n}{\partial t} = \left(\frac{1}{q}\nabla \cdot J_n - R + G\right)$$

$$\frac{\partial p}{\partial t} = \left(-\frac{1}{q}\nabla \cdot J_p - R + G\right)$$

In these equations, R and G describe the recombination and generation rates for the electrons and holes. The recombination process is influenced by factors such as the number of electrons and holes present as well as the doping and temperature. The generation rate is also dependent upon the carrier concentrations, but is most strongly influenced by the electric field, with increasing electric fields giving larger generation rates. Because this generation process is included, device simulators are capable of modeling the breakdown of devices at high voltage. J_n and J_p are the electron and hole current densities (in amperes per square centimeter). These current densities are given by another set of equations:

$$\mathbf{J}_n = q\mu\left(-n\nabla\Psi + \frac{kT_n}{q}\nabla n\right)$$

$$\mathbf{J}_p = q\mu\left(-p\nabla\Psi - \frac{kT_p}{q}\nabla p\right)$$

In this equation, k is Boltzmann's constant, μ is the carrier mobility, which is actually a complex function of the doping, n, p, electric field, temperature, and other factors. In silicon the electron mobility will range between 50 and 1000 and the hole mobility will normally be a factor of 2 smaller. In other semiconductors such as gallium arsenide the electron mobility can be as high as 5000. T_n and T_p are the electron and hole mean temperatures, which describe the average carrier energy. In many models these default to the device temperature (300 K). In the first term the current is proportional to the electric field ($\nabla\Psi$), and this term represents the drift of carriers with the electric field. In the second term the current is proportional to the gradient of the carrier concentration (∇n), so this term represents the diffusion of carriers from regions of high concentration to those of low concentration. The model is therefore called the drift-diffusion model.

In devices in which self-heating effects are important, a lattice heat equation can also be solved to give the internal device temperature:

$$\sigma(T)\frac{\partial T}{\partial t} = H + \nabla \cdot \lambda(T) \cdot \nabla T$$

$$H = -(J_n + J_p) \cdot \nabla\Psi + H_{\mathrm{R}}$$

where H is the heat generation term, which includes resistive (Joule) heating as well as recombination heating, H_R. The terms $\sigma(T)$, $\lambda(T)$ represent the specific heat and the thermal conductivity of the material (both temperature dependent). Inclusion of the heat equation is essential in many power device problems.

As with process simulation partial differential equations are involved, therefore, a spatial discretization is required. As with circuit simulation problems, various types of analysis are available:

- Steady-state (DC), used to calculate characteristic curves of MOSFETs, BJTs diodes, etc.
- AC analysis, used to calculate capacitances, *Y*-parameters, small signal gains, and *S*-parameters
- Transient analysis, used for calculation of switching and large signal behavior, and special types of analysis such as radiation effects

Example 8.5: NMOS IV Curves

The structure generated in the previous SUPREM-IV simulation is now passed into the device simulator and bias voltages are applied to the gate and drain. Models were included which account for Auger and Shockley Reed Hall recombination, doping and electric field-dependent mobility, and impact ionization. The set of drain characteristics obtained is shown in Figure 8.16. Observe how the curves bend upward at high V_{ds} as the device breaks down. The $V_{gs} = 1$ curve has a negative slope at $I_d = 1.5\text{e-}4$ A as the device enters snap-back. It is possible to model this type of behavior because impact ionization is included in the model.

Figure 8.17 shows the internal behavior of the device with $V_{gs} = 3$ V and $I_d = 3\text{e-}4$ A. The filled contours indicate impact ionization, with the highest rate being near the edge of the drain right beneath the gate. This is to be expected because this is the region in which the electric field is largest due to the drain depletion region. The dark lines indicate current flow from the source to the drain. Some current also flows from the drain to the substrate. This substrate current consists of holes generated by the impact ionization. The triangular grid used in the simulation can be seen in the source, drain, and gate electrodes. A similar grid was used in the oxide and silicon regions.

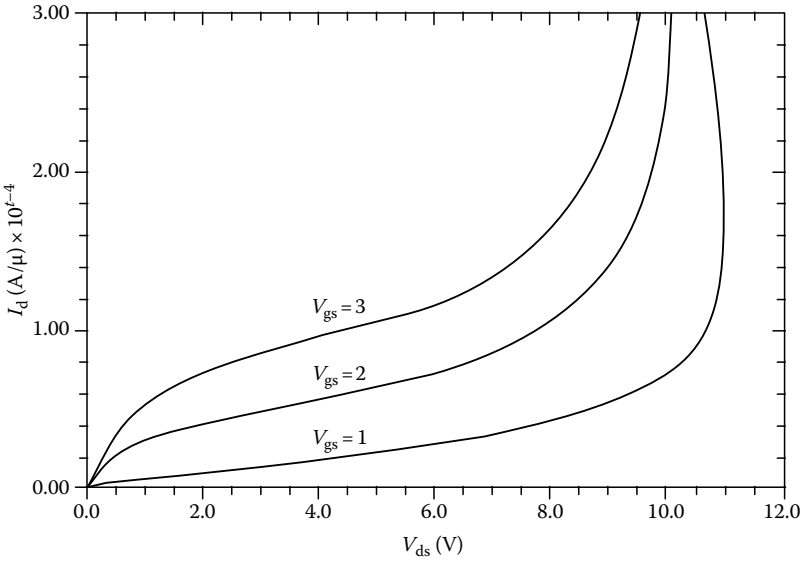


FIGURE 8.16 I_d vs. V_{ds} curves generated by device simulation, created with TMA MEDICI.

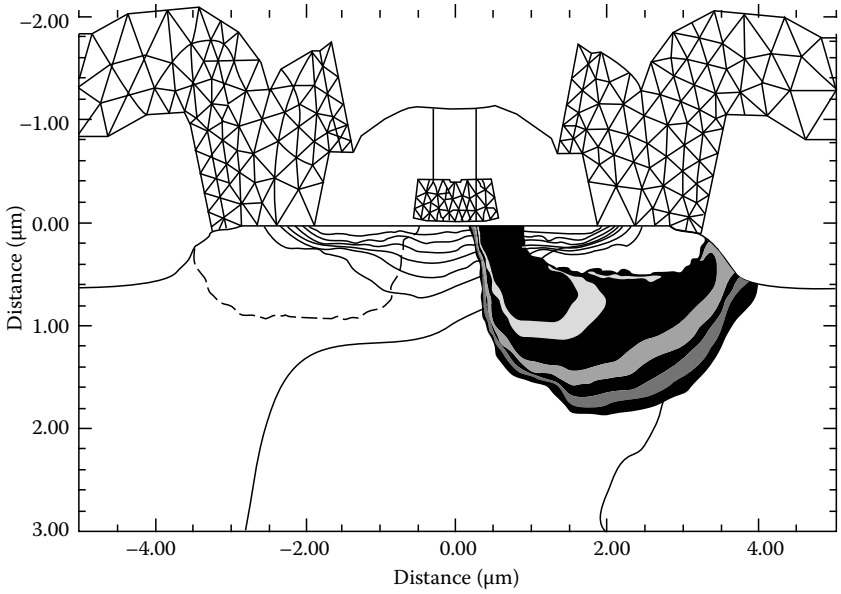


FIGURE 8.17 Internal behavior of MOSFET under bias, created with TMA MEDICI.

Appendix A

A.1 Circuit Analysis Software

SPICE2, SPICE3: University of California, Berkeley, CA
PSPICE: MicroSim Corporation, Irvine, CA (used in this chapter)
HSPICE: Meta Software, Campbell, CA
IsSPICE: Intusoft, San Pedro, CA
SPECTRE: Cadence Design Systems, San Jose, CA
SABRE: Analogy, Beaverton, OR

A.2 Process and Device Simulators

SUPREM-4, PISCES: Stanford University, Palo Alto, CA
MINIMOS: Technical University, Vienna, Austria
SUPREM-4, MEDICI, DAVINCI: Technology Modeling Associates, Palo Alto, CA (used in this chapter)
SEMICAD: Dawn Technologies, Sunnyvale, CA

References

1. P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*, New York: McGraw-Hill, 1988.
2. P. W. Tuinenga, *SPICE, A Guide to Circuit Simulation and Analysis Using PSPICE*, Englewood Cliffs, NJ: Prentice Hall, 1988.
3. J. A. Connelly and P. Choi, *Macromodeling with SPICE*, Englewood Cliffs, NJ: Prentice Hall, 1992.
4. S. Selberherr, *Analysis and Simulation of Semiconductor Devices*, Berlin: Springer-Verlag, 1984.
5. R. Dutton and Z. Yu, *Technology CAD, Computer Simulation of IC Process and Devices*, Boston: Kluwer Academic, 1993.

8.2 Parameter Extraction for Analog Circuit Simulation

Peter Bendix

8.2.1 Introduction

8.2.1.1 Definition of Device Modeling

We use various terms such as device characterization, parameter extraction, optimization, and model fitting to address an important engineering task. In all of these, we start with a mathematical model that describes the transistor behavior. The model has a number of parameters that are varied or adjusted to match the I–V (current–voltage) characteristics of a particular transistor or set of transistors. The act of determining the appropriate set of model parameters is what we call device modeling. We then use the model with the particular set of parameters that represent our transistors in a circuit simulator such as SPICE* to simulate how circuits with our kinds of transistors will behave. Usually, the models are supplied by the circuit simulator we chose. Occasionally, we may want to modify these models or construct our own models. In this case we need access to the circuit simulator model subroutines as well as the program that performs the device characterization.

Most people believe the preceding description covers device modeling. However, we feel this is a very narrow definition. One can obtain much more information by characterizing a semiconductor process than just providing models for circuit simulation. If the characterization is done correctly, it can provide detailed information about the fabrication process. This type of information is useful for process architects as an aid in developing future processes. In addition, the wealth of information from the parameters obtained can be used by process and product engineers to monitor wafers routinely. This provides much more information than what is usually obtained by doing a few crude electrical tests for process monitoring. Finally, circuit designers can use device characterization as a means of optimizing a circuit for maximum performance. Therefore, instead of just being a tool used by specialized device physicists, device characterization is a tool for process architects, process and product engineers, and circuit designers. It is a tool that can and should be used by most people concerned with semiconductor manufacture and design.

8.2.1.2 Steps Involved in Device Characterization

Device characterization begins with a test chip. Without the proper test chip structures, proper device modeling cannot be done from measured data. A good test chip for MOS technology would include transistors of varying geometries, gate oxide capacitance structures, junction diode capacitance structures, and overlap capacitance structures. This would be a minimal test chip. Additional structures might include ring oscillators and other circuits for checking the AC performance of the models obtained. It is very important that the transistors be well designed and their geometries be chosen appropriate for the technology as well as the desired device model. For bipolar technology modeling, the layout of structures used to get *S*-parameter measurements is very critical. Although a complete test chip description is beyond the scope of this book, be aware that even perfect device models cannot correct for a poor test chip.

Next, we need data that represent the behavior of a transistor or set of transistors of different sizes. These data can come from direct measurement or they can be produced by a device simulator such as PISCES.† Typically, one does voltage sweeps on various nodes of the transistor and measures the output current at some or all of these nodes. For example, in an MOS transistor, one might sweep the gate voltage and measure the resulting drain current, holding the drain and bulk voltages fixed. The equipment used to do DC device characterization measurements is usually a DC parametric tester.

---

* SPICE is a circuit simulation program from the Department of Electrical Engineering and Computer Science at the University of California, Berkeley.

† PISCES is a process simulation program from the Department of Electrical Engineering at Stanford University, Stanford, CA.

This equipment usually has a set of SMUs (source/measure units) that can force voltage and measure current, or force current and measure voltage. The measuring equipment can be manually run or controlled by a computer. In either case the measured data must be put onto a hard disk.

For oxide capacitors, junction diodes, and overlap capacitors, a capacitance or inductance, capacitance, resistance (LCR) meter is used to do the measurements. For junction capacitance models, a junction diode (source to bulk, for example) has a reverse bias sweep done and the capacitance is measured at each bias point. For oxide and overlap capacitances, only a single capacitance measurement at one bias voltage is required.

In a more automated environment, switching matrices, semiautomated probers, and temperature-controlled chucks are added. The same equipment is used for bipolar devices. In addition, network analyzers are often used for *S*-parameter characterization to obtain the related AC transit time parameters of bipolar transistors. A typical collection of measuring equipment might look like Figure 8.18.

It is also possible to use a device simulator like PISCES, or a combination of a process simulator like SUPREM-IV* coupled to a device simulator, to provide the simulated results. The process recipe steps are fed into SUPREM-IV, which produces a device structure and associated profiles of all dopants; this information is then fed into the device simulator along with a description of voltage sweeps to be applied to various device electrodes. The output of the device simulator is a set of IV characteristics which closely represent what would have been measured on a real device. The benefits of using simulation over measurement are that no expensive measurement equipment or fabricated wafers are necessary. This can be very helpful when trying to predict the device characteristics of a new fabrication process before any wafers have been produced.

Once the measured (or simulated) data are available, parameter extraction software is used to find the best set of model parameter values to fit the data. This can be done by extraction, optimization, or a combination of the two. Extraction is simpler and quicker, but not as accurate as optimization. In extraction simplifying assumptions are made about the model equations and parameter values are extracted using slope and intercept techniques, for example. In optimization general least-squares curve fitting routines are employed to vary a set or subset of parameters to fit the measured data. Extraction and optimization are covered in more detail in the following sections.

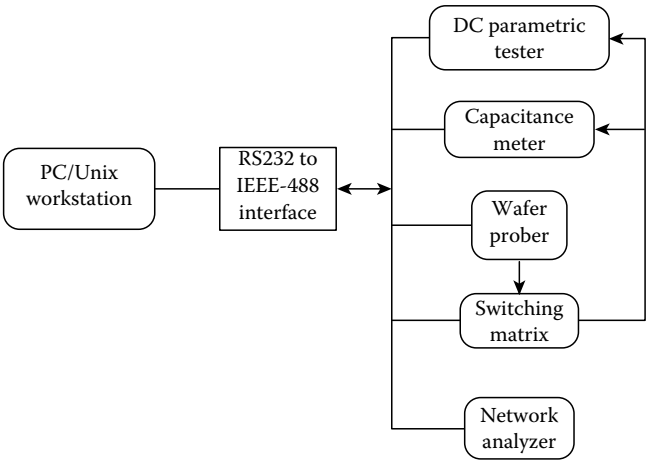


FIGURE 8.18 Typical hardware measuring equipment for device characterization.

* SUPREM-IV is a process simulation program from the Department of Electrical Engineering at Stanford University, Stanford, CA.

8.2.1.3 Least-Squares Curve Fitting (Analytical)

We begin this section by showing how to do least-squares curve fitting by analytical solutions, using a simple example to illustrate the method. We then discuss least-squares curve fitting using numerical solutions in the next section. We can only find analytical solutions to simple problems. The more complex ones must rely on numerical techniques.

Assume a collection of measured data, $m_1, \ldots, m_n$. For simplicity, let these measured data values be functions of a single variable, v, which was varied from v_1 through v_n, measuring each m_i data point at each variable value v_i, i running from 1 to n. For example, the m_i data points might be drain current of an MOS transistor, and the v_i might be the corresponding values of gate voltage. Assume that we have a model for calculating simulated values of the measured data points, and let these simulated values be denoted by $s_1, \ldots, s_n$. We define the least-squares, root mean square (RMS) error as

$$Error_{\text{rms}} = \left[\frac{\sum_{i=1}^{n} \{\text{weight}_i (s_i - m_i)\}^2}{\sum_{i=1}^{n} \{\text{weight}_i m_i\}^2} \right]^{1/2} \tag{8.1}$$

where a weighting term is included for each data point. The goal is to have the simulated data match the measured data as closely as possible, which means we want to minimize the RMS error. Actually, what we have called RMS error is really the relative RMS error, but the two terms are used synonymously. There is another way of expressing the error, called the absolute RMS error, defined as follows:

$$Error_{\text{rms}} = \left[\frac{\sum_{i=1}^{n} \{\text{weight}_i (s_i - m_i)\}^2}{\sum_{i=1}^{n} \{\text{weight}_i \; m_{\min}\}^2} \right]^{1/2} \tag{8.2}$$

where we have used the term $m_{\min}$ in the denominator to represent some minimum value of the measured data. The absolute RMS error is usually used when the measured values approach zero to avoid problems with small or zero denominators in Equation 8.1. For everything that follows, we consider only the relative RMS error. The best result is obtained by combining the relative RMS formula with the absolute RMS formula by taking the maximum of the denominator from Equation 8.1 or 8.2.

We have a simple expression for calculating the simulated data points, s_i, in terms of the input variable, v, and a number of model parameters, $p_1, \ldots, p_m$. That is,

$$s_i = f(v_i, p_1, \ldots, p_m) \tag{8.3}$$

where f is some function. Minimizing the RMS error function is equivalent to minimizing its square. Also, we can ignore the term in the denominator of Equation 8.1 as concerns minimizing, because it is a normalization term. In this spirit, we can define a new error term,

$$\text{Error} = (Error_{\text{rms}})^2 \left[\sum_{i=1}^{n} \{\text{weight}_i m_i\}^2 \right] \tag{8.4}$$

and claim that minimizing Error is equivalent to minimizing $Error_{\text{rms}}$. To minimize Error, we set all partial derivatives of it with respect to each model parameter equal to zero; that is, write

$$\frac{\partial(Error)}{\partial p_j} = 0 \quad \text{for } j = 1, \ldots, m \tag{8.5}$$

Then, solve the preceding equations for the value of p_j.

Least-squares curve fitting. Analytic example**:** Write a simple expression for the drain current of an MOS transistor in the linear region in terms of a single variable, the gate voltage, V_{gs}, and also in terms of two model parameters, β and V_{th};

$$I_{ds} = \beta(V_{gs} - V_{th})V_{ds} \tag{8.6}$$

We denote the measured drain current by I_{meas}. In terms of our previous notation, we have the following substitutions:

$$\begin{aligned} s &\rightarrow I_{ds} \\ v &\rightarrow V_{gs} \\ p_1 &\rightarrow \beta \\ p_2 &\rightarrow V_{th} \end{aligned}$$

We have two conditions to satisfy for minimizing the error:

$$\frac{\partial(Error)}{\partial \beta} = 0 \tag{8.7}$$

$$\frac{\partial(Error)}{\partial V_{th}} = 0 \tag{8.8}$$

Equations 8.7 and 8.8 imply, respectively,

$$\sum_{i=1}^{n} [\beta(V_{gs_i} - V_{th})V_{ds} - I_{meas_i}](V_{gs_i} - V_{th}) = 0 \tag{8.9}$$

$$\sum_{i=1}^{n} [\beta(V_{gs_i} - V_{th})V_{ds} - I_{meas_i}] = 0 \tag{8.10}$$

Solving Equations 8.9 and 8.10 is straightforward but tedious. The result is

$$V_{th} = \frac{[\text{Term}_1\text{Term}_2 - \text{Term}_3\text{Term}_4]}{[n\text{Term}_1 - \text{Term}_2\text{Term}_4]} \tag{8.11}$$

$$\beta = \left(\frac{1}{V_{ds}}\right)\frac{[n\text{Term}_1 - \text{Term}_2\text{Term}_4]}{\left[n\text{Term}_3 - \text{Term}_2^2\right]} \tag{8.12}$$

where n is the number of data points that i is summed over, and

$$\text{Term}_1 = \sum_{i=1}^{n} V_{gs_i} I_{meas_i} \tag{8.13}$$

$$\text{Term}_2 = \sum_{i=1}^{n} V_{\text{gs}_i} \tag{8.14}$$

$$\text{Term}_3 = \sum_{i=1}^{n} V_{\text{gs}_i}^2 \tag{8.15}$$

and

$$\text{Term}_4 = \sum_{i=1}^{n} I_{\text{meas}_i} \tag{8.16}$$

Thus, analytical solutions can become quite messy, even for the simplest model expressions. In fact, it is usually not possible to solve the system of equations analytically for more complicated models. Then, we must rely on numerical solutions.

One further word of caution is required. One may try comparing analytical solutions to those obtained by numerical techniques. They almost always will not match. The reason is that the weighting functions used to calculate $Error_{\text{rms}}$ are different for the two cases. In order to compare the two techniques, one must know the exact algorithm that the numerical least-squares curve fitting routine is using and be able to match the weighting functions. Often weighting is implicit in the numerical least-squares curve fitting algorithm that is not explicitly obvious.

8.2.1.4 Least-Squares Curve Fitting (Numerical)

For almost all practical applications we are forced to do least-squares curve fitting numerically, because the analytic solutions as previously discussed are not obtainable in closed form. What we are calling least-squares curve fitting is more generally known as nonlinear optimization. Many fine references on this topic are available. We highlight only the main features here, and paraphrase parts of [2] in what follows.

Basically, we want to solve the optimization problem whose goal is to minimize the function $F(\mathbf{x})$ where F is some arbitrary function of the vector $\mathbf{x}$. If extra conditions exist on the components of $\mathbf{x}$, we are then solving a constrained, nonlinear optimization problem. When using least-squares fitting for model parameter optimization, we usually constrain the range of the fitting parameters. This type of constraint is a very simple one and is a special case of what is called nonlinear constrained optimization. This or the unconstrained approach are the types normally used for model parameter optimization.

For simplicity, let F be a function of a single variable, x, instead of the vector, $\mathbf{x}$. We state without proof that a necessary condition for a minimum of $F(x)$ to exist is that

$$F' = \frac{\partial F}{\partial x} = 0 \tag{8.17}$$

This means we solve the minimization problem if we can solve for the zeroes of F'. This is exactly what we did when we went through the least-squares analytical method.

One numerical approach to solving for the zeroes of a function, f, is Newton's method. This is an iterative method in which we write

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{8.18}$$

Steepest descents method. Next consider the more general case in which F is now a function of many variables, $\mathbf{x}$; i.e., F is multivariate. We want to consider numerical techniques for finding the minimum of

F by iteration. We also would like to take iteration steps that decrease the value of F at each step. This imposes what is called a descent condition:

$$F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k) \tag{8.19}$$

where k is the iteration number. Solving the minimization problem amounts to choosing a search direction for the vector $\mathbf{x}$ such that at each iteration F decreases until some convergence criterion is satisfied with $\mathbf{x}_k$ as the solution. Let us write F as a Taylor series expanded around the point $\mathbf{x}$, with $\mathbf{p}$ a vector of unit length and h a scalar

$$F(\mathbf{x} + h\mathbf{p}) = F(\mathbf{x}) + hg^{\mathrm{T}}(\mathbf{x})\mathbf{p} + \frac{1}{2}h^2\mathbf{p}^{\mathrm{T}}G(\mathbf{x})\mathbf{p} + \cdots \tag{8.20}$$

where in Equation 8.20, $g(\mathbf{x})$ is the gradient (first derivative in the $\mathbf{p}$ direction) of $F(\mathbf{x})$ and $G(\mathbf{x})$ is the curvature (second derivative in the $\mathbf{p}$ direction) of $F(\mathbf{x})$. The uppercase "T" means matrix transpose.

We want to choose a direction and step length for the next iteration in which

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h_k\mathbf{p}_k \tag{8.21}$$

where in Equation 8.21, $\mathbf{p}$ is a unit vector in the search direction and h is the step size. In the method of steepest descents we set

$$\mathbf{p}_k = -g(\mathbf{x}_k) \tag{8.22}$$

That is, the method of steepest descents chooses the search direction to be opposite the first derivative in the direction $\mathbf{x}_k$.

Newton's method. The method of steepest descents is a first derivative method. Newton's method is a second derivative method (this is not to be confused with the simple Newton's method discussed previously). Referring to Equation 8.20, the Newton direction of search is defined to be the vector $\mathbf{p}$, which satisfies

$$G(\mathbf{x}_k)\mathbf{p}_k = -g(\mathbf{x}_k) \tag{8.23}$$

Finding $\mathbf{p}_k$ from the preceding equation involves calculating second derivatives to compute G.

Gauss–Newton method. We want to modify Newton's method so that second derivative calculations are avoided. To do this, consider the so-called least-squares problem in which F can be written as

$$F(\mathbf{x}) = \sum [f_i(\mathbf{x})]^2 \tag{8.24}$$

In this special case, we can write

$$g(\mathbf{x}) = J^{\mathbf{T}}(\mathbf{x})f(\mathbf{x}) \tag{8.25}$$

$$G(\mathbf{x}) = J^{\mathbf{T}}(\mathbf{x})J(\mathbf{x}) + Q(\mathbf{x}) \tag{8.26}$$

where J is the Jacobian of f and

$$Q(\mathbf{x}) = \sum f_i(\mathbf{x})G_i(\mathbf{x}) \tag{8.27}$$

where G_i is the so-called Hessian of f. Substituting Equations 8.25 and 8.26 into Equation 8.23 gives

$$\left[J^{\mathrm{T}}(\mathbf{x}_k)J(\mathbf{x}_k) + Q_k\right]\mathbf{p}_k = -J^{\mathbf{T}}(\mathbf{x}_k)f(\mathbf{x}_k) \tag{8.28}$$

If we throw away the Q term, Equation 8.28 becomes

$$\left[J^{\mathrm{T}}(\mathbf{x}_k)J(\mathbf{x}_k)\right]\mathbf{p}_k = -J^{\mathrm{T}}(\mathbf{x}_k)f(\mathbf{x}_k) \tag{8.29}$$

a condition for finding the search direction, $\mathbf{p}_k$, which does not involve second derivatives. Solving Equation 8.29 is tantamount to solving the linear least-squares problem of minimizing

$$F(\mathbf{x}) = \sum [\, j_i(\mathbf{x}_k)\mathbf{p}_k + f_i(\mathbf{x}_k)]^2 \tag{8.30}$$

When the search direction, $\mathbf{p}_k$, is found from Equation 8.29, we call this the Gauss–Newton method.

Levenberg–Marquardt algorithm. Both the method of steepest descents and the Gauss–Newton method have advantages and disadvantages. Briefly stated, the method of steepest descents will move toward the correct solution very quickly, but due to the large steps it takes, it can jump past the solution and not converge. The Gauss–Newton method, however, moves more slowly toward the correct solution, but because it tends to take smaller steps, it will converge better as its approaches the solution. The Levenberg–Marquardt algorithm was designed to combine the benefits of both the steepest descents and Gauss–Newton methods.

The Levenberg–Marquardt search direction is found by solving

$$\left[J^{\mathrm{T}}(\mathbf{x}_k)J(\mathbf{x}_k) + \lambda_k I\right]\mathbf{p}_k = -J^{\mathrm{T}}(\mathbf{x}_k)f(\mathbf{x}_k) \tag{8.31}$$

where λ_k is a nonnegative scalar and I is the identity matrix. Note that Equation 8.31 becomes Equation 8.29 when λ_k is zero, so that the Levenberg–Marquardt search direction becomes the Gauss–Newton in this case; as $\lambda_k \to \infty$, $\mathbf{p}_k$ becomes parallel to $-J^{\mathrm{T}}(\mathbf{x}_k)\, f(\mathbf{x}_k)$, which is just $-g(\mathbf{x}_k)$, the direction of steepest descents. Generally speaking, in the Levenberg–Marquardt algorithm the value of λ_k is varied continuously, starting with a value near the steepest descents direction (large λ_k) and moving toward smaller values of λ_k as the solution is approached.

8.2.1.5 Extraction (as Opposed to Optimization)

The terms ''extraction'' and ''optimization'' are, unfortunately, used interchangeably in the semiconductor industry; however, strictly speaking, they are not the same. By optimization, we mean using generalized least-squares curve fitting methods such as the Levenberg–Marquardt algorithm to find a set of model parameters. By extraction, we mean any technique that does not use general least-squares fitting methods. This is a somewhat loose interpretation of the term extraction, but perhaps the following discussion will justify it.

Suppose we have measured the drain current of an MOS transistor in the linear region at zero backbias (the bulk node bias) and we want to find the threshold voltage by extraction (as opposed to optimization). We could plot the drain current vs. gate voltage, draw a tangent line at the point where the slope of I_{ds} is a maximum, and find the V_{gs} axis intercept of this tangent line. This would give us a crude value of the threshold voltage, V_{th} (neglecting terms such as $V_{\mathrm{ds}}/2$). This would be a graphic extraction technique; it would not involve using least-squares optimization.

We could also apply the preceding graphic technique in an equivalent algorithmic way, without drawing any graphs. Let us write a linear region equation for the drain current

$$I_{\mathrm{ds}} = \beta(V_{\mathrm{gs}} - V_{\mathrm{th}})V_{\mathrm{ds}}[1 + Ke^{-\alpha(V_{\mathrm{gs}} - V_{\mathrm{th}})}] \tag{8.32}$$

In Equation 8.32, the exponential term is the subthreshold contribution to the drain current; it becomes negligible for $V_{\mathrm{gs}} \gg V_{\mathrm{th}}$. To find the value of V_{gs} where the maximum slope occurs, we set the derivative of I_{ds} with respect to V_{gs} equal to zero and solve for V_{gs}. The solution is

$$V_{gs}(\text{maximum slope}) = V_{th} + \left[\frac{1+K}{2\alpha K}\right] \tag{8.33}$$

where in obtaining Equation 8.33 we used the approximation

$$e^{-\alpha(V_{gs}-V_{th})} \cong 1 - \alpha(V_{gs} - V_{th}) \tag{8.34}$$

If we rewrite (8.33) for V_{th} in terms of V_{gs} (maximum slope) we have

$$V_{th} = V_{gs}(\text{maximum slope}) - \left[\frac{1+K}{2\alpha K}\right] \tag{8.35}$$

Therefore, the approach is to use a measurement search routine that seeks the maximum slope, $\Delta I_{ds}/\Delta V_{gs}$, where the Δ is calculated as numerical derivatives, and having found V_{gs} (maximum slope) through measurement, use Equation 8.35 to calculate V_{th} from V_{gs} (maximum slope).

We could continue by including the effects of mobility degradation, modifying Equation 8.32 to

$$I_{ds} = \left[\frac{\beta}{1+\theta(V_{gs} - V_{th})}\right](V_{gs} - V_{th})V_{ds}[1 + Ke^{-\alpha(V_{gs}-V_{th})}] \tag{8.36}$$

where we have now included the effects of mobility degradation with the θ term. The extraction analysis can be done again by including this term, with a somewhat different solution obtained using suitable approximations.

The main point is that we write the equations we want and then solve them by whatever approximations we choose, as long as these approximations allow us to get the extracted results in closed form. This is parameter extraction.

8.2.1.6 Extraction vs. Optimization

Extraction has the advantage of being much faster than optimization, but it is not always as accurate. It is also much harder to supply extraction routines for models that are being developed. Each time you make a change in the model, you must make suitable changes in the corresponding extraction routine. For optimization, however, no changes are necessary other than the change in the model itself, because least-squares curve fitting routines are completely general. Also, if anything goes wrong in the extraction algorithm (and no access to the source code is available), almost nothing can be done to correct the problem. With optimization, one can always change the range of data, weighting, upper and lower bounds, etc. A least-squares curve fitting program can be steered toward a correct solution.

Novices at device characterization find least-squares curve fitting somewhat frustrating because a certain amount of user intervention and intuition is necessary to obtain the correct results. These beginners prefer extraction methods because they do not have to do anything. However, after being burned by extraction routines that do not work, a more experienced user will usually prefer the flexibility, control, and accuracy that optimization provides.

Commercial software is available that provides both extraction and optimization together. The idea here is to first use extraction techniques to make reasonable initial guesses and then use these results as a starting point for optimization, because optimization can give very poor results if poor initial guesses for the parameters are used. Nothing is wrong with using extraction techniques to provide initial guesses for optimization, but for an experienced user this is rarely necessary, assuming that the least-squares curve fitting routine is robust (converges well) and the experienced user has some knowledge of the process under characterization. Software that relies heavily on extraction may do so because of the nonrobustness of its optimizer.

These comments apply when an experienced user is doing optimization locally, not globally. For global optimization (a technique we do not recommend), the previous comparisons between extraction and optimization are not valid. The following section contains more detail about local vs. global optimization.

8.2.1.7 Strategies: General Discussion

The most naive way of using an optimization would be to take all the measured data for all devices, put them into one big file, and fit to all these data with all model parameters simultaneously. Even for a very high quality, robust optimization program the chances of this method converging are slight. Even if the program does converge, it is almost certain that the values of the parameters will be very unphysical. This kind of approach is an extreme case of global optimization. We call any optimization technique that tries to fit with parameters to data outside their region of applicability a global approach. That is, if we try to fit to saturation region data with linear region parameters such as threshold voltage, mobility, etc., we are using a global approach. In general, we advise avoiding global approaches, although in the strategies described later, sometimes the rules are bent a little.

Our recommended approach is to fit subsets of relevant parameters to corresponding subsets of relevant data in a way that makes physical sense. For example, in the MOS level 3 model, VT0 is defined as the threshold voltage of a long, wide transistor at zero back-bias. It does not make sense to use this parameter to fit to a short channel transistor, or to fit at nonzero back-bias values, or to fit to anywhere outside the linear region. In addition, subsets of parameters should be obtained in the proper order so that those obtained at a later step do not affect those obtained at earlier steps. That is, we would not obtain saturation region parameters before we have obtained linear region parameters because the values of the linear region parameters would influence the saturation region fits; we should have to go back and reoptimize on the saturation region parameters after obtaining the linear region parameters. Finally, never use optimization to obtain a parameter value when the parameter can be measured directly. For example, the MOS oxide thickness, TOX, is a model parameter, but we would never use optimization to find it. Always measure its value directly on a large oxide capacitor provided on the test chip. The recommended procedure for proper device characterization follows:

1. Have all the appropriate structures necessary on your test chip. Without this, the job cannot be performed properly.
2. Always measure whatever parameters are directly measurable. Never use optimization for these.
3. Fit the subset of parameters to corresponding subsets of data, and do so in physically meaningful ways.
4. Fit the parameters in the proper order so that those obtained later do not affect those obtained previously. If this is not possible, iteration may be necessary.

Naturally, a good strategy cannot be mounted if one is not intimately familiar with the model used. There is no substitute for learning as much about the model as possible. Without this knowledge, one must rely on strategies provided by software vendors, and these vary widely in quality.

Finally, no one can provide a completely general strategy applicable to all models and all process technologies. At some point the strategy must be tailored to suit the available technology and circuit performance requirements. This not only requires familiarity with the available device models, but also information from the circuit designers and process architects.

8.2.2 MOS DC Models

8.2.2.1 Available MOS Models

A number of MOS models have been provided over time with the original circuit simulation program, SPICE. In addition, some commercially available circuit simulation programs have introduced their own proprietary models, most notably HSPICE.* This section concentrates on the standard MOS models

* HSPICE is a commercially available, SPICE-like circuit simulation from Meta Software, Campbell, CA.

provided by UC Berkeley's SPICE, not only because they have become the standard models used by all circuit simulation programs, but also because the proprietary models provided by commercial vendors are not well documented and no source code is available for these models to investigate them thoroughly.

MOS levels 1, 2, and 3. Originally, SPICE came with three MOS models known as level 1, level 2, and level 3. The level 1 MOS model is a very crude first-order model that is rarely used. The level 2 and level 3 MOS models are extensions of the level 1 model and have been used extensively in the past and present [11]. These two models contain about 15 DC parameters each and are usually considered useful for digital circuit simulation down to 1 μm channel length technologies. They can fit the drain current for wide transistors of varying length with reasonable accuracy (about 5% RMS error), but have very little advanced fitting capability for analog application. They have only one parameter for fitting the subthreshold region, and no parameters for fitting the derivative of drain current with respect to drain voltage, G_{ds} (usually considered critical for analog applications). They also have no ability to vary the mobility degradation with back-bias, so the fits to I_{ds} in the saturation region at back-bias are not very good. Finally, these models do not interpolate well over device geometry, e.g., if a fit is made to a wide-long device and a wide-short device, and then one observes how the models track for lengths between these two extremes, they usually do not perform well. For narrow devices, they can be quite poor as well. Level 3 has very little practical advantage over level 2, although the level 2 model is proclaimed to be more physically based, whereas the level 3 model is called semiempirical. If only one can be used, perhaps levels 3 is slightly better because it runs somewhat faster and does not have quite such an annoying kink in the transition region from linear to saturation as does level 2.

***Berkeley short-channel IGFET model* (*BSIM*).** To overcome the many shortcomings of level 2 and level 3, the BSIM and BSIM2 models were introduced. The most fundamental difference between these and the level 2 and 3 models is that BSIM and BSIM2 use a different approach to incorporating the geometry dependence [3,6]. In levels 2 and 3, the geometry dependence is built directly into the model equations. In BSIM and BSIM2, each parameter (except for a very few) is written as a sum of three terms:

$$\text{Parameter} = \text{Par}_0 + \frac{\text{Par}_L}{L_{\text{eff}}} + \frac{\text{Par}_W}{W_{\text{eff}}} \tag{8.37}$$

where

Par_0 is the zero-order term
Par_L accounts for the length dependence of the parameter
Par_W accounts for the width dependence
L_{eff} and W_{eff} are the effective channel width and length, respectively

This approach has a large influence on the device characterization strategy, as discussed later. Because of this tripling of the number of parameters and for other reasons as well, the BSIM model has about 54 DC parameters and the BSIM2 model has over 100.

The original goal of the BSIM model was to fit better than the level 2 and 3 models for submicron channel lengths, over a wider range of geometries, in the subthreshold region, and for nonzero back-bias. Without question, BSIM can fit individual devices better than level 2 and level 3. It also fits the subthreshold region better and it fits better for nonzero back-biases. However, its greatest shortcoming is its inability to fit over a large geometry variation. This occurs because Equation 8.37 is a truncated Taylor series in $1/L_{\text{eff}}$ and $1/W_{\text{eff}}$ terms, and in order to fit better over varying geometries, higher power terms in $1/L_{\text{eff}}$ and $1/W_{\text{eff}}$ are needed. In addition, no provision was put into the BSIM model for fitting G_{ds}, so its usefulness for analog applications is questionable. Many of the BSIM model parameters are unphysical, so it is very hard to understand the significance of these model parameters. This has profound implications for generating skew models (fast and slow models to represent the process corners) and for incorporating temperature dependence. Another flaw of the BSIM model is its wild

behavior for certain values of the model parameters. If model parameters are not specified for level 2 or 3, they will default to values that will at least force the model to behave well. For BSIM, not specifying certain model parameters, setting them to zero, or various combinations of values can cause the model to become very ill-behaved.

BSIM2. The BSIM2 model was developed to address the shortcomings of the BSIM model. This was basically an extension of the BSIM model, removing certain parameters that had very little effect, fixing fundamental problems such as currents varying the wrong way as a function of certain parameters, adding more unphysical fitting parameters, and adding parameters to allow fitting G_{ds}. BSIM2 does fit better than BSIM, but with more than twice as many parameters as BSIM, it should. However, it does not address the crucial problem of fitting large geometry variations. Its major strengths over BSIM are fitting the subthreshold region better, and fitting G_{ds} better. Most of the other shortcomings of BSIM are also present in BSIM2, and the large number of parameters in BSIM2 makes it a real chore to use in device characterization.

BSIM3. Realizing the shortcomings of BSIM2, UC Berkeley recently introduced the BSIM3 model. This is an unfortunate choice of name because it implies BSIM3 is related to BSIM and BSIM2. In reality, BSIM3 is an entirely new model that in some sense is related more to levels 2 and 3 than BSIM or BSIM2. The BSIM3 model abandons the length and width dependence approach of BSIM and BSIM2, preferring to go back to incorporating the geometry dependence directly into the model equations, as do levels 2 and 3. In addition, BSIM3 is a more physically based model, with about 30 fitting parameters (the model has many more parameters, but the majority of these can be left untouched for fitting), making it more manageable, and it has abundant parameters for fitting G_{ds}, making it a strong candidate for analog applications.

It is an evolving model, so perhaps it is unfair to criticize it at this early stage. Its greatest shortcoming is, again, the inability to fit well over a wide range of geometries. It is hoped that future modifications will address this problem. In all fairness, however, it is a large order to ask a model to be physically based, have not too many parameters, be well behaved for all default values of the parameters, fit well over temperature, fit G_{ds}, fit over a wide range of geometries, and still fit individual geometries as well as a model with over 100 parameters, such as BSIM2. Some of these features were compromised in developing BSIM3.

Proprietary models. A number of other models are available from commercial circuit simulator vendors, the literature, etc. Some circuit simulators also offer the ability to add a researcher's own models. In general, we caution against using proprietary models, especially those which are supplied without source code and complete documentation. Without an intimate knowledge of the model equations, it is very difficult to develop a good device characterization strategy. Also, incorporating such models into device characterization software is almost impossible. To circumvent this problem, many characterization programs have the ability to call the entire circuit simulator as a subroutine in order to exercise the proprietary model subroutines. This can slow program execution by a factor of 20 or more, seriously impacting the time required to characterize a technology. Also, if proprietary models are used without source code, the circuit simulator results can never be checked against another circuit simulator. Therefore, we want to stress the importance of using standard models. If these do not meet the individual requirements, the next best approach is to incorporate a proprietary model whose source code one has access to. This requires being able to add the individual model not only to circuit simulators, but also to device characterization programs; it can become a very large task.

8.2.2.2 MOS Level 3 Extraction Strategy in Detail

The strategy discussed here is one that we consider to be a good one, in the spirit of our earlier comments. Note, however, that this is not the only possible strategy for the level 3 model. The idea here is to illustrate basic concepts so that this strategy can be refined to meet particular individual requirements.

In order to do a DC characterization, the minimum requirement is one each of the wide-long, wide-short, and narrow-long devices. We list the steps of the procedure and then discuss them in more detail.

Step 1. Fit the wide-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VT0 (threshold voltage), U0 (mobility), and THETA (mobility degradation with V_{gs}).

Step 2. Fit the wide-short device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VT0, LD (length encroachment), and THETA. When finished with this step, replace VT0 and THETA with the values from step 1, but keep the value of LD.

Step 3. Fit the narrow-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VT0, DW (width encroachment), and THETA. When finished with this step, replace VT0 and THETA with the values from step 1, but keep the value of DW.

Step 4. Fit the wide-short device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters RS and RD (source and drain series resistance).

Step 5. Fit the wide-long device in the linear region at all back-biases, at V_{gs} values above the subthreshold region, with parameter NSUB (channel doping affects long channel variation of threshold voltage back-bias).

Step 6. Fit the wide-short device in the linear region at all back-biases, at V_{gs} values above the subthreshold region, with parameter XJ (erroneously called the junction depth; affects short-channel variation of threshold voltage with back-bias).

Step 7. Fit the narrow-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameter DELTA (narrow channel correction to threshold voltage).

Step 8. Fit the wide-short device in the saturation region at zero back-bias (or all back-biases) with parameters VMAX (velocity saturation), KAPPA (saturation region slope fitting parameter), and ETA (V_{ds} dependence of threshold voltage).

Step 9. Fit the wide-short device in the subthreshold region at whatever back-bias and drain voltage is appropriate (usually zero back-bias and low V_{ds}) with parameter NFS (subthreshold slope fitting parameter). One may need to fit with VT0 also and then VT0 is replaced after this step with the value of VT0 obtained from step 1.

This completes the DC characterization steps for the MOS level 3 model. One would then go on to do the junction and overlap capacitance terms (discussed later). Note that this model has no parameters for fitting over temperature, although temperature dependence is built into the model that the user cannot control.

In Step 1, VT0, U0, and THETA are defined in the model for a wide-long device at zero back-bias. They are zero-order fundamental parameters without any short or narrow channel corrections. We therefore fit them to a wide-long device. It is absolutely necessary that such a device be on the test chip. Without it, one cannot obtain these parameters properly. The subthreshold region must be avoided also because these parameters do not control the model behavior in subthreshold.

In Step 2, we use LD to fit the slope of the linear region curve, holding U0 fixed from step 1. We also fit with VT0 and THETA because without them the fitting will not work. However, we want only the value of LD that fits the slope, so we throw away VT0 and THETA, replacing them with the values from step 1.

Step 3 is the same as step 2, except that we are getting the width encroachment instead of the length.

In Step 1, the value of THETA that fits the high V_{gs} portion of the wide-long device linear region curve was found. Because the channel length of a long transistor is very large, the source and drain series resistances have almost no effect here, but for a short-channel device, the series resistance will also affect the high V_{gs} portion of the linear region curve. Therefore, in step 4 we fix THETA from step 1 and use RS and RD to fit the wide-short device in the linear region, high V_{gs} portion of the curve.

In Step 5, we fit with NSUB to get the variation of threshold voltage with back-bias. We will get better results if we restrict ourselves to lower values of V_{gs} (but still above subthreshold) because no mobility

degradation adjustment exists with back-bias, and therefore the fit may not be very good at higher V_{gs} values for the nonzero back bias curves.

Step 6 is just like step 5, except we are fitting the short-channel device. Some people think that the value of XJ should be the true junction depth. This is not true. The parameter XJ is loosely related to the junction depth, but XJ is really the short-channel correction to NSUB. Do not be surprised if XJ is not equal to the true junction depth.

Step 7 uses DELTA to make the narrow channel correction to the threshold voltage. This step is quite straightforward.

Step 8 is the only step that fits in the saturation region. The use of parameters VMAX and KAPPA is obvious, but one may question using ETA to fit in the saturation region. The parameter ETA adjusts the threshold voltage with respect to V_{ds}, and as such one could argue that ETA should be used to fit measurements of I_{ds} sweeping V_{gs} and stepping V_{ds} to high values. In doing so, one will corrupt the fit in the saturation region, and usually we want to fit the saturation region better at the expense of the linear region.

Step 9 uses NFS to fit the slope of the $\log(I_{ds})$ vs. V_{gs} curve. Often, the value of VT0 obtained from step 1 will prevent one from obtaining a good fit in the subthreshold region. If this happens, try fitting with VT0 and NFS, but replacing the final value of VT0 with that from step 1 at the end, keeping only NFS from this final step.

The preceding steps illustrate the concepts of fitting relevant subsets of parameters to relevant subsets of data to obtain physical values of the parameters, as well as fitting parameters in the proper order so that those obtained in the later steps will affect those obtained in earlier steps minimally. Please refer to Figures 8.19 and 8.20 for how the resulting fits typically appear (all graphs showing levels fits are provided by the device modeling software package Aurora, from Technology Modeling Associates, Inc., Palo Alto, CA).

An experienced person may notice that we have neglected some parameters. For example, we did not use parameters KP and GAMMA. This means KP will be calculated from U0, and GAMMA will be calculated from NSUB. In a sense U0 and NSUB are more fundamental parameters than KP and GAMMA. For example, KP depends on U0 and TOX; GAMMA depends on NSUB and TOX. If one is trying to obtain skew models, it is much more advantageous to analyze statistical distributions of

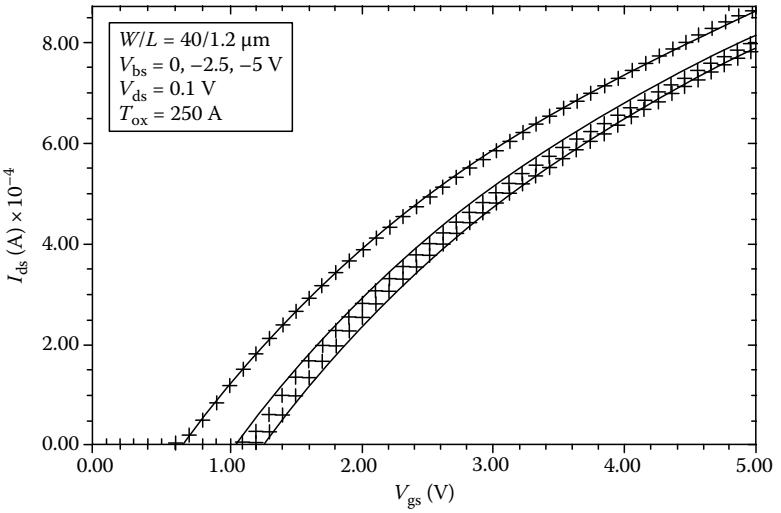


FIGURE 8.19 Typical MOS level 3 linear region measured and simulated plots at various V_{bs} values for a wide-short device.

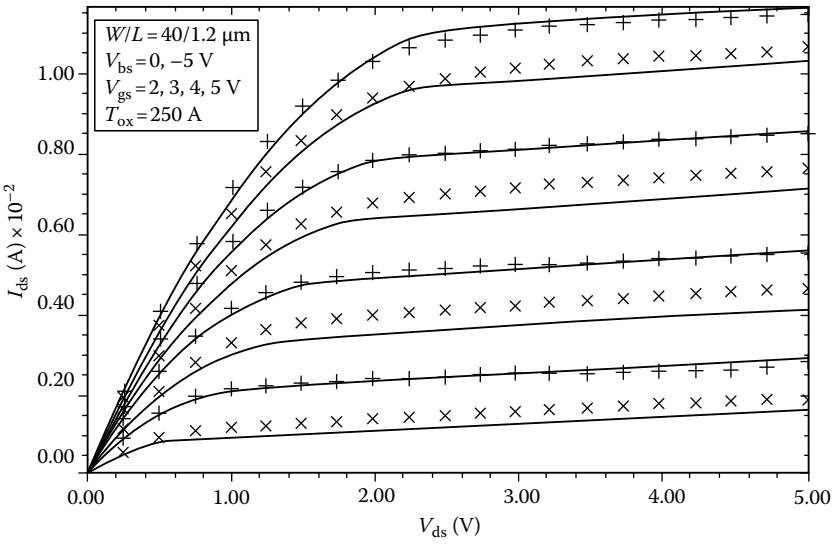


FIGURE 8.20 Typical MOS level 3 saturation region measured and simulated plots at various V_{gs} and V_{bs} values for a wide-short device.

parameters that depend on a single effect than those that depend on multiple effects. KP will depend on mobility and oxide thickness; U0 is therefore a more fundamental parameter. We also did not obtain parameter PHI, so it will be calculated from NSUB. The level 3 model is very insensitive to PHI, so using it for curve fitting is pointless. This illustrates the importance of being very familiar with the model equations. The kind of judgments described here cannot be made without such knowledge.

Test chip warnings. The following hints will greatly assist in properly performing device characterization:

1. Include a wide-long device; without this, the results will not be physically correct.
2. All MOS transistors with the same width should be drawn with their sources and drains identical. No difference should be seen in the number of source/drain contacts, contact spacing, source/drain contact overlap, poly gate to contact spacing, etc. In Figure 8.21, *c* is the contact size, *cs* is the contact space, *cov* is the contact overlap of diffusion, and *c2p* is the contact to poly spacing. All these dimensions should be identical for devices of different *L* but identical *W*. If not, extracting the series resistance will become more difficult.

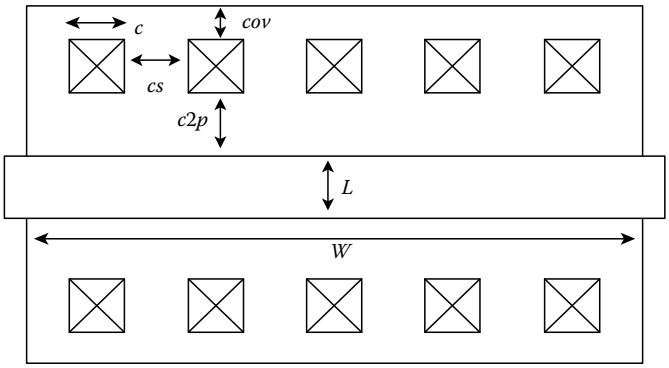


FIGURE 8.21 Mask layers showing dimensions that should be identical for all MOS devices of the same width.

3. Draw devices in pairs. That is, if the wide-long device is $W/L = 20/20$, make the wide-short device the same width as the wide-long device; e.g., make the short device 20/1, not 19/1. If the narrow-long device is 2/20, make the narrow-short device of the same width; i.e., make it 2/1, not 3/1, and similarly for the lengths. (Make the wide-short and the narrow-short devices have the same length.)
4. Draw structures for measuring the junction capacitances and overlap capacitances. These are discussed later in Section 8.2.5.

8.2.3 BSIM Extraction Strategy in Detail

All MOS model strategies have basic features in common; namely, fit the linear region at zero back-bias to get the basic zero-order parameters, fit the linear region at nonzero back-bias, fit the saturation region at zero back-bias, fit the saturation region at nonzero back-bias, and then fit the subthreshold region. It is possible to extend the type of strategy we covered for level 3 to the BSIM model, but that is not the way BSIM was intended to be used.

The triplet sets of parameters for incorporating geometry dependence into the BSIM model, Equation 8.37, allow an alternate strategy. We obtain sets of parameters without geometry dependence by fitting to individual devices without using the Par_L and Par_W terms. We do this for each device size individually. This produces sets of parameters relevant to each individual device. So, for device number 1 of width $W(1)$ and length $L(1)$ we would have a value for the parameter VFB which we will call VFB(1); for device number n of width $W(n)$ and length $L(n)$ we will have VFB(n). To get the Par_0, Par_L and Par_W terms, we fit to the parameters themselves, rather than the measured data. That is, to get VFB_0, VFB_L, and VFB_W, we fit to the "data points" VFB(1),...,VFB(n) with parameters VFB_0, VFB_L, and VFB_W using Equation 8.37, where L_{eff} and W_{eff} are different for each index, 1 through n.

Note that as L and W become very large, the parameters must approach Par_0. This suggests that we use the parameter values for the wide-long device as the Par_0 terms and only fit the other geometry sizes to get the Par_W and Par_L terms. For example, if we have obtained VFB(1) for our first device which is our wide-long device, we would set $\text{VFB}_0 = \text{VFB}(1)$, and then fit to VFB(2),...,VFB(n) with parameters VFB_L and VFB_W, and similarly for all the other triplets of parameters. In order to use a general least-squares optimization program in this way the software must be capable of specifying parameters as targets, as well as measured data points.

We now list a basic strategy for the BSIM model:

Step 1. Fit the wide-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VFB (flatband voltage), MUZ (mobility), and U0 (mobility degradation), with DL (length encroachment) and DW (width encroachment) set to zero.

Step 2. Fit the wide-short device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VFB, U0, and DL.

Step 3. Fit the narrow-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VFB, U0, and DW.

Step 4. Refit the wide-long device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VFB, MUZ, and U0, now that DL and DW are known.

Step 5. Fit the wide-short device in the linear region at zero back-bias, at V_{gs} values above the subthreshold region, with parameters VFB, RS, and RD. When finished, replace the value of VFB with the value found in step 4.

Step 6. Fit the wide-long device in the linear region at all back-biases, at V_{gs} values above the subthreshold region, with parameters K1 (first-order body effect), K2 (second-order body effect), U0, and X2U0 (V_{bs} dependence of U0).

Step 7. Fit the wide-long device in the saturation region at zero back-bias with parameters U0, ETA (V_{ds} dependence of threshold voltage), MUS (mobility in saturation), U1 (V_{ds} dependence of mobility), and X3MS (V_{ds} dependence cf MUS).

Step 8. Fit the wide-long device in the saturation region at all back-biases with parameter X2MS (V_{bs} dependence of MUS).

Step 9. Fit the wide-long device in the subthreshold region at zero back-bias and low V_{ds} value with parameter N0; then fit the subthreshold region nonzero back-bias low V_{ds} data with parameter NB; and finally fit the subthreshold region data at higher V_{ds} values with parameter ND. Or, fit all the subthreshold data simultaneously with parameters N0, NB, and ND.

Repeat steps 6 through 10 for all the other geometries, with the result of sets of geometry-independent parameters for each different size device. Then follow the procedure described previously for obtaining the geometry-dependent terms Par_0, Par_L, and Par_W.

In the preceding strategy, we have omitted various parameters either because they have minimal effect or because they have the wrong effect and were modified in the BSIM2 model. Because of the higher complexity of the BSIM model over the level 3 model, many more strategies are possible than the one just listed. One may be able to find variations of the above strategy that suit the individual technology better. Whatever modifications are made, the general spirit of the above strategy probably will remain.

Some prefer to use a more global approach with BSIM, fitting to measured data with Par_L and Par_W terms directly. Although this is certainly possible, it is definitely not a recommended approach. It represents the worst form of blind curve fitting, with no regard for physical correctness or understanding. The BSIM model was originally developed with the idea of obtaining the model parameters via extraction as opposed to optimization. In fact, UC Berkeley provides software for obtaining BSIM parameters using extraction algorithms, with no optimization at all. As stated previously, this has the advantage of being relatively fast and easy. Unfortunately, it does not always work. One of the major drawbacks of the BSIM model is that certain values of the parameters can cause the model to produce negative values of G_{ds} in saturation. This is highly undesirable, not only from a modeling standpoint, but also because of the convergence problems it can cause in circuit simulators. If an extraction strategy is used that does not guarantee nonnegative G_{ds}, very little can be done to fix the problem when G_{ds} becomes negative. Of course, the extraction algorithms can be modified, but this is difficult and time consuming. With optimization strategies, one can weight the fitting for G_{ds} more heavily and thus force the model to produce nonnegative G_{ds}. We, therefore, do not favor extraction strategies for BSIM, or anything else. As with most things in life, minimal effort provides minimal rewards.

8.2.3.1 BSIM2 Extraction Strategy

We do not cover the BSIM2 strategy in complete detail because it is very similar to the BSIM strategy, except more parameters are involved. The major difference in the two models is the inclusion of extra terms in BSIM2 for fitting G_{ds} (refer to Figure 8.22, which shows how badly BSIM typically fits $1/G_{ds}$ vs. V_{ds}). Basically, the BSIM2 strategy follows the BSIM strategy for the extraction of parameters not related to G_{ds}. Once these have been obtained, the last part of the strategy includes steps for fitting to G_{ds} with parameters that account for channel length modulation (CLM) and hot electron effects. The way this proceeds in BSIM2 is to fit I_{ds} first, and then parameters MU2, MU3, and MU4 are used to fit to $1/G_{ds}$ vs. V_{ds} curves for families of V_{ds} and V_{bs}. This can be a very time-consuming and frustrating experience, because fitting to $1/G_{ds}$ is quite difficult. Also, the equations describing how G_{ds} is modeled with MU2, MU3, and MU4 are very unphysical and the interplay between the parameters makes fitting awkward. The reader is referred to Figure 8.23, which shows how BSIM2 typically fits $1/G_{ds}$ vs. V_{ds}. BSIM2 is certainly better than BSIM, but it has its own problems fitting $1/G_{ds}$.

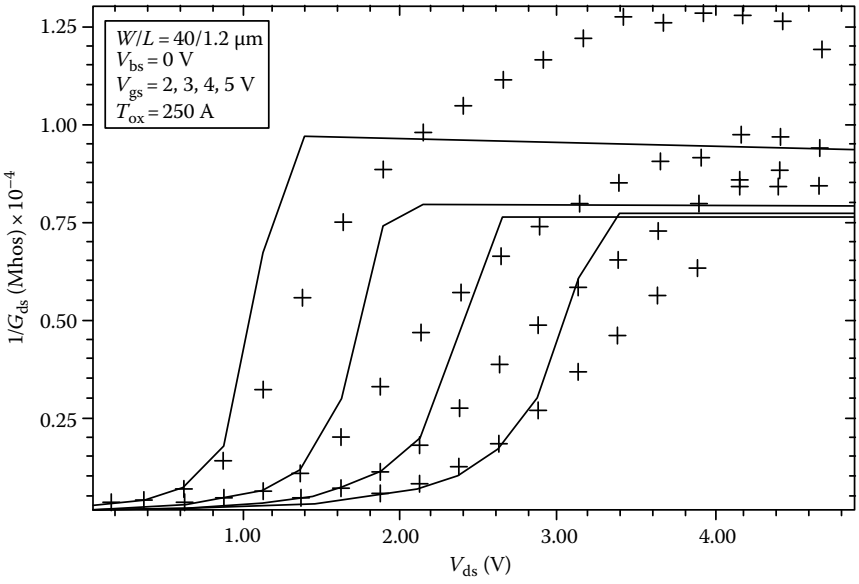


FIGURE 8.22 Typical BSIM $1/G_{ds}$ vs. V_{ds} measured and simulated plots at various V_{gs} values for a wide-short device.

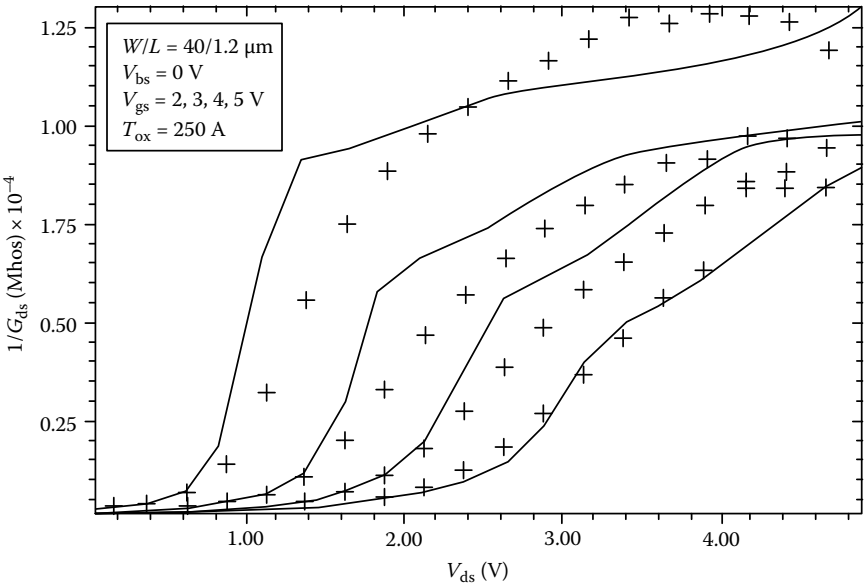


FIGURE 8.23 Typical BSIM2 $1/G_{ds}$ vs. V_{ds} measured and simulated plots at various V_{gs} values for a wide-short device.

8.2.3.2 BSIM3 Comments

The BSIM3 model is very new and will undoubtedly change in the future [5]. We will not list a BSIM3 strategy here, but focus instead on the features of the model that make it appealing for analog modeling.

BSIM3 has terms for fitting G_{ds} that relate to CLM, drain-induced barrier lowering (DIBL), and hot electron effects. They are incorporated completely differently from the G_{ds} fitting parameters of BSIM2.

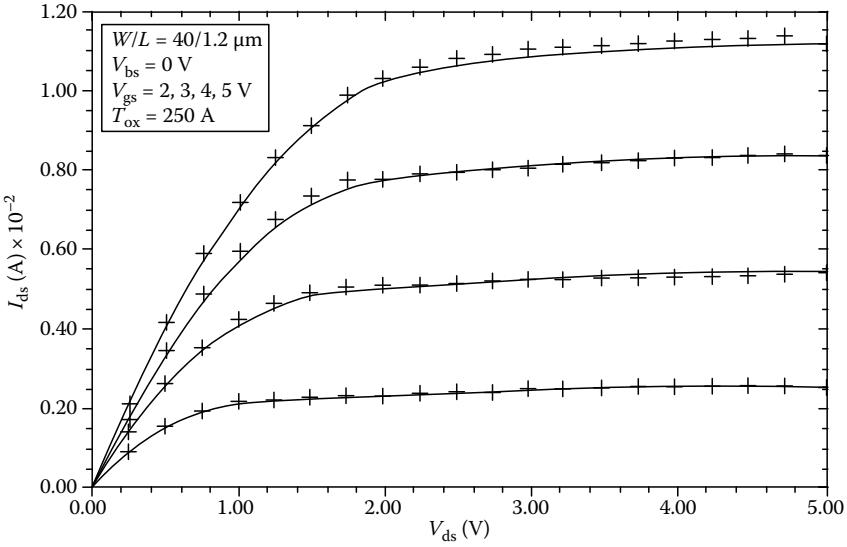


FIGURE 8.24 Typical BSIM3 saturation region measured and simulated plots at various V_{gs} values for a wide-short device.

In BSIM3, these parameters enter through a generalized Early voltage relation, with the drain current in saturation written as

$$I_{ds} = I_{d\,sat}\left[1 + \frac{(V_{ds} - V_{d\,sat})}{V_A}\right] \tag{8.38}$$

where V* is a generalized early voltage made up of three terms as

$$\frac{1}{V_A} = \frac{1}{V_{ACLM}} + \frac{1}{V_{ADIBL}} + \frac{1}{V_{AHCE}} \tag{8.39}$$

with the terms in Equation 8.39 representing generalized early voltages for CLM, DIBL, and hot carrier effects (HCE). This formulation is more physically appealing than the one used in BSIM2, making it easier to fit $1/G_{ds}$ vs. V_{ds}. curves with BSIM2. Figures 8.24 and 8.25 show how BSIM3 typically fits I_{ds} vs. V_{ds} and $1/G_{ds}$ vs. V_{ds}.

Most of the model parameters for BSIM3 have physical significance so they are obtained in the spirit of the parameters for the level 2 and 3 models. The incorporation of temperature dependence is also easier in BSIM3 because the parameters are more physical. All this, coupled with the fact that about 30 parameters exists for BSIM3 as compared to over 100 for BSIM2, makes BSIM3 a logical choice for analog design. However, BSIM3 is evolving, and shortcomings to the model may still exist that may be corrected in later revisions.

8.2.3.3 Which MOS Model to Use?

Many MOS models are available in circuit simulators, and the novice is bewildered as to which model is appropriate. No single answer exists, but some questions must be asked before making a choice:

1. What kind of technology am I characterizing?
2. How accurate a model do I need?

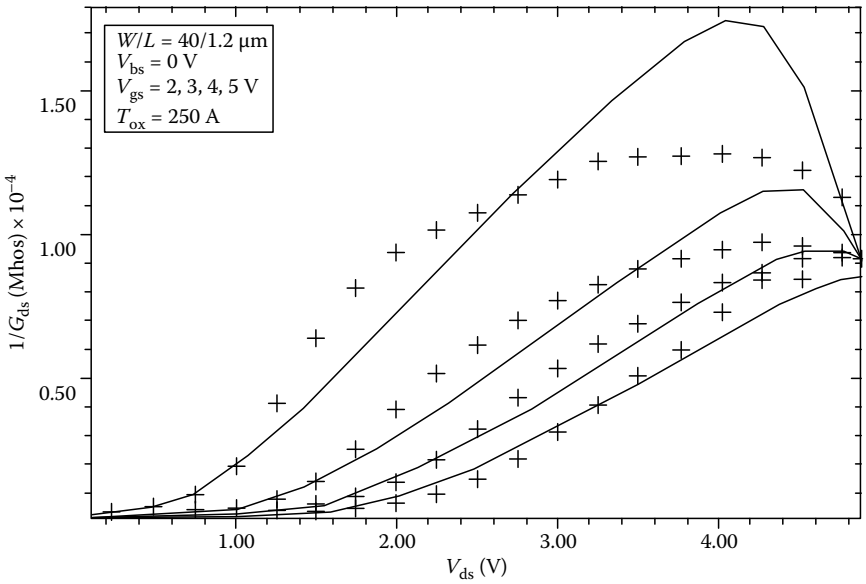


FIGURE 8.25 Typical BSIM3 $1/G_{ds}$ vs. V_{ds} measured and simulated plots at various V_{gs} values for a wide-short device.

3. Do I want to understand the technology?
4. How important are the skew model files (fast and slow parameter files)?
5. How experienced am I? Do I have the expertise to handle a more complicated model?
6. How much time can I spend doing device characterization?
7. Do I need to use this model in more than one circuit simulator?
8. Is the subthreshold region important?
9. Is fitting G_{ds} important?

Let us approach each question with regard to the models available. If the technology is not submicron, perhaps a simpler model such as level 3 is capable of doing everything needed. If the technology is deep submicron, then use a more complicated model such as BSIM, BSIM2, or BSIM3. If high accuracy is required, then the best choice is BSIM3, mainly because it is more physical than all the other models and is capable of fitting better.

For a good physical understanding of the process being characterized, BSIM and BSIM2 are not good choices. These are the least physically based of all the models. The level 2 and 3 models have good physical interpretation for most of the parameters, although they are relatively simple models. BSIM3 is also more physically based, with many more parameters than level 2 or 3, so it is probably the best choice.

If meaningful skew models need to be generated, then BSIM and BSIM2 are very difficult to use, again, because of their unphysical parameter sets. Usually, the simplest physically based model is the best for skew model generation. A more complicated physically based model such as BSIM3 may also be difficult to use for skew model generation.

If the user is inexperienced, none of the BSIM models should be used until the user's expertise improves. Our device is to practice using simpler models before tackling the harder ones.

If time is critical, the simpler models will definitely be much faster for use in characterization. The more complicated models require more measurements over wider ranges of voltages as well as wider ranges of geometries. This, coupled with the larger number of parameters, means they will take some time with which to work. The BSIM2 model will take longer than all the rest, especially if the G_{ds} fitting parameters are to be used.

The characterization results may need to be used in more than one circuit simulator. For example, if a foundry must supply models to various customers, they may be using different circuit simulators. In this case, proprietary models applicable to a single circuit simulator should not be used. Also, circuit designers may want to check the circuit simulation results on more than one circuit simulator. It is better to use standard Berkeley models (level 2, level 3, BSIM, BSIM2, and BSIM3) in such cases.

If the subthreshold region is important, then level 2 or level 3 cannot be used, and probably not even BSIM. BSIM2 or BSIM3 must be used instead. These two models have enough parameters for fitting the subthreshold region.

If fitting G_{ds} is important, BSIM2 and BSIM3 are, again, the only choices. None of the other models have enough parameters for fitting G_{ds}.

Finally, if a very unusual technology is to be characterized, none of the standard models may be appropriate. In this case commercially available specialized models or the user's own models must be used. This will be a large task, so the goals must justify the effort.

8.2.4 Bipolar DC Model

The standard bipolar model used by all circuit simulators is the Gummel–Poon model, often called the BJT (bipolar junction transistor) model [4]. Most circuit simulator vendors have added various extra parameters to this model, which we will not discuss, but they all have the basic parameters introduced by the original UC Berkeley SPICE model.

We now list a basic strategy for the BJT model:

Step 1. Fit a Gummel plot (log(IC) and log(IB) vs. VBE) for the IC curve in the straight line portion of the curve (low to middle VBE values) with parameters IS and NF. Note that it is best if the Gummel plots are done with VCB = 0.

Step 2. Fit plots of IC vs. VCE, stepping IB, in the high VCE region of the curves with parameters VAF, BF, and IKF.

Step 3. Fit plots of IC vs. VCE, stepping IB, in the low VCE region of the curves with parameter RC.

Step 4. Fit a Gummel plot for the IB curve for VBE values in the mid- to high-range with parameter BF.

Step 5. Fit a Gummel plot for the IB curve for low VBE values with parameters ISE and NE.

Step 6. Try to obtain parameter RE by direct measurement or by simulation using a device simulator such as PISCES. It is also best if RB can be obtained as a function of current by measurement on special structures, or by simulation. One may also obtain RB by *S*-parameter measurements (discussed later). Failing this, RB may be obtained by fitting to the Gummel plot for the IB curve for VBE values in the mid- to high-range with parameters RB, RBM, and IRB. This is a difficult step to perform with an optimization program.

Step 7. Fit a Gummel plot for the IC curve for high VBE values with parameter IKF.

Step 8. At this point, all the DC parameters have been obtained for the device in the forward direction. Next, obtain the reverse direction data by interchanging the emitter and collector to get the reverse Gummel and IC vs. VCE data.

Repeat Step 2 on the reverse data to get VAR, BR, IKR.
Repeat Step 4 on the reverse data to get BR.
Repeat Step 5 on the reverse data to get ISC, NC.
Repeat Step 7 on the reverse data to get IKR.

Fitting the reverse direction parameters is very frustrating because the BJT model is very poor in the reverse direction. After seeing how poor the fits can be, one may decide to obtain only a few of the reverse parameters.

8.2.5 MOS and Bipolar Capacitance Models

8.2.5.1 MOS Junction Capacitance Model

The MOS junction capacitance model accounts for bottom wall (area component) and sidewall (periphery component) effects. The total junction capacitance is written as a sum of the two terms

$$\mathrm{CJ(total)} = \frac{A(\mathrm{CJ})}{1 + [(VR/\mathrm{PB})]^{\mathrm{MJ}}} + \frac{P(\mathrm{CJSW})}{1 + [(VR/\mathrm{PB})]^{\mathrm{MJSW}}} \tag{8.40}$$

where

A is the area
P is the periphery of the junction diode capacitor
VR is the reverse bias (a positive number) across the device

The parameters of this model are CJ, CJSW, PB, MJ, and MJSW.

It is very easy to find parameter values for this model. Typically, one designs special large junction diodes on the test chip, large enough so that the capacitances being measured are many tens of picofarads. This means the diodes have typical dimensions of hundreds of microns. Two junction diodes are needed—one with a large area:periphery ratio, and another with a small area:periphery ratio. This is usually done by drawing a rectangular device for the large area:periphery ratio, and a multifingered device for the small area:periphery ratio (see Figure 8.26).

The strategy for this model consists of a single step, fitting to all the data simultaneously with all the parameters. Sometimes it is helpful to fix PB to some nominal value such as 0.7 or 0.8, rather than optimize on PB, because the model is not very sensitive to PB.

8.2.5.2 BJT Junction Capacitance Model

The BJT model is identical to the MOS, except it uses only the bottom wall (area term), not the sidewall. Three sets of BJT junction diodes need to be characterized: emitter–base, collector–base, and collector–substrate. Each of these has three model parameters to fit. For example, for the emitter–base junction diode, the parameters are CJE, VJE, and MJE (CJE corresponds to CJ, VJE to PB, and MJE to MJ of the MOS model). Similarly, for the collector–base junction diode, we have CJC, VJC, and MJC. Finally, for the collector–substrate junction diode, we have CJS, VJS, and MJS.

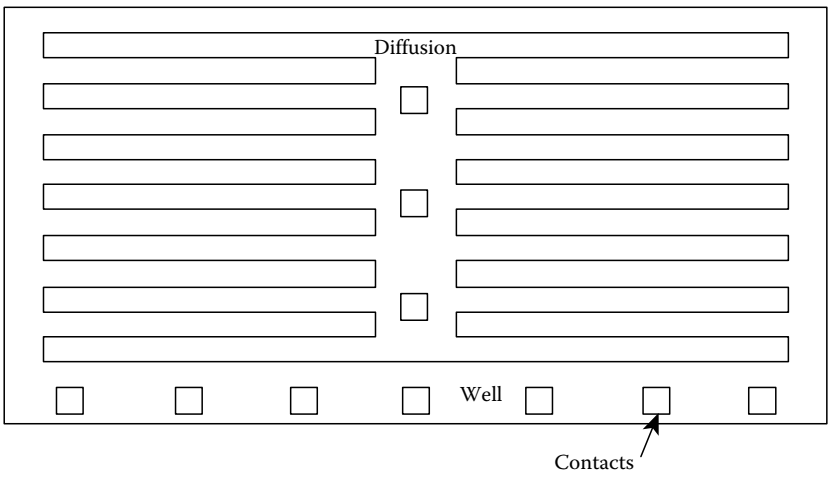


FIGURE 8.26 Mask layers showing the periphery multifingered junction diode structure.

These parameters are all fit in a single strategy step on the relevant data. Again, it is best if large test structures can be drawn rather than the values measured directly on transistor, because the transistor junction diodes have such small areas that the measurements are prone to accuracy problems.

8.2.5.3 MOS Overlap Capacitance Model

In addition to the junction capacitance model, the MOS models include terms for the overlap or Miller capacitance. This represents the capacitance associated with the poly gate overlap over the source and drain regions. As for the junction diodes, special test chip structures are necessary for obtaining the overlap capacitance terms CGS0, CGD0, and CGB0. These structures usually consist of large, multi-fingered devices that look just like the junction capacitance periphery diodes, except that where the junction diodes are diffusion over well structures, the overlap capacitors are poly over diffusion.

8.2.6 Bipolar High-Frequency Model

The SPICE BJT model has five parameters for describing the high-frequency behavior of bipolar devices. These are parameters TF, XTF, ITF, VTF, and PTF, all of which are associated with the base transit time, or equivalently, the cutoff frequency. Parameter TF is the forward transit time, XTF and ITF modify TF as a function of VBE (or IC), and VTF modifies TF as function of VBE (or IC), and VTF modifies TF as a function of VCB. Parameter PTF represents the excess phase at the cutoff frequency. The expression used by SPICE for the transit time is

$$\tau_f = \mathrm{TF}\left[1 + \mathrm{XTF}\left\{\frac{\mathrm{IF}}{\mathrm{IF} + \mathrm{ITF}}\right\}^2\right] e^{[\mathrm{VBC}/\{(1.44(\mathrm{VTF}))\}]} \tag{8.41}$$

where

$$\mathrm{IF} = \mathrm{IS}[e^{\{(q)(V_{\mathrm{be}})\}/\{(NF)(k)(T)\}} - 1] \tag{8.42}$$

SPICE uses the transit time to calculate the diffusion capacitance of the emitter–base junction in forward bias.

Obtaining the bipolar high frequency parameters is very complicated, time consuming, and difficult. It also requires very expensive measuring equipment, including a network analyzer for measuring *S*-parameters, and a quality high-frequency wafer prober with high-frequency probes. Having measured the *S*-parameters, after suitable data manipulation and optimization it is possible to find the high-frequency SPICE parameters.

It is beyond the scope of this book to cover high-frequency bipolar theory completely. We will, however, list the basic steps involved in obtaining the high frequency bipolar transit time parameters:

Step 1. Measure the *S*-parameters over an appropriate frequency range to go beyond the cutoff frequency, f_T. Do these measurements over a large range of IC values and stepping over families of VCB values as well. Be sure to include VCB = 0 in these data. Be sure that the network analyzer being used is well calibrated. These measurements can be done in either common emitter or common collector mode. Each has advantages and disadvantages.

Step 2. De-embed the *S*-parameter measurements by measurements dummy pad structures that duplicate the layout of the bipolar transistors, including pads and interconnect, but with no devices. The de-embedding procedure subtracts the effects of the pads and interconnects from the actual devices. This step is very important if reliable data are desired [10].

Step 3. Calculate the current gain, β from the *S*-parameter data. From plots of β vs. frequency, find the cutoff frequency, f_T. Also calculate parameter PTF directly from these data.

Step 4. Calculate τ_f from f_T by removing the effect of RE, RC, and the junction capacitances. This produces tables of τ_f vs. IC, or equivalently, τ_f vs. VBE, over families of VCB.

Step 5. Optimize on τ_f vs. VBE data over families of VCB with parameters TF, XTF, ITF, and VTF.

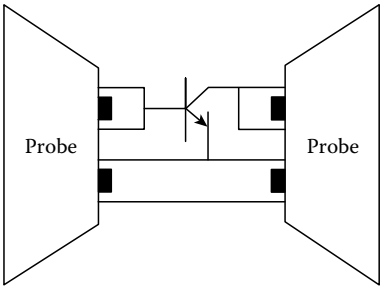


FIGURE 8.27 Typical bipolar transistor test chip layout for measuring *S*-parameters with high-frequency probes. This shows the common emitter configuration.

Note that many of the preceding steps are done automatically by some commercial device modeling programs.

It is very important to have well-designed and appropriate test structures for measuring *S*-parameters. The pad layout and spacing is very critical because of the special size and shape of the high-frequency probes (see Figure 8.27). The test chip structures must be layed out separately for common emitter and common collector modes if measurements are necessary for both these modes.

It is also possible to use the *S*-parameter measurements to obtain the base resistance as a function of collector current. From this information one can optimize to obtain parameters RB, RBM, and IRB. However, the base resistance obtained from *S*-parameters is not a true DC base resistance, and the model is expecting DC values of base resistance.

8.2.7 Miscellaneous Topics

8.2.7.1 Skew Parameter Files

This chapter discusses obtaining model parameters for a single wafer, usually one that has been chosen to represent a typical wafer for the technology being characterized. The parameter values obtained from this wafer correspond to a typical case. Circuit designers also want to simulate circuits with parameter values representing the extremes of process variation, the so-called fast and slow corners, or skew parameter files. These represent the best and worst case of the process variation over time.

Skew parameter values are obtained usually by tracking a few key parameters, measuring many wafers over a long period of time. The standard deviation of these key parameters is found and added to or subtracted from the typical parameter values to obtain the skew models. This method is extremely crude and will not normally produce a realistic skew model. It will almost always overestimate the process spread, because the various model parameters are not independent—they are correlated.

Obtaining realistic skew parameter values, taking into account all the subtle correlations between parameters, is more difficult. In fact, skew model generation is often more an art than a science. Many attempts have been made to utilize techniques from a branch of statistics called multivariate analysis [1]. In this approach, principal component or factor analysis is used to find parameters that are linear combinations of the original parameters. Only the first few of these new parameters will be kept; the others will be discarded because they have less significance. This new set will have fewer parameters than the original set and therefore will be more manageable in terms of finding their skews. The user sometimes must make many choices in the way the common factors are utilized, resulting in different users obtaining different results.

Unfortunately, a great deal of physical intuition is often required to use this approach effectively. To date, we have only seen it applied to the simpler MOS models such as level 3. It is not known if this is a viable approach for a much more complicated model such as BSIM2 [7].

8.2.7.2 Macro Modeling

Sometimes, a device is designed and manufactured that cannot be modeled by a single transistor model. In such cases, one may try to simulate the behavior of a single device by using many basic device elements together, representing the final device. For example, a single real bipolar device might be modeled using a combination of several ideal bipolar devices, diodes, resistors, and capacitors. This would be a macro model representing the real bipolar device.

Macro modeling usually uses least-squares optimization techniques. One chooses model parameter values to represent the ideal devices in the technology, and then optimizes on geometry size, capacitance size, resistance size, etc., to obtain the final macro model. This is similar to the optimization techniques used to find model parameters for a standard model, but in this case we are calling on an entire circuit simulator rather than a model subroutine to provide the output characteristics of our macro model.

Obviously, macro modeling can be very computation intensive. A complex macro model can take a very long time to optimize. Also, it may be impractical to use a macro model for every real device in the circuit. For example, if the circuit in question has 100,000 real transistors and each of these is being represented by a macro model with 10 components, the number of circuit nodes introduced by the macro model ($>10 \times 10{,}000$) might be prohibitive in terms of simulation time. Nevertheless, for critical components, macro modeling can provide tremendous insight.

8.2.7.3 Modeling in a TCAD Environment

Programs that do optimization are part of a larger set of software that is sometimes called TCAD (technology computer-aided design). Typically, TCAD encompasses software for process simulation (doping distributions), device simulation (electrical characteristics), device modeling, and circuit simulation. Other elements of TCAD can include lithography simulation, interconnect modeling (capacitance, resistance, and inductance), etc.

In the past, each of the software components of TCAD were used as stand-alone programs, with very little communication between the components. The trend now is to incorporate all these pieces into a complete environment that allows them to communicate with each other seamlessly, efficiently, and graphically, so that the user is not burdened with keeping track of file names, data, etc. In such a system, one can easily set up split lots and simulate from process steps up through circuit simulation. One will also be able to close the loop, feeding circuit simulation results back to the process steps and run optimization on an entire process to obtain optimal circuit performance characteristics (see Figure 8.28).

The future of device modeling will surely be pushed in this direction. Device modeling tools will be used not only separately, but within a total TCAD environment, intimately connected with process, device, interconnect, and circuit simulation tools.

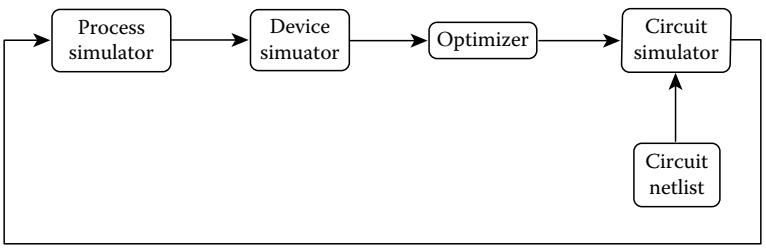


FIGURE 8.28 Block diagram of a TCAD system.

References

1. W. R. Dillon and M. Goldstein, *Multivariate Analysis Methods and Applications*, New York: John Wiley & Sons, 1984.
2. P. E. Gill, W. Murray, and M. Wright, *Practical Optimization*, Orlando, FL: Academic Press, 1981.
3. J. S. Duster, J.-C. Jeng, P. K. Ko, and C. Hu, User's guide for BSIM2 parameter extraction program and the SPICE3 with BSIM implementation, Electronic Research Laboratory, Berkeley, CA: University of California, 1988.
4. I. Getreu, *Modeling the Bipolar Transistor*, Beaverton, OR: Tektronix, 1976.
5. J.-H. Huang, Z. H. Liu, M.-C. Jeng, P. K. Ko, and C. Hu, BSIM3 manual, Berkeley, CA: University of California, 1993.
6. M.-C. Jeng, P. M. Lee, M. M. Kuo, P. K. Ko, and C. Hu, Theory, algorithms, and user's guide for BSIM and SCALP, Version 2.0, Electronic Research Laboratory, Berkeley, CA: University of California, 1987.
7. J. A. Power, A. Mathewson, and W. A. Lane, An approach for relating model parameter variabilities to process fluctuations, *Proc. IEEE Int. Conf. Microelectronic Test Struct*., vol. 6, Mar. Barcelona, Spain, 1993.
8. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge, U.K.: Cambridge University Press, 1988.
9. B. J. Sheu, D. L. Scharfetter, P. K. Ko, and M.-C. Jeng, BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors, *IEEE J. Solid-State Circuits*, SC-22(4), Aug. 1987.
10. P. J. van Wijnen, *On the Characterization and Optimization of High-Speed Silicon Bipolar Transistors*, Beaverton, OR: Cascade Microtech, Inc., 1991.
11. A. Vladimirescu and S. Liu, The simulation of MOS integrated circuits using SPICE2, memorandum no. UCB/ERL M80/7, Berkeley, CA: University of California, 1980.

Other recommended publications which are useful in device characterization are

L. W. Nagel. SPICE2: A computer program to simulate semiconductor circuits, memorandum no. ERL-M520, Berkeley, CA: University of California, 1975.

G. Massobrio and P. Antognetti, *Semiconductor Device Modeling with SPICE*, New York: McGraw-Hill, 1993.

9

Analog Circuit Simulation

9.1 Introduction **9**-1
9.2 Purpose of Simulation **9**-1
9.3 Netlists **9**-2
9.4 Formulation of the Circuit Equations **9**-3
9.5 Modified Nodal Analysis **9**-4
9.6 Active Device Models **9**-5
9.7 Types of Analysis **9**-7
DC (Steady-State) Analysis • AC Analysis • Transient Analysis
9.8 Verilog-A **9**-13
9.9 Fast Simulation Methods **9**-18
9.10 Commercially Available Simulators **9**-19
References **9**-19

J. Gregory Rollins
Technical Modeling Associates, Inc.

9.1 Introduction

Analog circuit simulation usually means simulation of analog circuits or very detailed simulation of digital circuits. The most widely known and used circuit simulation program is SPICE (simulation program with integrated circuit emphasis) of which it is estimated that there are over 100,000 copies in use. SPICE was first written at the University of California at Berkeley in 1975, and was based on the combined work of many researchers over a number of years. Research in the area of circuit simulation continues at many universities and industrial sites. Commercial versions of SPICE or related programs are available on a wide variety of computing platforms, from small personal computers to large mainframes. A list of some commercial simulator vendors can be found in the Appendix. The focus of this chapter is the simulators and the theory behind them. Examples are also given to illustrate their use.

9.2 Purpose of Simulation

Computer-aided simulation is a powerful aid for the design or analysis of VLSI circuits. Here, the main emphasis will be on analog circuits; however, the same simulation techniques may be applied to digital circuits, which are composed of analog circuits. The main limitation will be the size of these circuits because the techniques presented here provide a very detailed analysis of the circuit in question and, therefore, would be too costly in terms of computer resources to analyze a large digital system. However, some of the techniques used to analyze digital systems (like iterated timing analysis or relaxation methods) are closely related to the methods used in SPICE.

It is possible to simulate almost any type of circuit SPICE. The programs have built-in elements for resistors, capacitors, inductors, dependent and independent voltage and current sources, diodes, MOSFETs, JFETs, BJTs, transmission lines, and transformers. Commercial versions have libraries of standard components, which have all necessary parameters prefitted to typical specifications. These libraries include items such as discrete transistors, op-amps, phase-locked loops, voltage regulators, logic integrated circuits, and saturating transformer cores. Versions are also available that allow the inclusion of digital models (mixed-mode simulation) or behavioral models that allow the easy modeling of mathematical equations and relations.

Computer-aided circuit simulation is now considered an essential step in the design of modern integrated circuits. Without simulation, the number of ''trial runs'' necessary to produce a working IC would greatly increase the cost of the IC and the critical time-to-market. Simulation provides other advantages, including

- The ability to measure ''inaccessible'' voltages and currents that are buried inside a tiny chip or inside a single transistor.
- No loading problems are associated with placing a voltmeter or oscilloscope in the middle of the circuit, measuring difficult one-shot waveforms or probing a microscopic die.
- Mathematically ideal elements are available. Creating an ideal voltage or current source is trivial with a simulator, but impossible in the laboratory. In addition, all component values are exact and no parasitic elements exist.
- It is easy to change the values of components or the configuration of the circuit.

Unfortunately, computer-aided simulation has it own set of problems, including

- Real circuits are distributed systems, not the ''lumped element models'' which are assumed by simulators. Real circuits, therefore, have resistive, capacitive, and inductive parasitic elements present in addition to the intended components. In high-speed circuits, these parasitic elements can be the dominant performance-limiting elements in the circuit, and they must be painstakingly modeled.
- Suitable predefined numerical models have not yet been developed for certain types of devices or electrical phenomena. The software user may be required, therefore, to create his or her own models out of other models that are available in the simulator. (An example is the solid-state thyristor, which may be created from an npn and pnp bipolar transistor.)
- The numerical methods used may place constraints on the form of the model equations used. In addition, convergence difficulties can arise, making the simulators difficult to use.
- There are small errors associated with the solution of the equations and other errors in fitting the nonlinear models to the transistors that make up the circuit.

9.3 Netlists

Before simulating, a circuit must be coded into a netlist. Figure 9.1 shows the circuit for a simple differential pair. Circuit nodes are formed wherever two or more elements meet. This particular circuit has seven nodes, which are numbered 0–6. The ground or datum node is traditionally numbered as zero. The circuit elements (or branches) connect the nodes.

The netlist provides a description of the topography of a circuit and is simply a list of the branches (or elements) that make up the circuit. Typically, the elements may be entered in any order and each has a unique name, a list of nodes, and either a value or model identifier. For the differential amplifier of Figure 9.1, the netlist is shown in Figure 9.2.

The first line gives the title of the circuit (and is required in many simulators). The next three lines define the three voltage sources. The letter V at the beginning tells SPICE that this is a voltage source element. The list of nodes (two in this case) is next followed by the value in volts. The syntax for the resistor is similar to that of the voltage source; the starting letter R in the names of the resistors tells SPICE that these

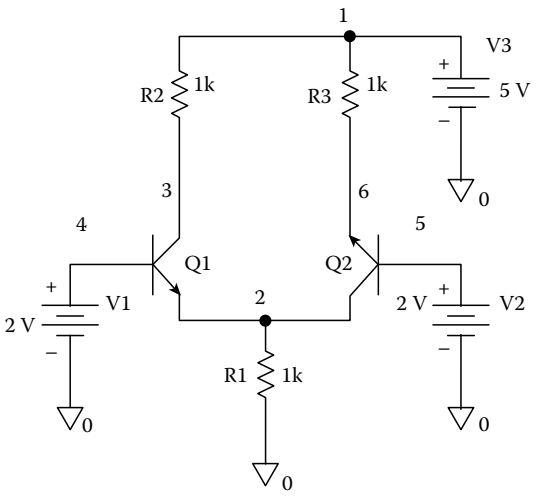


FIGURE 9.1 Circuit for differential pair.

are resistors. SPICE also understands that the abbreviation "k" after a value means 1000. For the two transistors Q1 and Q2, the starting letter Q indicates a bipolar transistor. Q1 and Q2 each have three nodes and in SPICE, the convention for their ordering is collector, base, and emitter. So, for Q1, the collector is connected to node 3, the base to node 4, and the emitter to node 2. The final entry "m2n2222" is a reference to the model for the bipolar transistor (note that both Q1 and Q2 reference the same model). The ".model" statement at the end of the listing defines this model. The model type is npn (for an npn bipolar junction transistor), and a list of "parameter = value" entries follow. These entries define the numerical values of constants in the mathematical models which are used for the

```
Differential pair circuit
V1 4 0 2V
V2 5 0 2V
V3 1 0 5V
R1 2 0 1k
R2 3 1 1K
R3 6 1 1K
Q1 3 4 2 m2n2222
Q2 6 5 2 mq2n2222
.model m2n2222 NPN IS = 1e-12 BF = 100 BR = 5 TF = 100pS
```

FIGURE 9.2 Netlist for differential pair.

bipolar transistor. (Models will be discussed later in more detail.) Most commercial circuit simulation packages come with "schematic capture" software that allows the user to draw the circuit by placing and connecting the elements with the mouse.

9.4 Formulation of the Circuit Equations

In SPICE, the circuits are represented by a system of ordinary differential equations. These equations are then solved using several different numerical techniques. The equations are constructed using Kirchhoff's voltage and current laws (KVL and KCL). The first system of equations pertains to the currents flowing into each node. One equation is written for each node in the circuit (except for the ground node), so the following equation is really a system of N equations for the N nodes in the circuit. The subscript i denotes the node index.

$$0 = F_i(V) = G_i(V) + \frac{\partial Q_i(V)}{\partial t} + W_i(t) \tag{9.1}$$

where

V is an N-dimensional vector that represents the voltages at the nodes

Q is another vector which represents the electrical charge (in Coulombs) at each node

The term W represents any independent current sources that may be attached to the nodes and has units of amperes

The function $G(V)$ represents the currents that flow into the nodes as a result of the voltages V

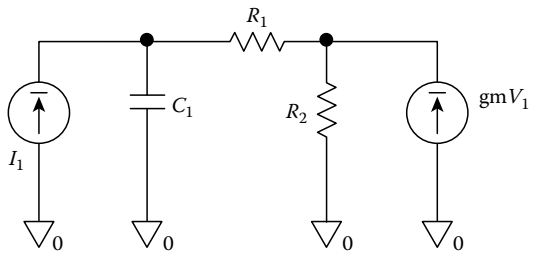


FIGURE 9.3 Example circuit for nodal analysis.

If the equations are formulated properly, a system of N equations in N unknown results.

For example, for the circuit of Figure 9.3 which has two nodes, we need to write two equations. At node 1:

$$0 = (V_1 - V_2)/R_1 + \frac{\mathrm{d}(C_1 V_1)}{\mathrm{d}t} + I_1 \qquad (9.2)$$

We can identify $G(V)$ as $(V_1 - V_2)/R$, the term $Q(V)$ is $C_1 V_1$ and $W(t)$ is simply I_1. Likewise at node 2:

$$0 = (V_2 - V_1)/R_1 + V_2/R_2 + \mathrm{gm}V_1 \qquad (9.3)$$

In this example, G and Q are simple linear terms; however, in general, they can be nonlinear functions of the voltage vector V.

9.5 Modified Nodal Analysis

Normal nodal analysis that uses only KCL cannot be used to represent ideal voltage sources or inductors. This is so because the branch current in these elements cannot be expressed as a function of the branch voltage. To resolve this problem, KVL is used to write a loop equation around each inductor or voltage source. Consider Figure 9.4 for an example of this procedure. The unknowns to be solved for are the voltage V_1 at node 1, V_2 the voltage at node 2, V_3 the voltage at node 3, the current flowing through voltage source V_1 which we shall call I_x, and the current flowing in the inductor L_1 which we shall call I_1 The system of equations is

$$\begin{aligned}
0 &= V_1/R_1 + I_x \\
0 &= V_2/R_2 - I_z + I_l \\
0 &= V_3/R_3 - I_l \\
0 &= V_1 - V_x + V_2 \\
0 &= V_2 + \frac{\mathrm{d}(L_1 I_1)}{\mathrm{d}t} - V_3
\end{aligned} \qquad (9.4)$$

The use of modified nodal analysis does have the disadvantage of requiring that an additional equation be included for each inductor or voltage source, but has the advantage that ideal voltage sources can be used.

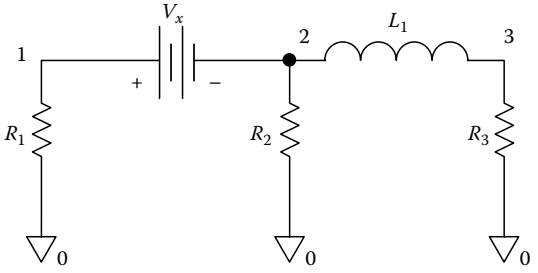


FIGURE 9.4 Circuit for modified nodal analysis.

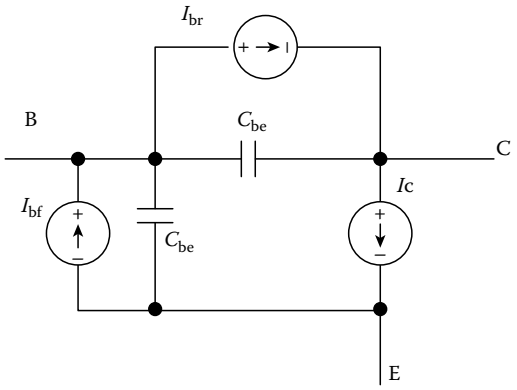


FIGURE 9.5 Ebers–Moll model for the bipolar transistor.

The total number of equations to be solved is therefore the number of nodes plus the number of voltages sources and inductors.

9.6 Active Device Models

VLSI circuits contain active devices like transistors or diodes, which act as amplifiers. These devices are normally described by a complicated set of nonlinear equations. We shall consider a simple model for the bipolar transistor—the Ebers–Moll model. This model is one of the first developed, and while it is too simple for practical application, it is useful for discussion.

A schematic of the Ebers–Moll model is shown in Figure 9.5. The model contains three nonlinear voltage-dependent current sources I_c, I_{bf}, and I_{br} and two nonlinear capacitances C_{be} and C_{bc}. The current flowing in the three current sources are given by the following equations:

$$I_c = I_s(\exp(V_{be}/V_t)) - \exp(V_{ce}/V_t) \tag{9.5}$$

$$I_{bf} = \frac{I_s}{B_f}(\exp(V_{be}/V_t) - 1) \tag{9.6}$$

$$I_{br} = \frac{I_s}{B_r}(\exp(V_{bc}/V_t) - 1) \tag{9.7}$$

The voltages V_{be} and V_{bc} are the voltages between base and emitter and the base and collector, respectively. I_s, B_f, and B_r are three user-defined parameters that govern the DC operation of the BJT. V_t is the "thermal voltage" or kT/q, which has the numerical value of approximately 0.26 V at room temperature. Observe that in the normal forward active mode, where $V_{be} > 0$ and $V_{ce} < 0$, I_{br} and the second term in I_c vanish and the current gain of the BJT, which is defined as I_c/I_b becomes numerically equal to B_f. Likewise, in the reverse mode where $V_{ce} > 0$ and $V_{be} < 0$, the reverse gain (I_e/I_b) is equal to B_r.

The two capacitances in Figure 9.5 contribute charge to the emitter, base, and collector, and this charge is given by the following equations:

$$Q_{be} = \tau_f I_s(\exp V_{be}/V_t - 1) + C_{je}\int_0^{V_{be}} (1 - V/V_{je})^{-m_e} \tag{9.8}$$

$$Q_{bc} = \tau_r I_s(\exp V_{bc}/V_t - 1) + C_{jc}\int_0^{V_{bc}} (1 - V/V_{jc})^{-m_c} \tag{9.9}$$

Q_{be} contributes positive charge to the base and negative charge to the emitter. Q_{bc} contributes positive charge to the base and negative charge to the collector. The first term in each charge expression is due to charge injected into the base from the emitter for Q_{be} and from the collector into the base for Q_{bc}. Observe that the exponential terms in the charge terms are identical to the term in I_c. This is so because the injected charge is proportional to the current flowing into the transistor. The terms τ_f and τ_r are the forward and reverse transit times, respectively, and correspond to the amount of time it takes the electrons (or holes) to cross the base. The second term in the charge expression (the term with the integral)

corresponds to the charge in the depletion region of the base–emitter junction for Q_{be} and in the base–collector junction for Q_{bc}. Recall that the depletion width in a pn junction is a function of the applied voltage. The terms V_{je} and V_{jc} are the "built-in" potentials with units of volts for the base–emitter and base–collector junctions. The terms m_c and m_e are the grading coefficients for the two junctions and are related to how rapidly the material changes from n-type to p-type across the junction.

This "simple" model has eleven constants I_s, B_f, B_r, C_{je}, C_{jc}, M_e, M_c, V_{je}, V_{jc}, T_f, and T_r that must be specified by the user. Typically, these constants would be extracted from measured I–V and C–V data taken from real transistors using a fitting or optimization procedure (typically a nonlinear least-squares fitting method is needed). The Ebers–Moll model has a number of shortcomings that are addressed in newer models like Gummel–Poon, Mextram, and VBIC. The Gummel–Poon model has over 40 parameters that must be adjusted to get a good fit to data in all regions of operation.

Models for MOS devices are even more complicated than the bipolar models. Modeling the MOSFET is more difficult than the bipolar transistor because it is often necessary to use a different equation for each of the four regions of operation (off, subthreshold, linear, and saturation) and the drain current and capacitance are functions of three voltages (V_{ds}, V_{bs}, and V_{gs}) rather than just two (V_{be} and V_{ce}) as in the case of the BJT. If a Newton–Raphson solver is to be used, the I–V characteristics and capacitances must be continuous and it is best if their first derivatives are continuous as well. Furthermore, MOS models contain the width (W) and length (L) of the MOSFET channel as parameters; and for the best utility the model should remain accurate for many values of W and L. This property is referred to as "scalability."

Over the years, literally hundreds of different MOS models have been developed. However, for modern VLSI devices, only three or four are commonly used today. These are the SPICE Level-3 MOS model, the HSPICE Level-28 model (which is a proprietary model developed by Meta Software), the public domain BSIM3 model developed at UC, Berkeley, and MOS9 developed at Phillips. These models are supported by many of the "silicon foundries," that is, parameters for the models are provided to chip designers by the foundries. BSIM3 has been observed to provide a good fit to measured data and its I–V curves to be smooth and continuous (thereby resulting in good simulator convergence). The main drawback of BSIM3 is that it has over 100 parameters which are related in intricate ways, making extraction of the parameter set a difficult process.

A process known as "binning" is used to provide greater accuracy. When binning is used, a different set of model parameters is used for each range of the channel length and width (L and W). An example of this is shown in Figure 9.6. For a given type of MOSFET, 12 complete sets of model parameters are extracted and each is valid for a given range. For example, in Figure 9.6, the set represented by the number "11" would only be valid for channel lengths between 0.8 and 2.0 μm and for channel widths between 0.5 and 0.8 μm. Thus, for a typical BSIM3 model with about 60 parameters, $12 \times 60 = 720$ parameters would need to be extracted in all and this just for one type of device.

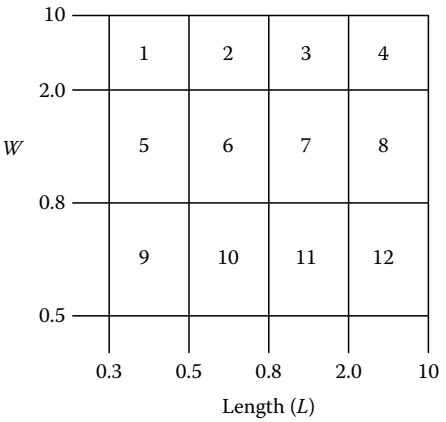


FIGURE 9.6 Binning of MOS parameters.

Many commercial simulators contain other types of models besides the traditional R, L, C, MOS, and BJT devices. Some simulators contain "behavioral" models which are useful for systems design or integration tasks; examples of these are integrators, multipliers, summation, and Laplace operator blocks. Some simulators are provided with libraries of prefitted models for commercially available operational amplifiers, logic chips, and discrete devices. Some programs allow "mixed-mode" simulation, which is a combination of logic simulation (which normally allows only a few discrete voltage states) and analog circuit simulation.

9.7 Types of Analysis

For analog circuits, three methods of analysis are commonly used: DC, AC, and transient analysis. DC analysis is used to examine the steady-state operation of a circuit; that is, what the circuit voltages and currents would be if all inputs were held constant for an infinite time. AC analysis (or sinusoidal steady state) examines circuit performance in the frequency domain using phasor analysis. Transient analysis is performed in the time domain and is the most powerful and computationally intensive of the three. For special applications, other methods of analysis are available such as the harmonic-balance method, which is useful for detailed analysis of nonlinear effects in circuits excited by purely periodic signals (like mixers and RF amplifiers).

9.7.1 DC (Steady-State) Analysis

DC analysis calculates the steady-state response of a circuit (with all inductors shorted and capacitors removed). DC analysis is used to determine the operating point (*Q*-point) of a circuit, power consumption, regulation and output voltage of power supplies, transfer functions, noise margin and fanout in logic gates, and many other types of analysis. In addition, a DC solution must be calculated to find the starting point for AC and transient analysis.

To calculate the DC solution, we need to solve Kirchhoff's equations formulated earlier. Unfortunately, since the circuit elements will be nonlinear in most cases, a system of transcendental equations will normally result and it is impossible to solve this system analytically. The method that has met with the most success is Newton's method or one of its derivatives.

9.7.1.1 Newton's Method

Newton's method is actually quite simple. We need is to solve the system of equations $F(X) = 0$ for X, where both F and X are vectors of dimension N. (F is the system of equations from modified nodal analysis, and X is the vector of voltages and current that we are solving for.) Newton's method states that given an initial guess for X^i, we can obtain a better guess X^{i+1} from the equation:

$$X^{i+1} = X^i - [J(X^i)]^{-1} F(X^i) \tag{9.10}$$

Note that all terms on the right side of the equation are functions only of the vector X^i. The term $J(X)$ is a $N \times N$ square matrix of partial derivatives of F, called the Jacobian. Each term in J is given by

$$J_{i,j} = \frac{\partial F_i(X)}{\partial X_j} \tag{9.11}$$

We assemble the Jacobian matrix for the circuit at the same time that we assemble the circuit equations. Analytic derivatives are used in most simulators.

The -1 in Equation 9.10 indicates that we need to invert the Jacobian matrix before multiplying by the vector F. Of course, we do not need to actually invert J to solve the problem; we only need to solve the linear problem $F = YJ$ for the vector Y and then calculate $X^{i+1} = X^i - Y$. A direct method such as the LU decomposition is usually employed to solve the linear system.

For the small circuit of Figure 9.3, analyzed in steady state (without the capacitor), the Jacobian entries are

$$\begin{array}{ll} J_{1,1} = 1/R_1 & J_{1,2} = -1/R_1 \\ J_{2,1} = 1/R_1 + gm & J_{2,2} = 1/R_1 + 1/R_2 \end{array} \tag{9.12}$$

```
Stead state analysis of differential pair.
V1 4 0 2V
V2 5 0 2V
V3 1 0 5V
R1 2 0 1k
R2 3 1 1K
R3 6 1 1K
Q1 3 4 2 m2n2222
Q2 6 5 2 m2n2222
.model m2n2222 NPN IS = 1e-12 BF = 100 BR = 5 TF = 100pS
.dc V1  1.0  3.0  0.01
```

FIGURE 9.7 Input file for DC sweep of V_1.

For a passive circuit (i.e., a circuit without gain), the Jacobian will be symmetric and for any row, the diagonal entry will be greater than the sum of all the other entries.

Newton's method converges quadratically, provided that the initial guess X^i is sufficiently close to the true solution. Quadratically implies that if the distance between X^i and the true solution is d, then the distance between X^{i+1} and the true solution will be d^2. Of course, we are assuming that d is small to start with. Still, programs like SPICE may require 50 or more iterations to achieve convergence. The reason for this is that, often times, the initial guess is poor and quadratic convergence is not obtained until the last few iterations. There are additional complications like the fact that the model equations can become invalid for certain voltages. For example, the BJT model will "explode" if a junction is forward-biased by more than 1 V or so since: $\exp(1/Vt) = 5e16$. Special limiting or damping methods must be used to keep the voltages and currents to within reasonable limits.

9.7.1.2 Example Simulation

Most circuit simulators allow the user to ramp one or more voltage sources and plot the voltage at any node or the current in certain branches. Returning to the differential pair of Figure 9.1, we can perform a DC analysis by simply adding a .dc statement (see Figure 9.7). A plot of the differential output voltage (between the two collectors) and the voltage at the two emitters is shown in Figure 9.8. Observe that the output voltage is zero when the differential pair is "balanced" with 2.0 V on both inputs. The output saturates at both high and low values for V_1, illustrating the nonlinear nature of the analysis. This simulation was run using the PSPICE package from MicroSim Corporation. The simulation run is a few seconds on a 486 type PC.

9.7.2 AC Analysis

AC analysis is performed in the frequency domain under the assumption that all signals are represented as a DC component V_{dc} plus a small sinusoidal component V_{ac}

$$V = V_{dc} + V_{ac} \exp(j\omega t) \tag{9.13}$$

where

$j = \sqrt{-1}$

ω is the radial frequency $(2\pi f)$

V_{ac} is a complex number

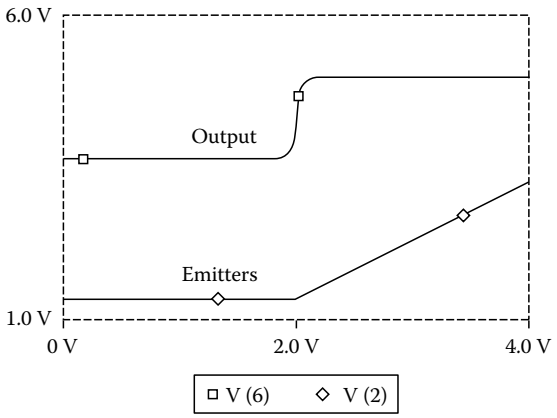


FIGURE 9.8 Output from DC analysis.

Expanding Equation 9.1 about the DC bias point V_{dc} (also referred to as the Q point), we obtain

$$F(V) = F(V_{dc}) + W_{dc} + W_{ac} + \frac{\partial G(V_{dc})}{\partial V_{dc}} V_{ac} + \frac{\partial}{\partial t}\left(\frac{\partial Q(V_{dc})}{\partial V_{dc}}\right) V_{ac} + \alpha V_{ac}^2 \Lambda \tag{9.14}$$

The series has an infinite number of terms; however, we assume that if V_{ac} is sufficiently small, all terms above first order can be neglected. The first two terms on the right-hand side are the DC solution and, when taken together, yield zero. The third term W_{ac} is the vector of independent AC sources which drive the circuit. The partial derivative in the fourth term is the Jacobian element, and the derivative of Q in parentheses is the capacitance at the node. When we substitute the exponential into Equation 9.14, each term will have an exponential term that can be canceled. The result of all these simplifications is the familiar result:

$$0 = W_{ac} + JV_{ac} + j\omega CV_{ac} \tag{9.15}$$

This equation contains only linear terms which are equal to the partial derivatives of the original problem evaluated at the Q point. Therefore, before we can solve the AC problem, we must calculate the DC bias point. Rearranging terms slightly, we obtain

$$V_{ac} = -(J + j\omega C)^{-1} W_{ac} \tag{9.16}$$

The solution at a given frequency can be obtained from a single matrix inversion. The matrix, however, is complex but normally the complex terms share a sparsity pattern similar to the real terms. It is normally possible (in FORTRAN and C++) to create a suitable linear solver by taking the linear solver which is used to calculate the DC solution and substituting ''complex'' variables for ''real'' variables. Since there is no nonlinear iteration, there are no convergence problems and AC analysis is straightforward and foolproof.

The same type of analysis can be applied to the equations for modified nodal analysis. The unknowns will of course be currents and the driving sources voltage sources.

$$I_{ac} = -(J + j\omega L)^{-1} E_{ac} \tag{9.17}$$

The only things that must be remembered with AC analysis are

1. AC solution is sensitive to the Q point, so if an amplifier is biased near its saturated DC output level, the AC gain will be smaller than if the amplifier were biased near the center of its range.
2. This is a linear analysis and therefore ''clipping'' and slew rate effects are not modeled. For example, if a 1 V AC signal is applied to the input of a small signal amplifier with a gain of 100 and a power supply voltage of 5 V, AC analysis will predict an output voltage of 100 V. This is of course impossible since the output voltage cannot exceed the power supply voltage of 5 V. If you want to include these effects, use transient analysis.

9.7.2.1 AC Analysis Example

In the following example, we will analyze the differential pair using AC analysis to determine its frequency response. To perform this analysis in SPICE, we need to only specify which sources are the AC-driving sources (by adding the magnitude of the AC signal at the end) and specify the frequency range on the .AC statement (see Figure 9.9). SPICE lets the user specify the range as linear or ''decade,'' indicating that we desire a logarithmic frequency scale. The first number is the number of frequency points per decade. The second number is the starting frequency, and the third number is the ending frequency.

```
AC analysis of differential pair.
V1 4 0 2V    AC 1
V2 5 0 2V
V3 1 0 5V
R1 2 0 1k
R2 3 1 1K
R3 6 1 1K
Q1 3 4 2 m2n2222
Q2 6 5 2 m2n2222
.model m2n2222 NPN IS = 1e-12 BF = 100 BR = 5 TF = 100pS
.AC DEC  10 1e3 1e9
```

FIGURE 9.9 Input file for AC analysis.

Figure 9.10 shows the results of the analysis. The gain begins to roll off at about 30 MHz due to the parasitic capacitances within the transistor models. The input impedance (which is plotted in kΩ) begins to roll off at a much lower frequency. The reduction in input impedance is due to the increasing current that flows in the base–emitter capacitance as the current increases. SPICE does not have a method of calculating input impedance, so we have calculated it as $Z = V_{\text{in}}/I(V_{\text{in}})$, where $V_{\text{in}} = 1.0$, using the postprocessing capability of PSPICE. This analysis took about 2 s on a 486-type PC.

9.7.2.2 Noise Analysis

Noise is a problem primarily in circuits that are designed for the amplification of small signals like the RF and IF amplifiers of a receiver. Noise is the result of random fluctuations in the currents which flow in the circuit and is generated in every circuit element. In circuit simulation, noise analysis, is an extension of AC analysis. During noise analysis, it is assumed that every circuit element contributes some small noise component either as a voltage V_{n} in series with the element or as a current I_{n} across the element. Since the noise sources are small in comparison to the DC signal levels, AC small signal analysis is an applicable analysis method.

Different models have been developed for the noise sources. In a resistor, thermal noise is the most important component. Thermal noise is due to the random motion of the electrons:

$$I_n^2 = \frac{4kT\Delta f}{R} \tag{9.18}$$

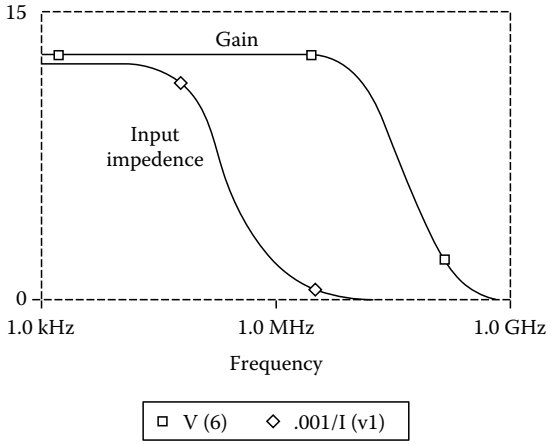


FIGURE 9.10 Gain and input impedance calculated by AC analysis.

where

T is the temperature
k is Boltzman's constant
Δf is the bandwidth of the circuit

In a semiconductor diode, shot noise is important. Shot noise is related to the probability that an electron will surmount the semiconductor barrier energy and be transported across the junction:

$$I_n^2 = 2qI_d\Delta f \tag{9.19}$$

Other types of noise occur in diodes and transistors; examples are flicker and popcorn noise. Noise sources, in general, are frequency-dependent.

Noise signals will be amplified or attenuated as they pass through different circuits. Normally, noise is referenced to some output point called the "summing node." This would normally be the output of the amplifier where we would actually measure the noise. We can call the gain between the summing node and the current flowing in an element j in the circuit $A_j(f)$. Here, f is the analysis frequency since the gain will normally be frequency dependent.

Noise signals are random and uncorrelated to each other so their magnitudes must be root-mean-squared summed rather than simply summed. Summing all noise sources in a circuit yields:

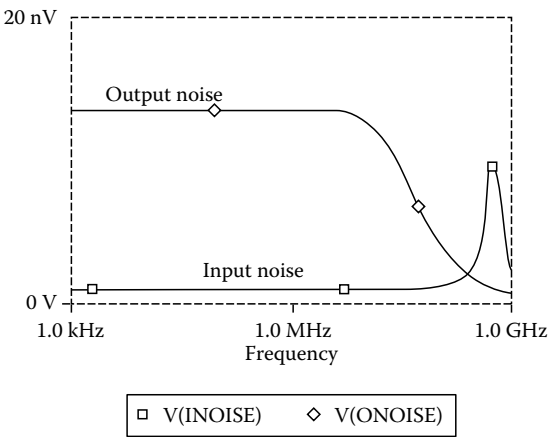


FIGURE 9.11 Noise referenced to output and input.

$$I_n(f) = \sqrt{\sum_j A_j^2(f) I_j^2(f)} \tag{9.20}$$

It is also common to reference noise back to the amplifier input and this is easily calculated by dividing the preceding expression by the amplifier gain. Specifying noise analysis in SPICE is simple. All the user needs to do is add a statement specifying the summing node and the input source. SPICE then calculates the noise at each as a function of frequency

```
.noise  v([6])  V1
```
(9.21)

See Figure 9.11 for example output. Many circuit simulators will also list the noise contributions of each element as part of the output. This is particularly helpful in locating the source of noise problems.

9.7.3 Transient Analysis

Transient analysis is the most powerful analysis capability because the transient response of a circuit is so difficult to calculate analytically. Transient analysis can be used for many types of analysis, such as switching speed, distortion, and checking the operation of circuits such as logic gates, oscillators, phase-locked loops, or switching power supplies. Transient analysis is also the most CPU intensive and can require 100 or 1000 times the CPU time of DC or AC analysis.

9.7.3.1 Numerical Method

In transient analysis, time is discretized into intervals called time steps. Typically, the time steps are of unequal length, with the smallest steps being taken during intervals where the circuit voltages and currents are changing most rapidly. The following procedure is used to discretize the time-dependent terms in Equation 9.1.

Time derivatives are replaced by difference operators, the simplest of which is the forward difference operator:

$$\frac{dQ(t_k)}{dt} = \frac{Q(t_{k+1}) - Q(t_k)}{h} \tag{9.22}$$

where h is the time step given by $h = t_{k+1} - t_k$. We can easily solve for the charge $Q(t_{k+1})$ at the next time point:

$$Q(t_{k+1}) = Q(t_k) - h(G_i(V(t_k)) + W_i(t_k)) \tag{9.23}$$

using only values from past time points. This means that it would be possible to solve the system simply by plugging in the updated values for V each time. This can be done without any matrix assembly or inversion and is very nice. (Note for simple linear capacitors, $V = Q/C$ at each node, so it is easy to get V back from Q.) However, this approach is undesirable for circuit simulation for two reasons: (1) The charge Q, which is a "state variable" of the system, is not a convenient choice since some nodes may not have capacitors (or inductors) attached, in which case they will not have Q values and (2) It turns out that forward (or explicit) time discretization methods like this one are unstable for "stiff" systems, and most circuit problems result in "stiff systems." The term "stiff system" refers to a system that has greatly varying time constants.

To overcome the stiffness problem, we must use implicit time discretization methods which, in essence, mean that the G and W terms in the above equations must be evaluated at t_{k+1}. Since G is nonlinear, we will need to use Newton's method once again.

The most popular implicit method is the trapezoidal method. The trapezoidal method has the advantage of requiring information only from one past time point and, furthermore, has the smallest error of any method requiring one past time point. The trapezoidal method states that if I is the current in a capacitor, then

$$I(t_{k+1}) = \frac{dQ}{dt} = 2\frac{Q(V(t_{k+1})) - Q(V(t_k))}{h} - I(t_k) \tag{9.24}$$

Therefore, we need to only substitute Equation 9.24 into Equation 9.1 to solve the transient problem. Observe that we are solving for the voltages $V(t_{k+1})$, and all terms involving t_k are constant and will not be included in the Jacobian matrix. An equivalent electrical model for the capacitor is shown in Figure 9.12. Therefore, the solution of the transient problem is in effect a series of DC solutions where the values of some of the elements depend on voltages from the previous time points.

All modern circuit simulators feature automatic time step control. This feature selects small time steps during intervals where changes occur rapidly and large time steps in intervals where there is little change. The most commonly used method of time step selection is based on the local truncation error (LTE) for each time step. For the trapezoidal rule, the LTE is given by

$$\varepsilon = \frac{h^3}{12}\frac{d^3x}{dt^3}(\xi) \tag{9.25}$$

and represents the maximum error introduced by the trapezoidal method at each time step. If the error (ε) is larger than some preset value, the step size is reduced. If the error is smaller, then the step size is increased. In addition, most simulators select time points so that they coincide with the edges of pulse-type waveforms.

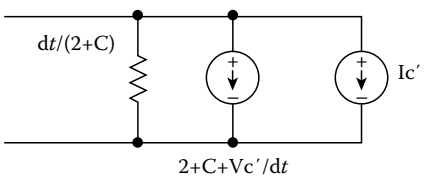


FIGURE 9.12 Electrical model for a capacitor; the two current sources are independent sources. The prime (′) indicates values from a preceding time point.

9.7.3.2 Transient Analysis Examples

As a simple example, we return to the differential pair and apply a sine wave differentially to the input. The amplitude (2 V p-p) is selected to drive the amplifier into saturation.

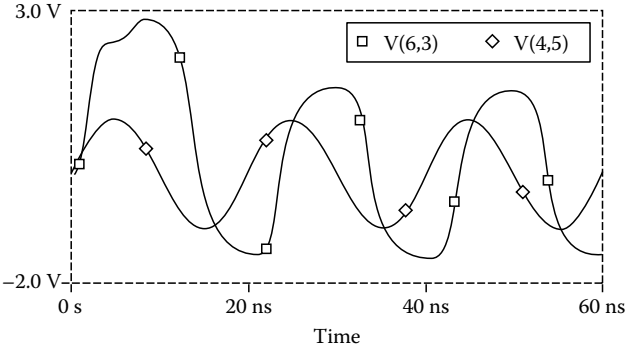


FIGURE 9.13 Transient response V(6,3) of differential amplifier to sinusoidal input at V(4,5).

In addition, we make the frequency (50 MHz) high enough to see phase shift effects. The output signal is therefore clipped due to the nonlinearities and shifted in phase due to the capacitive elements in the transistor models (see Figure 9.13). The first cycle shows extra distortion since it takes time for the ''zero-state'' response to die out. This simulation, using PSPICE, runs in about one second on a 486-type computer.

9.8 Verilog-A

Verilog-A is a new language designed for simulation of analog circuits at various levels. Mathematical equations can be entered directly as well as normal SPICE-type circuit elements.

Groups of equations and elements can be combined into reusable ''modules'' that are similar to subcircuits. Special functions are also provided for converting analog signals into digital equivalents, and vice versa. Systems-type elements such as Laplace operators, integrators, and differentiators are also provided. This makes it possible to perform new types of modeling that were not possible in simulators like SPICE:

- Equations can be used to construct new models for electrical devices (for example, the Ebers–Moll model described earlier could be easily implemented).
- Behavioral models for complex circuits like op-amps, comparators, phase detectors, etc. can be constructed. These models can capture the key behavior of a circuit and yet be simulated in a small fraction of the time it takes to simulate at the circuit level.
- Special interface elements make it possible to connect an analog block to a digital simulator, making mixed-mode simulation possible. Verilog-A is related to and compatible with the popular Verilog-D modeling language for digital circuits.

As an example, consider a phase-locked loop circuit which is designed as an 50X frequency multiplier. A block diagram for the PLL is shown in Figure 9.14 and the Verilog-A input listing is shown in Figures 9.15 and 9.16.

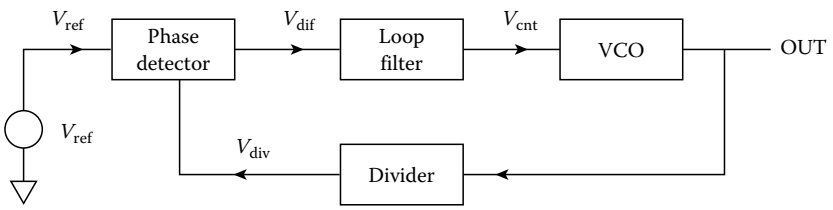


FIGURE 9.14 Block diagram of phase-locked loop.

```
'include "electrical.h"
'timescale 100ns/10ns

module top;                                  // Top level Module
        electrical Vdd, Vref, Vdif, Vcnt, Vout, Vdiv;
        pd #(.gain(15u), .rout (2e6), dir(–l)) p1(Vdif, Vdiv, Vref, Vdd);
        filter f1 (Vdif, Vcnt);
        vco #(.gain(2e7), .center(3e7)) v1 (Vout, Vcntm Vdd);
        divide #(.count(50)) d1 (Vdiv, Vout, Vdd);
        analog begin
                V(Vdd) <+ 5.0;
                V(Vref) <+ 2.5+2.5*sin(6.238*1e6*$realtime);
        end
endmodule

module pd (out, vco, ref, vdd);          //Phase detector Module.
        inout out, vco, ref, vdd;
        electrical out,vco, ref, vdd;
        parameter real gain = 15u, rout = 5e6;          // gain & output impedance
        parameter integer dir = 1;                      // 1,–1 pos or neg edge trigger
        integer state;
        analog begin
                @ (cross((V(ref) – V(vdd)/2),dir)) state = state – 1;
                @ (cross((V(vco) – V(vdd)/2),dir)) state = state + 1;
                if (state > 1 ) state = 1;
                if (state < –1) state = –1;
                if (state ! = 0)   I(out) <+ transition (state * gain, 0, 0, 0);
              I(out) <+ V(out)/rout;
        end
endmodule
```

FIGURE 9.15 Part one of Verilog-A listing for PLL.

Simulation of this system at the circuit level is very time consuming due to the extreme difference in frequencies. The phase detector operates at a low frequency of 1.0 MHz, while the VCO operates at close to 50 MHz. However, we need to simulate enough complete cycles at the phase detector output to verify that the circuit correctly locks onto the reference signal.

The circuit is broken up into five blocks or modules: the ''top module,'' VCO, divider, phase detector, and loop filter. The VCO has a simple linear dependence of frequency on the VCO input voltage and produces a sinusoidal output voltage. The VCO frequency is calculated by the simple expression freq = center + gain* ($V_{in} - V_{min}$). Center and gain are parameters which can be passed in when the VCO is created within the top module by the special syntax ``#(.gain(2e7),.center(3e7)'' in the top module. If the parameters are not specified when the module is created, then the default values specified within the module are used instead. The special V() operator is used to obtain the voltage at a node (in this case `V(in)` and `V(Vdd)`). The sinusoidal output is created using the SIN and IDT operators. SIN calculates the sine of its argument. I_{dt} calculates the integral of its argument with respect to time. The amplitude of the output is taken from the V_{dd} input, thus making it easy to integrate the VCO block with others. Given that $V_{dd} = 5$ V, gain = 2e7 Hz/V, center = 3e7 Hz, and in = 1.8, the final expression for the VCO output is

$$V_{out} = 2.5 + 2.5\sin\left(3e7 + 2e7\int 2\pi(V_{in} - 1.8)dt\right) \tag{9.26}$$

The phase detector functions as a charge pump which drives current into or out of the loop filter, depending on the phase difference between its two inputs. The `@cross(V1,dir)` function becomes

```
module divide (out, in, vdd);      // Divider module....
        inout out, in;
        electrical out, in;
        parameter real count = 4; // divide by this.
    integer n, state;
        analog begin

                @(cross((V(in) – V(vdd)/2.0),1)) n = n + 1;
                if (n >= count/2) begin
                        if (state == 0) state = 1;
                        else state = 0;
                        n = 0;
                end
                V(out) <+ transition(state*5, 0, 0, 0);
        end
endmodule

module vco (vout, vin, vdd);       // VCO module
        inout vin, vout, vdd;
        electrical vin, vout, vdd;
        parameter real gain = 5e6, center = 40e6, vmin = 1.8;
        real freq, vinp;
        analog begin
                vinp = V(vin);
                if (vinp < vm) vinp = vmin;
                freq = center + gain* (vinp-vmin);
                V(vout) <+ V(vdd)/2.0*sin(6.28318531*idt(freq,0)) + V(vdd)/2.0;
        end
endmodule

'language SPICE
.SUBCKT filter Vin Vout
Rf2 Vin Vout 100
Rfilter Vin VC2 200000
C1 VC2 0 58p
C2 Vin 0 5p
.ends
'endlanguage
```

FIGURE 9.16 Part two of Verilog-A PLL listing.

true whenever signal V_1 crosses zero in the direction specified by `dir`. This either increments or decrements the variable STATE. The ''transition'' function is used to convert the STATE signal, which is essentially digital and changes abruptly into a smoothly changing analog signal which can be applied to the rest of the circuit. The ''<+'' (or contribution) operator adds the current specified by the equation on the right to the node on the left. Therefore, the phase detector block forces current into the output node whenever the VCO signal leads the reference signal and forcing current out of the output node whenever the reference leads the VCO signal. The phase detector also has an output resistance which is specified by parameter ROUT.

The loop filter is a simple SPICE subcircuit composed of two resistors and one capacitor. Of course, this subcircuit could contain other types of elements as well and can even contain other Verilog-A modules. The divider block simply counts zero crossings and, when the count reaches the preset divisor, the output of the divider is toggled from 0 to 1, or vice versa. The transition function is used to ensure that a smooth, continuous analog output is generated by the divider.

This PLL was simulated using AMS from Antrim Design Systems. The results of the simulations are shown in Figure 9.17. The top of Figure 9.17 shows the output from the loop filter (V_{cnt}). After a few cycles, the PLL has locked onto the reference signal. The DC value of the loop filter output is

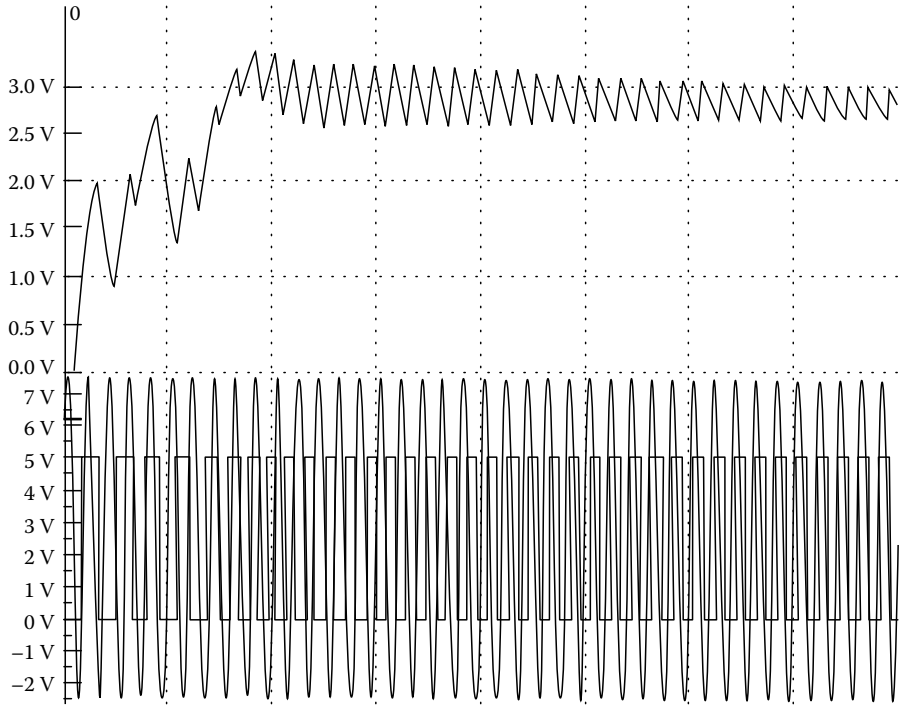


FIGURE 9.17 Loop filter output (top) V_{ref} and divider output (bottom) from PLL simulation.

approximately 2.8 V. Referring back to the VCO model, this gives an output frequency of 2e7* $(2.8 - 1.8) + 3e7 = 50$ MHz, which is as expected. The lower portion of Figure 9.17 shows the divider output (V_{div}) and the reference signal (V_{ref}). It can be seen that the two signals are locked in phase. Figure 9.18 shows the VCO output and the divider output. As expected, the VCO frequency is 50 times the divider frequency.

The behavioral models used in this example are extremely simple ones. Typically, more complex models must be used to accurately simulate the operation of an actual PLL. A better model might include effects such as the nonlinear dependence of the VCO frequency on the input voltage, the effects on signals introduced through power supply lines, delays in the divider and phase detector, and finite signal rise and fall times. These models can be built up from measurements, or transistor-level simulation of the underlying blocks (a process known as characterization). Of course, during the simulation, any of the behavioral blocks could be replaced by detailed transistor-level models or complex Verilog-D digital models.

Another Verilog-A example is shown in Figure 9.19. Here, the Ebers–Moll model developed earlier is implemented as a module. This module can then be used in a circuit in the same way as the normal built-in models. Verilog-A takes care of calculating all the derivatives needed to form the Jacobian matrix. The ''parameter'' entries can be used in the same way as the parameters on a SPICE.MODEL statement. Observe the special ''ddt'' operator. This operator is used to take the time derivative of its argument. In this case, the time derivative of the charge (a current) is calculated and summed in with the other DC components. The ''$limexp'' operation is a special limited exponential operator designed to give better convergence when modeling pn junctions. Of course, this module could be expanded and additional features could be added.

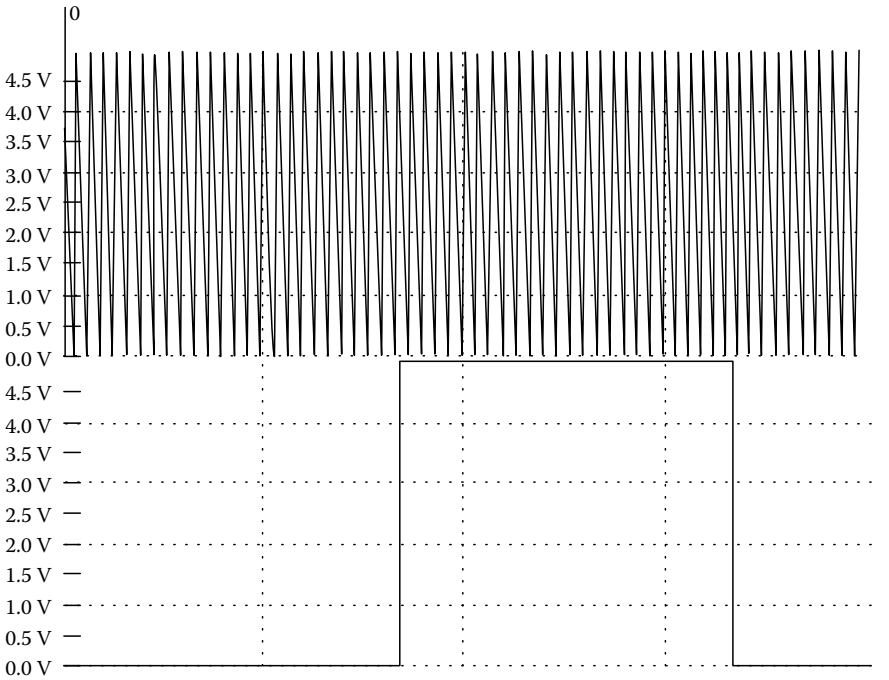


FIGURE 9.18 VCO output and divider output from PLL simulation.

```
module mybjt (C,B,E);
        inout C,B,E;
        electrical C,B,E;
        parameter real is = le–16, bf = 100, br = 10, tf = 1n, tr = l0n, cje = 1p cjc = lp,
                vje = 0.75, vjc = 0.75, mje = 0.33 mjc = 0.33;
        real ibc, ibe, qbc, qbe, vbe, vbc, cc, cb, x, y;
        analog begin
                vbe = V(B,E);
                vbc = V(B,C);
                ibe = is*($limexp(vbe/$vt)–1);
                ibc = is*($limexp(vbc/$vt)–1);
                cb = ibe/bf + ibc/br;
                cc = ibe – ibc – ibc/br
                if(vbe < 0) begin
                        x = 1–vbe/vje;
                        y = exp(–mje*ln(x));
                        qbe = vje*cje*(1–x*y)/(l.0–mje)+tf*ibe;
                end
                else qbe = cje*(vbe*+0.5*mje*vbe*vbe/vje) + tf*ibe;
                if(vbc < 0) begin
                        x = 1 –vbc/vjc;
                        y = exp(–mjc*ln(x));
                        qbc = vjc*cjc*(l–x*y)/(1.0–mjc) + tr*ibc;
                end
                else qbc = cjc*(vbc+0.5*mjc*vbc*vbc/vjc) + tr*ibc;
                ibe = ddt(qbe);
                ibc = ddt(qbc);
                I(C,E) <+ cc – ibc;
                I(B,E) <+ cb + ibe + ibc;
        end
endmodule
```

FIGURE 9.19 Verilog-A implementation of the Ebers–Moll model.

9.9 Fast Simulation Methods

As circuits get larger, simulation times become larger. In addition, as integrated circuit feature sizes shrink, second-order effects become more important and many circuit designers would like to be able to simulate large digital systems at the transistor level (requiring 10,000 to 100,000 nodes). Numerical studies in early versions of SPICE showed that the linear solution time could be reduced to 26% for relatively small circuits with careful coding. The remainder is used during the assembly of the matrix, primarily for model evaluation. The same studies found that the CPU time for the matrix solution was proportional to $n^{1.24}$, where n is the number of nodes. The matrix assembly time on the other hand should increase linearly with node count. Circuits have since grown much bigger, but the models (particularly for MOS devices) have also become more complicated.

Matrix assembly time can be reduced by a number of methods. One method is to simplify the models; however, accuracy will be lost as well. A better way is to precompute the charge and current characteristics for the complicated models and store them into tables. During simulation, the actual current and charges can be found from table lookup and interpolation, which can be done quickly and efficiently. However, there are some problems:

1. To assure convergence of Newton's method, both the charge and current functions and their derivatives must be continuous. This rules out most simple interpolation schemes and means that something like a cubic spline must be used.
2. The tables can become large. A MOS device has four terminals, which means that all tables will be functions of three independent variables. In addition, the MOSFET requires four separate tables (I_d, Q_g, Q_d, Q_b). If we are lucky, we can account for simple parameteric variations (like channel width) by a simple multiplying factor. However, if there are more complex dependencies as is the case with channel length, oxide thickness, temperature, or device type, we will need one complete set of tables for each device.

If the voltages applied to an element do not change from the past iteration to the present iteration, then there is no need to recompute the element currents, charges, and their derivatives. This method is referred to as taking advantage of latency and can result in large CPU time savings in logic circuits, particularly if coupled with a method that only refractors part of the Jacobian matrix. The tricky part is knowing when the changes in voltage can be ignored. Consider, for example, the input to a high-gain op-amp, here ignoring a microvolt change at the input could result in a large error at the output. Use of sophisticated latency-determining methods could also cut into the savings.

Another set of methods are the waveform relaxation techniques which increase efficiency by temporarily ignoring couplings between nodes. The simplest version of the method is as follows. Consider a circuit with n nodes which requires m time points for its solution. The circuit can be represented by the vector equation:

$$F_i(V(t)) + \frac{dQ_i(V(t))}{dt} = 0 \tag{9.27}$$

Using trapezoidal time integration gives a new function:

$$W_i(V(k)) = F_i(V(k)) + F_i(V(k-1)) + 2[Q_i(V(k)) - Q_i(V(k-1))]/dt = 0 \tag{9.28}$$

We need to find the $V(k)$ that makes W zero for all k time points at all i nodes. The normal method solves for all n nodes simultaneously at each time point before advancing k. Waveform relaxation solves for all m time points at a single node (calculates the waveform at that node) before advancing to the next node. An outer loop is used to assure that all the individual nodal waveforms are consistent with each other.

Waveform relaxation is extremely efficient as long as the number of outer loops is small. The number of iterations will be small if the equations are solved in the correct order; that is, starting on nodes which are signal sources and following the direction of signal propagation through the circuit. This way, the waveform at node $i+1$ will depend strongly on the waveform at node i, but the waveform at node i will depend weakly on the signal at node $i+1$. The method is particularly effective if signal propagation is unidirectional, as is sometimes the case in logic circuits. During practical implementation, the total simulation interval is divided into several subintervals and the subintervals are solved sequentially. This reduces the total number of time points which must be stored. Variants of the method solve small numbers of tightly coupled nodes as a group; such a group might include all the nodes in a TTL gate or in a small feedback loop. Large feedback loops can be handled by making the simulation time for each subinterval less than the time required for a signal to propagate around the loop.

The efficiency of this method can be further improved using different time steps at different nodes, yielding a multirate method. This way, during a given interval, small time steps are used at active nodes while large steps are used at inactive nodes (taking advantage of latency).

9.10 Commercially Available Simulators

The simulations in this chapter were performed with the evaluation version of PSPICE from Microsim and AMS from Antrim design systems. The following vendors market circuit simulation software. The different programs have strengths in different areas and most vendors allow you to try their software in-house for an ''evaluation period'' before you buy.

SPICE2-SPICE3	University of California, Berkeley, CA
AMS	Antrim Design Systems, Scotts Valley, CA, www.antrim.com
PSPICE	Orcad Corporation, Irvine, CA, www.orcad.com
HSPICE	Avant! Corporation, Fremont, CA, www.avanticorp.com
ISPICE	Intusoft, SanPedro, CA, www.intusoft.com
SABER	Analogy, Beaverton, OR, www.analogy.com
SPECTRE	Cadence Design Systems, San Jose, CA, www.cadence.com
TIMEMILL	Synopsys Corporation, Sunnyvale, CA, www.synopsys.com
ACCUSIM II	Mentor Graphics, Wilsonville, OR, www.mentorg.com

References

On general circuit simulation:

1. J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, New York, 1983.
2. A. E. Ruehli, Ed., *Circuit Analysis, Simulation and Design, Part 1*, Elsevier Science, B.V. North Holland, 1981.
3. L. Nagel, SPICE2: A computer program to simulate semiconductor circuits, PhD thesis, University of California, Berkeley, CA, 1975.
4. K. Kundert, *The Designers Guide to SPICE and SPECTRE*, Kluwer Academic, Boston, MA, 1995.
5. P. W. Tuinenga, *SPICE, A Guide to Circuit Simulation and Analysis Using PSPICE*, Prentice Hall, Englewood Cliffs, NJ, 1988.
6. J. A. Connelley and P. Choi, *Macromodeling with SPICE*, Prentice Hall, Engelwood Cliffs, NJ, 1992.
7. P. Gray and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, Wiley, New York, 1977.
8. A. Vladimiresch, *The SPICE Book*, Wiley, New York, 1994.

On modern techniques:

9. A. E. Ruehli, Ed., *Circuit Analysis, Simulation and Design, Part 2*, Elsevier Science, B.V. North Holland, 1981.

On device models:

10. P. Antognetti and G. Massobrio, *Semiconductor Modeling with SPICE2*, 2nd edn., McGraw-Hill, New York, 1993.
11. D. Foty, *MOSFET Modeling with SPICE*, Prentice Hall, Englewood Cliffs, NJ, 1997.
12. BSIM3 Users Guide, www-device.EECS.Berkeley.EDU/∼bsim3.
13. MOS-9 models, www-us2.semiconductors.philips.com/Philils_Models.

On Verilog-A

14. D. Fitzpatrick, *Analog Behavior Modeling with the Verilog-A Language*, Kluwer Academic, Boston, MA, 1997.

II

Design Automation

Wai-Kai Chen
University of Illinois at Chicago

10 Internet-Based Microelectronic Design Automation Framework
Moon-Jung Chung and Heechul Kim **10**-1
Introduction • Functional Requirements of Framework • IMEDA System • Formal Representation of Design Process • Execution Environment of the Framework • Implementation • Conclusion • References

11 System-Level Design *Alice C. Parker, Yosef Tirat-Gefen, and Suhrid A. Wadekar* **11**-1
Introduction • System Specification • System Partitioning • Scheduling and Allocating Tasks to Processing Modules • Allocating and Scheduling Storage Modules • Selecting Implementation and Packaging Styles for System Modules • Interconnection Strategy • Word-Length Determination • Predicting System Characteristics • Survey of Research in System Design • References

12 Performance Modeling and Analysis Using VHDL and SystemC
Robert H. Klenke, Jonathan A. Andrews, and James H. Aylor **12**-1
Introduction • ADEPT Design Environment • Simple Example of an ADEPT Performance Model • Mixed-Level Modeling • Performance and Mixed-Level Modeling Using SystemC • Conclusions • References

13 Embedded Computing Systems and Hardware/Software Codesign
Wayne Wolf **13**-1
Introduction • Uses of Microprocessors • Embedded System Architectures • Hardware/Software Codesign •

14 Design Automation Technology Roadmap *Donald R. Cottrell* **14**-1
Introduction • Design Automation: Historical Perspective • The Future • Summary • References

10

Internet-Based Microelectronic Design Automation Framework

10.1 Introduction **10**-1
10.2 Functional Requirements of Framework **10**-3
Building Blocks of Process • Functional Requirements of Workflow Management • Process Specification • Execution Environment • Literature Surveys
10.3 IMEDA System **10**-9
10.4 Formal Representation of Design Process **10**-12
Process Flow Graph • Process Grammars
10.5 Execution Environment of the Framework **10**-16
Cockpit Program • Manager Programs • Execution Example • Scheduling
10.6 Implementation **10**-23
System Cockpit • External Tools • Communications Model • User Interface
10.7 Conclusion **10**-30
References **10**-30

Moon-Jung Chung
Michigan State University

Heechul Kim
Hankuk University of Foreign Studies

10.1 Introduction

As the complexity of VLSI systems continues to increase, the microelectronic industry must possess an ability to reconfigure design and manufacturing resources and integrate design activities so that it can quickly adapt to the market changes and new technology. Gaining this ability imposes a twofold challenge: (1) to coordinate design activities that are geographically separated and (2) to represent an immense amount of knowledge from various disciplines in a unified format. The Internet can provide the catalyst by abridging many design activities with the resources around the world not only to exchange information but also to communicate ideas and methodologies.

In this chapter, we present a collaborative engineering framework that coordinates distributed design activities through the Internet. Engineers can represent, exchange, and access the design knowledge and carry out design activities. The crux of the framework is the formal representation of process flow using the process grammar, which provides the theoretical foundation for representation, abstraction, manipulation, and execution of design processes. The abstraction of process representation provides mechanisms to represent hierarchical decomposition and alternative methods, which enable designers to manipulate

the process flow diagram and select the best method. In the framework, the process information is layered into separate specification and execution levels so that designers can capture processes and execute them dynamically. As the framework is being executed, a designer can be informed of the current status of design such as updating and tracing design changes and be able to handling exception. The framework can improve design productivity by accessing, reusing, and revising the previous process for a similar. The cockpit of our framework interfaces with engineers to perform design tasks and to negotiate design trade-off. The framework has the capability to launch whiteboards that enable the engineers in a distributed environment to view the common process flows and data and to concurrently execute dynamic activities such as process refinement, selection of alternative process, and design reviews. The proposed framework has a provision for various browsers where the tasks and data used in one activity can be organized and retrieved later for other activities.

One of the predominant challenges for microelectronic design is to handle the increased complexity of VLSI systems. At the turn of the century, it is expected that there will be 100 million transistors in a single chip with 0.1 μm features, which will require an even shorter design time (Spiller and Newton, 1997). This increase of chip complexity has given impetus to trends such as system on a chip, embedded system, and hardware/software codesign. To cope with this challenge, industry uses custom-off-the-shelf (COTS) components, relies on design reuse, and practices outsourcing design. In addition, design is highly modularized and carried out by many specialized teams in a geographically distributed environment. Multifacets of design and manufacturing, such as manufacturability and low power, should be considered at the early stage of design. It is a major challenge to coordinate these design activities (Fairbairn, 1994). The difficulties are caused by due to the interdependencies among the activities, the delay in obtaining distant information, the inability to respond to errors and changes quickly, and general lack of communications. At the same time, the industry must contend with decreased expenditures on manufacturing facilities while maintaining rapid responses to market and technology changes.

To meet this challenge, the U.S. government has launched several programs. The Rapid Prototyping of Application Specific Signal Processor (RASSP) Program was initiated by the Department of Defense to bring about the timely design and manufacturing of signal processors. One of the main goals of the RASSP program was to provide an effective design environment to achieve a four-time improvement in the development cycle of digital systems (Chung et al., 1996). DARPA also initiated a program to develop and demonstrate key software elements for Integrated Product and Process Development (IPPD) and agile manufacturing applications. One of the foci of the earlier program was the development of infrastructure for distributed design and manufacturing. Recently, the program is continued to Rapid Design Exploration & Optimization (RaDEO) to support research, development, and demonstration of enabling technologies, tools, and infrastructure for the next generation of design environments for complex electromechanical systems. The design environment of RaDEO is planned to provide cognitive support to engineers by vastly improving their ability to explore, generate, track, store, and analyze design alternatives (Lyons, 1997).

The new information technologies, such as the Internet and mobile computing, are changing the way we communicate and conduct business. More and more design centers use PCs, and link them on the Internet/intranet. The web-based communication allows people to collaborate across space and time, between humans, humans and computers, and computers in a shared virtual world (Berners-Lee et al., 1994). This emerging technology holds the key to enhance design and manufacturing activities. The Internet can be used as the medium of a virtual environment where concepts and methodologies can be discussed, accessed, and improved by the participating engineers. Through the medium, resources and activities can be reorganized, reconfigured, and integrated by the participating organizations. This new paradigm certainly impacts the traditional means for designing and manufacturing a complex product. Using Java, programs can be implemented in a platform-independent way so that they can be executed in any machine with a Web browser. Common Object Request Broker Architecture (CORBA) (Yang and Duddy, 1996) provides distributed services for tools to communicate through the Internet (Vogel and Duddy). Designers may be able to execute remote tools through the Internet and see the visualization of design data (Erkes et al., 1996; Chan et al., 1998; Chung and Kwon, 1998).

Even though the potential impact of this technology will be great on computer aided design, Electronic Design Automation (EDA) industry has been slow in adapting this new technology (Spiller and Newton, 1997). Until recently, EDA frameworks used to be a collection of point tools. These complete suites of tools are integrated tightly by the framework using their proprietary technology. These frameworks have been suitable enough to carry out a routine task where the process of design is fixed. However, new tools appear constantly. To mix and match various tools outside of a particular framework is very difficult. Moreover, tools, expertise, and materials for design and manufacturing of a single system are dispersed geographically. Now we have reached the stage where a single tool or framework is not sufficient enough to handle the increasing complexity of a chip and emerging new technology. A new framework is necessary which is open and scalable. It must support collaborative design activities so that designers can add new tools to the framework, and interface them with other CAD systems. There are two key functions of the framework: (1) managing the process and (2) maintaining the relationship among many design representations. For design data management, refer to Katz et al. (1987). In this chapter, we will focus on the process management aspect.

To cope with the complex process of VLSI system design, we need a higher level of viewing of a complete process, i.e., the abstraction of process by hiding all details that need not to be considered for the purpose at hand. As pointed out in National Institute of Standards and Technology reports (Schlenoff et al., 1996; Knutilla et al., 1998), a ''unified process specification language'' should have the following major requirements: abstraction, alternative task, complex groups of tasks, and complex sequences.

In this chapter, we first review the functional requirements of the process management in VLSI system design. We then present the Internet-based Microelectronic Design Automation (IMEDA) System. IMEDA is a web-based collaborative engineering framework where engineers can represent, exchange, and access design knowledge and perform the design activities through the Internet. The crux of the framework is a formal representation of process flow using process grammar. Similar to the language grammar, production rules of the process grammar map tasks into admissible process flows (Baldwin and Chung, 1995a). The production rules allow a complex activity to be represented more concisely with a small number of high-level tasks. The process grammar provides the theoretical foundation for representation, abstraction, manipulation, and execution of design and manufacturing processes. It facilitates the communication at an appropriate level of complexity. The abstraction mechanism provides a natural way of browsing the process repository and facilitates process reuse and improvement. The strong theoretical foundation of our approach allows users to analyze and predict the behavior of a particular process. The cockpit of our framework interfaces with engineers to perform design tasks and to negotiate design trade-off. The framework guides the designer in selecting tools and design methodologies, and it generates process configurations that provide optimal solutions with a given set of constraints. The just-in-time binding and the location transparency of tools maximize the utilization of company resources. The framework is equipped with whiteboards so that engineers in a distributed environment can view the common process flows and data and concurrently execute dynamic activities such as process refinement, selection of alternative processes, and design reviews. With the grammar, the framework gracefully handles exceptions and alternative productions. A layered approach is used to separate the specification of design process and execution parameters. One of the main advantages of this separation is freeing designers from the overspecification and graceful exception handling. The framework, implemented using Java, is open and extensible. New process, tools, and user-defined process knowledge and constraints can be added easily.

10.2 Functional Requirements of Framework

Design methodology is defined as a collection of principles and procedures employed in the design of engineering systems. Baldwin and Chung (1995a) define design methodology management as selecting and executing methodologies so that the input specifications are transformed into desired output specifications. Kleinfeldth et al. (1994) state that ''design methodology management provides for the

definition, presentation, execution, and control of design methodology in a flexible, configured way.'' Given a methodology, we can select a process or processes for that particular methodology.

Each design activity, whether big or small, can be treated as a task. A complex design task is hierarchically decomposed into simpler subtasks, and each subtask in turn may be further decomposed. Each task can be considered as a transformation from input specification to output specification. The term *workflow* is used to represent the details of a process including its *structure in terms of all the required tasks and their interdependencies*. Some process may be ill-structured, and capturing it as a workflow may not be easy. Exceptions, conditional executions, and human involvement during the process make it difficult to model the process as a workflow.

There can be many different tools or alternative processes to accomplish a task. Thus, a design process requires design decisions such as selecting tools and processes as well as selecting appropriate design parameters. At a very high level of design, the input specifications and constraints are very general and may even be ill-structured. As we continue to decompose and perform the tasks based on design decisions, the output specifications are refined and the constraints on each task become more restrictive. When the output of a task does not meet certain requirements or constraints, a new process, tools, or parameters must be selected. Therefore, the design process is typically iterative and based on previous design experience. Design process is also a collaborative process, involving many different engineering activities and requiring the coordination among engineers, their activities, and the design results.

Until recently, it was the designer's responsibility to determine which tools to use and in what order to use them. However, managing the design process itself has become difficult, since each tool has its own capabilities and limitations. Moreover, new tools are developed and new processes are introduced continually. The situation is further aggravated because of incompatible assumptions and data formats between tools. To manage the process, we need a framework to monitor the process, carry out design tasks, support cooperative teamwork, and maintain the relationship among many design representations (Katz et al., 1987; Chiueh and Katz, 1990). The framework must support concurrent engineering activities by integrating various CAD tools and process and component libraries into a seamless environment. Figure 10.1 shows the RASSP enterprise system architecture (Welsh et al., 1995). It integrates tools, tool frameworks, and data management functions into an enterprise environment. The key functionality of the RASSP system is managing the RASSP design methodology by ''process automation'', that is, controlling CAD program execution through workflow.

10.2.1 Building Blocks of Process

The lowest level of a building block of a design process is a tool. A *tool* is an unbreakable unit of a CAD program. It usually performs a specific task by transforming given input specifications into output specifications. A *task* is defined as design activities that include information about what tools to use and how to use them. It can be decomposed into smaller subtasks. The simplest form of the task, called an *atomic task*, is the one that cannot be decomposed into subtasks. In essence, an atomic task is defined as an encapsulated tool. A task is called *logical* if it is not atomic. A workflow of a logical task describes the details of how the task is decomposed into subtasks, and the data and control dependencies such as the relationship between design data used in the subtasks. For a given task, there can be several workflows, each of which denotes a possible way of accomplishing the task. A *methodology* is a collection of workflow supported together with information on which workflow should be selected in a particular instance.

10.2.2 Functional Requirements of Workflow Management

To be effective, a framework must integrate many design automation tools and allow the designer to specify acceptable methodologies and tools together with information such as when and how they may be used. Such a framework must not only guide the designer in selecting tools and design methodologies, but also aid the designer in constructing a workflow that is suitable to complete the design under given

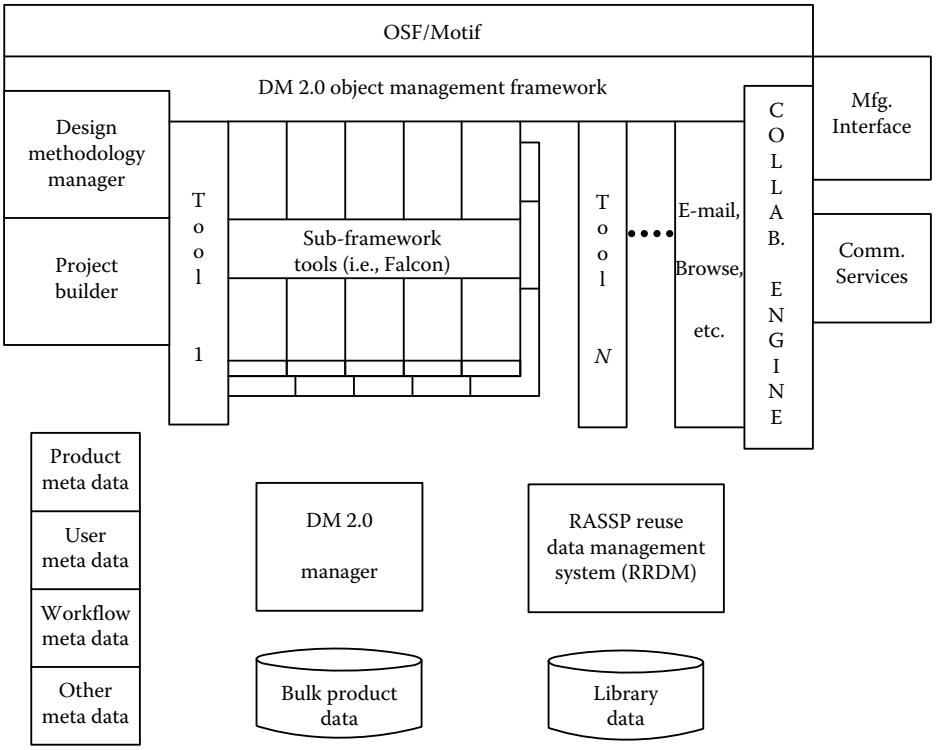


FIGURE 10.1 RASSP enterprise system architecture.

constraints. The constructed workflow should guarantee that required steps are not skipped; built-in design checks are incorporated into the workflow. The framework must also keep the relationships between various design representations, maintain the consistency between designs and support cooperative teamwork, and allow the designer to interact with the system to adjust design parameters or to modify the previous design process. The framework must be extendible to accommodate rapidly changing technologies and emerging new tools. Such a framework can facilitate developing new hardware systems as well as redesigning a system from a previous design.

During a design process, a particular methodology or workflow selected by a designer must be based on available tools, resources (computing and human), and design data. For example, a company may impose a rule that if input is a VHDL behavioral description, then designers should use Model Technology's VHDL simulator, but if the input is Verlig, they must use ViewLogic simulator. Or, if a component uses Xilinx, then all other components must also use Xilinx. Methodology must be driven by local expertise and individual preference, which in turn, are based on the designer's experience.

The process management should not constrain the designer. Instead, it must free designers from routine tasks, and guide the execution of workflow. User interaction and a designer's freedom are especially important when exceptions are encountered during the execution of flows, or when designers are going to modify the workflow locally. The system must support such activities through ''controlled interactions'' with designers.

Process management can be divided into two parts:

- Formal specification of supported methodologies and tools that must show the tasks and data involved in a workflow and their relationships
- Execution environment that helps designers to construct workflow and execute them

10.2.3 Process Specification

Methodology management must provide facilities to specify design processes. Specification of processes involves tasks and their structures (i.e., workflow). The task involved and the flow of process, that is the way the process can be accomplished in terms of its subtasks, must be defined. Processes must be encapsulated and presented to designers in a usable way. Designers want an environment to guide them in building a workflow and to help them execute it during the design process. Designers must be able to browse related processes, and compare, analyze, and modify them.

10.2.3.1 Tasks

Designers should be able to define the tasks that can be logical or atomic, organize the defined tasks, and retrieve them. Task abstraction refers to using and viewing a task for specific purposes and ignoring the irrelevant aspects of the task. In general, object-oriented approaches are used for this purpose. Abstraction of the task may be accomplished by defining tasks in terms of ''the operations the task is performing'' without detailing the operations themselves. Abstraction of tasks allows users to clearly see the behavior of them and use them without knowing the details of their internal implementations. Using the generalization–specialization (GS) hierarchy (Chung and Kim, 1990), similar tasks can be grouped together. In the hierarchy, a node in the lower level inherits its attributes from its predecessors. By inheriting the behavior of a task, the program can be shared, and by inheriting the representation of a task (in terms of its flow), the structure (workflow) can be shared. The Process Handbook (Malone et al., in press) embodies concepts of specialization and decomposition to represent processes.

There are various approaches associated with binding a specific tool to an atomic task. A tool can be bound to a task statically at the compile time, or dynamically at the run time based on available resources and constraints. When a new tool is installed, designers should be able to modify the existing bindings. The simplest approach is to modify the source code or write a script file and recompile the system. The ideal case is plug and play, meaning that CAD vendors address the need of tool interoperability, e.g., the Tool Encapsulation Specification (TES) proposed by CFI (1995).

10.2.3.2 Workflow

To define a workflow, we must specify the tasks involved in the workflow, data, and their relationship. A set of workflows defined by methodology developers enforces the user to follow the flows imposed by the company or group. Flows may also serve to guide users in developing their own flows. Designers would retrieve the cataloged flows, modify them, and use them for their own purposes based on the guidelines imposed by the developer. It is necessary to generate legal flows. A blackboard approach was used in (Lander et al., 1996) to generate a particular flow suitable for a given task. In Nelsis (ten Bosch et al., 1991), branches of a flow are explicitly represented using ''or'' nodes and ''merge'' nodes. A task can be accomplished in various ways. It is necessary to represent alternative methodologies for the task succinctly so that designers can access alternative methodologies and select the best one based on what-if analysis. IDEF3.X (IDEF) is used to graphically model workflow in RASSP environment. Figure 10.2 shows an example of workflow using IDEF3.X. A node denotes a task. It has inputs, outputs, mechanisms, and conditions. IDEF definition has been around for 20 years mainly to capture flat modeling such as a shop floor process. IDEF specification, however, requires complete information such as control mechanisms and scheduling at the specification time, making the captured process difficult to understand. In IDEF, ''or'' nodes are used to represent the alternative paths. It does not have an explicit mechanism to represent alternative workflow. IDEF is ideal only for documenting the current practice and not suitable for executing iterative process which is determined during the execution of the process. Perhaps, the most important aspect missing from most process management systems is the abstraction mechanism (Schlenoff et al., 1996).

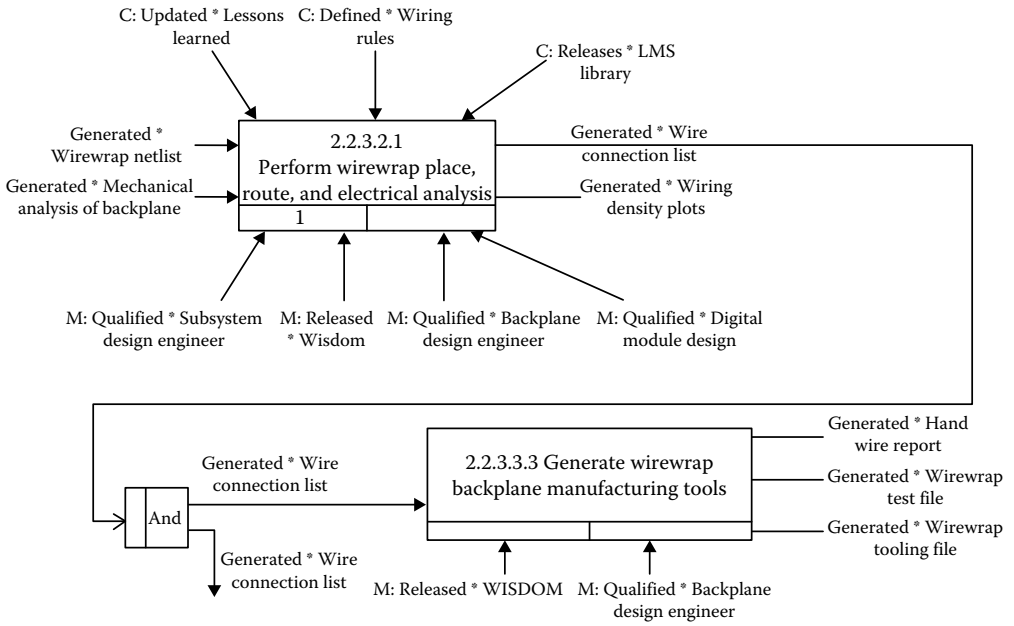


FIGURE 10.2 Workflow example using IDEF definition.

10.2.4 Execution Environment

The execution environment provides dynamic execution of tasks and tools and binds data to tools, either manually or automatically. Few frameworks separate the execution environment from the specification of design process. There are several modes in which a task can be executed (Kleinfeldth et al., 1994): manual mode, manual execution of flow, automatic flow execution, and automatic flow generation. In manual flow execution, the environment executes a task in the context of a flow. In an automatic flow execution environment, tasks are executed based on the order specified on the flow graph. In automatic flow generation, the framework generates workflow dynamically and executes them without the guidance of designers. Many frameworks use blackboard or knowledge-based approaches to generate workflow. However, it is important for designers to be able to analyze the workflow created and share it with others. That is, repeatability and predictability are important factors if frameworks support dynamic creation of workflow.

Each task may be associated with pre and postconditions. Before a task is executed, the precondition of the task is evaluated. If the condition is not satisfied, the framework either waits until the condition is met, or aborts the task and selects another alternative. After the task is executed, its postcondition is evaluated to determine if the result meets the exit criteria. If the evaluation is unsatisfactory, another alternative should be tried.

When a task is complex involving many subtasks and each subtask in turn has many alternatives, generating a workflow for the task that would successfully accomplish the task is not easy. If the first try of an alternative is not successful, another alternative should be tried. In some cases, backtrack occurs which nullifies all the executions of previous workflow.

10.2.5 Literature Surveys

Many systems have been proposed to generate design process (Knapp and Parker, 1991) and manage workflow (Dellen et al., 1997; Lavana et al., 1997; Schurmann and Altmeyer, 1997; Sutton and Director,

1998). Many of them use the Web technology to coordinate various activities in business (Andreoli, 1998), manufacturing (Berners-Lee et al., 1994; Cutkosy et al., 1996; Erkes et al., 1996), and microelectronic design (Rastogi et al., 1993; Chan et al., 1998). WELD (Chan et al., 1998) is a network infrastructure for a distributed design system that offers users the ability to create a customizable and adaptable virtual design system that can couple tools, libraries, design, and validation services. It provides support not only for designing but also for manufacturing, consulting, component acquisition, and product distribution, encompassing the developments of companies, universities, and individuals throughout the world. Lavana et al. (1997) proposed an Internet-based collaborative design. They use Petri nets as a modeling tool for describing and executing workflow. User teams, at different sites, control the workflow execution by selection of its path. Minerva II (Sutton and Director, 1998) is a software tool that provides design process management capabilities serving multiple designers working with multiple CAD frameworks. The proposed system generates design plan and realizes unified design process management across multiple CAD frameworks and potentially across multiple design disciplines. ExPro (Rastogi et al., 1993) is an expert-system-based process management system for the semiconductor design process.

There are several systems that automatically determine what tools to execute. OASIS (1992) uses Unix make file style to describe a set of rules for controlling individual design steps. The Design Planning Engine of the ADAM system (Knapp and Parker, 1986, 1991) produces a plan graph using a forward chaining approach. Acceptable methodologies are specified by listing preconditions and postconditions for each tool in a lisp-like language. Estimation programs are used to guide the chaining. Ulysses (Bushnell and Director, 1986) and Cadweld (Daniel and Director, 1991) are blackboard systems used to control design processes. A knowledge source, which encapsulates each tool, views the information on the blackboard and determines when the tool would be appropriate. The task management is integrated into the CAD framework and Task Model is interpreted by a blackboard architecture instead of a fixed inference mechanism. Minerva (Jacome and Director, 1992) and the OCT task manager (Chiueh and Katz, 1990) use hierarchical strategies for planning the design process. Hierarchical planning strategies take advantage of knowledge about how to perform abstract tasks which involve several subtasks.

To represent design process and workflow, many languages and schema have been proposed. NELSIS (ten Bosch et al., 1991) framework is based on a central, object-oriented database and on a flow management. It uses a dataflow graph as Flow Model and provides the hierarchical definition and execution of design flow. PLAYOUT (Schurmann and Altmeyer, 1997) framework is based on separate Task and Flow Models which are highly interrelated among themselves and the Product Model. In Barthelmann (1996), graph grammar is proposed in defining the task of software process management. Westfechtel (1996) proposed ''process-net'' to generate the process flow dynamically. However, in many of these systems, the relationship between task and data is not explicitly represented. Therefore, representing the case in which a task generates more than one datum and each of them goes to a different task is not easy. In Schurmann and Altmeyer (1997), Task Model (describing the I/O behavior of design tools) is used as a link between the Product Model and the Flow Model. The proposed system integrates data and process management to provide traceability. Many systems use IDEF to represent a process (Chung et al., 1996; Stavas et al.; IDEF). IDEF specification, however, requires complete information such as control mechanisms and scheduling at the specification time, making the captured process difficult to understand.

Although there are many other systems that address the problem of managing process, most proposed system use either a rule-based approach or a hard-coded process flow. They frequently require source code modification for any change in process. Moreover, they do not have mathematical formalism. Without the formalism, it is difficult to handle the iterative nature of the engineering process and to simulate the causal effects of any changes in parameters and resources. Consequently, coordinating the dynamic nature of processes is not well supported in most systems. It is difficult to analyze the rationale how an output is generated and where a failure has occurred. They also lack a systematic way of generating all permissible process flows at any level of abstraction while providing means to hide

the details of the flow when they are not needed. Most systems have the tendency to overspecify the flow information, requiring complete details of a process flow before executing the process. In most real situations, the complete flow information may not be known after the process has been executed: they are limited in their ability to address the underlying problem of process flexibility. They are rather rigid and not centered on users, and do not handle exceptions gracefully. Thus, the major functions for the collaborative framework such as adding new tools and sharing and improving the process flow cannot be realized. Most of them are weak in at least one of the following criteria suggested by NIST (Schlenoff et al., 1996): process abstraction, alternative tasks, complex groups of tasks, and complex sequences.

10.3 IMEDA System

The Internet-based Microelectronic Design Automation (IMEDA) System is a general management framework for performing various tasks in design and manufacturing of complex microelectronic systems. It provides a means to integrate many specialized tools such as CAD and analysis packages, and allows the designer to specify acceptable methodologies and tools together with information such as when and how they may be used. IMEDA is a collaborative engineering framework that coordinates design activities distributed geographically. The framework facilitates the flow of multimedia data sets representing design process, production, and management information among the organizational units of a virtual enterprise. IMEDA uses process grammar (Baldwin and Chung, 1995a) to represent the dynamic behavior of the design and manufacturing process. In a sense, IMEDA is similar to agent-based approach such as Redux (Petrie, 1996). Redux, however, does not provide process abstraction mechanism or facility to display the process flow explicitly.

The major functionality of the framework includes

- Formal representation of the design process using the process grammar that captures a complex sequence of activities of microelectronic design
- Execution environment that selects a process, elaborates the process, invokes tools, pre- and postevaluates the productions if the results meet the criterion, and notifies designers
- User interface that allows designers to interact with the framework, guides the design process, and edits the process and productions
- Tool integration and communication mechanism using Internet Socket and HTTP
- Access control that provides a mechanism to secure the activity and notification and approval that provide the mechanisms to disperse design changes to, and responses from, subscribers

IMEDA is a distributed framework design knowledge, including process information, manager programs, etc., are maintained in a distributed fashion by local servers. Figure 10.3 illustrates how IMEDA links tools and sites for distributed design activities. The main components of IMEDA are

- **System Cockpit:** It controls all interactions between the user and the system and between the system components. The cockpit will be implemented as a Java applet and may be executable on any platform for which a Java-enabled browser is available. It keeps track of the current design status and informs the user of possible actions. It allows users to collaboratively create and edit process flows, production libraries, and design data.
- **Manager Programs:** These encapsulate design knowledge. Using preevaluation functions, managers estimate the possibility of success for each alternative. They invoke tools and call postevaluation functions to determine if a tool's output meets the specified requirements. The interface servers allow cockpits and other Java-coded programs to view and manipulate production, task and design data libraries. Manager programs must be maintained by tool integrators to reflect site-specific information such as company design practices and different ways of installing tools.

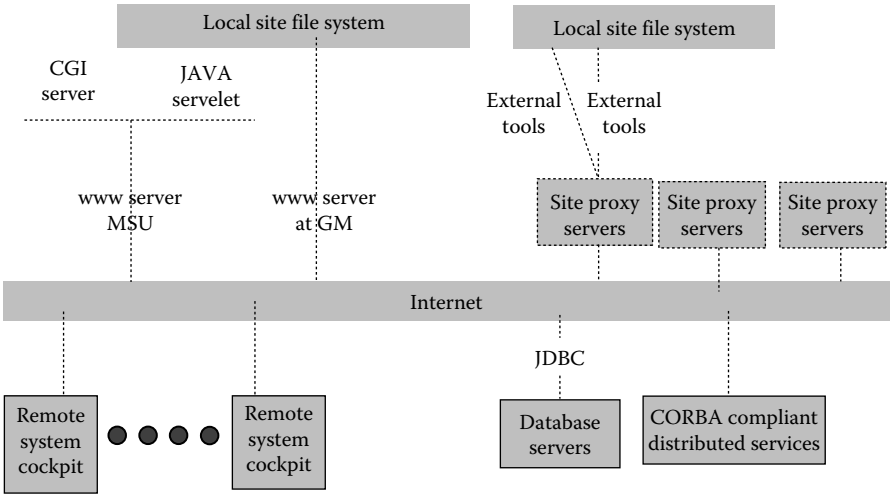


FIGURE 10.3 Architecture of IMEDA.

- **Browsers:** The task browser organizes the tasks in a GS hierarchy and contains all the productions available for each task. The data-specification browser organizes the data-specifications in a GS hierarchy and contains all the children.
- **External Tools:** These programs are the objects invoked by the framework during DM activities. Each atomic task in a process flow is bound to an external tool. External tools are written typically by the domain experts.
- **Site Proxy Server:** Any physical site that will host external tools must have a site proxy server running. These servers provide an interface between the cockpit and the external tools. The site server receives requests from system cockpits, and invokes the appropriate tool. Following the tool completion, the site server notifies the requesting cockpit, returning results, etc.
- **CGI Servers and Java Servlets:** The system cockpit may also access modules and services provided by CGI servers or the more recently introduced Java servlets. Currently, the system integrates modules of this type as direct components of the system (as opposed to external tools that may vary with the flow).
- **Database Servers:** Access to component data is a very important function. Using an API called JDBC, the framework can directly access virtually any commercially available database server remotely.
- **Whiteboard:** The shared cockpit or ''whiteboard'' is a communication medium to share information among users in a distributed environment. It allows designers to interact with the system and guides the design process collaboratively. Designers will be able to examine design results and current process flows, post messages, and carry out design activities both concurrently and collaboratively. Three types of whiteboards are the process board, the chat board, and the freeform drawing board. Their functionality includes (1) process board to the common process flow graph indicating the current task being executed and the intermediate results arrived at before the current task; (2) drawing board to load visual design data, and to design and simulate process; and (3) chat board to allow participants to communicate with each other via text-based dialog box.

IMEDA uses a methodology specification based on a process flow graphs and process grammars (Baldwin and Chung, 1995). Process grammars are the means for transforming high-level process flow graphs into progressively more detailed graphs by applying a set of substitution rules, called productions,

to nodes that represent logical tasks. It provides not only the process aspect of design activities but also a mechanism to coordinate them. The formalism in process grammar facilitates abstraction mechanisms to represent hierarchical decomposition and alternative methods, which enable designers to manipulate the process flow diagram and select the best method. The formalism provides the theoretical foundations for the development of IMEDA.

IMEDA contains the database of admissible flows, called process specifications. With the initial task, constraints, and execution environment parameters, including personal profile, IMEDA guides designers in constructing process flow graphs in a top-down manner by applying productions. It also provides designers with the ability to discover process configurations that provide optimal solutions. It maintains consistency among designs and allows the designer to interact with the system and adjust design parameters, or modify the previous design process. As the framework is being executed, a designer can be informed of the current status of design such as updating and tracing design changes and be able to handling exception.

Real-world processes are typically very complex by their very nature; IMEDA provides designers the ability to analyze, organize, and optimize processes in a way never before possible. More importantly, the framework can improve design productivity by accessing, reusing, and revising the previous process for a similar design.

The unique features of our framework include

Process Abstraction/Modeling: Process grammars provide abstraction mechanism for modeling admissible process flows. The abstraction mechanism allows a complex activity to be represented more concisely with a small number of higher-level tasks, providing a natural way of browsing the process repository. The strong theoretical foundation of our approach allows users to analyze and predict the behavior of a particular process. With the grammar, the process flow gracefully handles exceptions and alternative productions. When a task has alternative productions, backtracking occurs to select other productions.

Separation of Process Specification and Execution Environment: Execution environment information such as complex control parameters and constraints is hidden from the process specification. The information of these two layers is merely linked together to show the current task being executed on a process flow. The represented process flow can be executed in both automatic and manual modes. In the automatic mode, the framework executes all possible combinations to find a solution. In the manual mode, users can explore design space.

Communication and Collaboration: To promote real-time collaboration among participants, the framework is equipped with the whiteboard, a communication medium to share information. Users can browse related processes, compare them with other processes, analyze, and simulate them. Locally managed process flows and productions can be integrated by the framework in the central server. The framework manages the production rules governing the higher level tasks, while lower level tasks and their productions are managed by local servers. This permits the framework to be effective in orchestrating a large-scale activity.

Efficient Search of Design Process and Solution: IMEDA is able to select the best process and generate a process plan, or select a production dynamically and create a process flow. The process grammar easily captures design alternatives. The execution environment selects and executes the best one. If the selected process does not meet the requirement, then the framework backtracks and selects another alternative. This backtrack occurs recursively until a solution is found. If you allow a designer to select the best solution among many feasible ones, the framework may generate many multiple versions of the solution.

Process Simulation: The quality of a product depends on the tools (maturity, speed, and special strength of the tool), process (or workflow selected), and design data (selected from the reuse library). Our framework predicts the quality of results (product) and assesses the risk and reliability. This information can be used to select the best process/workflow suitable for a project.

Parallel Execution of Several Processes and Multiple Versions: To reduce the design time and risk, it is necessary to execute independent tasks in parallel whenever they are available. Sometimes, it is necessary to investigate several alternatives simultaneously to reduce the design time and risk. Or the designer may want to execute multiple versions with different design parameters. The key issue in this case is scheduling the tasks to optimize the resource requirements.

Life Cycle Support of Process Management: The process can be regarded as a product. A process (such as airplane designing or shipbuilding) may last many years. During this time, it may be necessary for the process itself to be modified because of new tools and technologies. Life cycle support includes updating the process dynamically, and testing/validating the design process, version history and configuration management of the design process. Tests and validations of the design processes, the simulation of processes, and impact analysis are necessary tools.

10.4 Formal Representation of Design Process*

IMEDA uses a methodology specification based on a process flow graphs and process grammars (Baldwin and Chung, 1995). The grammar is an extension of graph grammar originally proposed by Ehrig (1979) and has been applied to interconnection network (Derk and DeBrunner, 1998) and software engineering (Heiman et al., 1997).

10.4.1 Process Flow Graph

A process flow graph depicts tasks, data, and the relationships among them, describing the sequence of tasks for an activity. Three basic symbols are used to represent a process flow graph. Oval nodes represent logical tasks, two-concentric oval nodes represent atomic tasks, rectangular nodes represent data specifications and diamond nodes represent selectors. A task that can be decomposed into subtasks is called *logical*. Logical task nodes represent abstract tasks that could be done with several different tools or tool combinations. A task that cannot be decomposed is *atomic*. An atomic task node, commonly called a tool invocation, represents a run of an application program.

A *selector* is a task node that selects data or parameter. Data specifications are design data, where the output specification produced by a task can be consumed by another task as an input specification. Each data specification node, identified by a rectangle, is labeled with a data specification type. Using the graphical elements of the flow graph, engineers can create a process flow in a top-down fashion. These elements can be combined into a process flow graph using directed arcs. The result is a bipartite acyclic directed graph that identifies clearly the task and data flow relationships among the tasks in a design activity. The set of edges indicates those data specifications used and produced by each task. Each specification must have at most one incoming edge. Data specifications with no incoming edges are inputs of the design exercise. $T(G)$, $S(G)$, and $E(G)$ are the sets of task nodes, specification nodes, and edges of graph G, respectively. Figure 10.4 shows a process flow graph that describes a possible rapid prototyping design process, in which a state diagram is transformed into a field-programmable gate array (FPGA) configuration file.

The various specification types form a class hierarchy where each child is a specialization of the parent. There may be several incompatible children. For example, VHDL and Verilog descriptions are both children of simulation models. We utilize these specification types to avoid data format incompatibilities between tools (Figure 10.5a). Process flow graphs can describe design processes to varying levels of detail. A graph containing many logical nodes abstractly describes what should be done without describing how

* Materials in this section are excerpted from Baldwin, R. and Chung, M.J., *IEEE Comput.*, February, 54, 1995a. With permission.

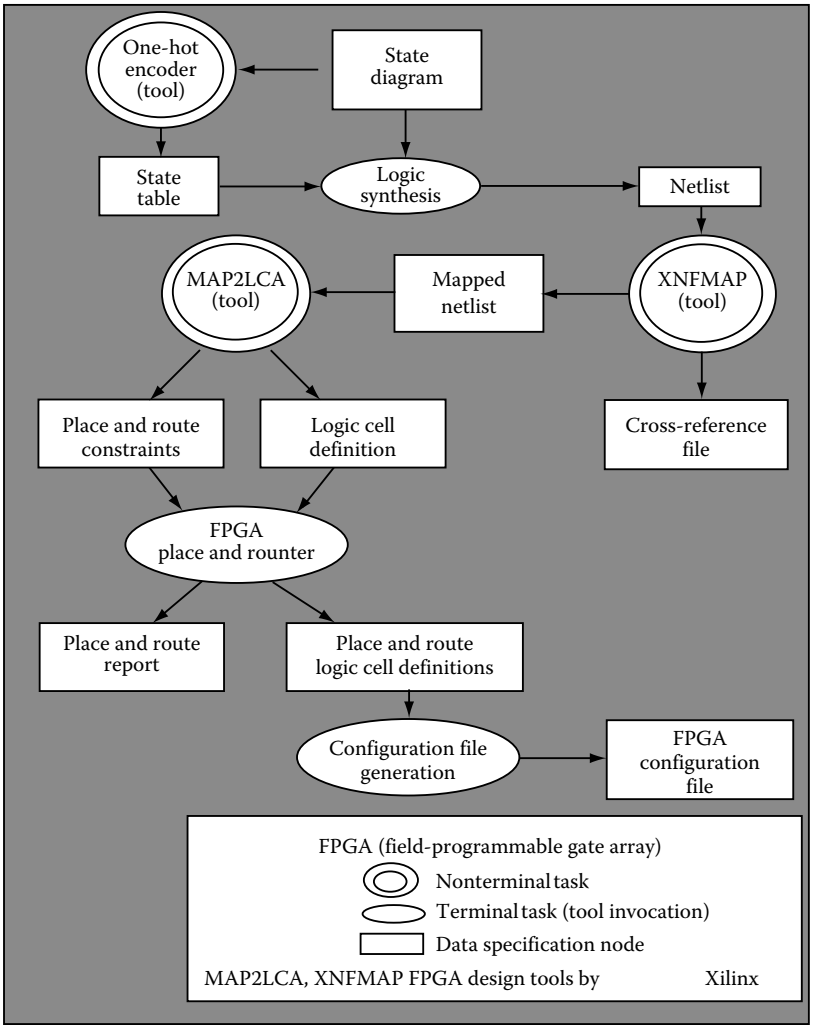


FIGURE 10.4 Sample process flow graph in which a state diagram is transformed into a field-programmable gate array configuration file.

it should be done (i.e., specifying which tools to use). Conversely, a graph in which all task nodes are atomic completely describes a methodology.

In our prototype, we use the following definitions: In(N) is the set of input nodes of node N: In(N) = $\{M \mid (M,N) \in E\}$. Out(N) is the set of output nodes of node N: Out(N) = $\{M \mid (N,M) \in E\}$. $I(G)$ is the set of input specifications of graph G: $\{N \in S(G) \mid \text{In}(N) = \emptyset\}$.

10.4.2 Process Grammars

The designer specifies the overall objectives with the initial graph that lists available input specifications, desired output specifications, and the logical tasks to be performed. By means of process grammars, logical task nodes are replaced by the flows of detailed subtasks and intermediate specifications. The output specification nodes are also replaced by nodes that may have a child specification type.

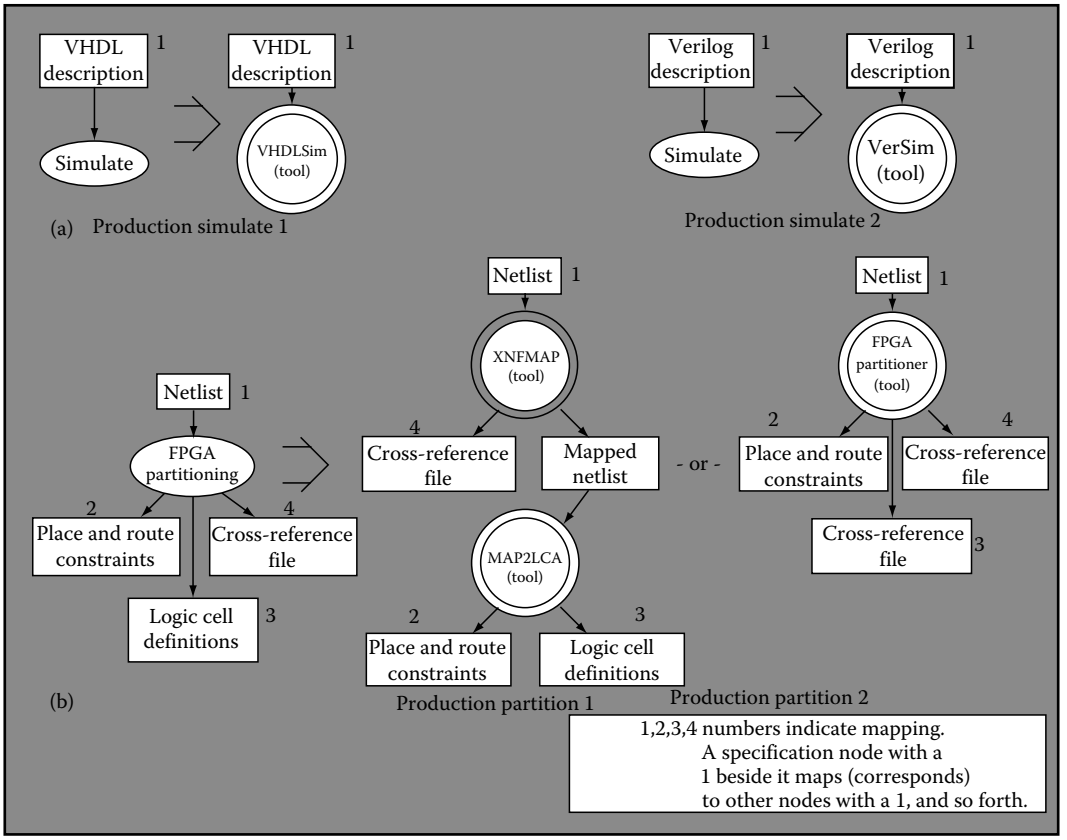


FIGURE 10.5 Graph production from a design process grammar. Two simulation alternatives based on input format are portrayed in (a); two partition alternatives representing different processes for an abstract task are portrayed in (b).

The productions in a graph grammar permit the replacement of one subgraph by another. A production in a design process grammar can be expressed formally as a tuple $P = (G_{LHS}, G_{RHS}, \sigma_{in}, \sigma_{out})$, where G_{LHS} and G_{RHS} are process flow graphs for the left side and the right side of the production, respectively, such that (1) G_{LHS} has one logical task node representing the task to be replaced, (2) σ_{in} is a mapping from the input specifications $I(G_{LHS})$ to $I(G_{RHS})$, indicating the relationship between two input specifications (each input specification of $I(G_{RHS})$ is a subtype of $I(G_{LHS})$), and (3) σ_{out} is a mapping from the output specifications of G_{LHS} to output specifications of G_{RHS} indicating the correspondence between them (each output specification must be mapped to a specification with the same type or a subtype). Figure 10.5 illustrates productions for two tasks, simulate and FPGA partitioning. The mappings are indicated by the numbers beside the specification nodes. Alternative productions may be necessary to handle different input specification types (as in Figure 10.5a), or because they represent different processes—separated by the word "or"—for performing the abstract task (as in Figure 10.5b).

Let A be the logical task node in G_{LHS} and A' be a logical task node in the original process flow graph G such that A has the same task label as A'. The production rule P can be applied to A', which means that A' can be replaced with G_{RHS} only if each input and output specifications of A' matches to input and output specifications of G_{LHS}, respectively. If there are several production rules with the same left side flow graph, it implies that there are alternative production rules for the logical task. Formally, the production matches A' if

1. A' has the same task label as A.
2. There is a mapping ρ_{in}, from In(A) to In(A'), indicating how the inputs should be mapped. For all nodes $N \in$ In(A), $\rho_{in}(N)$ should have the same type as N or a subtype.
3. There is a mapping, ρ_{out}, from Out(A') to Out(A), indicating how the outputs should be mapped. For all nodes $N \in$ Out(A'), $\rho_{out}(N)$ should have the same type as N or a subtype.

The mappings are used to determine how edges that connected the replaced subgraph to the remainder should be redirected to nodes in the new subgraph. Once a match is found in graph G, the production is applied as follows:

1. Insert $G_{RHS} - I(G_{RHS})$ into G. The inputs of the replaced tasks are not replaced.
2. For every N in $I(G_{RHS})$ and edge (N,M) in G_{RHS}, add edge ($\rho_{in}[\sigma_{in}(N)]$,$M$) to G. That is to connect the inputs of A' to the new task nodes that will use them.
3. For every N in Out(A') and edge (N,M) in G, replace edge (N,M) with edge ($\sigma_{out}[\rho_{out}(N)]$,$M$). That is to connect the new output nodes to the tasks that will use them.
4. Remove A' and Out(A') from G, along with all edges incident on them.

Figure 10.6 illustrates a derivation in which the FPGA partitioning task is planned, using a production from Figure 10.5b.

The process grammar provides mechanism of specifying alternative methods for a logical task. A high-level flow graph can then be decomposed into detailed flow graphs by applying *production rules* to a logical task. A *production rule* is a substitution that permits the replacement of a logical task node with a flow graph that represents a possible way of performing the task. The concept of applying productions to

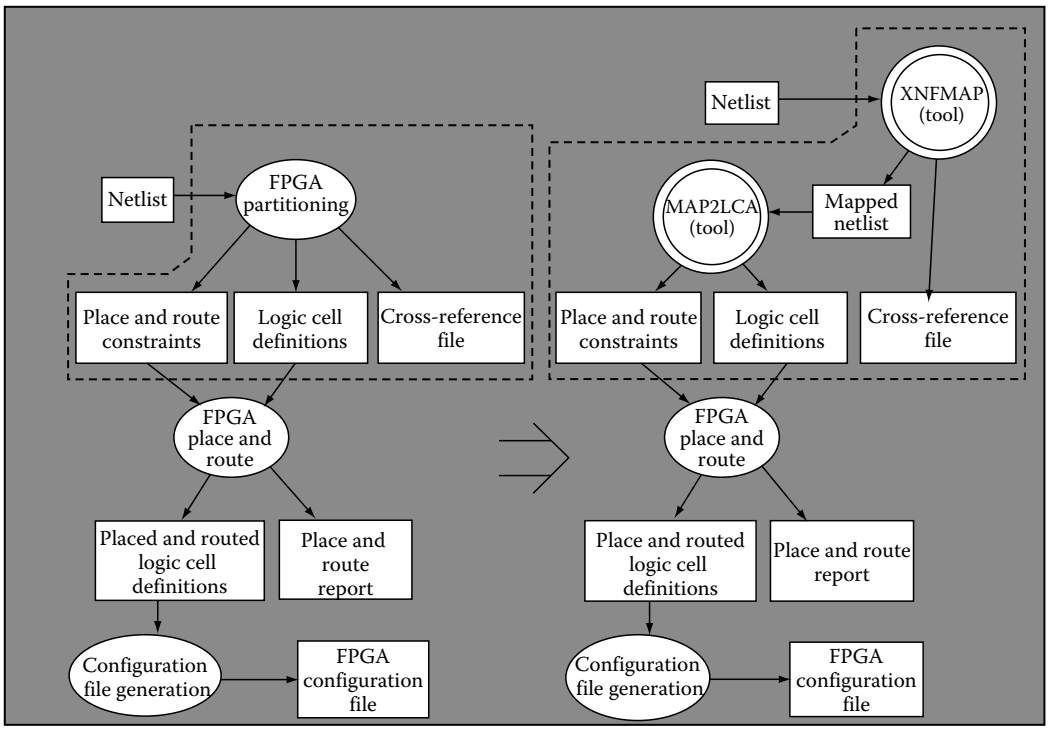


FIGURE 10.6 Sample graph derivation. Nodes in the outlined region, left, are replaced with nodes in the outlined region, right, according to production partition 1 in Figure 10.5.

logical tasks is somewhat analogous to the idea of productions in traditional (i.e., nongraph) grammars. In this sense, logical tasks correspond to *logical symbols* in grammar, and atomic tasks correspond to *terminal symbols*.

10.5 Execution Environment of the Framework

Figure 10.7 illustrates the architecture of our proposed system, which applies the theory developed in the previous section. Decisions to select or invoke tools are split between the designers and a set of manager programs, where manager programs are making the routine decisions and the designers make decisions that requires higher-level thinking. A program called Cockpit coordinates the interaction among manager programs and the designers. Tool sets and methodology preferences will differ among sites and over time. Therefore, our assumption is that each unit designates a person (or group) to act as system integrator, who writes and maintains the tool-dependent code in the system. We provide the tool-independent code and template to simplify the task of writing tool-dependent code.

10.5.1 Cockpit Program

The designer interacts with Cockpit, a program which keeps track of the current process flow graph and informs the designer of possible actions such as productions that could be applied or tasks that could be executed. Cockpit contains no task-specific knowledge; its information about the design process comes entirely from a file of graph productions. When new tools are acquired or new design processes are developed, the system integrator modifies this file by adding, deleting, and editing productions.

To assist the designer in choosing an appropriate action, Cockpit interacts with several manager programs which encapsulate design knowledge. There are two types of manager programs: task managers and production managers. Task managers invoke tools and determine which productions to execute for

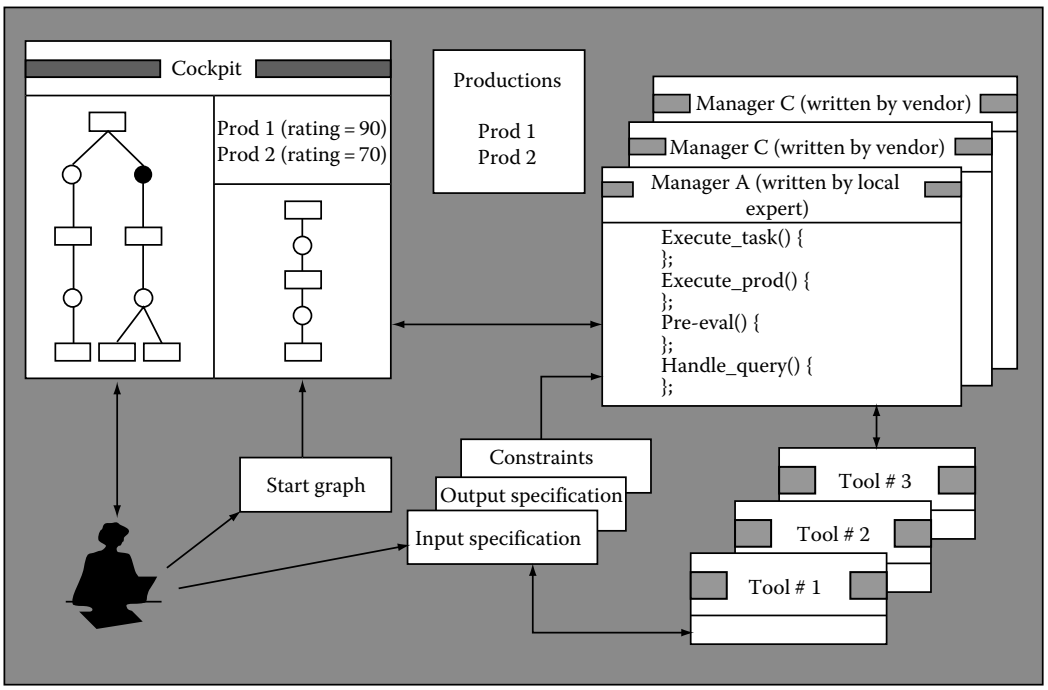


FIGURE 10.7 Proposed system based on Cockpit.

logical task nodes. Production managers provide ratings for the productions and schedule the execution of tasks on the right-hand side of the production. Managers communicate with each other using messages issued by Cockpit.

Our prototype system operates as follows. Cockpit reads the initial process flow graph from an input file generated by using a text editor. Cockpit then iteratively identifies when productions can be applied to logical task nodes and requests that the production managers assign the ratings to indicate how appropriate the productions are for those tasks. The process flow graph and the ratings of possible production applications are displayed for the designer, who directs Cockpit through a graphical user interface to apply a production or execute a task at any time. When asked to execute a task, Cockpit sends a message to a task manager. For an atomic task node, the task manager simply invokes the corresponding tool. For a logical task, the task manager must choose one or more productions, as identified by a Cockpit. The Cockpit applies the production and requests that the production manager executes it.

10.5.2 Manager Programs

Manager programs must be maintained by system integrators to reflect site-specific information, such as company design practices and tool installation methods. Typically, a manager program has its own thread. A Cockpit may have several manager programs, and therefore multithreads. We define a communication protocol between Cockpit and manager programs and provide templates for manager programs. The manager programs provide five operations: preevaluation, tool invocation, logical task execution, production execution, and query handling. Each operation described below corresponds to a C++ or Java function in the templates, which system integrators can customize as needed.

Preevaluation: Production managers assign ratings to help designers and task managers select the most appropriate productions. The rating indicates the likelihood of success from applying this production. The strategies used by the system integrator provide most of the code to handle the rating. In some cases, it may be sufficient to assign ratings statically, based on the success of past productions. These static ratings can be adjusted downward when the production has already been tried unsuccessfully on this task node (which could be determined using the query mechanism). Alternatively, the ratings may be an arbitrarily complex function of parameters obtained through the query mechanism or by examining the input files. Sophisticated manager programs may continuously gather and analyze process metrics that indicate those conditions leading to success, adjust adjusting ratings accordingly.

Tool Invocation: Atomic task mangers must invoke the corresponding software tool when requested by Cockpit, then determine whether the tool completed successfully. In many cases, information may be predetermined and entered in a standard template, which uses the tool's result status to determine success. In other cases, the manager must determine tool parameters using task-specific knowledge or determine success by checking task-specific constraints. Either situation would require further customization of the manager program.

Logical Task Execution: Logical task managers for logical tasks must select productions to execute the logical task. Cockpit informs the task manager of available productions and their ratings. The task manager can either direct Cockpit to apply and execute one or more productions, or it can decide that none of the productions is worthwhile and report failure. The task manager can also request that the productions be reevaluated when new information has been generated that might influence the ratings, such as a production's failure. If a production succeeds, the task manager checks any constraints; if they are satisfied, it reports success.

Production Execution: Production managers execute each task on the right-hand side of the production at the appropriate time and possibly check constraints. If one of the tasks fails or a constraint is violated, backtracking can occur. The production manager can use task-specific knowledge to determine which

tasks to repeat. If the production manager cannot handle the failure itself, it reports the failure to Cockpit, and the managers of higher level tasks and productions attempt to handle it.

Query Handling: Both production and task managers participate in the query mechanism. A production manager can send queries to its parent (the task manager for the logical task being performed) or to one of its children (a task manager of a subtask). Similarly, a task manager can send a query to its parent production manager or to one of its children (a production manager of the production it executed). The manager templates define C functions, which take string arguments, for sending these queries. System integrators call these functions but do not need to modify them. The manager templates also contain functions which are modified by system integrators for responding to queries. Common queries can be handled by template code; for example, a production manager can frequently ask its parent whether the production has already been attempted for that task and whether it succeeded. The manager template handles any unrecognized query from a child manager by forwarding it to the parent manager. Code must be added to handle queries for task-specific information such as the estimated circuit area or latency.

10.5.3 Execution Example

Now we describe a synthesis scenario that illustrates our prototype architecture in use. In this scenario, the objective is to design a controller from a state diagram, which will ultimately be done following the process flow graph in Figure 10.4. There are performance and cost constraints on the design, and the requirement to produce a prototype quickly. The productions used are intended to be representative but not unique. For simplicity, we assume that a single designer is performing the design with, therefore, only one Cockpit.

The start graph for this scenario contains only the primary task, chip synthesis, and specification nodes for its inputs and outputs (like the graph in the left in Figure 10.8). Cockpit tells us that the production of Figure 10.8 can be applied. We ask Cockpit to apply it. The chip synthesis node is then replaced by nodes for state encoding, logic synthesis, and physical synthesis, along with intermediate specification nodes.

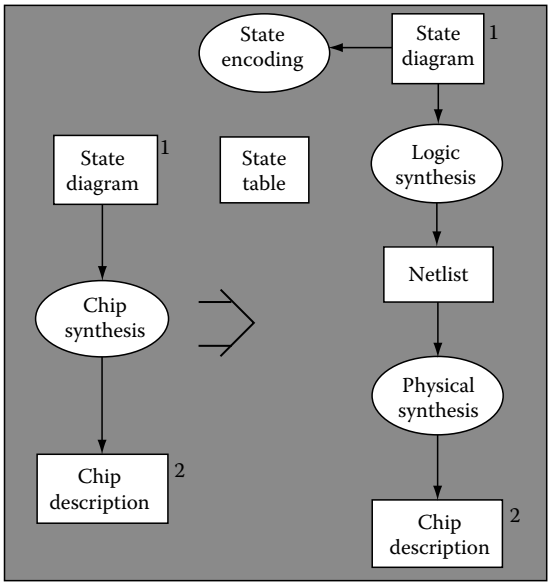


FIGURE 10.8 Productions for chip synthesis.

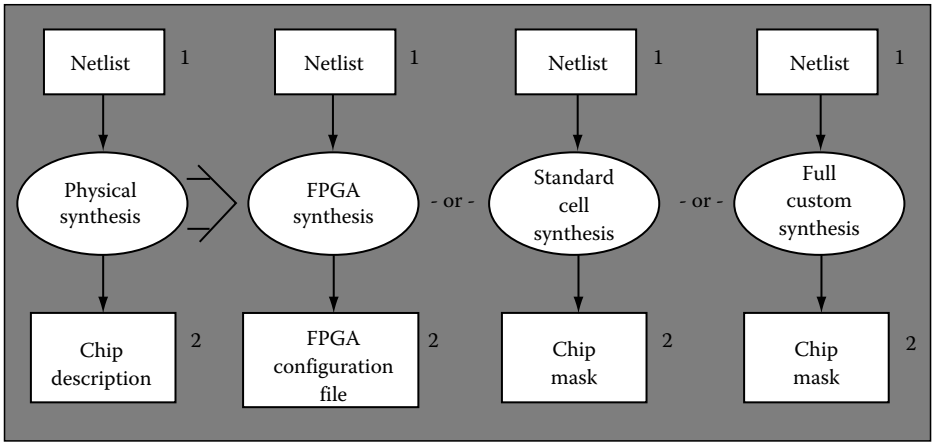


FIGURE 10.9 Productions for physical synthesis.

Next, we want to plan the physical synthesis task. Tasks can be planned in an order other than they are to be performed. Cockpit determines that any of the productions shown in Figure 10.9 may be applied, then queries each production's task manager program asking it to rate the production's appropriateness in the current situation. Based on the need to implement the design quickly, the productions for standard cell synthesis and full custom synthesis are rated low while the production for FPGA synthesis is rated high. Ratings are displayed to help us decide.

When we plan the state encoding task, Cockpit finds two productions: one to use the tool Minbits encoder and the other to use the tool One-hot encoder. One-hot encoder works well for FPGAs, while Minbits encoder works better for other technologies. To assign proper ratings to these productions, their production managers must find out which implementation technology will be used. First, they send a query to their parent manager, the state encoding task manager. This manager forwards the message to its parent, the chip synthesis production manager. In turn, this manager forwards the query to the physical synthesis task manager for an answer. All messages are routed by Cockpit, which is aware of the entire task hierarchy. This sequence of actions is illustrated in Figure 10.10.

After further planning and tool invocations, a netlist is produced for our controller. The next step is the FPGA synthesis task. We apply the production in Figure 10.11 and proceed to the FPGA partitioning task. The knowledge to automate this task has already been encoded into the requisite manager programs, so we direct Cockpit to execute the FPGA partitioning task. It finds the two productions illustrated in Figure 10.5b and requests their ratings, Next, Cockpit sends an execute message, along with the ratings, to the FPGA partitioning task manager. This manager's strategy is to always execute the highest-rated production, which in this case is production Partition 1. (Other task managers might have asked that both productions be executed or, if neither were promising, immediately reported failure.) This sequence of actions is shown in Figure 10.12.

Because the Partition 1 manager used an as-soon-as-possible task scheduling strategy, it asks Cockpit to execute XNFMAP immediately. The other subtask, MAP2LCA, is executed when XNFMAP complete successfully. After both tasks complete successfully, Cockpit reports success to the FPGA partitioning task manager. This action sequence is illustrated in Figure 10.13.

10.5.4 Scheduling

In this section, we describe a detailed description and discussion of auto-mode scheduling, including the implementation of the *linear scheduler*. The ability to search through the configuration space of a design process for a design configuration that meets user-specified constraints is important. For example,

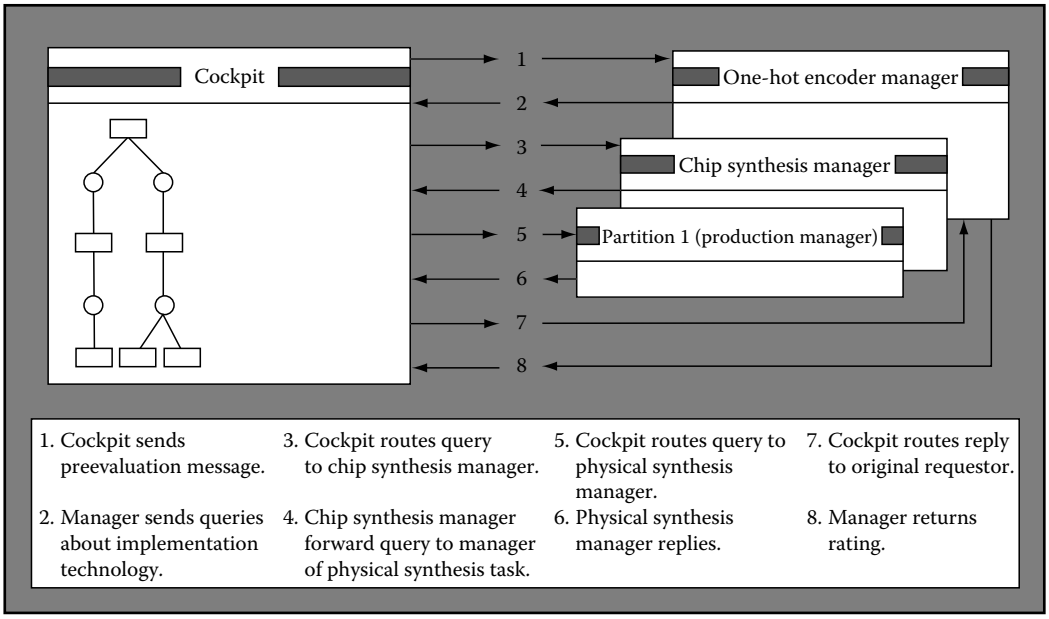


FIGURE 10.10 Sequence of actions for query handling.

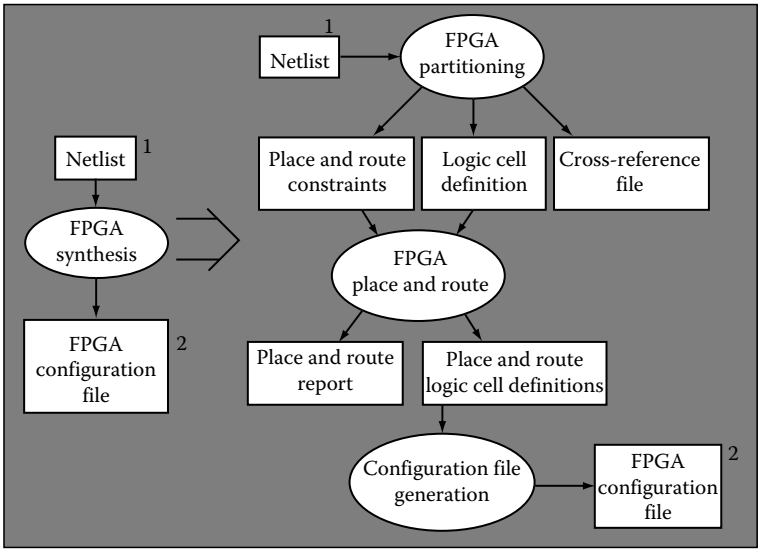


FIGURE 10.11 Production for field-programmable gate array.

assume that a user has defined a process for designing a digital filter with several different alternative ways of performing logical tasks such as ''FPGA Partitioning'' and ''Select the Filter Architecture.'' One constraint that an engineer may wish to place on the design might be: ''Find a process configuration that produces a filter that has maximum delay at most 10 ns.'' Given such a constraint, the framework must search through the configuration space of the filter design process, looking for a sequence of valid atomic tasks that produces a filter with ''maximum delay at most 10 ns.'' We call the framework component that

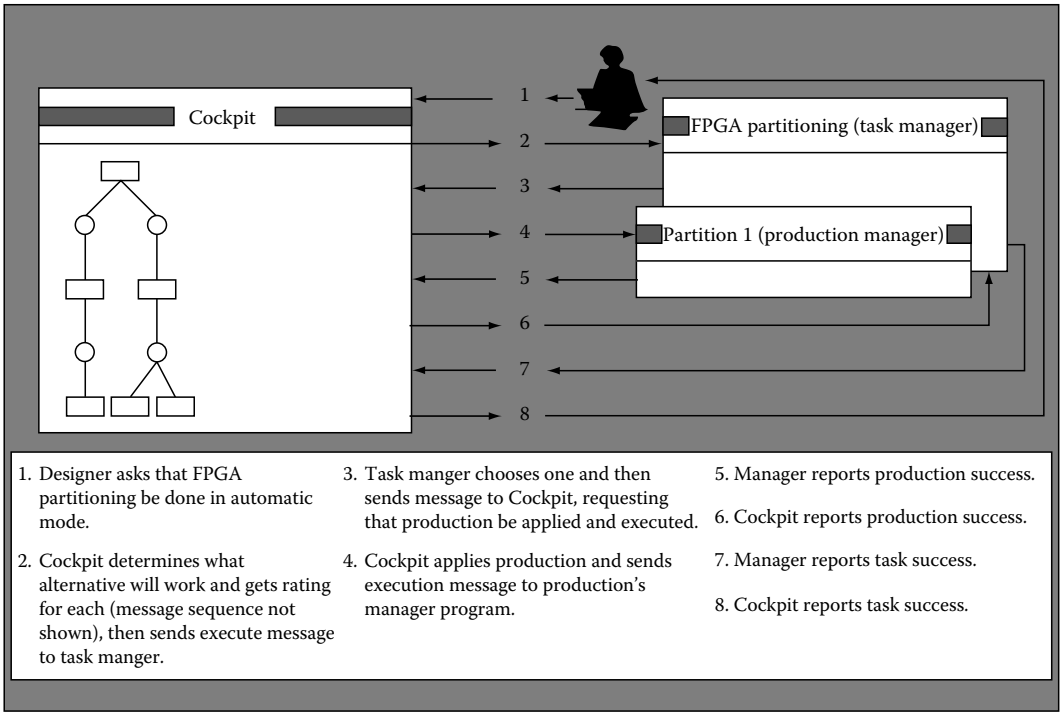


FIGURE 10.12 Sequence of action during automatic task execution.

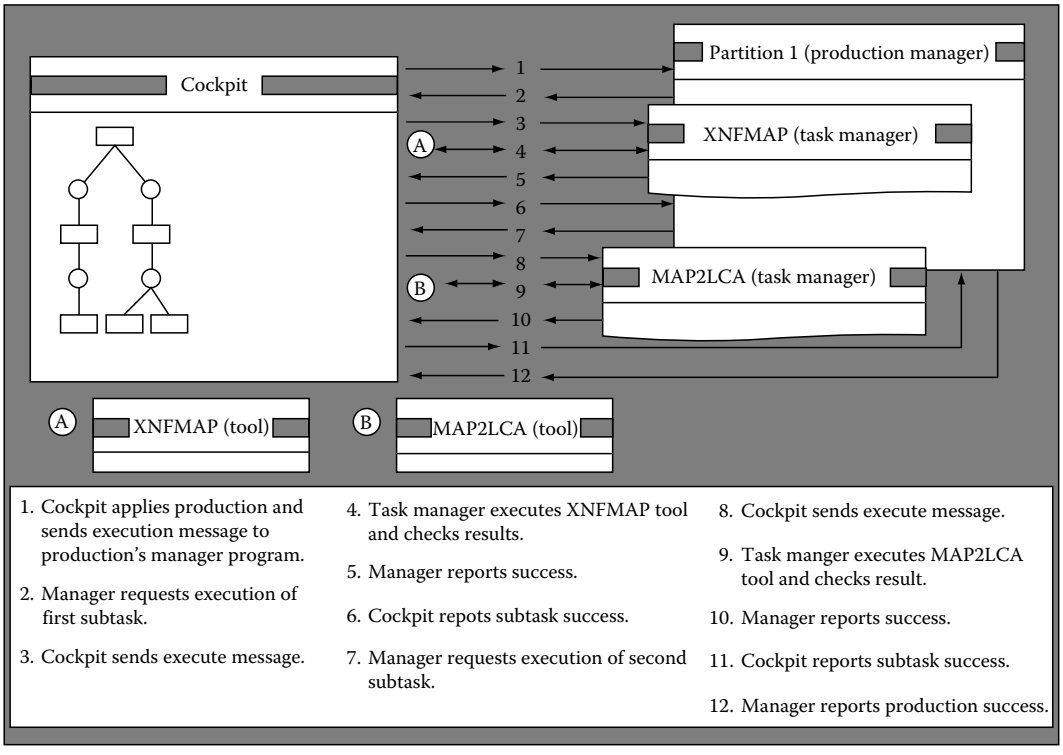


FIGURE 10.13 Sequence of actions during automatic production execution.

performs this search a *scheduler*. There are, of course, many different ways of searching through the design process configuration space. In general, a successful scheduler will provide the following functionality:

- **Completeness (Identification of Successful Configurations):** Given a particular configuration of a process, the correct scheduler will be able to conclusively determine whether the configuration meets user-specified constraints. The scheduler must guarantee that before reporting failure, all possible process configurations have been considered, and if there is a successful configuration, the algorithm must find it.
- **Reasonable Performance:** The configuration space of a process grows exponentially (in the number of tasks). Ideally, a scheduler will be able to search the configuration space using an algorithm that requires less than exponential time.

The Linear Scheduling Algorithm is very simple yet complete and meets most of the above criteria. In this algorithm, for each process flow graph (corresponding to an initial process flow graph or a production), it has a scheduler. Each scheduler is a separate thread with a *Task Schedule List* (TSL) representing the order in which tasks are to be executed. The tasks in a scheduler's TSL are called its children tasks. A scheduler also has a *task pointer* to indicate the child task being executed in the TSL. The algorithm is recursive such that with each new instantiation of a production of a given task, a new scheduler is created to manage the flow graph representing the production selected. A liner scheduler creates a TSL by performing a topological sort of the initial process flow graph and executes its children tasks in order. If a child task is atomic, the scheduler executes the task without creating a new scheduler; otherwise, it selects a new alternative, creates a new child scheduler to manage the selected alternative, and waits for a signal from the child scheduler indicating success or failure. When a child task execution is successful, the scheduler increments the *task pointer* in its TSL and proceeds to execute the next task. If a scheduler reaches the end of its TSL, it signals success to its own parent, and awaits signals from its parent if it should terminate itself (all successful) or rollback (try another to find new configurations). If a child task fails, the scheduler tries another alternative for the task. If there is no alternatives left, it rolls back (by decrementing the task pointer) until it finds a logical task that has another alternative to try. If a scheduler rolls back to the beginning of the TSL and cannot find an alternative, then its flow has failed. In this case, it signals a failure to its parent and terminates itself.

In the linear scheduling algorithm, each scheduler can send or receive any of five signals: PROCEED, ROLLBACK, CHILD-SUCCESS, CHILD-FAILURE, and DIE. These signals comprise scheduler-to-scheduler communication, including self-signaling. Each of the five signals is discussed below.

- **PROCEED:** This signal tells the scheduler to execute the next task in the TSL. It can be self-sent or received from a parent scheduler. For example, a scheduler increments its task pointer and sends itself a PROCEED signal when a child task succeeds, whereas it sends a PROCEED signal to its children to start its execution.
- **ROLLBACK:** This is signaled when a task execution has failed. This signal may be self-sent or received from a parent scheduler. Scheduler self-signals ROLLBACK whenever a child task fails. A Rollback can result in either trying the next alternative of a logical task, or decrementing the task pointer and trying the previous task in the TSL. If rollback results in decrementing the task pointer to point to a child task node which has received a success-signal, the parent scheduler will send a rollback signal to that child task scheduler.
- **CHILD-SUCCESS:** A child scheduler sends a CHILD-SUCCESS to its parent scheduler if it has successfully completed the execution of all of the tasks in its TSL. After sending the child-success signal, the scheduler remains active, listening for possible rollback signals from the parent. After receiving a child-success signal, parent schedulers self-send a proceed signal.
- **CHILD-FAILURE:** A child-failure signal is sent from a child scheduler to its parent in the event that the child's managed flow fails. After sending a child-failure signal, children

schedulers terminate. Upon receiving child-failure signals, parent scheduler self-send a rollback signal.

- **DIE:** This signal may be either self-sent, or sent from parent schedulers to their children schedulers.

10.6 Implementation

In this section, a high level description of the major components of IMEDA and their organization and functionality will be presented. Detailed explanations of the key concepts involved in the architecture of the Process Management Framework will also be discussed, including external tool integration, the tool invocation process, the Java File System, and state properties.

10.6.1 System Cockpit

The System Cockpit, as its name suggests, is shown in Figure 10.14 where nearly all user interaction with the framework takes place. It is here that users create, modify, save, load, and simulate process flow graphs representing design processes. This system component has been implemented as a Java applet. As such, it is possible to run the cockpit in any Java-enabled Web browser such as Netscape's Navigator

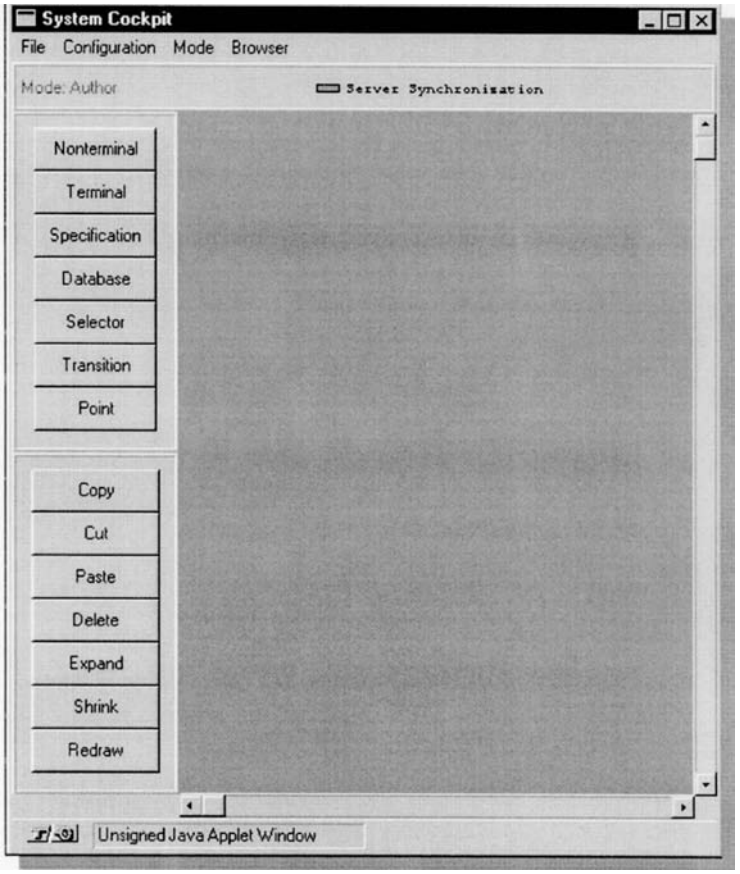


FIGURE 10.14 System Cockpit window.

or Microsoft's Internet Explorer. It is also possible to run the cockpit in some Java-enabled operating systems such as IBM's OS/2.

Each cockpit component also has the following components:

- **Root Flow:** Every cockpit has a Root Flow. The Root Flow is the flow currently being edited in the Cockpit's Flow Edit Panel. Notice that the Root Flow may change as a result of applying a production to a flow graph, in which case the Root Flow becomes a derivation of itself.
- **Flow Edit Panel:** The Flow Edit Panel is the interactive Graphical User Interface for creating and editing process flow graphs. This component also acts as a display for animating process simulations performed by various schedulers such as the manual or auto-mode linear scheduler.
- **Class Directory:** The Cockpit has two Class Directories: Task Directory and the Specification Directory. These directories provide the ''browser'' capabilities of the framework, allowing users to create reusable general-to-specific hierarchies of task classes. Class Directories are implemented using a *tree* structure.
- **Production Database:** The Production Database acts as a warehouse for logical task productions. These productions document the alternative methods available for completing a logical task. Each Production Database has a list of Productions. The Production Database is implemented as a tree-like structure, with Productions being on the root trunk, and Alternatives being leaves.
- **Browser:** Browsers provide the tree-like graphical user interface for users to edit both Class Directories and Databases. There are three Browsers: Database Browser for accessing the Production Database, Directory Browser for accessing the Task Directory, and Directory Browser for accessing the Spec Browser. Both Database Browsers and Directory Browsers inherit properties from object Browser, and offer the user nearly identical editing environments and visual representations. This deliberate consolidation of Browser interfaces allowed us to provide designers with an interface that was consistent and easier to learn.
- **Menu:** A user typically performs and accesses most of the system's key function from the Cockpit's Menu.
- **Scheduler:** The Cockpit has one or more schedulers. Schedulers are responsible for searching the configuration space of a design process for configurations that meet user specified design constraints. The Scheduler animates its process simulations by displaying them in the Flow Edit Panel of the Cockpit.

10.6.2 External Tools

External Tools are the concrete entities to which atomic tasks from a production flow are bound. When a flow task object is expanded in the Cockpit Applet (during process simulation), the corresponding external tool is invoked. The external tool uses a series of inputs and produces a series of outputs (contained in files). These inputs and outputs are similarly bound to specifications in a production flow. Outputs from one tool are typically used as inputs for another. IMEDA can handle the transfer of input and output files between remote sites. The site proxy servers, in conjunction with a remote file server (also running at each site) automatically handle the transfer of files from one system to another. External tools may be implemented using any language, and on any platform that has the capability of running a site server. While performing benchmark tests of IMEDA, we used external tools written in C, Fortran, Perl, csh (a Unix shell script), Java applications, and Mathematica scripts.

10.6.2.1 External Tool Integration

One of the primary functionality of IMEDA is the integration of user-defined external tools into an abstract process flow. IMEDA then uses these tools both in simulating the process flow to find a flow configuration that meets specific constraints, and in managing selected flow configurations during actual design execution.

There are two steps to integrating tools with a process flow defined in IMEDA: *association* and *execution*. Association involves ''linking'' or ''binding'' an abstract flow item (e.g., an atomic task) to an external tool. Execution describes the various steps that IMEDA takes to actually invoke the external tool and process the results.

10.6.2.2 Binding Tools

External tools may be bound to three types of flow objects: atomic tasks, selectors, and multiple version selectors. Binding an external tool to a flow object is a simple and straightforward job, involving simply defining certain properties in the flow object.

The following properties must be defined in an object that is to be bound to an external tool:

- **SITE:** Due to the fact that IMEDA can execute tools on remote systems, it is necessary to specify the site where the tool is located on. Typically, a default SITE will be specified in the system defaults, and making it unnecessary to define the site property unless the default is to be overriden. Note that the actual site ID specified by the SITE property must refer to a site that is running a site proxy server listening on that ID. See the ''Executing External Tools'' section below for more details.
- **CMDLINE:** The CMDLINE property specifies the command to be executed at the specified remote site. The CMDLINE property should include any switches or arguments that will always be sent to the external tool. Basically, the CMDLINE argument should be in the same format that would be used if the command were executed from a shell/DOS prompt.
- **WORKDIR:** The working directory of the tool is specified by the WORKDIR property. This is the directory in which IMEDA will actually execute the external tool, create temporary files, etc. This property is also quite often defined in the global system defaults, and thus may not necessarily have to be defined for every tool.
- **WRAPPERPATH:** The JDK 1.0.2 does not allow Java applications to execute a tool in an arbitrary directory. To handle remote tool execution, a wrapper is provided. It is a ''go-between'' program that would simply change directories and then execute the external tool. This program can be as simple as a DOS/NT batch file, a shell script, or a perl program. The external tool is wrapped in this simple script, and executed. Since IMEDA can execute tools at remote and heterogeneous sites, it was very difficult to create a single wrapper that would work on all platforms (WIN32, Unix, etc.). Therefor, the wrapper program may be specified for each tool, defined as global default, or a combination of the two.

Once the properties above have been defined for a flow object, the object is said to be ''bound'' to an external tool. If no site, directory, or filename is specified for the outputs of the flow object, IMEDA automatically creates unique file names, and stores the files in the working directory of the tool on the site that the tool was run. If a tool uses as inputs data items that are not specified by any other task, then the data items must be bound to static files on some site.

10.6.2.3 Executing External Tools

Once flow objects have been bound to the appropriate external tools, IMEDA can be used to perform process simulation or process management. IMEDA actually has several ''layers'' that lie between the Cockpit (a Java applet) and the external tool that is bound to a flow being viewed by a user in the Cockpit. A description of each of IMEDA components for tool invocations is listed below:

- **Tool Proxy:** The tool proxy component acts as a liaison between flow objects defined in Cockpits and the site proxy server. All communication is done transparently through the communication server utilizing TCP/IP sockets. The tool proxy ''packages'' information from Cockpit objects (atomic tasks, selectors, etc.) into string messages that the proxy server will recognize. It also listens

for and processes messages from the proxy server (through the communications server) and relays the information back to the Cockpit object that instantiated the tool proxy originally.

- **Communications Server:** Due to various security restrictions in the 1.0.2 version of Sun Microsystem's Java Development Kit (JDK), it is impossible to create TCP/IP socket connections between a Java applet and any IP address other than the address from which the applet was loaded. Therefore, it was necessary to create a ''relay server'' in order to allow cockpit applets to communicate with remote site proxy servers. The sole purpose of the communications server is to receive messages from one source and then to rebroadcast them to all parties that are connected and listening on the same channel.
- **Site Proxy Server:** Site proxy servers are responsible for receiving and processing invocation requests from tool proxies. When an invocation request is received, the site proxy server checks to see that the request is formatted correctly, starts a tool monitor to manage the external tool invocation, and returns the exit status of the external tool after it has completed.
- **Tool Monitors:** When the site proxy server receives an invocation request and invokes an external tool, it may take a significant amount of time for the tool to complete. If the proxy server had to delay the handling of other requests while waiting for each external tool to complete, IMEDA would become very inefficient. For this reason, the proxy server spawns a tool monitor for each external tool that is to be executed. The tool monitor runs as a separate thread, waiting on the tool, storing its stdout and stderr, and moving any input or output files that need moving to their appropriate site locations, and notifying the calling site proxy server when the tool has completed. This allows the site proxy server to continue receiving and processing invocation requests in a timely manner.
- **Tool Wrapper:** Tool wrapper changes directories into the specified WORKDIR, and then executes the CMDLINE.
- **External Tool:** External tools are the actual executable programs that run during a tool invocation. There is very little restriction on the nature of the external tools.

10.6.3 Communications Model

The Communications Model of IMEDA is perhaps the most complex portion of the system in some respects. This is where truly *distributed* communications come into play. One system component is communicating with another via network messages rather than function calls.

The heart of the communications model is the Communications Server. This server is implemented as a broadcast server. All incoming messages to the server are simply broadcast to all other connected parties. FlowObjects communicate with the Communications Server via ToolProxys. A ToolProxy allows a FlowObject to abstract all network communications and focus on the functionality of invoking tasks. A ToolProxy takes care of constructing a network message to invoke an external tool. That message is then sent to the Communications Server via a Communications Client. The Communication Client takes care of the low-level socket based communication complexities. Finally, the Communications Client sends the message to the Communications Server, which broadcasts the message to all connected clients. The client for which the message was intended (typically a site proxy server) decodes the message and, depending on its type, creates either a ToolMonitor (for an Invocation Message) or an External Redraw Monitor (for a Redraw Request).

The site proxy server creates these monitors to track the execution of external programs, rather than monitoring them itself. In this way, the proxy server can focus on its primary job—receiving and decoding network messages. When the monitors invoke an external tool, they must do so within a wrapper. Once the monitors have observed the termination of an external program, they gather any output on *stdout* or *stderr* and return these along with the exit code of the program to the site proxy server. The proxy server returns the results to the Communications Server, then the Communications Client, then the ToolProxy, and finally to the original calling FlowObject.

10.6.4 User Interface

The Cockpit provides both the user interface and core functionality of IMEDA. While multiple users may use different instances of the Cockpit simultaneously, there is currently no provision for direct collaboration between multiple users. Developing efficient means of real-time interaction between IMEDA users is one of the major thrusts of the next development cycle.

Currently the GUI of the Cockpit provides the following functionalities:

- **Flow Editing:** Users may create and edit process flows using the flow editor module of the Cockpit. The flow editor provides the user with a simple graphical interface that allows the use of a template of tools for ''drawing'' a flow. Flows can be optimally organized via services provided by a remote Layout Server written in Perl.
- **Production Library Maintenance:** The Cockpit provides functionality for user maintenance of collections of logical task productions, called libraries. Users may organize productions, modify input/output sets, or create/edit individual productions using flow editors.
- **Class Library Maintenance:** Users are provided with libraries of task and specification classes that are organized into a GS hierarchy. Users can instantiate a class into an actual task, specification, selector, or database when creating a flow by simply dragging the appropriate class from a class browser and dropping it onto a flow editor's canvas. The Cockpit provides the user with a simple tree structure interface to facilitate the creation and maintenance of class libraries.
- **Process Simulation:** Processes may be simulated using the Cockpit. The Cockpit provides the user with several scheduler modules that determine how the process configuration space will be explored. The schedulers control the execution of external tools (through the appropriate site proxy servers) and simulation display (flow animation for user monitoring of simulation progress). There are multiple schedulers for the user to choose from when simulating a process, including the manual scheduler, comprehensive linear scheduler, etc.
- **Process Archival:** The Cockpit allows processes to be archived on a remote server using the Java File System (JFS). The Cockpit is enabled by a JFS client interface to connect to a remote JFS server where process files are saved and loaded. While the JFS has its clear advantages, it is also awkward to not allow users to save process files, libraries, etc. on their local systems. Until version 1.1 of the Java Development Kit, local storage by a Java applet was simply not an option—the browser JVM definition did not allow access to most local resources. With version 1.1 of the JDK, however, comes the ability to electronically sign an applet. Once this has been done, users can grant privileged resource access to specific applets after a signature has been verified.

10.6.4.1 Design Flow Graph Properties

Initially, a flow graph created by a user using GUI is not associated with any system-specific information. For example, when a designer creates an atomic task node in a flow graph, there is initially no association with any external tool. The framework must provide a mechanism for users to bind flow graph entities to the external tools or activities that they represent.

We have used the concept of *properties* to allow users to bind flow graph objects to external entities. In an attempt to maintain flexibility, properties have been implemented in a very generic fashion. Users can define any number of properties for flow object. There are a number of key properties that the framework recognizes for each type of flow object. The user defines these properties to communicate needed configuration data to the framework.

A property consists of a *property label* and *property contents*. The label identifies the property, and consists of an alphanumeric string with no white space. The content of a property is any string. Currently users define properties using a freeform text input dialog, with each line defining a property. The first word on a line represents the property label, and the remainder of the line constitutes the property contents.

10.6.4.2 Property Inheritance

To further extend the flexibility of flow object properties, the framework requires that each flow object be associated with a *flow object class*. Classes allow designers to define properties that are common to all flow objects that inherit from that flow object class. Furthermore, classes are organized into a general-to-specific hierarchy, with children classes inheriting properties from parent classes.

Therefore, the properties of a particular class consist of any properties defined locally for that object, in addition to properties defined in the object's inherited class hierarchy. If a property is defined in both the flow object and one of its parent classes, the property definition in the flow object takes precedence. If a property is defined in more than one class in a class hierarchy, the ''youngest'' class (e.g., the child in a parent–child relationship) takes precedence.

Classes are defined in the Class Browsers of IMEDA. Designers that have identified a clear general-to-specific hierarchy of flow object classes can quickly create design flow graphs by dragging and dropping from class browsers onto flow design canvases. The user would then need only to overload those properties in the flow objects that are different from their respective parent classes (Figure 10.15).

For example, consider a class hierarchy of classes that all invoke the same external sort tool, but pass different flags to the tool, based on the context. It is likely that all of these tools will have properties in common, such as a common working directory and tool site. By defining these common properties in a common ancestor of all of the classes, such as *Search*, it is unnecessary to redefine the properties in the children classes.

Of course, children classes can define new properties that are not contained in the parent classes, and may also overload property definitions provided by ancestors. Following these rules, class *Insertion* would have the following properties defined: WORKDIR, SITE, WRAPPERPATH, and CMDLINE (Figure 10.16).

10.6.4.3 Macro Substitution

While performing benchmarks on IMEDA, one cumbersome aspect of the framework that users often pointed out was the need to reenter properties for tasks or specifications if, for example, a tool name or working directory changed. Finding every property that needed to be changed was a tedious job, and prone to errors.

In an attempt to deal with this problem, we came up with the idea of *property macros*. That is, a property macro is any macro that is not a key system macro. A macro is a textual substitution rule that

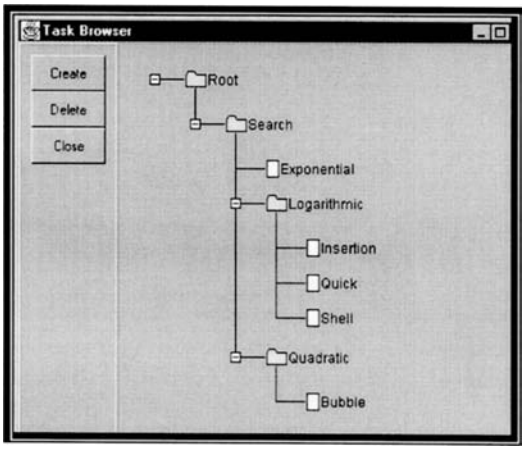


FIGURE 10.15 Task browser.

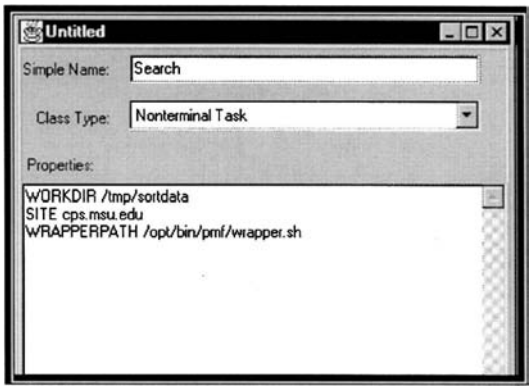


FIGURE 10.16 Property window and property inheritance.

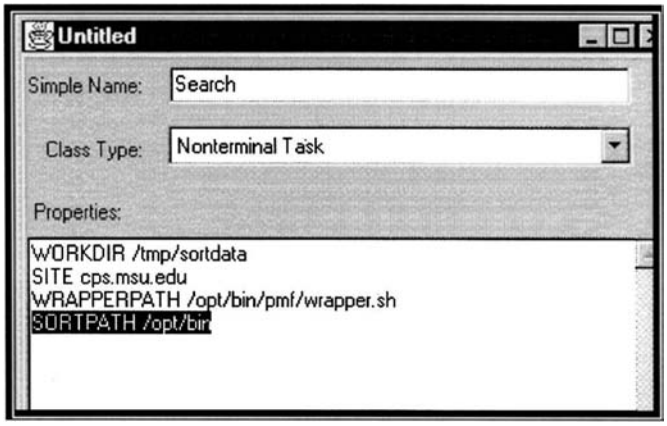


FIGURE 10.17 Macro definition.

can be created by users. By using macros in the property databases of flow objects, design flows can be made more flexible and more amiable to future changes.

As an example, consider a design flow that contains many atomic tasks bound to an external tool. Our previous example using searches is one possible scenario. On one system, the path to the external tool may be ''/opt/bin/sort,'' while on another system the path is ''/user/keyesdav/public/bin/sort.'' Making the flow object properties flexible is easy if a property macro named SORTPATH is defined in an ancestor of all affected flow objects (Figure 10.17). Children flow objects can then use that macro in place of a static path when specifying the flow object properties. As a further example, consider a modification to the previous ''Search task hierarchy'' where we define a macro SORTPATH in the Search class, and then use that macro in subsequent children classes, such as the Insertion class.

In the highlighted portion of the Property Database text area, a macro called ''SORTPATH'' is defined. In subsequent class' Property Databases, this macro can be used in place of a static path. This makes it easy to change the path for all tools that use the SORTPATH property macro—just the property database dialog where SORTPATH is originally defined needs to be modified (Figure 10.18).

10.6.4.4 Key Framework Properties

In our current implementation of IMEDA, there are a number of key properties defined. These properties allow users to communicate needed information to the framework in a flexible fashion. Most importantly,

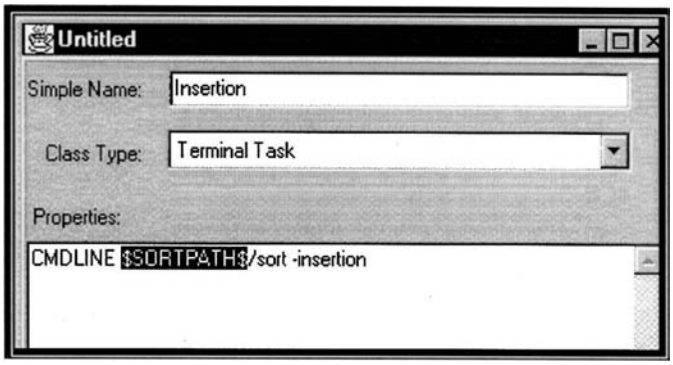


FIGURE 10.18 Macro substitution.

it allows system architects to define or modify system properties quickly. This is an important benefit when working with evolving software such as IMEDA.

10.7 Conclusion

Managing the design process is the key factor to improve the productivity in the microelectronic industry. We have presented an Internet-based Microelectronic Design Automation (IMEDA) framework to manage the design process. IMEDA uses a powerful formalism, called design process grammars, for representing design processes. We have also proposed an execution environment that utilizes this formalism to assist designers in selecting and executing appropriate design processes. The proposed approach is applicable not only in rapid prototyping but also in any environment where a design is carried out hierarchically and many alternative processes are possible.

The primary advantages of our system are

- **Formalism:** A strong theoretical foundation enables us to analyze how our system will operate with different methodologies.
- **Parallelism:** In addition to performing independent tasks within a methodology in parallel, our system also allows multiple methodologies to be executed in parallel.
- **Extensibility:** New tools can be integrated easily by adding productions and manager programs.
- **Flexibility:** Many different control strategies can be used. They can even be mixed within the same design exercise.

The prototype of IMEDA is implemented using Java. We are currently integrating more tools into our prototype system and developing manager program templates that implement more sophisticated algorithms for preevaluation, logical task execution, and query handling. Our system will become more useful as CAD vendors to adapt open software systems and allow greater tool interoperability.

References

Andreoli, J.-M., Pacull, F., and Pareschi, R., XPECT: A framework for electronic commerce, *IEEE Internet Comput.*, 1(4): 40–48, 1998.

Baldwin, R. and Chung, M.J., A formal approach to mangaging design processes, *IEEE Comput.*, 28: 54–63, February 1995a.

Baldwin, R. and Chung, M.J., Managing engineering data for complex products, *Res. Eng. Des.*, 7: 215–231, 1995b.

Barthelmann, K., Process specification and verification, *Lect. Notes Comput. Sci.*, 1073, 225–239, 1996.

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F., and Secret, A., The World-Wide Web, *Commn. ACM*, 37(8): 76–82, 1994.

Bushnell, M.L. and Director, S.W., VLSI CAD tool integration using the Ulysses environment, *23rd ACM/IEEE Design Automation Conference*, Las Vegas, NV, pp. 55–61, 1986.

Casotto, A., Newton, A.R., and Sangiovanni-Vincentelli, A., Design management based on design traces, *27th ACM/IEEE Design Automation Conference*, Orlando, FL, pp. 136–141, 1990.

CFI, Tool Encapsulation Specification, Draft Standard, Version 2.0, released by the CFI TES Working Group, 1995.

Chan, F.L., Spiller, M.D., and Newton, A.R., WELD—An environment for web-based electronic design, *35th ACM/IEEE Design Automation Conference*, ACM Press, San Francisco, CA, pp. 146–151, June 1998.

Chiueh, T.F. and Katz, R.H., A history model for managing the VLSI design process, *International Conference on Computer-Aided Design*, Santa Clara, CA, pp. 358–361, 1990.

Chung, M.J. and Kim, S., An object-oriented VHDL environment, *Proceedings of the 27th ACM/IEEE Design Automation Conference*, Orlando, FL, June 24–28, IEEE Computer Society Press, pp. 431–436, 1990.

Chung, M.J. and Kim, S., Configuration management and version control in an object-oriented VHDL environment, *International Conference on Computer-Aided Design '91*, Santa Clara, CA, pp. 258–261, 1991.

Chung, M.J. and Kwon, P., A web-based framework for design and manufacturing a mechanical system, *1998 ASME Design Engineering Technical Conference (DETC) and Computer Engineering Conference*, Atlanta, GA, September 1998, paper number DAC-5599.

Chung, M.J., Charmichael, L., and Dukes, M., Managing a RASSP design process, *Comp. Ind.*, 30: 49–61, 1996.

Cutkosy, M.R., Tenenbaum, J.M., and Glicksman, J., Madefast: Collaborative engineering over the Internet, *Commn. ACM*, 39(9): 78–87, 1996.

Daniel, J. and Director, S.W., An object oriented approach to CAD tool control, *IEEE Trans. Comput. Aided Des.*, 10: 698–713, June 1991.

Dellen, B., Maurer, F., and Pews, G., Knowledge-based techniques to increase the flexibility of workflow management, *Data and Knowledge Engineering*, North-Holland Publishing Company, North-Holland, The Netherlands, 1997.

Derk, M.D. and DeBrunner, L.S., Reconfiguartion for fault tolerance using graph grammar, *ACM Trans. Comput. Syst.*, 16(1): 41–54, February 1998.

Di Janni, A., A monitor for complex CAD systems, *23rd Design Automation Conference*, Las Vegas, NV, pp. 145–151, 1986.

Ehrig, H., Introduction to the algebraic theory of graph grammars, *1st Workshop on Graph Grammars and Their Applications to Computer Science and Biology*, Springer, LNCS, pp. 1–69, 1979.

Erkes, J.W., Kenny, K.B., Lewis, J.W., Sarachan, B.D., Sobololewski, M.W., and Sum, R.N., Implementing shared manufacturing services on the World-Wide Web, *Commn. ACM*, 39(2): 34–45, 1996.

Fairbairn, D.G., 1994 keynote address, *31st Design Automation Conference*, San Diego, CA, pp. xvi–xvii, 1994.

Hardwick, M., Spooner, D.L., Rando, T., and Morris, K.C., Sharing manufacturing information in virtual enterprises, *Commn. ACM*, 39(2): 46–54, 1996.

Hawker, S., SEMATECH Computer Integrated Manufacturing (CIM) Framework Architecture Concepts, Principles, and Guidelines, Version 0.7.

Heiman, P. et al., Graph-based software process management, *Int. J. Software Eng. Knowledge Eng.*, 7(4): 1–24, December 1997.

Hines, K. and Borriello, G., A geographically distributed framework for embedded system design and validation, *35th Annual Design Automation Conference*, ACM Press, San Francisco, CA, pp. 140–145, June 1998.

Hsu, M. and Kleissner, C., Objectflow: Towards a process management infrastructure, *Distributed Parallel Databases*, 4: 169–194, 1996.

IDEF, http://www.idef.com.

Jacome, M.F. and Director, S.W., Design process management for CAD frameworks, *29th Design Automation Conference*, Anaheim, CA, pp. 500–505, 1992.

Jacome, M.F. and Director, S.W., A formal basis for design process planning and management, *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*, 15(10): 1197–1211, October 1996.

Katz, R.H., Bhateja, R., E-Li Chang, E., Gedye, D., and Trijanto, V., Design version management, *IEEE Des. Test*, 4(1): 12–22, February 1987.

Kleinfeldth, S., Guiney, M., Miller, J.K., and Barnes, M., Design methodology management, *Proc. IEEE*, 82(2): 231–250, February 1994.

Knapp, D. and Parker, A., The ADAM design planning engine, *IEEE Trans. Comput. Aided Des. Integrated Circuits Syst.*, 10(7): 829–846, July 1991.

Knapp, D.W. and Parker, A.C., A design utility manager: The ADAM planning engine, *23rd ACM/IEEE Design Automation Conference*, Las Vegas, NV, pp. 48–54, 1986.

Knutilla, A., Schlenoff, C., Ray, S., Polyak, S.T., Tate, A., Chiun Cheah, S., and Anderson, R.C., Process specification language: An analysis of existing representations, NISTIR 6160, National Institute of Standards and Technology, Gaithersburg, MD, 1998.

Kocourek, C., An architecture for process modeling and execution support, *Comput. Aided Syst. Theory—EUROCAST*, Springer Berlin/Heidelberg, 1030: 199–216, 1995a.

Kocourek, C., Planning and execution support for design process, *IEEE International Symposium and Workshop on Systems Engineering of Computer Based System Proceedings*, 1995b.

Lander, S.E., Staley, S.M., and Corkill, D.D., Designing integrated engineering environments: Blackboard-based integration of design and analysis tools, *Concurrent Engineering*, 4(1): 59–71, 1996.

Lavana, H., Khetawat, A., Brglez, F., and Kozminski, K., Executable workflows: A paradigm for collaborative design on the Internet, *34th ACM/IEEE Design Automation Conference*, ACM Press, Anaheim Convention Center, Anaheim, CA, pp. 553–558, June 9–13, 1997.

Lyons, K., RaDEO Project Overview, http://www.cs.utah.edu/projects/alpha1/arpa/mind/index.html, 1997.

Malone, T.W., Crowston, K., and Herman, G.A. (Eds.), *The MIT Process Handbook*, MIT Press, Cambridge, MA, 2003.

OASIS, *OASIS Users Guide and Reference Manual*, MCNC, Research Triangle Park, NC, 1992.

Petrie, C.J., Agent based engineering, the Web, and intelligence, *Proceedings of IEEE Expert*, pp. 24–29, December 1996.

Rastogi, P., Koziki, M., and Golshani, F., ExPro: An expert system based process management system, *IEEE Trans. Semiconductor Manuf.*, 6(3): 207–218, 1993.

Schlenoff, C., Knutilla, A., and Ray, S., Unified process specification language: Requirements for modeling process, NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD, 1996.

Schurmann, B. and Altmeyer, J., Modeling design tasks and tools—The link between product and flow model, *Proceedings of the 34th ACM/IEEE Design Automation Conference*, Anaheim, CA, pp. 564–569, June 1997.

Spiller, M.D. and Newton, A.R., EDA and network, *International Conference on Computer-Aided Design*, San Jose, CA, pp. 470–475, 1997.

Stavas, J., Wedgwood, J., Forte, M., Selvidge, W., Tuck, M.C., Bard, A., and Finnie, E., Workflow modeling for implementing complex, CAD-based, design methodologies, Proceedings of the Mentor Graphics Users Group International Conference, 1996.

Sutton, P.R. and Director, S.W., A description language for design process management, *33rd ACM/IEEE Design Automation Conference*, Las Vegas, NV, pp. 175–180, June 1996.

Sutton, P.R. and Director, S.W., Framework encapsulations: A new approach to CAD tool interoperability, *35th ACM/IEEE Design Automation Conference*, San Francisco, CA, pp. 134–139, June 15–19, 1998.

ten Bosch, K.O., Bingley, P., and Van der Wolf, P., Design flow management in the NELSIS CAD framework, *Proceedings of the 28th Design Automation Conference*, San Francisco, CA, pp. 711–716, 1991.

Toye, G., Cutkosky, M.R., Leifer, L.J., Tenenbaum, J.M., and Glicksman, J., SHARE: A methodology and environment for collaborative product development, *Proceedings of the 2nd Workshop Enabling Technology: Infrastructure Collaborative Enterprises*, Los Alamitos, CA, IEEE Computer Society Press, pp. 33–47, 1993.

Vogel, A. and Duddy, K., *Java Programming with CORBA*, Third Edition, Wiley Computer Publishing, New York, 2001.

Welsh, J., Kalathil, B., Chanda, B., Tuck, M.C., Selvidge, W., Finnie, E., and Bard, A., Integrated process control and data management in RASSP enterprise system, *Proceedings of 1995 RASSP Conference*, *Second Annual RASSP Conference*, Crystal City, VA, July 24–28, 1995.

Westfechtel, B., Integrated product and process management for engineering design applications, *Integrated Comput. Aided Eng.*, 3(1): 20–35, 1996.

Yang, Z. and Duddy, K., CORBA: A platform for distributed object computing, *ACM Oper. Syst. Rev.*, 30(2): 4–31, 1996.

11

System-Level Design

11.1 Introduction **11**-1
Design Philosophies and System-Level Design • System Design Space
11.2 System Specification **11**-3
11.3 System Partitioning **11**-5
Constructive Partitioning Techniques • Iterative Partitioning Techniques
11.4 Scheduling and Allocating Tasks to Processing Modules **11**-7
11.5 Allocating and Scheduling Storage Modules **11**-8
11.6 Selecting Implementation and Packaging Styles for System Modules **11**-9
11.7 Interconnection Strategy **11**-9
11.8 Word-Length Determination **11**-10
11.9 Predicting System Characteristics **11**-10
11.10 Survey of Research in System Design **11**-10
System Specification • Partitioning • Nonpipelined Design • Macropipelined Design • Genetic Algorithms • Imprecise Computation • Probabilistic Models and Stochastic Simulation • Performance Bounds Theory and Prediction • Word-Length Selection • Embedded Systems • System on Chip and Network on Chip
References **11**-16

Alice C. Parker
University of Southern California

Yosef Tirat-Gefen
University of Southern California

Suhrid A. Wadekar
University of Southern California

11.1 Introduction

A *system* is a collection of interdependent operational components that together accomplish a complex task. Examples of systems range from cellular phones to camcorders to satellites. Projections point to a continuous increase in the complexity of systems in the coming years.

The term system, when used in the digital design domain, connotes many different entities. A system can consist of a processor, memory, and input/output, all on a single integrated circuit (IC), or can consist of a network of processors, geographically distributed, all performing a specific application. There can be a single clock, with modules communicating synchronously and multiple clocks with asynchronous communication or entirely asynchronous operation. The design can be general purpose, or specific to a given application, i.e., application-specific. The above variations together constitute the *system style*. System style selection is determined to a great extent by the physical technologies used, the environment in which the system operates, designer experience, and corporate culture, and is not automated to any great extent.

System-level design covers a wide range of design activities and design situations. It includes the more specific activity *system engineering*, which involves the requirements, development, test planning,

subsystem interfacing, and end-to-end analysis of systems. System-level design is sometimes called *system architecting*, a term used widely in the aerospace industry.

General-purpose system-level design involves the design of programmable digital systems including the basic modules containing storage, processors, input/output, and system controllers. At the system level, the design activities include determining the following:

- *Power budget* (the amount of power allocated to each module in the system)
- Cost and performance budget allocated to each module in the system
- Interconnection strategy
- Selection of commercial off-the-shelf (COTS) modules
- Packaging of each module
- Overall packaging strategy
- Number of processors, storage units, and input/output interfaces required
- Overall characteristics of each processor, storage unit, and I/O interface

For example, memory system design focuses on the number of memory modules required, how they are organized, and the capacity of each module. A specific system-level decision in this domain can be how to partition the memory between the processor chip and the off-chip memory. At a higher level, a similar decision might involve configuration of the complete storage hierarchy, including memory, disk drives, and archival storage.

For each general-purpose system designed, many more systems are designed to perform specific applications. *Application-specific system design* involves the same activities as described above, but can involve many more decisions, since there are usually more custom logic modules involved. Specifications for application-specific systems contain not only requirements on general capabilities, but also contain the functionality required in terms of specific tasks to be executed. Major application-specific system-level design activities include not only the above general-purpose system design activities, but also the following activities:

- Partitioning an application into multiple functional modules
- Scheduling the application tasks on shared functional modules
- Allocating functional modules to perform the application tasks
- Allocating and scheduling storage modules to contain blocks of data as they are processed
- Determining the implementation styles of functional modules
- Determining the word lengths of data necessary to achieve a given accuracy of computation
- Predicting resulting system characteristics once the system design is complete

Each of the system design tasks given in the two lists above will be described in detail below. Since the majority of system design activities are application-specific, this section will focus on system-level design of application-specific systems. Related activities, hardware–software codesign, verification, and simulation are covered in other sections.

11.1.1 Design Philosophies and System-Level Design

Many design tools have been constructed with a *top-down design* philosophy. Top-down design represents a design process whereby the design becomes increasingly detailed until final implementation is complete. Considerable prediction of resulting system characteristics is required to make the higher-level decisions with some degree of success.

Bottom-up design, on the other hand, relies on designing a set of primitive elements, and then forming more complex modules from those elements. Ultimately, the modules are assembled into a system. At each stage of the design process, there is complete knowledge of the parameters of the lower-level elements. However, the lower-level elements may be inappropriate for the tasks at hand.

System designers in industry describe the design process as being much less organized and considerably more complex than the top-down and bottom-up philosophies suggest. There is a mixture of top-down and bottom-up activities with major bottlenecks of the system receiving detailed design consideration while other parts of the system still exist only as abstract specifications. For this reason, the system-level design activities we present in detail here support such a complex design situation. Modules, elements, and components used to design at the system level might exist, or might only exist as abstract estimates along with requirements. The system can be designed after all modules have been designed and manufactured, prior to any detailed design, or with a mixture of existing and new modules.

11.1.2 System Design Space

System design, like data path design, is quite straightforward as long as the constraints are not too severe. However, most designs must solve harder problems than problems solved by existing systems. Designers must race to produce working systems faster than competitors, systems that are also less expensive. More variations in design are possible than ever before and such variations require a large *design space* to be explored. The dimensions of the design space (its axes) are system properties such as cost, power, design time, and performance. The design space contains a population of designs, each of which possesses different values of these system properties. There are literally millions of system designs for a given specification, each of which exhibits different cost, performance, power consumption, and design time. Straightforward solutions that do not attempt to optimize system properties are easy to obtain, but may be inferior to designs that are produced by system-level design tools and have undergone many iterations of design. The complexity of system design is not because system design is an inherently difficult activity, but because so many variations in design are possible and time does not permit exploration of all of them.

11.2 System Specification

Complete system specifications contain a wide range of information, including

- Constraints on the system power, performance, cost, weight, size, and delivery time
- Required functionality of the system components
- Any required information about the system structure
- Required communication between system components
- Flow of data between components
- Flow of control in the system
- Specification of input precision and desired output precision

Most systems specifications that are reasonably complete exist first in a natural language. However, natural language interfaces are not currently available with commercial system-level design tools.

More conventional system specification methods used to drive system-level design tools include formal languages, graphs, and a mixture of the two. Each of the formal system specification methods described here contains some of the information found in a complete specification, i.e., most specification methods are incomplete. The designer can provide the remaining information necessary for full system design interactively, can be entered later in the design process, or can be provided in other forms at the same time the specification is processed. The required design activities determine the specification method used for a given system design task.

There are no widely adopted formal languages for system-level hardware design although System-Level Design Language (SLDL) was developed by an industry group. Hardware descriptive languages such as VHSIC Verilog Hardware Descriptive Language (VHDL) [1] and Verilog [2] are used to describe the functionality of modules in an application-specific system. High-level synthesis tools can then synthesize

such descriptions to produce register-transfer designs. Extensions of VHDL and Verilog have been proposed to encompass more system-level design properties. Apart from system constraints, VHDL specifications can form complete system descriptions. However, the level of detail required in VHDL and to some extent in Verilog requires the designer to make some implementation decisions. In addition, some information that is explicit in more abstract specifications such as the flow of control between tasks is implicit in HDLs.

Graphical tools have been used for a number of years to describe system behavior and structure. *Block diagrams* are often used to describe system structure. Block diagrams assume that tasks have already been assigned to basic blocks and their configuration in the system has been specified. Block diagrams generally cannot represent the flow of data or control, or design constraints. The processor memory switch (PMS) notation invented by Bell and Newell was an early attempt to formalize the use of block diagrams for system specification [3].

Petri nets have been used for many years to describe system behavior using a *token-flow* model. A token-flow model represents the flow of control with tokens, which flow from one activity of the system to another. Many tokens can be active in a given model concurrently, representing asynchronous activity and parallelism, important in many system designs. Timed Petri nets have been used to model system performance, but Petri nets cannot easily be used to model other system constraints, system behavior, or any structural information.

State diagrams and graphical tools such as *State charts* [4] provide alternative methods for describing systems. Such tools provide mechanisms to describe the flow of control, but do not describe system constraints, system structure, data flow, or functionality.

Task-flow graphs, an outgrowth from the control/data-flow graphs (CDFG) used in high-level synthesis are often used for system specification. These graphs describe the flow of control and data between tasks. When used in a hierarchical fashion, task nodes in the task-flow graph can contain detailed functional information about each task, often in the form of a CDFG. Task-flow graphs contain no mechanisms for describing system constraints or system structure.

Spec charts [5] incorporate VHDL descriptions into state-chart-like notation, overcoming the lack of functional information found in state charts.

Figure 11.1 illustrates the use of block diagrams, Petri nets, task-flow graphs, and spec charts.

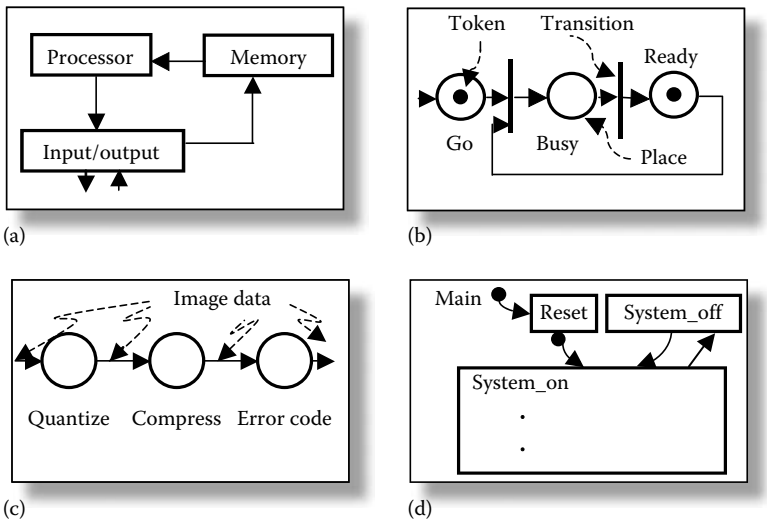


FIGURE 11.1 Use of (a) block diagrams, (b) Petri nets, (c) task-flow graphs, and (d) spec charts, shown in simplified form.

11.3 System Partitioning

Most systems are too large to fit on a single substrate. If the complexity of the system tasks and the capacities of the system modules are of the same order, then partitioning is not required. All other systems must be partitioned so that they fit into the allowed dies, packages, boards, multichip modules (MCMs), and cases. *Partitioning determines the functions, tasks, or operations in each partition of a system*. Each partition can represent a substrate, package, MCM, or larger component. Partitioning is performed with respect to a number of goals, including minimizing cost, design time or power, or maximizing performance. Any of these goals can be reformulated as specific constraints such as meeting given power requirements.

When systems are partitioned, resulting communication delays must be taken into account, affecting performance. Limitations on interconnection size must be taken into account, affecting performance as well. Pin and interconnection limitations force the multiplexing of inputs and outputs, reducing performance and sometimes affecting cost. Power consumption must also be taken into account. Power balancing between partitions and total power consumption might both be considerations. To meet market windows, system partitions can facilitate the use of COTS, programmable components, or easily fabricated components such as gate arrays. To meet cost constraints, functions that are found in the same partition might share partition resources. Such functions or tasks cannot execute concurrently, affecting performance.

Partitioning is widely used at the logic level as well as on physical designs. In these cases, much more information is known about the design properties and the interconnection structure has been determined. System partitioning is performed when information about properties of the specific components might be uncertain, and the interconnection structure undetermined. For these reasons, techniques used at lower levels must be modified to include predictions of design properties not yet known and prediction of the possible interconnection structure as a result of the partitioning.

The exact partitioning method used depends on the type of specification available. If detailed CDFG or HDL specifications are used, the partitioning method might be concerned with which register-transfer functions (e.g., add, multiply, and shift) are found in each partition. If the specification primitives are tasks, as in a task-flow graph specification, then the tasks must be assigned to partitions. Generally, the more detailed the specification, the larger the size of the partitioning problem. Powerful partitioning methods can be applied to problems of small size ($n < 100$). Weaker methods such as incremental improvement must be used when the problem size is larger.

Partitioning methods can be based on *constructive* partitioning or *iterative improvement*. Constructive partitioning involves taking an unpartitioned design and assigning operations or tasks to partitions. Basic constructive partitioning methods include *bin packing* using a first-fit decreasing heuristic, *clustering* operations into partitions by assigning nearest neighbors to the same partition until the partition is full, *random* placement into partitions, and *integer programming* approaches.

11.3.1 Constructive Partitioning Techniques

Bin packing involves creating a number of bins equal in number to the number of partitions desired and equal in size to the size of partitions desired. One common approach involves the following steps: the tasks or operations are sorted by size. The largest task in the list is placed in the first bin, and then the next largest is placed in the first bin, if it will fit, or else into the second bin. Each task is placed into the first bin in which it will fit, until all tasks have been placed in bins. More bins are added if necessary. This simple heuristic is useful to create an initial set of partitions to be improved iteratively later.

Clustering is a more powerful method to create partitions. Here is a simple clustering heuristic. Each task is ranked by the extent of ''connections'' to other tasks either owing to control flow, data flow, or physical position limitations. The most connected task is placed in the first partition and then the tasks connected to it are placed in the same partition, in the order of the strength of their connections to the

first task. Once the partition is full, the task with the most total connections remaining outside a partition is placed in a new partition, and other tasks are placed there in the order of their connections to the first task. This heuristic continues until all tasks are placed.

Random partitioning places tasks into partitions in a greedy fashion until the partitions are full. Some randomization of the choice of tasks is useful in producing a family of systems, each member of which is partitioned randomly. This family of systems can be used successfully in iterative improvement techniques for partitioning, as described later in this section.

The most powerful technique for constructive partitioning is mathematical programming. Integer and mixed integer-linear programming (MILP) techniques have been used frequently in the past for partitioning. Such powerful techniques are computationally very expensive and are successful only when the number of objects to be partitioned is small. The basic idea behind integer programming used for partitioning is the following: an integer, TP(i, j), is used to represent the assignment of tasks to partitions. When TP = 1, task i is assigned to partition j. For each task in this problem, there would be an equation

$$\sum_{j=1}^{\text{Partition total}} \text{TP}(i,j) = 1 \tag{11.1}$$

This equation states that each task must be assigned to one and only one partition. There would be many constraints of this type in the integer program, some of which were inequalities. There would be one function representing cost, performance, or other design property to be optimized. The simultaneous solution of all constraints, given some minimization or maximization goal, would yield the optimal partitioning.

Apart from the computational complexity of this technique, the formulation of the mathematical programming constraints is tedious and error prone if performed manually. The most important advantage of mathematical programming formulations is the discipline it imposes on the CAD programmer in formulating an exact definition of the CAD problem to be solved. Such problem formulations can prove useful when applied in a more practical environment, as described in Section 11.3.2.

11.3.2 Iterative Partitioning Techniques

Of the many iterative partitioning techniques available, two have been applied most successfully at the system level. These are *min-cut partitioning*, first proposed by Kernigan and Lin, and *genetic algorithms*.

Min-cut partitioning involves exchanging tasks or operations between partitions to minimize the total amount of ''interconnections'' cut. The interconnections can be computed as the sum of data flowing between partitions, or as the sum of an estimate of the actual interconnections that will be required in the system. The advantage of summing the data flowing is that it provides a quick computation, since the numbers are contained in the task-flow graph. Better partitions can be obtained if the required physical interconnections are taken into account since they are related more directly to cost and performance than is the amount of data flowing. If a partial structure exists for the design, predicting the unknown interconnections allows partitioning to be performed on a mixed design, one that contains existing parts as well as parts under design.

Genetic algorithms, highly popular for many engineering optimization problems, are especially suited to the partitioning problem. The problem formulation is similar in some ways to mathematical programming formulations. A simple genetic algorithm for partitioning is described here. In this example, a chromosome represents each partitioned system design, and each chromosome contains genes, representing information about the system. A particular gene TP(i, j) might represent the fact that task i is contained in partition j when it is equal to 1, and is set to 0 otherwise. A family of designs created by some constructive partitioning technique then undergoes mutation and crossover as new designs evolve.

A *fitness function* is used to check the quality of the design and the evolution is halted when the design is considered fit or when no improvement has occurred after some time. In the case of partitioning, the fitness function might include the estimated volume of interconnections, the predicted cost or performance of the system, or other system properties.

The reader might note some similarity between the mathematical programming formulation of the partitioning problem presented here and the genetic algorithm formulation. This similarity allows the CAD developer to create a mathematical programming model of the problem to be solved, find optimal solutions to small problems, and then create a genetic algorithm version. The genetic algorithm version can be checked against the optimal solutions found by the mathematical program. However, genetic algorithms can take into account many more details than can mathematical program formulations, can handle nonlinear relationships better, and can even handle *stochastic parameters*.*

Partitioning is most valuable when there is a mismatch between the sizes of system tasks and the capacities of system modules. When the system tasks and system modules are more closely matched, then the system design can proceed directly to *scheduling* and *allocating* tasks to processing modules.

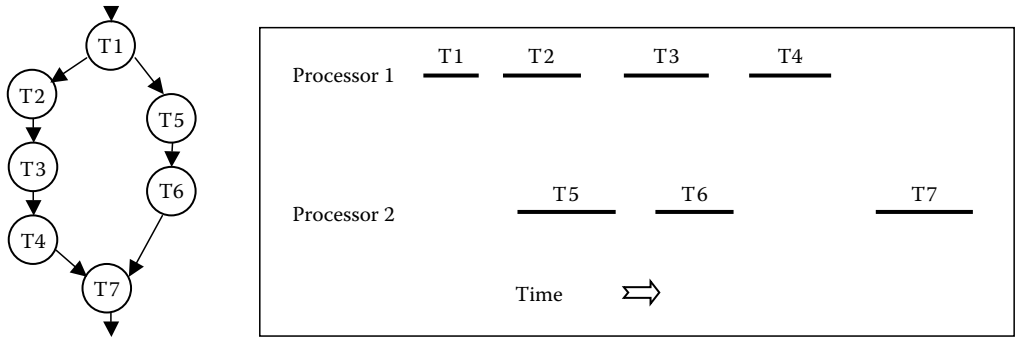
11.4 Scheduling and Allocating Tasks to Processing Modules

Scheduling and allocating tasks to processing modules involves the determination of how many processing modules are required, which modules execute which tasks, and the order in which the tasks are processed by the system. In the special case where only a single task is processed by each module, the scheduling becomes trivial. Otherwise, if the tasks share modules, the order in which the tasks are processed by the modules can affect system performance or cost. If the tasks are ordered inappropriately, some tasks might wait too long for input data, and performance might be affected. Alternatively, to meet performance constraints, additional modules must be added to perform more tasks in parallel, increasing system cost.

A variety of modules might be available to carry out each task, with differing cost and performance parameters. As each task is allocated to a module, that module is selected from a set of modules available to execute the task. This is analogous to the task *module selection*, which occurs as part of high-level synthesis. For the system design problem considered here, the modules can be general-purpose processors, special-purpose processors (e.g., signal-processing processors), or special-purpose hardware. If all (or most) modules used are general purpose, the systems synthesized are known as heterogeneous application-specific multiprocessors.

A variety of techniques can be used for scheduling and allocation of system tasks to modules. Just as with partitioning, these techniques can be constructive or iterative. Constructive scheduling techniques for system tasks include greedy techniques such as ASAP (as soon as possible) and ALAP (as late as possible). In ASAP scheduling, the tasks are scheduled as early as possible on a free processing module. The tasks scheduled first are the ones with the longest paths from their outputs to final system outputs or system completion. Such techniques, with variations, can be used to provide starting populations of system designs to be further improved iteratively. The use of such greedy techniques for system synthesis differs from the conventional use in high-level synthesis, where the system is assumed to be synchronous, with tasks scheduled into time steps. System task scheduling assumes no central clock, and tasks take a wide range of time to complete. Some tasks could even complete stochastically, with completion time being a random variable. Other tasks could complete basic calculations in a set time, but could perform a finer grain (more accurate) of computations if more time were available. A simple task-flow graph is shown in Figure 11.2, along with a Gantt chart illustrating the ASAP scheduling of tasks onto two processors. Note that two lengthy tasks are performed in parallel with three shorter tasks and that no two tasks take the same amount of time.

* Stochastic parameters represent values that are uncertain. There is a finite probability of a parameter taking a specific value that varies with time, but that probability is less than 1, in general.



(a) Task-flow graph (b) Gantt chart showing schedule

FIGURE 11.2 Example of task-flow graph and schedule.

Similar to partitioning, scheduling, allocation, and module selection, can be performed using mathematical programming. In this case, since the scheduling is asynchronous, time becomes a linear rather than integer quantity. Therefore, MILP is employed to model system-level scheduling and allocation. A typical MILP timing constraint is the following:

$$T_{\mathrm{OA}}(i) + C_{\mathrm{delay}} \leq T_{\mathrm{IR}}(j) \tag{11.2}$$

where

$T_{\mathrm{OA}}(i)$ is the time the output is available from task i
C_{delay} is the communication delay
$T_{\mathrm{IR}}(j)$ is the time the input is required by task j

Unfortunately, the actual constraints used in scheduling and allocation are mostly more complex than this, because the design choices have yet to be made. Here is another example:

$$T_{\mathrm{OA}}(i) \geq T_{\mathrm{IR}}(i) + \sum_{k} \left[P_{\mathrm{delay}}(k) * M(i, k)\right] \tag{11.3}$$

This constraint states that the time an output from task i is available is greater than or equal to the time necessary inputs are received by task i, and a processing delay P_{delay} has occurred. $M(i, k)$ indicates that task i is allocated to module k. P_{delay} can take on a range of values, depending on which of the k modules is being used to implement task i. The summation is actually a linearized select function that picks the value of P_{delay} to use depending on which value of $M(i, k)$ is set to 1.

As with partitioning, mathematical programming for scheduling and allocation is computationally intensive, and impractical for all but the smallest designs, but it does provide a baseline model of design that can be incorporated into other tools.

The most frequent technique used for iterative improvement in scheduling and allocation at the system level is a genetic algorithm. The genes can be used to represent task allocation and scheduling. To represent asynchronous scheduling accurately, time is generally represented as a linear quantity in such genes rather than an integer quantity.

11.5 Allocating and Scheduling Storage Modules

In digital systems, all data require some form of temporary or permanent storage. If the storage is shared by several data sets, the use of the storage by each data set must be scheduled. The importance of this task in system design has been overlooked in the past, but has now become an important system-level task.

Modern digital systems usually contain some multimedia tasks and data. The storage requirements for multimedia tasks sometimes result in systems where processing costs are dwarfed by storage costs, particularly caching costs. For such systems, storage must be scheduled and allocated either during or after task scheduling and allocation. If storage is scheduled and allocated concurrently with task scheduling and allocation, the total system costs are easier to determine and functional module sharing can be increased if necessary to control total costs. Alternatively, if storage allocation and scheduling are performed after task scheduling and allocation, then both programs are simpler, but the result may not be as close to optimal.

Techniques similar to those used for task scheduling and allocation can be used for storage scheduling and allocation.

11.6 Selecting Implementation and Packaging Styles for System Modules

Packaging styles can range from single-chip dual-in-line packages (DIPs) to MCMs, boards, racks, and cases. Implementation styles include general-purpose processor, special-purpose programmable processor (e.g., signal processor), COTS modules, field programmable gate arrays (FPGAs), gate array, standard cell, and custom ICs. System cost, performance, power, and design time constraints determine selection of implementation and packaging styles for many system designs. Tight performance constraints favor custom ICs, packaged in MCMs. Tight cost constraints favor off-the-shelf processors and gate array implementations, with small substrates and inexpensive packaging. Tight power constraints favor custom circuits. Tight design time constraints favor COTS modules and FPGAs. If a single design property has high priority, the designer can select the appropriate implementation style and packaging technology. If, however, design time is crucial, but the system to be designed must process video signals in real time, then trade-offs in packaging and implementation style must be made. The optimality of system cost and power consumption might be sacrificed: the entire design might be built with FPGAs, with much parallel processing and at great cost and large size. Because time-to-market is so important, early market entry systems may sacrifice the optimality of many system parameters initially, and then improve them in the next version of the product.

Selection of implementation styles and packaging can be accomplished by adding some design parameters to the scheduling and allocation program, if that program is not already computationally intensive. The parameters added would include a variable indicating that a particular

- Functional module was assigned a certain implementation style
- Storage module was assigned a certain implementation style
- Functional module was assigned a certain packaging style
- Storage module was assigned a certain packaging style

Some economy of processing could be obtained if certain implementation styles precluded certain packaging styles.

11.7 Interconnection Strategy

Modules in a digital system are usually interconnected in some carefully architected, consistent manner. If point-to-point interconnections are used, they are used throughout the system, or in a subsystem. In the same manner, buses are not broken arbitrarily to insert point-to-point connections or rings. For this reason, digital system design programs usually assume an interconnection style and determine the system performance relative to that style. The most common interconnection styles are bus, point-to-point, and ring.

11.8 Word-Length Determination

Functional specifications for system tasks are frequently detailed enough to contain the algorithm to be implemented. To determine the implementation costs of each system task, knowledge of the word widths to be used is important as system cost varies almost quadratically with word width.

Tools to automatically select task word width are currently experimental, but the potential for future commercial tools exists.

In typical hardware implementations of an arithmetic-intensive algorithm, designers must determine the word lengths of resources such as adders, multipliers, and registers. Wadekar and Parker [6] in a recent publication, present algorithm-level optimization techniques to select distinct word lengths for each computation. These techniques meet the desired accuracy and minimize the design cost for the given performance constraints. The cost reduction is possible by avoiding unnecessary bit-level computations that do not contribute significantly to the accuracy of the final results. At the algorithm level, determining the necessary and sufficient precision of an individual computation is a difficult task since the precision of various predecessor/successor operations can be traded off to achieve the same desired precision in the final result. This is achieved using a mathematical model [7] and a genetic selection mechanism [6]. There is a distinct advantage to word-length optimization at the algorithmic level. The optimized operation word lengths can be used to guide high-level synthesis or designers to achieve an efficient utilization of resources of distinct word lengths and costs. Specifically, only a few resources of larger word lengths and high cost may be needed for operations requiring high precision to meet the final accuracy requirement. Other relatively low-precision operations may be executed by resources of smaller word lengths. If there is no timing conflict, a large word-length resource can also execute a small word-length operation, thus improving the overall resource utilization further. These high-level design decisions cannot be made without the knowledge of word lengths prior to synthesis.

11.9 Predicting System Characteristics

In system-level design, early prediction gives designers the freedom to make numerous high-level choices (such as die size, package type, and latency of the pipeline) with confidence that the final implementation will meet power and energy as well as cost and performance constraints. These predictions can guide power budgeting and subsequent synthesis of various system components, which is critical in synthesizing systems that have low power dissipation, or long battery life. The use by synthesis programs of performance and cost lower bounds allows smaller solution spaces to be searched, which leads to faster computation of the optimal solution.

System cost, performance, power consumption, and design time can be computed if the properties of each system module are known. System design using existing modules requires little prediction. If system design is performed prior to design of any of the contained system modules, however, their properties must be predicted or estimated. Owing to the complexities of prediction techniques, describing these techniques is a subject worthy of an entire chapter. A brief survey of related readings is found in Section 11.10.

The register-transfer and subsequent lower-level power prediction techniques such as gate- and transistor-level techniques are essential for validation before fabricating the circuit. However, these techniques are less efficient for system-level design as a design must be generated before prediction can be done.

11.10 Survey of Research in System Design

Many researchers have investigated the problem of system design, dating back to the early 1970s. This section highlights work that is distinctive, along with tutorial articles covering relevant topics.

Much good research is not referenced here, and the reader is reminded that the field is dynamic, with new techniques and tools appearing almost daily.

Issues in top-down versus bottom-up design approaches were highlighted in the design experiment reported by Gupta et al. [8].

11.10.1 System Specification

System specification has received little attention historically except in the specific area of software specifications. Several researchers have proposed natural language interfaces capable of processing system specifications and creating internal representations of the systems that are considerably more structured. Of note is the work by Granacki [9] and Cyre [10]. One noteworthy approach is the design specification language (DSL), found in the design analysis and synthesis environment [11]. One of the few books on the subject concerns the design of embedded systems—systems with hardware and software designed for a particular application set [12]. In one particular effort, Petri nets were used to specify the interface requirements in a system of communicating modules, which were then synthesized [13]. The SIERA system designed by Srivastava, Richards, and Broderson [14] supports specification, simulation, and interactive design of systems.

The Rugby model [15] represents hardware/software systems and the design process, using four dimensions to represent designs: time, computation, communication, and data. da Silva [16] describes the system data structure (SDS), used for internal representation of systems. da Silva also presents an external language for system descriptions called OSCAR. SDS is general and comprehensive, covering behavior and structure. OSCAR is a visual interface to SDS with a formal semantics and syntax based on a visual coordination language paradigm. It is used to capture the behavior and the coordination aspects of concurrent and communicating processes.

SystemC [17] provides hardware-oriented constructs as a class library implemented in standard C++. Although the use of SystemC is claimed to span design and verification from concept to implementation in hardware and software, the constructs provided are somewhat of lower level than most of the system design activities described in this chapter. Extensions to VHDL and Verilog also support some system design activities at the lower levels of system design.

11.10.2 Partitioning

Partitioning research covers a wide range of system design situations. Many early partitioning techniques dealt with assigning register-level operations to partitions. APARTY, a partitioner designed by Lagnese and Thomas, partitions CDFG designs for single-chip implementation to obtain efficient layouts [18]. Vahid [19] performed a detailed survey of techniques for assigning operations to partitions. CHOP assigns CDFG operations to partitions for multichip design of synchronous, common clocked systems [20]. Vahid and Gajski developed an early partitioner, SpecPart, which assigns processes to partitions [21]. Chen and Parker reported on a process-to-partition technique called ProPart [22].

11.10.3 Nonpipelined Design

Although research on *system design* spans more than two decades, most of the earlier works focus on single aspects of design-like task assignment, and not on the entire design problem. We cite some representative works here. These include graph theoretical approaches to task assignment [23,24], analytical modeling approaches for task assignment [25], and probabilistic modeling approaches for task partitioning [26,27], scheduling [28], and synthesis [29]. Two publications of note cover application of heuristics to system design [30,31].

Other publications of note include mathematical programming formulations for task partitioning [32] and communication channel assignment [33]. Early efforts include those done by soviet researchers since

the beginning of the 1970s such as Linsky and Kornev [34] and others, where each model only included a subset of the entire synthesis problem. Chu et al. [35] published one of the first MILP models for a subproblem of system-level design, scheduling. The program Synthesis of Systems (SOS) including a compiler for MILP models [36,37] was developed, based on a comprehensive MILP model for system synthesis. SOS takes a description of a system described using a task-flow graph, a processor library, and some cost and performance constraints, and generates an MILP model to be optimized by an MILP solver. The SOS tool generates MILP models for the design of nonperiodic (nonpipelined) heterogeneous multiprocessors. The models share a common structure, which is an extension of the previous work by Hafer and Parker for high-level synthesis of digital systems [38].

Performance bounds of solutions found by algorithms or heuristics for system-level design are proposed in many papers, including the landmark papers by Fernandez and Bussel [39], Garey and Graham [40], and more recent publications [41].

The work of Gupta et al. [8] reported the successful use of system-level design tools in the development of an application-specific heterogeneous multiprocessor (ASHM) for image processing. Gupta and Zorian [42] describe the design of systems using cores, silicon cells with at least 5000 gates. The same issue of *IEEE Design and Test of Computers* contains a number of useful articles on the design of embedded core-based systems. Li and Wolf [43] report on a model of hierarchical memory and a multiprocessor synthesis algorithm, which takes into account the hierarchical memory structure.

A major project, RASSP, is a rapid-prototyping approach whose development is funded by the US Department of Defense [44]. RASSP addresses the integrated design of hardware and software for signal-processing applications.

An early work on board-level design, MICON, is of particular interest [45]. Other research results solving similar problems with more degrees of design freedom include the research by Chen [46] and Heo [47]. GARDEN, written by Heo, finds the design with the shortest estimated time-to-market that meets cost and performance constraints.

All the MILP synthesis works cited up to this point address only the nonperiodic case.

Synthesis of ASHMs is a major activity in the general area of system synthesis. One of the most significant system-level design efforts is Lee's Ptolemy project at the University of California, Berkeley. Representative publications include papers by Lee and Bier describing a simulation environment for signal processing [48] and the paper by Kalavede et al. in 1995 [49]. Another prominent effort is the SpecSyn project of Gajski et al. [50] which is a system-level design methodology and framework.

11.10.4 Macropipelined Design

Macropipelined (periodic) multiprocessors execute tasks in a pipelined fashion, with tasks executing concurrently on different sets of data. Most research work on design of *macropipelined* multiprocessors has been restricted to homogeneous multiprocessors having negligible communication costs. This survey divides the previous contributions according to the execution mode: preemptive or *nonpreemptive*.

11.10.4.1 Nonpreemptive Mode

The nonpreemptive mode of execution assumes that each task is executed without interruption. It is used quite often in low-cost implementations. Much research has been performed on system scheduling for the nonpreemptive mode. A method to compute the minimum possible value for the initiation interval for a task-flow graph given an unlimited number of processors and no communication costs was found by Renfors and Neuvo [51].

Wang and Hu [52] use heuristics for the allocation and *full static scheduling* (meaning that each task is executed on the same processor for all iterations) of generalized perfect-rate task-flow graphs on homogeneous multiprocessors. Wang and Hu apply planning, an artificial intelligence method, to the task scheduling problem. The processor allocation problem is solved using a *conflict-graph* approach.

Gelabert and Barnwell [53] developed an optimal method to design macropipelined homogeneous multiprocessors using *cyclic-static scheduling*, where the task-to-processor mapping is not time-invariant as in the full static case, but is periodic, i.e., the tasks are successively executed by all processors. Gelabert and Barnwell assume that the delays for *intraprocessor* and *interprocessor* communications are the same, which is an idealistic scenario. Their approach is able to find an optimal implementation (minimal iteration interval) in exponential time in the worst case.

Tirat-Gefen [54], in his doctoral thesis, extended the SOS MILP model to solve for optimal macropipelined ASHMs. He also proposed an ILP model allowing simultaneous optimal retiming and processor/module selection in high- and system-level synthesis [55].

Verhauger [56] addresses the problem of periodic multidimensional scheduling. His thesis uses an ILP model to handle the design of homogeneous multiprocessors without communication costs implementing data-flow programs with nested loops. His work evaluates the complexity of the scheduling and allocation problems for the multidimensional case, which were both found to be NP-complete. Verhauger proposes a set of heuristics to handle both problems.

Passos et al. [57] evaluate the use of multidimensional retiming for synchronous data-flow graphs. However, their formalism can only be applied to homogeneous multiprocessors without communication costs.

11.10.4.2 Preemptive Mode of Execution

Peng and Shin [58] address the optimal static allocation of periodic tasks with precedence constraints and preemption on a homogeneous multiprocessor. Their approach has an exponential time complexity. Ramamritham [59] developed a heuristic method that has a more reasonable computational cost. Rate-monotonic scheduling (RMS) is a commonly used method for allocating periodic real-time tasks in distributed systems [60]. The same method can be used in homogeneous multiprocessors.

11.10.5 Genetic Algorithms

Genetic algorithms are becoming an important tool for solving the highly nonlinear problems related to system-level synthesis. The use of genetic algorithms in optimization is well discussed by Michalewicz [61] where formulations for problems such as bin packing, processor scheduling, traveling salesman, and system partitioning are outlined.

Research works involving the use of genetic algorithms to system-level synthesis problems are starting to be published, as for example, the results of

- Hou et al. [62]—scheduling of tasks in a homogeneous multiprocessor without communication costs
- Wang et al. [63]—scheduling of tasks in heterogeneous multiprocessors with communication costs, but not allowing cost versus performance trade-off, i.e., all processors have the same cost
- Ravikumar and Gupta [64]—mapping of tasks into a reconfigurable homogeneous array processor without communication costs
- Tirat-Gefen and Parker [65]—a genetic algorithm for design of ASHMs with nonnegligible communications costs specified by a nonperiodic task-flow graph representing both control and data flow
- Tirat-Gefen [54]—introduced a full-set of genetic algorithms for system-level design of ASHMs incorporating new design features such as imprecise computation and probabilistic design

11.10.6 Imprecise Computation

The main results in imprecise *computation* theory are due to Liu et al. [66] who developed polynomial time algorithms for optimal scheduling of preemptive tasks on homogeneous multiprocessors without

communications costs. Ho et al. [67] proposed an approach to minimize the total error, where the error of a task being imprecisely executed is proportional to the amount of time that its optional part was not allowed to execute, i.e., the time still needed for its full completion. Polynomial time-optimal algorithms were derived for some instances of the problem [66].

Tirat-Gefen et al. [68] presented in 1997 a new approach for ASHM design that allows trade-offs between cost, performance, and data-quality through incorporation of imprecise computation into the system-level design cycle.

11.10.7 Probabilistic Models and Stochastic Simulation

Many probabilistic models for solving different subproblems in digital design have been proposed recently. The problem of task and data-transfer scheduling on a multiprocessor when some tasks (data transfers) have nondeterministic execution times (communication times) can be modeled by PERT networks, which were introduced by Malcolm et al. [69] along with the critical path method (CPM) analysis methodology.

A survey on PERT networks and their generalization to conditional PERT networks is done by Elmaghraby [70]. In system-level design, the completion time of a PERT network corresponds to the *system latency*, whose cumulative distribution function (cdf) is a nonlinear function of the probability density distributions of the computation times of the tasks and the communication times of the data transfers in the task-flow graph.

The exact computation of the cdf of the completion time is computationally expensive for large PERT networks, therefore it is important to find approaches that approximate the value of the expected time of the completion time and its cdf. One of the first of these approaches was due to Fulkerson [71], who derived an algorithm in 1962 to find a tight estimate (lower bound) of the expected value of the completion time. Robillard and Trahan [72] proposed a different method using the characteristic function of the completion time in approximating the cdf of the completion time.

Mehrotra et al. [73] proposed a heuristic for estimating the moments of the probabilistic distribution of the system latency t_c. Kulkarni and Adlakha [74] developed an approach based on Markov processes for the same problem. Hagstrom [75] introduced an exact solution for the problem when the random variables modeling the computation and communication times are finite discrete random variables. Kamburowski [76] developed a tight upper bound on the expected completion time of a PERT network.

An approach using random graphs to model distributed computations was introduced by Indurkhya et al. [26], whose theoretical results were improved by Nicol [27]. Purushotaman and Subrahmanyam [77] proposed formal methods applied to concurrent systems with a probabilistic behavior. An example of modeling using queueing networks instead of PERT networks is given by Thomasian and Bay [78]. Estimating errors owing to the use of PERT assumptions in scheduling problems is discussed by Lukaszewicz [79].

Tirat-Gefen developed a set of genetic algorithms using stratified stochastic sampling allowing simultaneous probabilistic optimization of the scheduling and allocation of tasks and communications on ASHM with nonnegligible communication costs [54].

11.10.8 Performance Bounds Theory and Prediction

Sastry [80] developed a stochastic approach for estimation of wireability (routability) for gate arrays. Kurdahi [81] created a discrete probabilistic model for area estimation of VLSI chips designed according to a standard cell methodology. Küçükçakar [82] introduced a method for partitioning of behavioral specifications onto multiple VLSI chips using probabilistic area/performance predictors integrated into a package called BEST (behavioral estimation). BEST provides a range of prediction techniques that can be applied at the algorithm level and includes references to prior research. These predictors provide information required by Tirat-Gefen's system-level probabilistic optimization methods [54].

Lower bounds on the performance and execution time of task-flow graphs mapped to a set of available processors and communication links were developed by Liu and Liu [83] for the case of heterogeneous processors, but no communication costs and by Hwang et al. [84] for homogeneous processors with communication costs. Tight lower bounds on the number of processors and execution time for the case of homogeneous processors in the presence of communication costs were developed by Al-Mouhamed [85]. Yen and Wolf [86] provide a technique for performance estimation for real-time distributed systems.

At the system and register-transfer level, estimating power consumption by the interconnect is important [87]. Wadekar et al. [88] reported ''Freedom,'' a tool to estimate system energy and power that accounts for functional-resource, register, multiplexer, memory, input/output pads, and interconnect power. This tool employs a statistical estimation technique to associate low-level, technology-dependent, physical, and electrical parameters with expected circuit resources and interconnect. At the system level, Freedom generates predictions with high accuracy by deriving an accurate model of the load capacitance for the given target technology—a task reported as critical in high-level power prediction by Brand and Visweswariah [89]. Methods to estimate power consumption prior to high-level synthesis were also investigated by Mehra and Rabaey [90]. Liu and Svensson [91] reported a technique to estimate power consumption in CMOS VLSI chips. The reader is referred to an example of a publication that reports power prediction and optimization techniques at the register-transfer level [92].

11.10.9 Word-Length Selection

Many researchers studied word-length optimization techniques at the register-transfer level. A few example publications are cited here. These techniques can be classified as statistical techniques applied to digital filters [93], simulated annealing-based optimization of filters [94], and simulation-based optimization of filters, digital communication, and signal-processing systems [95]. Sung and Kum reported a simulation-based word-length optimization technique for fixed-point digital signal-processing systems [96]. The objective of these particular architecture-level techniques is to minimize the number of bits in the design that is related to, but not the same as the overall hardware cost.

11.10.10 Embedded Systems

Embedded systems are becoming ubiquitous. The main factor differentiating embedded systems from other electronic systems is the focus of attention on the application rather than on the system as a computing engine. Typically, the I/O in an embedded system is to end users, sensors, and actuators. Sensors provide information on environmental conditions, for example, and actuators control the mechanical portion of the system. In an autonomous vehicle, an embedded system inputs the vehicle's GPS location via a sensor, and outputs control information to actuators that control the acceleration, braking, and steering. Embedded systems are typically real-time systems, falling into the classes hard real-time systems, where failure is catastrophic, and soft real-time systems, where failure is not catastrophic.

In an embedded system, there is typically at least one general-purpose processor or microcontroller, and one or more coprocessors that are COTS chips, DSPs, FPGAs, or custom VLSI chips. The main tasks to be performed for embedded systems are partitioning into hardware/software tasks, assigning tasks to processors, scheduling the tasks, and simulation.

11.10.11 System on Chip and Network on Chip

The level of integration of microelectronics has allowed entire systems to be fabricated on a single IC. The physical design of on-chip processing elements called cores can be obtained from vendors. Thus, along with the traditional system design issues, there are issues of intellectual property (IP) for such cores.

System on Chip (SoC) tend to be used for embedded systems, where hardware/software codesign is a major activity.

Researchers and organizations are also examining the connection of large quantities of cores on a single chip, and proposing to interconnect the cores with various types of interconnection networks. Benini and DeMicheli [97] proposed the use of packet switched on-chip networks, and there has been much publication activity in this area since. Raghavan proposed a hierarchical, heterogeneous approach, and described an alternative network architecture with torus topology and token ring protocol [98].

References

1. *IEEE Standard VHDL Language Reference Manual*, IEEE Std. 1076, IEEE Press, New York, 1987.
2. Bhasker, J., *A Verilog HDL Primer*, Star Galaxy Press, Allentown, PA, 1997.
3. Bell, G. and Newell, A., *Computer Structures: Readings and Examples*, McGraw-Hill, New York, 1971.
4. Harel, D., Statecharts: A visual formalism for complex systems, *Science of Computer Programming*, 8, 231–274, 1987.
5. Vahid, F., Narayan, S., and Gajski, D.D., SpecCharts: A VHDL front-end for embedded systems, *IEEE Transactions on CAD*, 14, 694, 1995.
6. Wadekar, S.A. and Parker, A.C., Accuracy sensitive word-length selection for algorithm optimization, in *Proceedings of the International Conference on Circuit Design [ICCD]*, 1998, 54.
7. Wadekar, S.A. and Parker, A.C., Algorithm-level verification of arithmetic-intensive application-specific hardware designs for computation accuracy, in *Digest Third International High-Level Design Validation and Test Workshop*, 1998.
8. Gupta, P., Chen, C.T., DeSouza-Batista, J.C., and Parker, A.C., Experience with image compression chip design using unified system construction tools, in *Proceedings of the 31st Design Automation Conference*, 1994.
9. Granacki, J. and Parker, A.C., PHRAN—Span: A natural language interface for system specifications, in *Proceedings of the 24th Design Automation Conference*, 1987, 416.
10. Cyre, W.R., Armstrong, J.R., and Honcharik, A.J., Generating simulation models from natural language specifications, *Simulation*, 65, 239, 1995.
11. Tanir, O. and Agarwal, V.K., A specification-driven architectural design environment, *Computer*, 6, 26–35, 1995.
12. Gajski, D.D., Vahid, F., Narayan, S., and Gong, J., *Specification and Design of Embedded Systems*, Prentice Hall, Englewood Cliffs, NJ, 1994.
13. de Jong, G. and Lin, B., A communicating Petri net model for the design of concurrent asynchronous modules, *ACM/IEEE Design Automation Conference*, June 1994.
14. Srivastava, M.B., Richards, B.C., and Broderson, R.W., System-level hardware module generation, *IEEE Transactions on Very Large-Scale Integration [VLSI] Systems*, 3, 20, 1995.
15. Jantsch, A., Kumar, S., and Hemani, A., The RUGBY model: A conceptual frame for the study of modeling, analysis and synthesis concepts of electronic systems, in *Proceedings of the Conference on Design, Automation and Test in Europe*, Munich, Germany, 1999, 72–78.
16. da Silva, D., Jr., A comprehensive framework for the specification of hardware/software systems, PhD dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, December 2001.
17. Grötker, T., Liao, S., Martin, G., and Swan, S., *System Design with SystemC*, Kluwer Academic, Boston, MA, 2002.
18. Lagnese, E. and Thomas, D., Architectural partitioning for system level synthesis of integrated circuits, *IEEE Transactions on Computer-Aided Design*, 1991.
19. Vahid, F., A survey of behavioral-level partitioning systems, Technical Report TR ICS 91–71, University of California, Irvine, 1991.

20. Kucukcakar, K. and Parker, A.C., Chop: A constraint-driven system-level partitioner, in *Proceedings of the 28th Design Automation Conference*, 1991, 514.
21. Vahid, F. and Gajski, D.D., Specification partitioning for system design, in *Proceedings of the 29th Design Automation Conference*, 1992.
22. Parker, A.C., Chen, C.-T., and Gupta, P., Unified system construction, in *Proceedings of the SASIMI Conference*, 1993.
23. Bokhari, S.H., Assignment problems in parallel and distributed computing, Kluwer Academic Publishers, Dordrecht, 1987.
24. Stone, H.S. and Bokhari, S.H., Control of distributed processes, *Computer*, 11, 97, 1978.
25. Haddad, E.K., Optimal load allocation for parallel and distributed processing, Technical Report TR 89–12, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, April 1989.
26. Indurkhya, B., Stone, H.S., and Cheng, L.X., Optimal partitioning of randomly generated distributed programs, *IEEE Transactions on Software Engineering*, SE-12, 483, 1986.
27. Nicol, D.M., Optimal partitioning of random programs across two processors, *IEEE Transactions on Software Engineering*, 15, 134, 1989.
28. Lee, C.Y., Hwang, J.J., Chow, Y.C., and Anger, F.D., Multiprocessor scheduling with interprocessor communication delays, *Operations Research Letters*, 7, 141, 1988.
29. Tirat-Gefen, Y.G., Silva, D.C., and Parker, A.C., Incorporating imprecise computation into system-level design of application-specific heterogeneous multiprocessors, in *Proceedings of the 34th Design Automation Conference*, 1997.
30. DeSouza-Batista, J.C. and Parker, A.C., Optimal synthesis of application-specific heterogeneous pipelined multiprocessors, in *Proceedings of the International Conference on Application-Specific Array Processors*, 1994.
31. Mehrotra, R. and Talukdar, S.N., Scheduling of tasks for distributed processors, Technical Report DRC-18-68-84, Design Research Center, Carnegie-Mellon University, Pittsburgh, PA, December 1984.
32. Agrawal, R. and Jagadish, H.V., Partitioning techniques for large-grained parallelism, *IEEE Transactions on Computers*, 37, 1627, 1988.
33. Barthou, D., Gasperoni, F., and Schwiegelshon, U., Allocating communication channels to parallel tasks, *Environments and Tools for Parallel Scientific Computing*, Elsevier Science Publishers B.V., Amsterdam, 1993, 275.
34. Linsky, V.S. and Kornev, M.D., Construction of optimum schedules for parallel processors, *Engineering Cybernetics*, 10, 506, 1972.
35. Chu, W.W., Hollaway, L.J., and Efe, K., Task allocation in distributed data processing, *Computer*, 13, 57, 1980.
36. Prakash, S. and Parker, A.C., SOS: Synthesis of application-specific heterogeneous multiprocessor systems, *Journal of Parallel and Distributed Computing*, 16, 338, 1992.
37. Prakash, S., Synthesis of application-specific multiprocessor systems, PhD thesis, Department of Electrical Engineering and Systems, University of Southern California, Los Angeles, CA, January 1994.
38. Hafer, L. and Parker, A., Automated synthesis of digital hardware, *IEEE Transactions on Computers*, C-31, 93, 1981.
39. Fernandez, E.B. and Bussel, B., Bounds on the number of processors and time for multiprocessor optimal schedules, *IEEE Transactions on Computers*, C-22, 745, 1975.
40. Garey, M.R. and Graham, R.L., Bounds for multiprocessor scheduling with resource constraints, *SIAM Journal of Computing*, 4, 187, 1975.
41. Jaffe, J.M., Bounds on the scheduling of typed task systems, *SIAM Journal of Computing*, 9, 541, 1991.
42. Gupta, R. and Zorian, Y., Introducing core-based system design, *IEEE Design and Test of Computers*, October–December 15, 1997.

43. Li, Y. and Wolf, W., A task-level hierarchical memory model for system synthesis of multiprocessors, in *Proceedings of the Design Automation Conference*, 1997, 153.
44. *IEEE Design and Test*, Vol. 13, No. 3, Fall 1996, published by IEEE Computer Society, Washington, DC.
45. W. Birmingham and D. Siewiorek, MICON: A single board computer synthesis tool, in *Proceedings of the 21st Design Automation Conference*, 1984.
46. Chen, C-T., System-level design techniques and tools for synthesis of application-specific digital systems, PhD thesis, Department of Electrical Engineering and Systems, University of Southern California, Los Angeles, CA, January 1994.
47. Heo, D.H., Ravikumar, C.P., and Parker, A., Rapid synthesis of multi-chip systems, *in Proceedings of the 10th International Conference on VLSI Design*, 1997, 62.
48. Lee, E.A. and Bier, J.C., Architectures for statically scheduled dataflow, *Journal of Parallel and Distributed Computing*, 10, 333–348, 1990.
49. Kalavede, A., Pino, J.L., and Lee, E.A., Managing complexity in heterogeneous system specification, simulation and synthesis, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 1995.
50. Gajski, D.D., Vahid, F., and Narayan, S., A design methodology for system-specification refinement, in *Proceedings of the European Design Automation Conference*, 1994, 458.
51. Renfors, M. and Neuvo, Y., The maximum sampling rate of digital filters under hardware speed constraints, *IEEE Transactions on Circuits and Systems*, CAS-28, 196, 1981.
52. Wang, D.J. and Hu, Y.H., Multiprocessor implementation of real-time DSP algorithms, *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 3, 393, 1995.
53. Gelabert, P.R. and Barnwell, T.P., Optimal automatic periodic multiprocessor scheduler for fully specified flow graphs, *IEEE Transactions on Signal Processing*, 41, 858, 1993.
54. Tirat-Gefen, Y.G., Theory and practice in system-level of application-specific heterogeneous multiprocessors, PhD dissertation, Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, 1997.
55. CasteloVide-e-Souza, Y.G., Potkonjak, M. and Parker, A.C., Optimal ILP-based approach for throughput optimization using algorithm/architecture matching and retiming, *Proceedings of the 32nd Design Automation Conference*, June 1995.
56. Verhauger, W.F., Multidimensional periodic scheduling, PhD thesis, Eindhoven University of Technology, Holland, 1995.
57. Passos, N.L., Sha, E.H., and Bass S.C., Optimizing DSP flow-graphs via schedule-based multidimensional retiming, *IEEE Transaction on Signal Processing*, 44, 150, 1996.
58. Peng, D.T. and Shin, K.G., Static allocation of periodic tasks with precedence constraints in distributed real-time systems, in *Proceedings of the 9th International Conference of Distributed Computing*, 1989, 190.
59. Ramamritham, K., Allocation and scheduling of precedence-related periodic tasks, *IEEE Transactions on Parallel and Distributed Systems*, 6, 1995.
60. Ramamritham, K., Stankovic, J.A., and Shiah, P.F., Efficient scheduling algorithms for real-time multiprocessors systems, *IEEE Transaction on Parallel and Distributed Systems*, 1, 184, 1990.
61. Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, 1994.
62. Hou, E.S.H, Ansari, N., and Ren, H., A Genetic algorithm for multiprocessor scheduling, *IEEE Transactions on Parallel and Distributed Systems*, 5, 113, 1994.
63. Wang, L., Siegel, H.J., and Roychowdhury, V.P., A genetic-algorithm-based approach for task matching and scheduling in heterogeneous computing environments, in *Proceedings of the Heterogeneous Computing Workshop, International Parallel Processing Symposium*, 1996, 72.
64. Ravikumar, C.P. and Gupta, A., Genetic algorithm for mapping tasks onto a reconfigurable parallel processor, *IEEE Proceedings in Computing Digital Technology*, 142, 81, 1995.

65. Tirat-Gefen, Y.G. and Parker, A.C., MEGA: An approach to system-level design of application-specific heterogeneous multiprocessors, in *Proceedings of the Heterogeneous Computing Workshop, International Parallel Processing Symposium*, 1996, 105.
66. Liu, J.W.S., Lin, K.-J., Shih, W.-K., Yu, A.C.-S., Chung, J.-Y., and Zhao, W., Algorithms for scheduling imprecise computations, *IEEE Computer*, 24, 58, 1991.
67. Ho, K., Leung, J.Y-T., and Wei, W-D., Minimizing maximum weighted error for imprecise computation tasks, Technical Report UNL-CSE-92-017, Department of Computer Science and Engineering, University of Nebraska, Lincoln, 1992.
68. Tirat-Gefen, Y.G., Silva, D.C., and Parker, A.C., Incorporating imprecise computation into system-level design of application-specific heterogeneous multiprocessors, in *Proceedings of the 34th Design Automation Conference*, 1997.
69. Malcolm, D.G., Roseboom, J.H., Clark, C.E., and Fazar, W., Application of a technique for research and development program evaluation, *Operations Research*, 7, 646, 1959.
70. Elmaghraby, S.E., The theory of networks and management science: Part II, *Management Science*, 17, B.54, 1970.
71. Fulkerson, D.R., Expected critical path lengths in pert networks, *Operations Research*, 10, 808, 1962.
72. Robillard, P. and Trahan, M., The completion time of PERT networks, *Operations Research*, 25, 15, 1977.
73. Mehrotra, K., Chai, J., and Pillutla, S., A study of approximating the moments of the job completion time in PERT networks, Technical Report, School of Computer and Information Science, Syracuse University, New York, 1991.
74. Kulkarni, V.G. and Adlakha, V.G., Markov and Markov-regenerative pert networks, *Operations Research*, 34, 769, 1986.
75. Hagstrom, J.N., Computing the probability distribution of project duration in a PERT network, *Networks*, 20, 231, 1990.
76. Kamburowski, J., An upper bound on the expected completion time of PERT networks, *European Journal of Operational Research*, 21, 206, 1985.
77. Purushothaman, S. and Subrahmanyam, P.A., Reasoning about probabilistic behavior in concurrent systems, *IEEE Transactions on Software Engineering*, SE-13, 740, 1987.
78. Thomasian, A., Analytic queueing network models for parallel processing of task systems, *IEEE Transactions on Computers*, C-35, 1045, 1986.
79. Lukaszewicz, J., On the estimation of errors introduced by standard assumptions concerning the distribution of activity duration in PERT calculations, *Operations Research*, 13, 326, 1965.
80. Sastry, S. and Parker, A.C., Stochastic models for wireability analysis of gate arrays, *IEEE Transactions on Computer-Aided Design*, CAD-5, 1986.
81. Kurdahi, F.J., Techniques for area estimation of VLSI layouts, *IEEE Transaction on Computer-Aided Design*, 8, 81, 1989.
82. Küçükçakar, K. and Parker, A.C., A methodology and design tools to support system-level VLSI design, *IEEE Transactions on Very Large-Scale Integration [VLSI] Systems*, 3, 355, 1995.
83. Liu, J.W.S. and Liu, C.L., Performance analysis of multiprocessor systems containing functionally dedicated processors, *Acta Informatica* 10, 95, 1978.
84. Hwang, J.J., Chow, Y.C., Ahnger, F.D., and Lee, C.Y., Scheduling precedence graphs in systems with interprocessor communication times, *SIAM Journal of Computing*, 18, 244, 1989.
85. Mouhamed, M., Lower bound on the number of processors and time for scheduling precedence graphs with communication costs, *IEEE Transactions on Software Engineering*, 16, 1990.
86. Yen, T.-Y. and Wolf, W., Performance estimation for real-time embedded systems, in *Proceedings of the International Conference on Computer Design*, 1995, 64.
87. Landman, P.E. and Rabaey, J.M., Activity-sensitive architectural power analysis, *IEEE Transactions on CAD*, 15, 571, 1996.

88. Wadekar, S.A., Parker, A.C., and Ravikumar, C.P., FREEDOM: Statistical behavioral estimation of system energy and power, in *Proceedings of the Eleventh International Conference on VLSI Design*, 1998, 30.
89. Brand, D. and C. Visweswariah, Inaccuracies in power estimation during logic synthesis, in *Proceedings of the European Design Automation Conference (EURO-DAC)*, 1996, 388.
90. Mehra, R. and Rabaey, J., Behavioral level power estimation and exploration, in *Proceedings of the First International Workshop on Low Power Design*, 1994, 197.
91. Liu, D. and Svensson, C., Power consumption estimation in CMOS VLSI chips, *IEEE Journal of Solid-State Circuits*, 29, 663, 1994.
92. Landman, P.E. and Rabaey, J.M., Activity-sensitive architectural power analysis, *IEEE Transactions on Computer-Aided Design*, 15, 571, 1996.
93. Zeng, B. and Neuvo, Y., Analysis of floating point roundoff errors using dummy multiplier coefficient sensitivities, *IEEE Transactions on Circuits and Systems*, 38, 590, 1991.
94. Catthoor, F., Vandewalle, J., and De Mann, H., Simulated annealing based optimization of coefficient and data word lengths in digital filters, *International Journal of Circuit Theory Applications*, 16, 371, 1988.
95. Grzeszczak, A., Mandal, M.K., Panchanathan, S., and Yeap, T., VLSI implementation of discrete wavelet transform, *IEEE Transactions on VLSI Systems*, 4, 421, 1996.
96. Sung, W. and Kum, Ki-II., Simulation-based word-length optimization method for fixed-point digital signal processing systems, *IEEE Transactions on Signal Processing*, 43, 3087, 1995.
97. Benini, L. and De Micheli, G., Networks on chip: A new paradigm for systems on chip design, in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2002, 418–419.
98. Raghavan, D., Extending the design space for networks on chip, MS thesis, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, May 2004.

12

Performance Modeling and Analysis Using VHDL and SystemC

12.1 Introduction **12**-1
Multilevel Modeling
12.2 ADEPT Design Environment **12**-9
Token Implementation • ADEPT Handshaking and Token Passing Mechanism • Module Categories • ADEPTs
12.3 Simple Example of an ADEPT Performance Model **12**-18
Three-Computer System • Simulation Results
12.4 Mixed-Level Modeling **12**-23
Mixed-Level Modeling Taxonomy • Interface for Mixed-Level Modeling with FSMD Components • Interface for Mixed-Level Modeling with Complex Sequential Components
12.5 Performance and Mixed-Level Modeling Using SystemC **12**-42
SystemC Background • SystemC Performance Models • Processor Model • Channels • Shared Bus Model • Fully Connected Model • Crossbar Model • SystemC Performance Modeling Examples • Mixed-Level Processor Model • Mixed-Level Examples • Random Data Generation • Mixed-Level Example Summary
12.6 Conclusions **12**-64
References **12**-64

Robert H. Klenke
Virginia Commonwealth University

Jonathan A. Andrews
Virginia Commonwealth University

James H. Aylor
University of Virginia

12.1 Introduction

It has been noted by the digital design community that the greatest potential for additional cost and iteration cycle time savings is through improvements in tools and techniques that support the early stages of the design process [1]. As shown in Figure 12.1, decisions made during the initial phases of a product's development cycle determine up to 80% of its total cost. The result is that accurate, fast analysis tools must be available to the designer at the early stages of the design process to help make these decisions. Design alternatives must be effectively evaluated at this level with respect to multiple metrics, such as performance, dependability, and testability. This analysis capability will allow a larger portion of the design space to be explored yielding higher quality as well as lower cost designs.

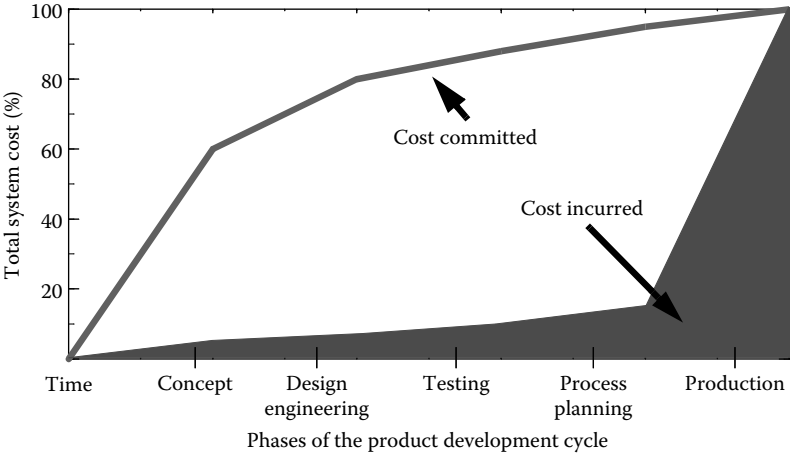


FIGURE 12.1 Product costs over the development cycle.

In 1979, Hill and vanCleemput [2] of the SABLE environment identified three or four stages or phases of design, and noted that each traditionally uses their own simulator. While the languages used have changed and some of the phases can be simulated together these phases still exist in most industry design flows: first an initial high-level simulation, then an intermediate level roughly at the instruction set simulation level, and then a gate-level phase. The authors add a potential fourth stage to deal with custom integrated circuit (IC) simulation. While acknowledging the potential for more efficient simulation at a given level due to potential optimization, they point out five key disadvantages to having different simulators and languages for the various levels. They are

1. Design effort is multiplied by the necessity of learning several simulator systems and recoding a design in each.
2. Possibility of error is increased as more human manipulation enters the system.
3. Each simulator operates at just one level of abstraction. Because it is impossible to simulate an entire computer at a low level of abstraction, only small fragments can be simulated at any one time.
4. Each fragment needs to be driven by a supply of realistic data and its output needs to be interpreted. Often, writing the software to serve these needs is more effortful than developing the design itself.
5. As the design becomes increasingly fragmented to simulate, it becomes difficult for any designer to see how his own particular piece fits into the overall function of the system.

While SABLE was developed almost three decades ago, it is worth noting that today designers are still struggling to address the same basic issues expressed above. Much of the industry has at least two or three design phases requiring separate models and simulators, which still results in a multiplication of design effort. This multiplication of design effort still increases the possibility of error due to inherently fallible human manipulation. While it may not be impossible to simulate an entire computer at one low level of abstraction, it is still impractical and inefficient. Thus, designs are often still fragmented for development and simulation, these fragments still need realistic data, and generating it is still time-consuming. Designers working on a fragment often have difficulty seeing how his or her piece fits into overall function, and this often leads to differing assumptions between fragments and expensive design revision.

There are a number of current tools and techniques that support analysis of these metrics at the system level to varying degrees. A major problem with these tools is that they are not integrated into the engineering design environment in which the system will ultimately be implemented. This problem leads

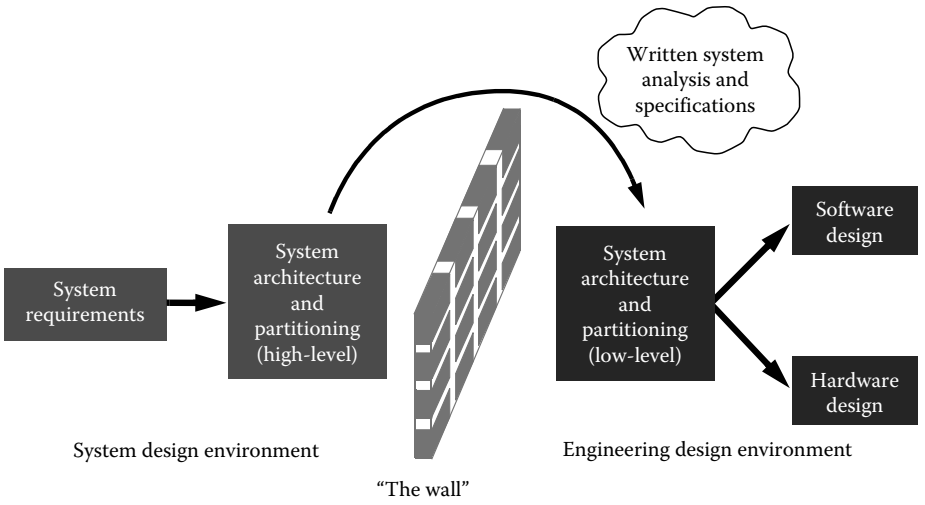


FIGURE 12.2 Disconnect between system-level design environments and engineering design environments.

to a major disconnect in the design process. Once the system-level model is developed and analyzed, the resulting high-level design is specified on paper and thrown ''over the wall'' for implementation by the engineering design team, as illustrated in Figure 12.2. As a result, the engineering design team has to often interpret this specification to implement the system, which often leads to design errors. It also has to develop their own initial ''high-level'' model from which to begin the design process in a top-down manner. Additionally, there is no automated mechanism by which feedback on design assumptions and estimations can be provided to the system design team by the engineering design team.

For systems that contain significant portions of both application-specific hardware and software executing on embedded processors, design alternatives for competing system architectures and hardware/software (HW/SW) partitioning strategies must be effectively and efficiently evaluated using high-level performance models. Additionally, the selected hardware and software system architecture must be refined in an integrated manner from the high-level models to an actual implementation to avoid implementation mistakes and the associated high redesign costs. Unfortunately, most existing design environments lack the ability to model and design a system's hardware and software in the same environment. A similar wall to that between the system design environment and the engineering design environment exists between the hardware and the software design environments. This results in a design path as shown in Figure 12.3, where the hardware and software design process begins with a common system requirement and specification, but proceeds through a separate and isolated design process until final system integration. At this point, assumptions on both sides may prove to be drastically wrong resulting in incorrect system function and poor system performance.

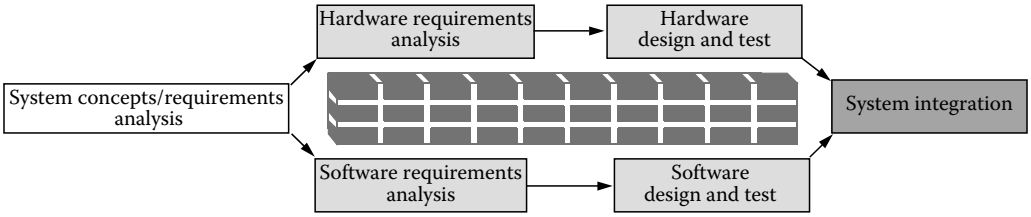


FIGURE 12.3 Current hardware/software system development methodology.

A unified, cooperative approach in which the hardware and software options can be considered together is required to increase the quality and decrease the design time for complex HW/SW systems. This approach is called *hardware/software codesign*, or simply *codesign* [3–5]. Codesign leads to more efficient implementations and improves overall system performance, reliability, and cost-effectiveness [5]. Also, because decisions regarding the implementation of functionality in software can impact hardware design (and vice versa), problems can be detected and changes made earlier in the development process [6].

Codesign can especially benefit the design of *embedded systems* [7], which contain hardware and software tailored for a particular application. As the complexity of these systems increases, the issue of providing design approaches that scale up to more complicated systems becomes of greater concern. A detailed description of a system can approach the complexity of the system itself [8], and the amount of detail present can make analysis intractable. Therefore, decomposition techniques and abstractions are necessary to manage this complexity.

What is needed is a design environment in which the capability for performance modeling of HW/SW systems at a high level of abstraction is fully integrated into the engineering design environment. To completely eliminate the ''over the wall'' problem and the resulting model discontinuity, this environment must support the incremental refinement of the abstract system-level performance model into an implementation-level model. Using this environment, a design methodology based on incremental refinement can be developed.

The design methodology illustrated in Figure 12.4 was proposed by Lockheed Martin Advanced Technology Laboratory as a new way to design systems [9]. This methodology uses the level of the risk of not meeting the design specifications as the metric for driving the design process. In this spiral-based design methodology, there are two iteration cycles. The major cycles (or spirals), denoted as Cycle 1, Cycle 2, . . . , Cycle *N* in the figure, correspond to the design iterations where major architectural changes are made in response to some specification metrics and the system as a whole is refined and more design detail is added to the model. Consistent with the new paradigm of system design, these iterations will actually produce virtual or simulation-based prototypes. A virtual prototype is simply a simulatable model of the system with stimuli described at a given level of design detail or design abstraction that describes the system's operation. Novel to this approach are the mini spirals. The mini spiral cycles, denoted by the levels on the figure labeled Systems, Architecture, and Detailed design, correspond to the refinement of only those portions of the design that are deemed to be ''high risk.'' High risk is obviously defined by the designer but is most often the situation where if one

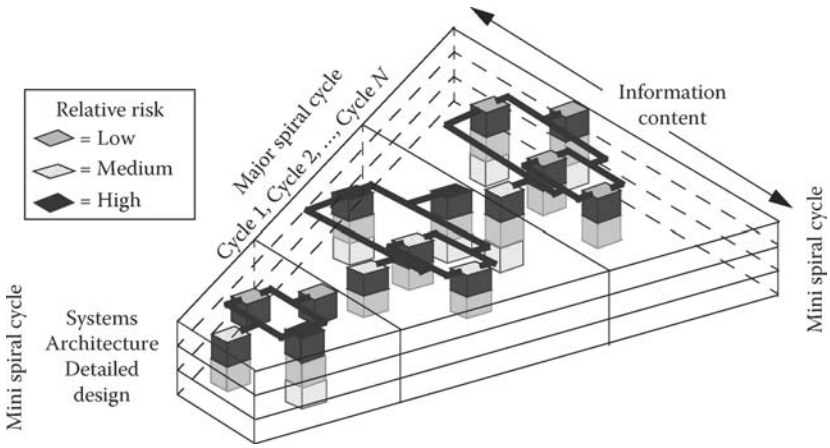


FIGURE 12.4 RDEIM.

or more of these components fail to meet their individual specifications, the system will fail to meet its specifications. The way to minimize the risk is to refine these components to possibly make an implementation so that the actual performance is known. Unlike the major cycles where the entire design is refined, the key to the mini spirals is the fact that only the critical portions of design are refined. For obvious reasons, the resulting models have been denoted as ''Risk Driven Expanding Information Models'' (RDEIMs).

The key to being able to implement this design approach is to be able to evaluate the overall design with portions of the system having been refined to a detailed level, while the rest of the system model remains at the abstract level. For example, in the first major cycle of Figure 12.4 the element with the highest relative risk is fully implemented (detailed design level) while the other elements are described at more abstract levels (system level or architectural level). If the simulation of the model shown in the first major cycle detects that the overall system will not meet its performance requirements, then the ''high risk'' processing element could be replaced by two similar elements operating in parallel. This result is shown in the second major cycle and at this point, another element of the system may become the new ''bottleneck,'' i.e., the highest relative risk, and it will be refined in a similar manner.

Implied in the RDEIM approach is a solution to the ''over the wall'' problem including hardware/software codesign. The proposed solution is to fully integrate performance modeling into the design process.

Obviously, one of the major capabilities necessary to implement a top-down design methodology such as the RDEIM is the ability to cosimulate system models which contain some components that are modeled at an abstract performance level (uninterpreted models) and some that are modeled at a detailed behavioral level (interpreted models). This capability to model and cosimulate uninterpreted models and interpreted models is called *mixed-level modeling* (sometimes referred to as hybrid modeling). Mixed-level modeling requires the development of interfaces that can resolve the differences between uninterpreted models that, by design, do not contain a representation of all of the data or timing information of the final implementation, and interpreted models which require most, or possibly all, data values and timing relationships to be specified. Techniques for systematic development of these mixed-level modeling interfaces and resolution of these differences in abstraction is the focus of some of the latest work in mixed-level modeling.

In addition to the problem of mixed-level modeling interfaces, another issue that may have to be solved is that of different modeling languages being used at different levels of design abstraction. While VHDL, and to some extent various extensions to Verilog, can be used to model at the system level, many designers constructing these types of models prefer to use a language with a more programming language-like syntax. As a result of this and other factors, the SystemC language [10] was developed. SystemC is a library extension to C++ that builds key concepts from existing hardware description languages (HDLs)—primarily a timing model that allows simulation of concurrent events—into an intrinsically object oriented HDL. Its use of objects, pointers, and other standard C++ items makes it a versatile language. Particularly the ability to describe and use complex data types, and the concepts of interfaces and channels make it much more intuitive for describing abstract behaviors than existing HDLs such as VHDL or Verilog. Since SystemC is C++, existing C or C++ code that models behavior can be imported with little or no modification.

12.1.1 Multilevel Modeling

The need for multilevel modeling was recognized almost three decades ago. Multilevel modeling implies that representations at different levels of detail coexist within a model [8,11,12]. Until the early 1990s, the term multilevel modeling was used for integrating behavioral or functional models with lower level models. The objective was to provide a continuous design path from functional models down to implementation. This objective was achieved and the VLSI industry utilizes it today. An example is the tool called Droid, developed by Texas Instruments [13].

The mixed-level modeling approach, as described in this chapter, is a specific type of multilevel modeling which integrates *performance* and *behavioral* models. Thus, only related research on multilevel modeling systems that spans both the performance and the functional/behavioral domains will be described.

Although behavioral or functional modeling is typically well understood by the design community, performance modeling is a foreign topic to most designers. Performance modeling, also called uninterpreted modeling, is utilized in the very early stages of the design process in evaluating such metrics as throughput and utilization. Performance models are also used to identify bottlenecks within a system and are often associated with the job of a system engineer. The term ''uninterpreted modeling'' reflects the view that performance models lack value-oriented data and functional (input/output) transformations. However, in some instances, this information is necessary to allow adequate analysis to be performed.

A variety of techniques have been employed for performance modeling. The most common techniques are Petri nets [14–16] and queuing models [17,18]. A combination of these techniques, such as a mixture of Petri nets and queuing models [19], has been utilized to provide more powerful modeling capabilities. All of these models have mathematical foundations. However, models of complex systems constructed using these approaches can quickly become unwieldy and difficult to analyze.

Examples of a Petri net and a queuing model are shown in Figure 12.5. A queuing model consists of queues and servers. Jobs (or customers) arrive at a specific arrival rate and are placed in a queue for service. These jobs are removed from the queue to be processed by a server at a particular service rate. Typically, the arrival and service rates are expressed using probability distributions. There is a queuing discipline, such as first-come-first-serve, which determines the order in which jobs are to be serviced. Once they are serviced, the jobs depart and arrive at another queue or simply leave the system. The number of jobs in the queues represents the model's state. Queueing models have been used successfully for modeling many complex systems. However, one of the major disadvantages of queuing models is their inability to model synchronization between processes.

As a system modeling paradigm, Petri nets overcome this disadvantage of queuing models. Petri nets consist of places, transitions, arcs, and a marking. The places are equivalent to conditions and hold tokens, which represent information. Thus, the presence of a token in the place of a Petri net corresponds to a particular condition being true. Transitions are associated with events, and the ''firing'' of a transition indicates that some event has occurred. A marking consists of a particular placement of tokens within the places of a Petri net and represents the state of the net. When a transition fires, tokens are removed from the input places and are added to the output places, changing the marking (the state) of the net and allowing the dynamic behavior of a Petri net to be modeled.

Petri nets can be used for performance analysis by associating a time with the transitions. Timed and stochastic Petri nets contain deterministic and probabilistic delays, respectively. Normally, these Petri nets are uninterpreted, since no *interpretation* (values or value transformations) are associated with the tokens or transitions. However, values or value transformations can be associated with the various elements of Petri net models as described below.

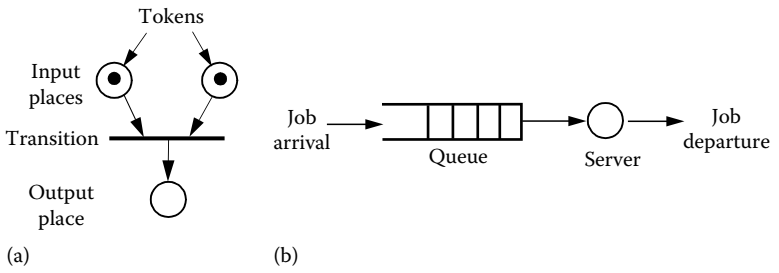


FIGURE 12.5 (a) Petri net and (b) queuing model.

Petri nets that have values associated with tokens are known as colored Petri nets (CPNs). In the colored Petri nets, each token has an attached ''color,'' indicating the identity of the token. The net is similar to the basic definition of the Petri net except that a functional dependency is specified between the color of the token and the transition firing action. In addition, the color of the token produced by a transition may be different from the color of the tokens on the input places. Colored Petri nets have an increased ability to efficiently model real systems with small nets which are equivalent to much larger plain Petri nets due to their increased descriptive powers.

Numerous multilevel modeling systems exist based on these two performance modeling techniques. Architecture Design and Assessment System (ADAS) is a set of tools specifically targeted for high-level design [20]. ADAS models both hardware and software using directed graphs based on timed Petri nets. The flow of information is quantified by identifying discrete units of information called tokens. ADAS supports two levels of modeling. The more abstract level is a dataflow description and is used for performance estimation. The less abstract level is defined as a behavioral level but still uses tokens which carry data structures with them. The functionality is embedded into the models using C or Ada programs. The capability of generating high-level VHDL models from the C or Ada models is provided. These high-level VHDL models can be further refined and developed in a VHDL environment but the refined models cannot be integrated back into the ADAS performance model. The flow of information in these high-level VHDL models is still represented by tokens. Therefore, implementation-level components cannot be integrated into an ADAS performance model. Another limitation is that all input values to the behavioral node must be contained within the token data structure.

Scientific and engineering software (SES)/Workbench is a design specification modeling and simulation tool [21]. It is used to construct and evaluate proposed system designs and to analyze their performance. A graphical interface is used to create a structural model which is then converted into a specific simulatable description (SES/sim). SES/Workbench enables the transition across domains of interpretation by using a *user node*, in which C-language and SES/sim statements can be executed. Therefore, SES/Workbench has similar limitations to ADAS; the inability to simulate a multilevel model when input values of behavioral nodes are not fully specified and the inadequacy of simulating components described as implementation-level HDLs (the capability of integrating VHDL models has been introduced later in the next paragraph). In addition, multiple simulation languages (both SES/sim and C) are required for multilevel models.

The Reveal interactor is a tool developed by Redwood Design Automation [22]. A model constructed in Reveal is aimed at the functional verification of RTL VHDL and Verilog descriptions and, therefore, does not include a separate transaction-based performance modeling capability. However, Reveal can work in conjunction with SES/Workbench. By mixing models created in Reveal and SES/Workbench, a multilevel modeling capability exists. Again, these multilevel models are very limited due to the fact that all the required information at the lower level part of the model must be available within the higher level model.

Integrated Design Automation System (IDAS) is a multilevel design environment which allows for rapid prototyping of systems [23]. Although the behavioral specifications need to be expressed as Ada, C, or FORTRAN programs, IDAS provides the capability of automatically translating VHDL description to Ada. However, the user cannot create abstract models in which certain behavior is unspecified. Also, it does not support classical performance models (such as queuing models and Petri nets) and forces the user to specify a behavioral description very early in the design process.

Transcend claims to integrate multilevel descriptions into a single environment [24,25]. In the more abstract level, T-flow models are used, in which tokens are used to represent flow of data. The capability of integrating VHDL submodels into a T-flow model is provided. However, interfacing between the two models requires a ''C++ like'' language, which map variables to/from VHDL signals, resulting in a heterogeneous simulation environment. Although their approach is geared toward the same objective as mixed-level modeling, the T-flow model must also include all the necessary data to activate the

VHDL submodels. Therefore, the upper-level model cannot be ''too abstract'' and must include lower level details.

Methodology for integrated design and simulation (MIDAS) supports the design of distributed systems via iterative refinement of partially implemented performance specification (PIPS) models [26]. A PIPS model is a partially implemented design where some components exist as simulation models and others as operational subsystems (i.e., implemented components). Although they use the term ''hybrid model'' in this context it refers to a different type of modeling. MIDAS is an ''integrated approach to software design'' [26]. It supports the performance evaluation of software being executed on a given machine. It does not allow the integration of components expressed in an HDL into the model.

The Ptolemy project is an academic research effort being conducted at the University of California at Berkeley [27,28]. Ptolemy, a comprehensive system prototyping tool, is actually constructed of multiple domains. Most domains are geared toward functional verification and have no notion of time. Each domain is used for modeling a different type of system. They also vary in the modeling level (level of abstraction). Ptolemy provides limited capability of mixing domains within one design. The execution of a transition across domains is accomplished with a ''wormhole.'' A wormhole is the mechanism for supporting the simulation of heterogeneous models. Thus, a multilevel modeling and analysis capability is provided. There are two major limitations to this approach compared with the mixed-level modeling approach being described. The first one is the heterogeneity—several description languages. Therefore, translation between simulators is required. The second one is that the interface between domains only translates data. Therefore, all the information required by the receiving domain must be generated by the transmitting domain.

Honeywell Technology Center (HTC) conducted a research effort that specifically addressed the mixed-level modeling problem [29]. This research had its basis in the UVa mixed-level modeling effort. The investigators at HTC developed a performance modeling library (PML) [30,31] and added a partial mixed-level modeling capability to this environment. The PML is used for performance models at a relatively low level of abstraction. Therefore, it assumes that all the information required by the interpreted element is provided by the performance model. In addition, their interface between uninterpreted and interpreted domains allows for bidirectional data flow.

Transaction-level modeling is a model paradigm that combines key concepts from interface-based design [32] with the system-level modeling to develop a refineable model of the system as a whole. Essentially, the system as a whole is represented with the model of computation done in components separated as much as possible from the model of communication between components.

As explained by Cai and Gajski [33],

> In a transaction-level model (TLM), the details of communication among computation components are separated from the details of computational components. Communication is modeled by channels, while transaction requests take place by calling interface functions of these channel models. Unnecessary details of communication and computation are hidden in a TLM and may be added later. TLMs speed up simulation and allow exploring and validating design alternatives at the higher level of abstraction.

There are a number of advantages to such an approach. Of course you get the advantages of a system-level model, but the separation of the modeling of the communication from the modeling of the computation also has a number of advantages. The primary advantage is that it allows the designer to focus on one aspect of the design at a time. The separation limits the complexity that the designer must contend with just as encapsulation and abstraction of component behaviors do. This also allows the designer to develop or refine the communication model of the system in the same way that they would do for the computation model.

To summarize, numerous multilevel modeling efforts exist. However, they are being developed, not addressing the issue of lack of information at the transition between levels of abstraction. The solution of

this problem is essential for true stepwise refinement of the performance models to behavioral models. In addition, integration of performance modeling level and behavioral modeling level was mostly performed by mixing different simulation environments, which results in heterogeneous modeling approach.

12.2 ADEPT Design Environment

A unified end-to-end design environment based solely on VHDL that allows designers to model systems from the abstract level to the gate level as described above has been developed. This environment supports the development of system-level models of digital systems that can be analyzed for multiple metrics like performance and dependability, and can then be used as a starting point for the actual implementation. A tool called Advanced design environment prototype tool (ADEPT) has been developed to implement this environment. ADEPT actually supports both system-level performance and dependability analysis in a common design environment using a collection of predefined library elements. ADEPT also includes the capability to simulate both system- and implementation-level (behavioral) models in a common simulation environment. This capability allows the stepwise refinement of system-level models into implementation-level models.

Two approaches to creating a unified design environment are possible. An evolutionary solution is to provide an environment that ''translates'' data from different models at various points in the design process and creates interfaces for the noncommunicating software tools used to develop these models. With this approach, users must be familiar with several modeling languages and tools. Also, analysis of design alternatives is difficult and is likely to be limited by design time constraints.

A revolutionary approach, the one being developed in ADEPT, is to use a single modeling language and mathematical foundation. This approach uses a common modeling language and simulation environment which decreases the need for translators and multiple models, reducing inconsistencies and the probability of errors in translation. Finally, the existence of a mathematical foundation provides an environment for complex system analysis using analytical approaches.

Simulators for hardware description languages accurately and conveniently represent the physical implementation of digital systems at the circuit, logic, register-transfer, and algorithmic levels. By adding a system-level modeling capability based on extended Petri nets and queuing models to the hardware description language, a single design environment can be used from concept to implementation. The environment would also allow for the mixed simulation of both *uninterpreted* (performance) models and *interpreted* (behavioral) models because of the use of a common modeling language. Although it would be possible to develop the high-level performance model and the detailed behavioral model in two different modeling languages and then develop some sort of ''translator'' or foreign language interface to hook them together, a better approach is to use a single modeling language for both models. A single modeling language that spans numerous design phases is much easier to use, encouraging more design analysis and consequently better designs.

ADEPT implements an end-to-end unified design environment based upon the use of the VHSIC hardware description language (VHDL), IEEE Std. 1076 [34]. VHDL is a natural choice for this single modeling language in that it has high-level language constructs, but unlike other programming languages, it has a built-in timing and concurrency model. VHDL does have some disadvantages in terms of simulation execution time, but techniques have been developed to help address this problem [35].

ADEPT supports the integrated performance and dependability analysis of system-level models and includes the capability to simulate both uninterpreted and interpreted models through mixed-level modeling. ADEPT also has a mathematical basis in Petri nets thus providing the capability for analysis through simulation or analytical approaches [36].

In the ADEPT environment, a system model is constructed by interconnecting a collection of predefined elements called ADEPT modules. The modules model the information flow, both data and control, through a system. Each ADEPT module has a VHDL behavioral description and a corresponding mathematical description in the form of a CPN based on Jensen's CPN model [37]. The modules

communicate by exchanging *tokens*, which represent the presence of information, using a fully interlocked, four-state handshaking protocol [38]. The basic ADEPT modules are intended to be building blocks from which useful modeling functionality can be constructed. In addition, custom modules can be developed by the user if required and incorporated into a system model as long as the handshaking protocol is adhered to. Finally, some libraries of application-specific, high-level modeling modules such as Multiprocessor Communications Network Modeling Library [39] have been developed and included in ADEPT.

The following sections discuss the VHDL implementation of the token data type and transfer mechanism used in ADEPT, and the modules provided in the standard ADEPT modeling library.

12.2.1 Token Implementation

The modules defined in this chapter have been implemented in VHDL, and use a modified version of the token passing mechanism defined by Hady [38]. Signals used to transport tokens must be of the type *token*, which has been defined as a record with two fields. The first field, labeled STATUS, is used to implement the token passing mechanism. The second field, labeled COLOR, is an array of integers that is used to hold user-defined color information in the model. The ADEPT allow the user to select from a predefined number of color field options. The tools then automatically link in the proper VHDL design library so that the VHDL descriptions of the primitive modules operate on the defined color field. The default structure of the *data-type* token used in the examples discussed in this document is shown in Figure 12.6.

There are two types of outputs and two types of inputs in the basic ADEPT modules. *Independent* outputs are connected to *control* inputs, and the resulting connection is referred to as a *control-type* signal. *Dependent* outputs are connected to *data* inputs, and the resulting connection is referred to as a *data-type* signal. To make the descriptions more intuitive, outputs are often referred to as the "source side" of a signal, and inputs are referred to as the "sink side" of a signal.

```
type handshake is (removed, acked, released, present);
type token_fields is (status,
                      tag1, tag2, tag3, tag4, tag5,
                      tag6, tag7, tag8, tag9, tag10,
                      tag11, tag12, tag13, tag14, tag15,
                      boole1, boole2, boole3,
                      fault, module_info,
                      index, act_time, color);
type color_type is array (token_fields range tag1 to act_time) of integer;

type token is
      record
         status    : handshake;
         color     : color_type;
      end record;

type op_fields is (add, sub, mul, div);
type cmp_fields is (lt, le, gt, ge, eq, ne);
type tkf_array is array (1 to 25) of token_fields;-- used in file_read

type token_vector is array (integer range <>) of token;

--  resolution function for token
function protocol (input : token_vector) return token;
subtype token_res is protocol token;
```

FIGURE 12.6 Token type definition.

Tokens on *independent* outputs may be written over by the next token, so the ''writing'' process is *independent* of the previous value on the signal. In contrast, new tokens may not be placed on *dependent* outputs until the previous token has been removed, so the ''writing'' process is *dependent* on the previous value on the signal. Data-type signals use the four-step handshaking process to ensure that tokens do not get overwritten, but no handshaking occurs on *control-type* signals.

The STATUS field of a token signal can take on four values. Signals connecting *independent* outputs to *control* inputs (control-type signals) make use of only two of the four values. *Independent* outputs place a value PRESENT on the status field to indicate that a token is present. Since *control* inputs only copy the token but do not remove it, the *independent* output only needs to change the value of the status field to RELEASED to indicate that the token is no longer present on the signal, and the *control* input can no longer consider the signal to contain valid information. The signals connecting *dependent* outputs to *data* inputs (data-type signals) need all four values (representing a fully interlocked handshaking scheme) to ensure that a dependent output does not overwrite a token before the data input to which it is connected has read and removed it. This distinction is important since a token on control-type signals represents the presence or absence of a condition in the network while tokens on data-type signals, in contrast, represent information or data that cannot be lost or overwritten. In addition, fanout is not permitted on *dependent* outputs and is permitted on *independent* outputs.

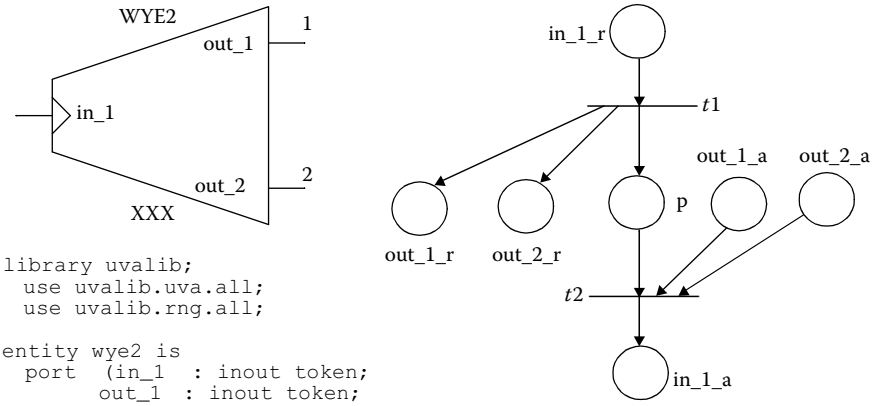
The signals used in the implementation are of type *token*. The token passing mechanism for data-type signals is implemented by defining a VHDL bus resolution function. This function is called each time a signal associated with a dependent output and a data input changes value. The function essentially looks at the value of the STATUS field at the ports associated with the signal and decides the final value of the signal. At the beginning of simulation the STATUS field is initialized to REMOVED. This value on a signal corresponds to an idle link in the nets defined by Dennis [40]. This state of a signal indicates that a signal is free and a request token may be placed on it. Upon seeing the REMOVED value, the requesting module may set the value of the signal to PRESENT. This operation corresponds to the ready signal in Dennis' definition. The requested module acknowledges a ready signal (PRESENT) by setting the signal value to ACKED, after the requested operation has been completed. Upon sensing the value of ACKED on the signal, the requesting module sets the value of the signal to RELEASED, which in turn causes the requested module to place the value of REMOVED on the signal. This action concludes one request/acknowledge cycle, and the next request/acknowledge cycle may begin.

In terms of Petri Nets, the models described here can be described as one-safe Petri nets since, at any given time, no place in the net can contain more than one token. The correspondence between the signal values and transitions in a Petri net may be defined, in general, in the following manner:

1. *PRESENT*. A token arrives at the input place of a transition.
2. *ACKED*. The output place of a transition is empty.
3. *RELEASED*. The transition has fired.
4. *REMOVED*. The token has been transferred from the input place to the output place.

The modules to be described in this chapter may be defined in terms of Petri nets. As an example, a Petri net description of the Wye module is shown in Figure 12.7.

The function of the Wye module is to copy the input token to both outputs. The input token is not acknowledged until both output tokens have been acknowledged. In the Petri net of Figure 12.7, the ''r'' and ''a'' labels correspond to ''ready'' and ''acknowledge,'' respectively. When a token arrives at the place labeled ''0r,'' the top transition is enabled and a token is placed in the ''1r,'' ''2r,'' and center places. The first two places correspond to a token being placed on the module outputs. Once the output tokens are acknowledged (corresponding to tokens arriving at the ''1a'' and ''2a'' places), the lower transition is enabled and a token is placed in ''0a,'' corresponding to the input token being acknowledged. The module is then ready for the next input token. Note that since this module does not manipulate the color fields, no color notation appears on the net.




```
library uvalib;
  use uvalib.uva.all;
  use uvalib.rng.all;

entity wye2 is
  port  (in_1  : inout token;
         out_1  : inout token;
         out_2  : inout token);
end wye2;

architecture ar_wye2 of wye2 is
begin
  pr_wye2 : process (in_1, out_1, out_2)
  begin
    if token_present (in_1)
      and token_removed (out_1)
      and token_removed (out_2) then
      out_1 <= in_1;
      out_2 <= in_1;
    end if;

    if token_acked (out_1) and token_acked (out_2) then
     ack_token (in_1);
     release_token (out_1);
     release_token (out_2);
    end if;

    if token_released (in_1) then
      remove_token (in_1);
    end if;
  end process pr_wye2;
end ar_wye2;
```

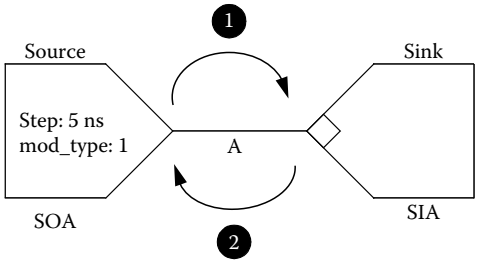
FIGURE 12.7 Wye module ADEPT symbol, its behavioral VHDL description, and its CPN representation.

The specific CPN description used is based on the work of Jensen [37]. The complete CPN descriptions of each of the ADEPT building blocks can be found in Ref. [41].

12.2.2 ADEPT Handshaking and Token Passing Mechanism

Recall that the ADEPT standard token has a *status* field which can take on four values: *PRESENT, ACKED, RELEASED,* and *REMOVED*. These values reflect which stage of the token passing protocol is currently in progress. Several examples will now be presented to show how the handshaking and token passing occurs between ADEPT modules. Figure 12.8 illustrates the handshaking process between a Source module connected to a Sink module.

Figure 12.8a describes the ''simplified'' event sequence. Here, we can think of the four-step handshaking as consisting of simply a forward propagation of the token and a backward propagation of the token acknowledgment. Thinking of the handshaking in this simplified manner, Table A in Figure 12.8 shows the simplified event sequence and times. At time 0 ns, the Source module places a token on A, corresponding to Event 1. The token is immediately acknowledged by the Sink module, corresponding

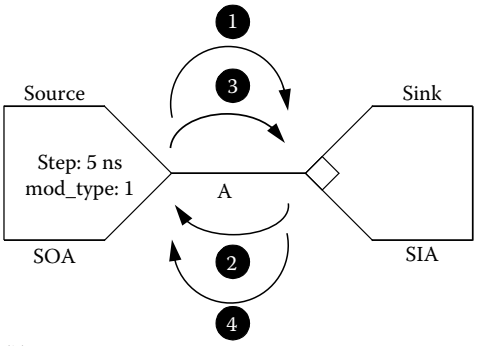


Time between new tokens on A = step + path delay = 5 ns

(a)

TABLE A Simplified Event Sequence

Event	Time (ns)	Description
1	0	Source module places token on signal A
2	0	Sink module acknowledges token on A
1	5	Source module places next token on signal A



(b)

TABLE B Detailed Event Sequence

Event	Time (ns)	Delta	Description	Resolved Signal A
1	0	1	Source module places token on A	*PRESENT*
2	0	2	Sink module acknowledges token on A	*ACKED*
3	0	3	Source module release token on A	*RELEASED*
4	0	4	Sink module removes token on A	*REMOVED*
1	5	1	Source module places token on A	*PRESENT*

FIGURE 12.8 Two module handshaking examples.

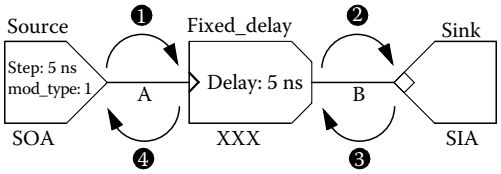
to Event 2. Since the Source module has a *step* generic of 5 ns, and there is no other delay along the path to the Sink, the next token will be output by the Source module at time 5 ns.

Figure 12.8b details the entire handshaking process. Table B in Figure 12.8 lists the detailed event sequence for this example. Since handshaking occurs in zero simulation time, the events are listed by *delta cycles* within each simulation time. The function of delta cycles in VHDL is to provide a means for ordering and synchronizing events (such as handshaking) which occur in zero time. The actual event sequence for this example consists of four steps, as shown in Figure 12.8b. Event 1 is repeated at time 5 ns, which starts the handshaking process over again.

12.2.2.1 Three-Module Example

To illustrate how tokens are passed through intermediate modules, a three-module example will now be examined. Consider the case where a Fixed Delay module is placed between a Source and Sink. This situation is illustrated in Figure 12.9.

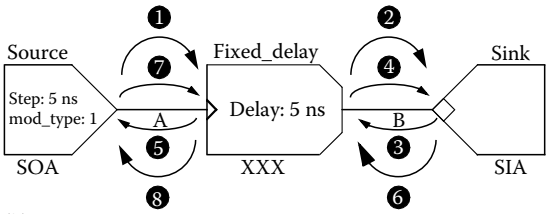
Figure 12.9a shows the simplified event sequence, where we can think of the four-step handshaking as consisting of simply a forward propagation of the token and a backward propagation of the token acknowledgment. Table A in Figure 12.9 lists this simplified event sequence. Notice that since now there is a path delay from the Source to the Sink, the time between new tokens from the Source is *step* + *path_delay* = 5 + 5 = 10 ns. At time 0 ns, the Source module places the first token on Signal A (Event 1). This token is read by the Delay module and placed on Signal B after a delay of 5 ns (Event 2). The Sink module then immediately acknowledges the token on Signal B (Event 3). The Delay module then passes



(a)

TABLE A Simplified Event Sequence

Event	Time (ns)	Description
1	0	Source module places token on signal A
2	5	Delay module places token on signal B
3	5	Sink module acknowledges token on signal B
4	5	Delay module acknowledges token on signal A
1	10	Source module places next token on signal A



(b)

TABLE B Detailed Event Sequence

Event	Time (ns)	Delta	Description	Resolved Signal A	Resolved Signal B
1	0	1	Source module places token on A	*PRESENT*	*REMOVED*
2	5	1	Delay module places token on B	—	*PRESENT*
3	5	2	Sink module acknowledges token on B	—	*ACKED*
4	5	3	Delay module releases token on B	—	*RELEASED*
5			Delay module acknowledges token on A	*ACKED*	—
6	5	4	Sink module removes token on B	—	*REMOVED*
7			Source module relases token on A	*RELEASED*	—
8	5	5	Delay module removes token on A	*REMOVED*	—
1	10	1	Source module places token on A	*PRESENT*	—

FIGURE 12.9 Three-module handshaking examples.

this acknowledgment back through to Signal A (Event 4). At time 10 ns, the Source module places the next token on Signal A, starting the process over again.

Figure 12.9b shows the detailed event sequence for this example. Since there are two signals, there are a total of eight detailed events. Table B in Figure 12.9 lists the detailed event sequence for this example. Again since the handshaking occurs in zero simulation time, the events are listed by delta cycles within each simulation time. Table B lists the resolved values on both Signal A and Signal B, where a value of "—" indicates no change in value. Notice that the sequence of events on each signal proceeds in this order: place the token (*PRESENT*) by the "source side," acknowledge the token (*ACKED*) by the "sink side," release the token (*RELEASED*) by the "source side," and finally remove the token (*REMOVED*) by the "sink side." The important concept to note in this example is the ordering of events. Notice that the Delay module releases the token on its output (Event 4) before it acknowledges the token on its input (Event 5), even though the two events occur in the same simulation time. These actions trigger the Sink module to then remove the token on Signal B (Event 6) and the Source module to release the token on Signal A (Event 7), both in the next delta cycle. Thus Signal B is ready for another token (removed) a full delta cycle before Signal A is ready.

12.2.2.2 Token Passing Example

To illustrate how tokens propagate in a larger system, consider the model shown in Figure 12.10. This figure shows the simplified event sequence, where again we can think of the four-step handshaking as consisting of simply a forward propagation of the token and a backward propagation of the token acknowledgment. In this system, the Source module provides tokens to the Sequence A (SA) module (Event 1), who first passes them to the Sequence B (SB) module (Event 2). The SB module first passes the token to the Fixed Delay A (FDA) module (Event 3), who passes it to the Sink A (SIA) module after a delay of 5 ns (Event 4). The SIA module then immediately acknowledges the token (Event 5), which causes the FDA module to pass the acknowledgment back to the SB module (Event 6). At this point the SB module places a copy of the token on its second output (Event 7), where it is passed through the

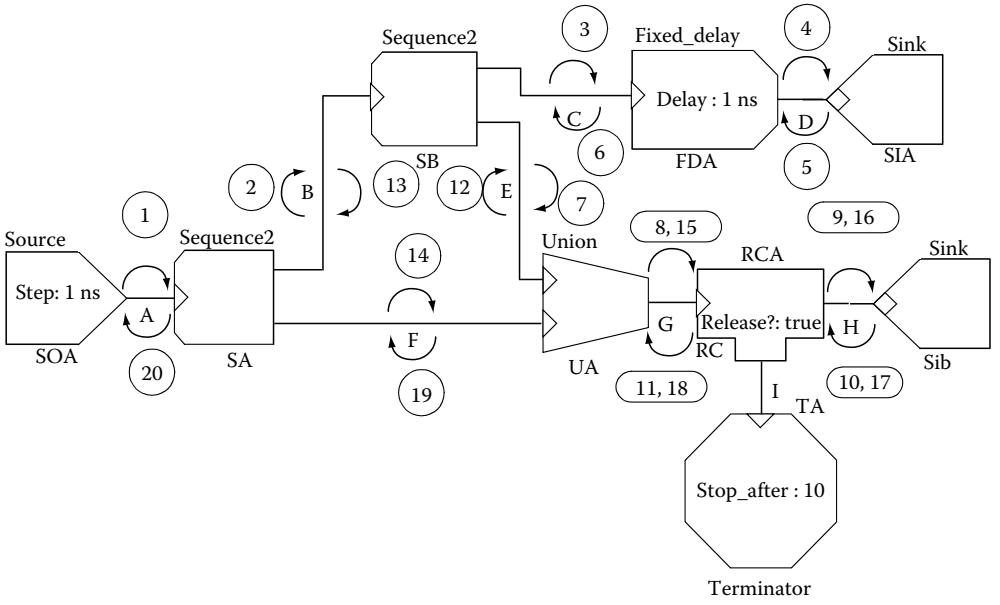


FIGURE 12.10 Token passing example showing simplified event sequence.

Union A and onto its output (Event 8). The Read Color A module then places the token on its out_1 output (Event 9) and simultaneously places the token on its *independent* output. The Sink B module then acknowledges the token on Signal H (Event 10). Upon seeing this acknowledgment, the Read Color A module releases the token on its *independent* output since the Read Color module has no ''memory'' (the *release?* generic equals *true*). The acknowledgment is then propagated back to the SB module (Events 11 and 12). At this point, the SB module acknowledges its input token (Event 13), freeing the SA module to place a token on its second output (Event 14). This token is then passed through the Union A, Read Color A, and Sink module (Events 15 and 16) and the acknowledgment is returned (Events 17–19). The SA module can then acknowledge the token on its input (Event 20), allowing the Source module to generate the next token 5 ns (*step*) later.

If we examine the activity on Signal I (the *independent* output of the Read Color A module), we see that it goes present after Event 8, released after Event 10, present after Event 15, and released again after Event 17. Consider the operation of the Terminator module attached to this signal. Since the Terminator module halts simulation after the number of active events specified by the *stop_after* generic, if we were to simulate this example, the simulation would halt after Event 15 (the second active event).

12.2.3 Module Categories

The ADEPT primitive modules may be divided into six categories: *control modules*, *color modules*, *delay modules*, *fault modules*, *miscellaneous parts modules*, and *mixed-level modules*.

As the name implies, the *control modules* are used to control the flow of tokens through the model. As such, the control modules form the major portion of the performance model. The control modules operate only on the STATUS field of a signal and do not alter the color of a token on a signal. Further, no elapsed simulation time results from the activation of the control modules since they do not have any delay associated with them. There are several control modules whose names end in a numeral, such as ''union2.'' These modules are part of a family of modules that have the same function, but differ in the number of inputs or outputs that they possess. For example, there are eight ''union'' modules, ''union2,'' ''union3,'' ''union4,'' ..., ''union8.'' With the exception of the Switch, Queue, and logical modules, the control modules have been adapted from those proposed by Dennis [40].

The control modules described above process colored tokens but do not alter the color fields of the tokens passing through them. Manipulation of the color field of a token is reserved to the *color modules*. These color modules permit various operations on the color fields such as allowing the user to read and write the color fields of the tokens. The color modules also permit the user to compare the color information carried by the tokens and to control the flow of the tokens based on the result of the comparisons. The use of these color modules enables high-level modeling of systems at different levels of detail. These modules permit the designer to add more detail or information to the model by placing and manipulating information placed on the color fields of tokens flowing through the model. The color fields of these tokens can be set and read by these modules to represent such things as destination or source node addresses, data length, data type (e.g., ''digital'' or ''analog''), or any type of such information that the designer feels is important to the design.

The *delay modules* facilitate the description of data path delays at the conceptual or block level of a design. As an example, the Fixed Delay module, or the FD module, may be used in conjunction with the control modules to model the delay in the control structure of a design which is independent on the type or size of an operation being performed. In contrast, the Data-Dependent Delay module may be used to model processes in which the time to complete a particular operation is dependent on the amount or type of data being processed. The input to the Data-Dependent Delay module is contained on one of the color fields of the incoming token.

The *fault modules* are used to represent the presence of faults and errors in a system model. The modules allow the user to model fault injection, fault/error detection, and error correction processes. The *miscellaneous parts* category contains modules that perform ''convenience'' functions in ADEPT.

Examples include the Collector, Terminator, and Monitor modules. The Collector module is used to write input activation times to a file, and the Terminator module is used to halt simulation after a specified number of events have occurred. The Monitor module is a data-collection device that can be connected across other modules to gather statistical information during a VHDL simulation.

The *mixed-level modules* support mixed-level modeling in the ADEPT environment by defining the interfaces around the interpreted and uninterpreted components in a system model. The functions of these modules and their use in creating mixed-level models are described in more detail in Sections 12.4.2 and 12.4.3.

The complete functionality of each primitive module is defined and the generics associated with each of the modules are described in the *ADEPT Library Reference Manual* [42]. The standard ADEPT module symbol convention is also explained in more detail in the *ADEPT Library Reference Manual.*

In addition to the primitive modules, there are libraries of more complex modules included in ADEPT. In general, these libraries contain modules for modeling systems in a specific applications area. These libraries are also discussed in more detail in the *ADEPT Library Reference Manual.*

12.2.4 ADEPTs

The ADEPT system is currently available on Sun platforms using Mentor Graphics' *Design Architect* as the front end schematic capture system, or on Windows PCs using OrCAD's *Capture* as the front end schematic capture system. The overall architecture of the ADEPT system is shown in Figure 12.11.

The schematic front end is used to graphically construct the system model from a library of ADEPT module symbols. Once the schematic of the model has been constructed, the schematic capture system's netlist generation capability is used to generate an electronic design interchange format (EDIF) 2.0.0 netlist of the model. Once the EDIF netlist of the model is generated, the ADEPT software is used to translate the model into a structural VHDL description consisting of interconnections of ADEPT

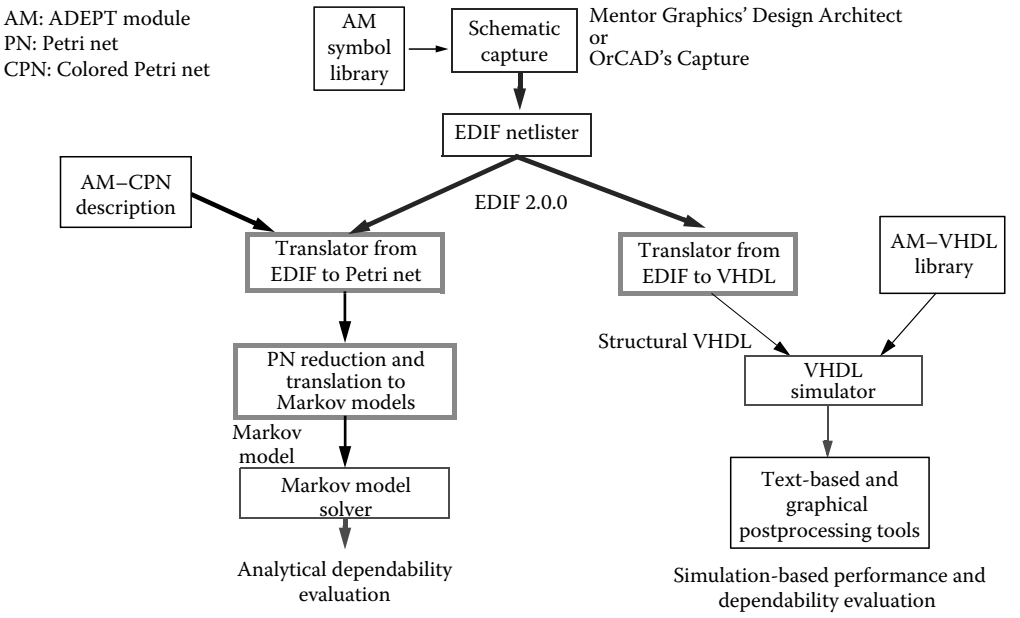


FIGURE 12.11 ADEPT design flow.

modules. The user can then simulate the structural VHDL that is generated using the compiled VHDL behavioral descriptions of the ADEPT modules to obtain performance and dependability measures.

In addition to VHDL simulation, a path exists that allows the CPN description of the system model to be constructed from the CPN descriptions of the ADEPT modules. This CPN description can then be translated into a Markov model using well-known techniques and then solved using commercial tools to obtain reliability, availability, and safety information.

12.3 Simple Example of an ADEPT Performance Model

This section presents a simple example of the usage of the primitive building blocks for performance modeling. The example is a three-computer system wherein the three-computers share a common bus. The example also presents simulation results and system performance evaluations.

12.3.1 Three-Computer System

This section discusses a simple example to illustrate how the modules discussed previously may be interconnected to model and evaluate the performance of a complete system. The system to be modeled consists of three computers communicating over a common bus, as shown in Figure 12.12. Each block representing a computer can be thought to contain its own processor, memory, and peripheral devices. The actual ADEPT schematic for the three-computer system is shown in Figure 12.13.

Computer C1 contains some sensors and preprocessing capabilities. It collects data from the environment, converts it into a more compact form and then sends it to computer C2 via the bus. The computer C2 further processes the data and passes it to computer C3 where the data are appropriately utilized. It is assumed that data are transferred in packets and each packet of data is of varying length. In the example described here, computers C1 and C2 receive packets whose sizes are uniformly distributed between 0 and 100. The packet size of computer C3 is uniformly distributed between 0 and 500. The external environment in this example is modeled by a Source module in C1 and a Sink module in C3.

C1 has an output queue, C2 has both an input queue and an output queue, while C3 has one input queue. All queues in this example are assumed to be of length 8. If the input queues of C2 or C3 are full the corresponding Q_free signal is released (value = *RELEASED*). This interconnection prevents the

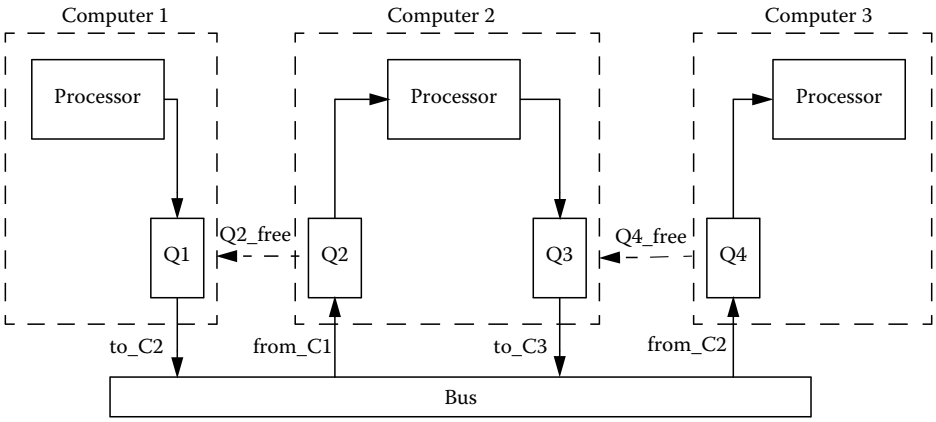


FIGURE 12.12 Three-computer system.

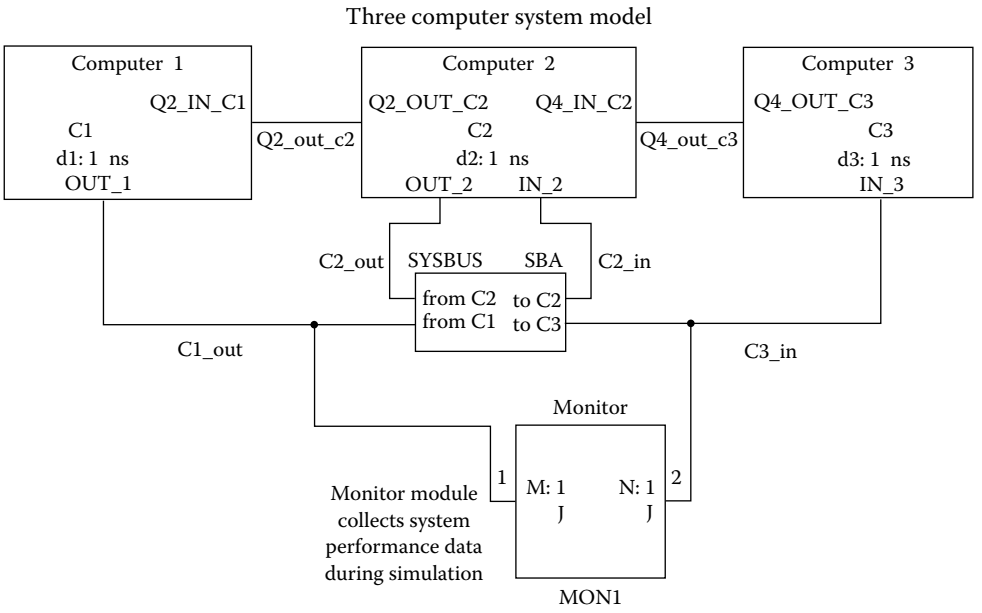


FIGURE 12.13 Three-computer system ADEPT schematic.

computer writing into the corresponding queue from placing data on the bus when the queue is full. This technique not only prevents the bus from being unnecessarily held up but also eliminates the possibility of a deadlock.

The ADEPT model of Computer 1 (C1) is shown in Figure 12.14. The Source A (SOA) module along with the Sequence A (SA), Set_Color A (SCA) and Random A (RA) modules generate tokens whose *tag1* field is set according to a distribution which is representative of the varying data sizes that C1 receives. The Data-Dependent Delay A (DDA) models the processing time of C1 which is directly proportional to the packet size. The *unit_step* delay for the DDA module is passed down as a generic *d1**1 ns. As soon as a token, representing one packet of data, appears at the output of the DDA module, the Sequence B (SB), Set_Color B (SCB), and Random B (RB) modules together set the *tag1* field of the token to represent the packet size after processing. The Constant A (COA) and Set_Color C (SCC) modules set the *tag2* field of the token to 2. This coloring indicates to the bus arbitration unit that the token is to be transferred to C2. The token is then placed in the output Queue (Q1) of C1 and the token is acknowledged. This acknowledge signal is passed back to the Source A module which then produces the next token. The Fixed Delay (FDA and FDB) modules represent the read and write times associated with the Queue. The Switch A (SWA) element is controlled by the Q_free signal from C2 and prevents a token from the output queue of C1 from being placed on the bus if the incoming Q_free signal is inactive.

Figure 12.15 shows the ADEPT model of Computer 2 (C2). When a token arrives at the *data* input of C2 the token is placed at the input of the Queue (Q2). The *control* output of the Queue becomes the Q2_Free output of C2. The remaining modules perform the same function as in C1 except that the tag2 field of the tokens output by C2 is set to 3 which indicates to the bus arbitration unit that the token is to be sent to C3. The DDA module represents the relative processing speed of Computer C2. The *unit_step* delay for the DDA module is passed down as a generic *d2**1 ns.

The modules defining Computer 3 (C3) are shown in Figure 12.16. A token arriving at the input is placed in the queue. The DDA element reads one token at a time and provides the delay associated with

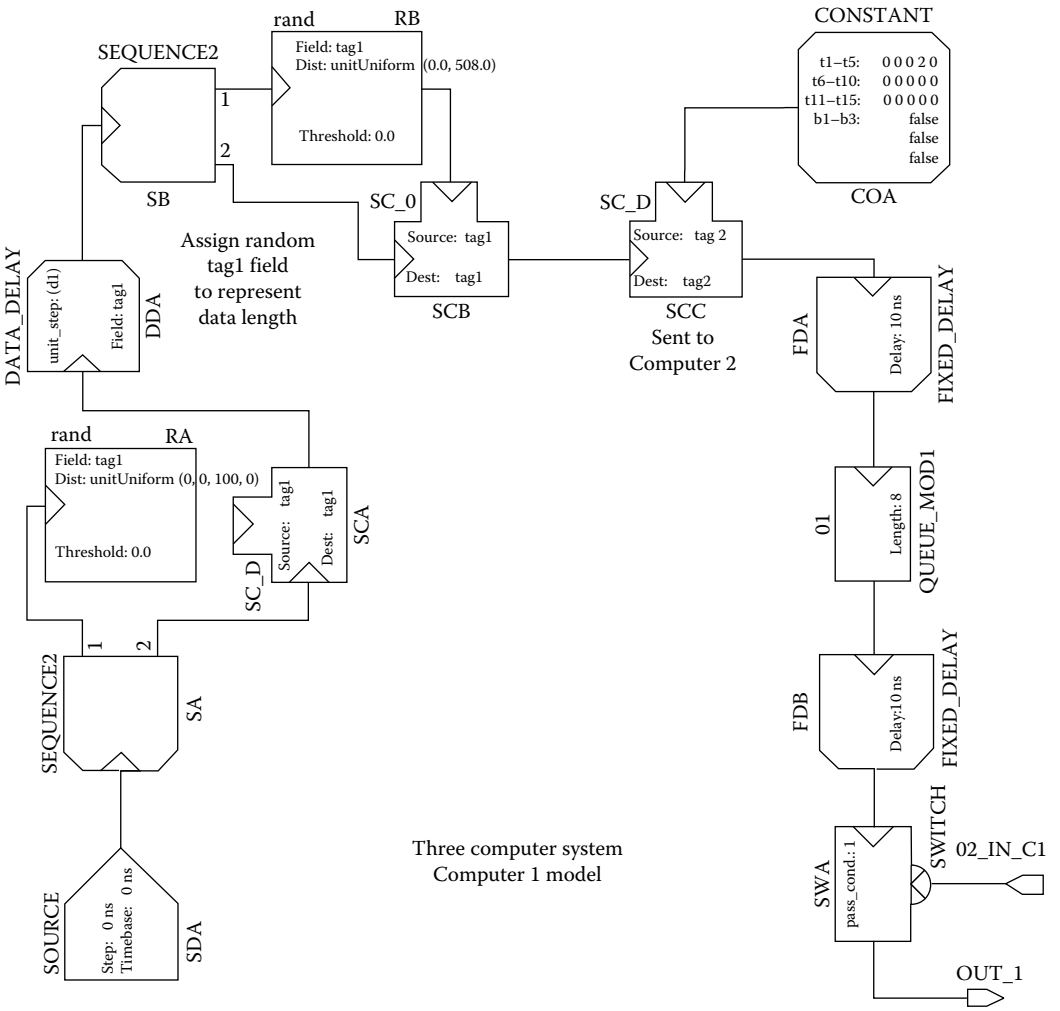


FIGURE 12.14 ADEPT model of Computer 1.

the processing time of C3 before the Sink A (SIA) module removes the token. The *unit_step* delay for the DDA module is passed down as a generic *d3**1 ns.

The bus is shown in Figure 12.17. Arbitration is provided to ensure that both C1 and C2 do not write onto the bus at the same time. The Arbiter A (AA) element provides the required arbitration. Since the output of C2 is connected to the IN_1(1) input of the AA element, it has a higher priority over C1. The Union A (UA) element passes a token present at either of its inputs to its output. The output of the UA element is connected to the Data-Dependent Delay A (DDA) element, which models the packet transfer delay associated with moving packets over the bus. The delay through the bus is dependent on the size of the packet of information being transferred (size stored on the *tag1* field). Note that in this analysis, the bus delay was set to 0. The *independent* output of the Read_Color A(RCA) element is connected to the *control* input of the Decider A (DA) module. The base of the DA element is set to 2. Since the *tag2* field of the token is set to 2 or 3 depending on whether it originated from C1 or C2, the first output of the Decider A module is connected to C2 and the second output is connected to C3. This technique ensures that the tokens are passed on to the correct destination, C2 or C3.

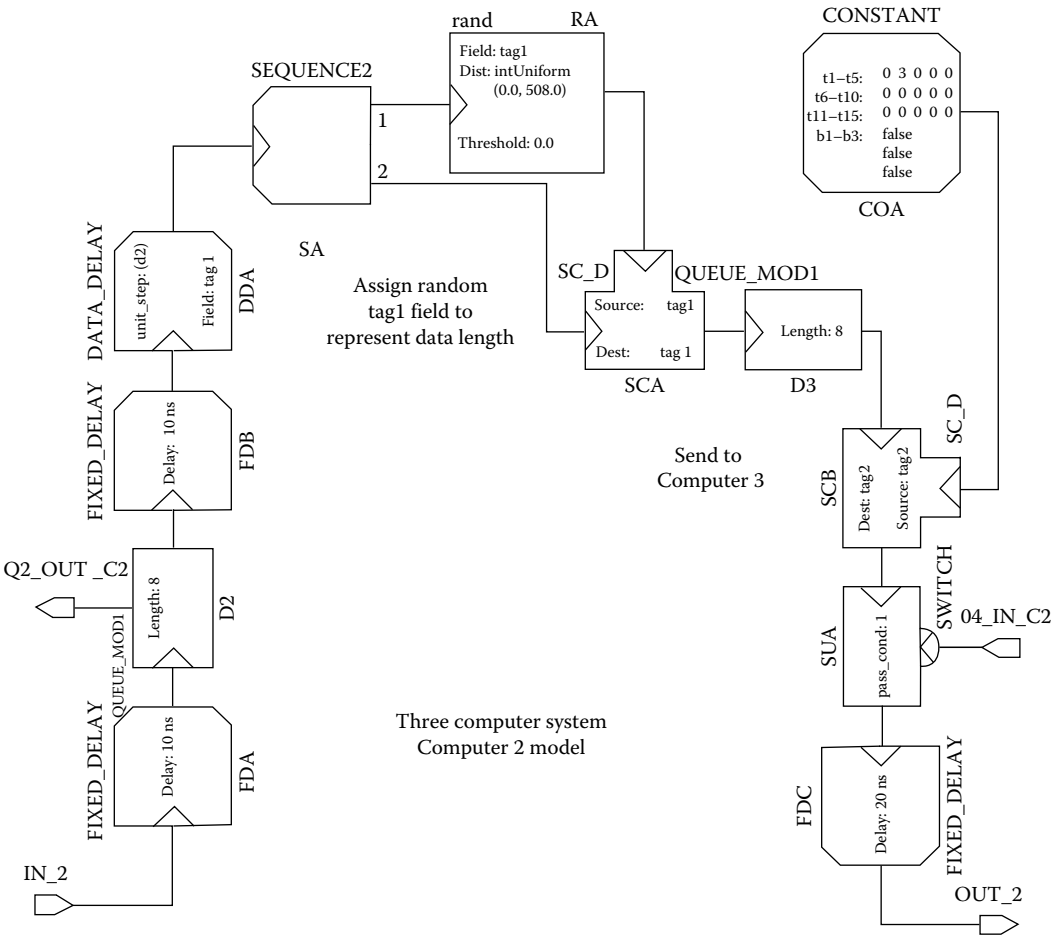


FIGURE 12.15 ADEPT model of Computer 2.

12.3.2 Simulation Results

This section presents simulation results that illustrate the queuing model capabilities of the ADEPT system. The model enables the study of the overall speed of the system in terms of the number of packets of information transferred in a given time. It also allows the study of the average number of tokens present in each queue during simulation, and the effect of varying system parameters on the number of items in the queues and overall throughput of the system. The generic unit_step delay of the DD elements (*d1*, *d2*, and *d3*) associated with the three computers is representative of the processing speed of the computers. The optimal relative speeds of the computers may also be deduced by simulation of this model. Figure 12.18 shows graphs of the number of items in each queue versus simulation time. These graphs were generated using the BAARS postsimulation analysis tool. The upper graph shows the queue lengths when *d1*, *d2*, and *d3* was set to 5, 5, and 2 ns. The lower graph shows the queue lengths when *d1*, *d2*, and *d3* was set to 5, 4, and 1 ns. In the first case, because the processing time of Computer 3 was so much longer than Computer 1 or 2, the queue in Computer 3 became full at ∼4000 ns of simulation time. The filling of the queue in Computer 3 delayed items coming out of Computers 1 and 2 thus causing their queues to also become full. In the second case, the processing time ratios were such that Computer 3 could keep up with the incoming tokens and the queues never got completely full.

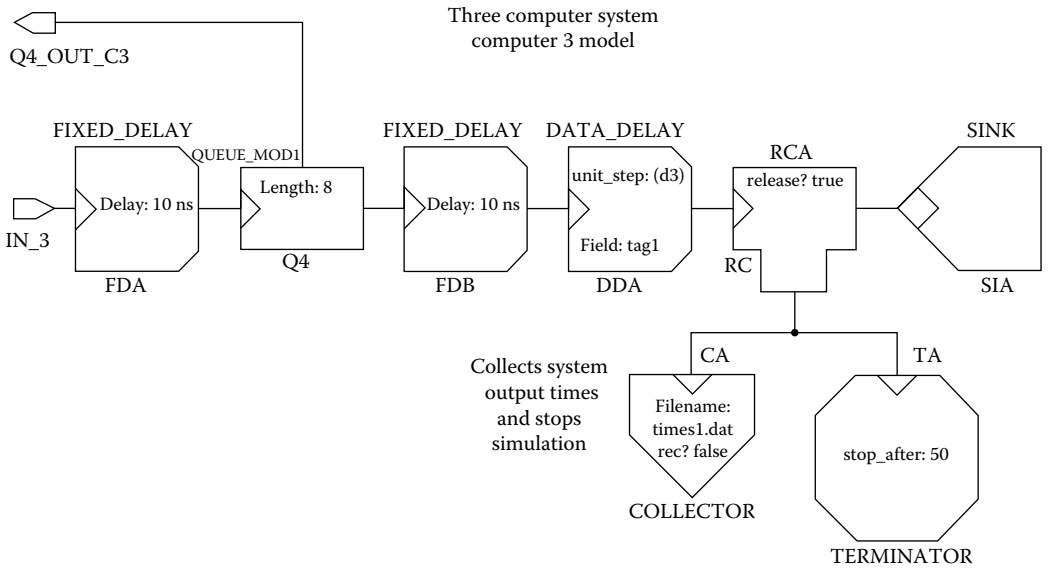


FIGURE 12.16 ADEPT model of Computer 3.

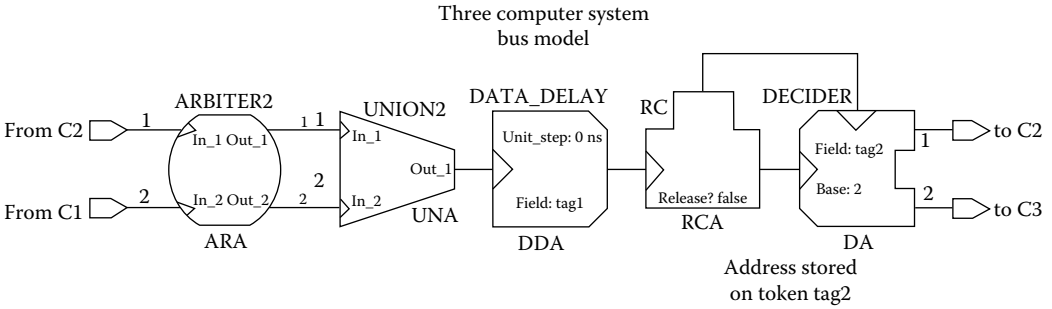


FIGURE 12.17 ADEPT model of system bus.

Table 12.1 summarizes the effect of relative speeds of the computers on the number of packets transferred. Since the size of the packets received by C3 is uniformly distributed between 0 and 500 while the size of the packets received by C1 and C2 is uniformly distributed between 0 and 100, it is intuitively obvious that the overall throughput of the system is largely determined by the speed of Computer C3. The results do indicate this behavior. For example, when the relative instruction execution times for C1, C2, and C3 are 5, 5, and 2, respectively, a total of 197 packets are transferred. By increasing the instruction execution time of C2 by one time unit and decreasing the instruction execution time of C3 by one time unit, it is seen that a total of 321 packets are transferred, an increase of slightly over 60%.

This example has illustrated the use of the various ADEPT modules to model a complete system. Note how the interconnections between modules describing a component of the system are similar to a flow chart describing the behavior of the component. This example also demonstrated that complex systems at varying levels of abstraction and interpretation can easily be modeled using the VHDL-based ADEPTs.

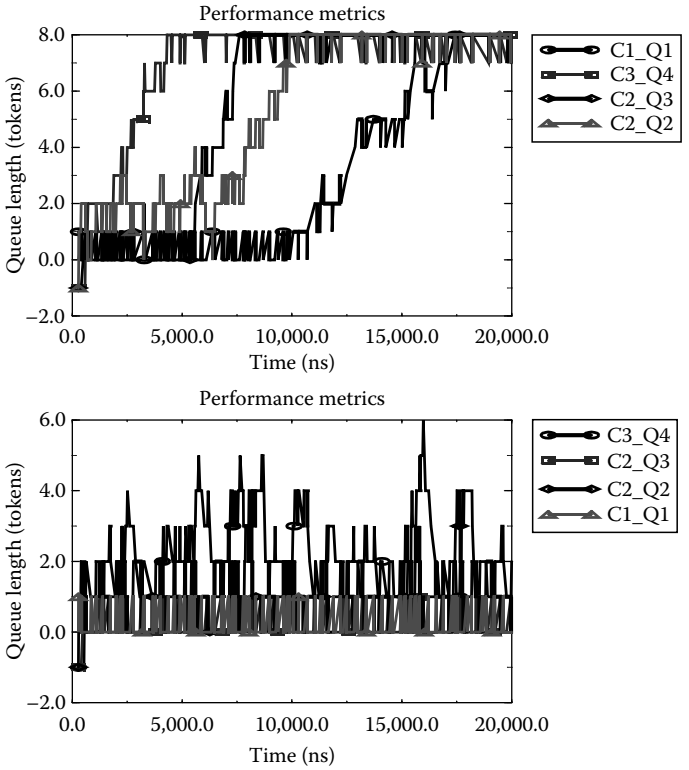


FIGURE 12.18 Queue lengths versus simulation time for various processing delay times.

TABLE 12.1 System Throughput versus Computer Delay Times

C1 Delay (ns)	C2 Delay (ns)	C3 Delay (ns)	Packets per 100,000 Time Units (ns)
8	8	3	131
9	10	2	194
5	5	2	197
3	5	2	197
2	2	2	197
5	6	1	321
5	4	1	322
4	3	1	384
3	2	1	385
2	2	1	386

12.4 Mixed-Level Modeling

As described earlier, performance (uninterpreted) modeling has been previously used primarily by systems engineers when performing design analysis at the early stages of the design process. Although most of the design detail is not included in these models since this detail does not yet exist, techniques such as token coloring can be used to include that design detail that is necessary for an accurate model.

However, most of the detail, especially very low-level information such as word widths and bit encoding, are not present. In fact, many additional design decisions must be made before an actual implementation could ever be constructed. In performance models constructed in ADEPT, abstract tokens are used to represent this information (data and control) and its flow in the system. An illustration of such a performance model with its analysis was given the three-computer system described in Section 12.3.1. In contrast, behavioral (interpreted) models can be thought of as including much more design detail often to the point that an implementation could be constructed. Therefore, data values and system timing are available and variables typically take on integer, real, or bit values. For example, a synthesizable model of a carry-lookahead adder would be one extreme of a behavioral model.

It should be obvious that the two examples described above are extremes of the two types of models. Extensive design detail can be housed in the tokens of a performance model and information in a behavioral model can be encapsulated in a very abstract type. However, there is always a difference in the abstraction level of the two modeling types. Therefore, to develop a model that can include both performance and behavioral models, interfaces between the different levels of design abstraction, called *mixed-level interfaces,* must be included in the overall model to resolve differences between these two modeling domains.

The mixed-level interface is divided into two primary parts which function together; the part that handles the transition from the uninterpreted domain to the interpreted domain (U/I) and the part that handles the transition from the interpreted domain to the uninterpreted domain (I/U). The general structure of a mixed-level model is shown in Figure 12.19.

In addition to the tokens-to-values (U/I) and values-to-tokens (I/U) conversion processes, the mixed-level interface must resolve the differences in design detail that naturally exist at the interface between uninterpreted and interpreted elements. These differences in detail appear as differences in data and timing abstraction across the interface. The differences in timing abstraction across the interface arise because the components at different levels model timing events at different granularities. For example, the passing of a token that represents a packet of data being transferred across a network in

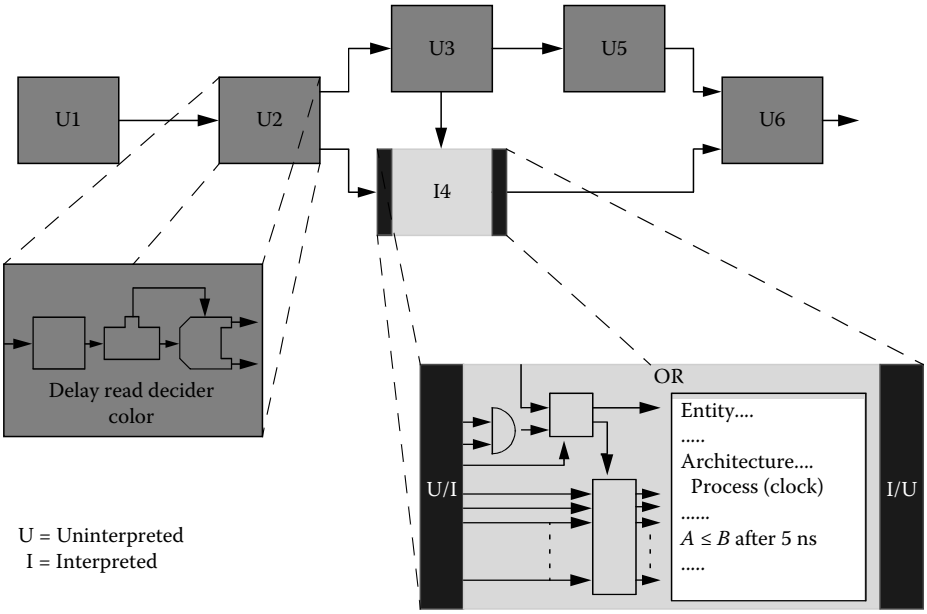


FIGURE 12.19 General structure of a mixed-level model.

a performance model may correspond to hundreds or thousands of bus cycles for a model at the behavioral level.

The differences in data abstraction across the interface are due to the fact that typically, performance models do not include full functional details whereas behavioral models require full functional data (in terms of values on their inputs) to be present before they will execute correctly. For example, a performance level modeling token arriving at the inputs to the mixed-level interface for a behavioral floating point coprocessor model may contain information about the operation the token represents, but may not contain the data upon which the operation is to take place. In this case, the mixed-level interface must generate the data required by the behavioral model and do it in a way that a meaningful performance metric, such as best- or worst-case delays, is obtained.

12.4.1 Mixed-Level Modeling Taxonomy

The functions that the mixed-level interface must perform and the most efficient structure of the interface are affected by several attributes of the system being modeled and the model itself. To partition the mixed-level modeling space and better define the specific solutions, a taxonomy of mixed-level model classes has been developed [43]. The classes of mixed-level modeling are defined by those model attributes which fundamentally alter the development and the implementation of the mixed-level interface. The mixed-level modeling space is partitioned according to three major characteristics:

1. Evaluation *objective* of the mixed-level model
2. *Timing mechanism* of the uninterpreted model
3. *Nature* of the interpreted element

For a given mixed-level model, these three characteristics can be viewed as attributes of the mixed-level model and the analysis effort. Figure 12.20 summarizes the taxonomy of mixed-level models.

Mixed-level modeling objectives. The structure and the functionality of the mixed-level interface are strongly influenced by the objective that the analysis of the mixed-level model will be targeted toward. For the purposes of this work, these objectives were broken down into two major categories:

1. *Performance analysis* and *timing verification*. To analyze the performance of the system (as defined previously) and verify that the specific components under consideration meet system timing constraints. Note that other metrics, such as power consumption, could be analyzed using mixed-level models, but these were outside the scope of this classification.
2. *Functional verification*. To verify that the function (input-to-output value transformation) of the interpreted component is correct within the context of the system model.

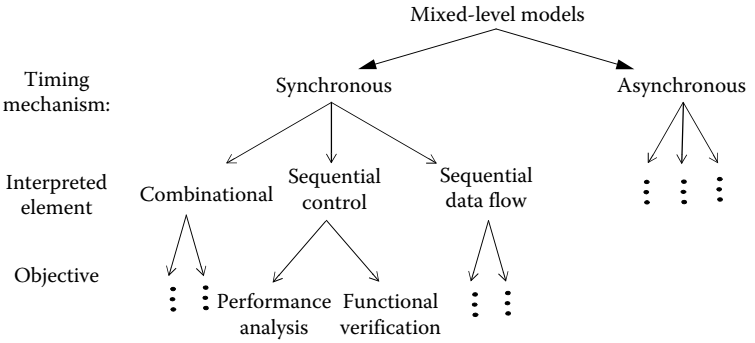


FIGURE 12.20 Mixed-level modeling categories.

Timing mechanisms. Typically, performance (uninterpreted) models are asynchronous in nature and the flow of tokens depends on the handshaking protocol. However, in modeling a system that is globally synchronous (all elements are synchronized to a global clock) a mechanism to synchronize the flow of tokens across the model can be introduced. This synchronization of the performance model will require different mixed-level modeling approaches. Thus the two types of system models that affect the mixed-level interface are

1. *Asynchronous models*. Tokens on independent signal paths within the system model move asynchronously with respect to each other and arrive at the interface at different times.
2. *Synchronous models*. The flow of tokens in the system model is synchronized by some global mechanism and they arrive at the interface at the same time, according to that mechanism.

Interpreted component. The mixed-level modeling technique strongly depends upon the type of the interpreted component that is introduced into the performance model. It is natural to partition interpreted models into those that model combinational elements and sequential elements. Techniques for constructing mixed-level interfaces for models of combinational interpreted elements have been developed previously [44]. In the combinational element case, the techniques for resolving the timing across the interface were more straightforward because of the asynchronous nature of the combinational elements, and the data-abstraction problem was solved using a methodology similar to the one presented here. However, using sequential interpreted elements in mixed-level models requires more complex interfaces to solve the synchronization problem and different specific techniques for solving the data-abstraction problem. Further research into the problem of mixed-level modeling with sequential elements suggested that interpreted components be broken down into three classifications as described below:

1. *Combinational elements*. Unclocked (with no states) elements
2. *Sequential control elements* (SCEs). Clocked elements (with states) that are used for controlling data flow, e.g., a control unit or a controller
3. *Sequential data flow elements* (SDEs). Elements that include datapath elements *and* clocked elements that control the data flow, e.g., control unit and datapath

The major reason for partitioning the sequential elements into SDEs and SCEs is based on the timing attributes of these elements. In a SCE control input values are read every cycle and control output values (that control a datapath) are generated every cycle. In contrast, SDEs have data inputs and may have some control inputs but the output data are usually generated several clock cycles later. This difference in the timing attributes will dictate a different technique for mixed-level modeling. Because the solution for the timing and data-abstraction problem for SDEs is more complex, and the solution for SCEs can be derived from the solution for SDEs, developing a solution for SDEs was the focus of this work.

12.4.2 Interface for Mixed-Level Modeling with FSMD Components

This section describes the interface structure and operation for sequential interpreted elements that can be described as finite state machines with datapaths (FSMDs) [45]. They consist of a finite state machine (FSM) used as a control unit, and a datapath, as shown in Figure 12.21. System models are quite often naturally partitioned to blocks which adhere to the FSMD structure. Each of these FSMDs processes the data and have some processing delays associated with them. These FSMDs indicate the completion of a processing task to the rest of the system either by asserting a set of control outputs or by the appearance of valid data on their outputs. This characteristic of being able to determine when the data-processing task of the FSMD is completed is a key property in the methodology of constructing mixed-level models with FSMD interpreted components.

The functions performed by the U/I operator include placing the proper values on the inputs to the FSMD interpreted model, and generating a clock signal for the FSMD interpreted model. The required

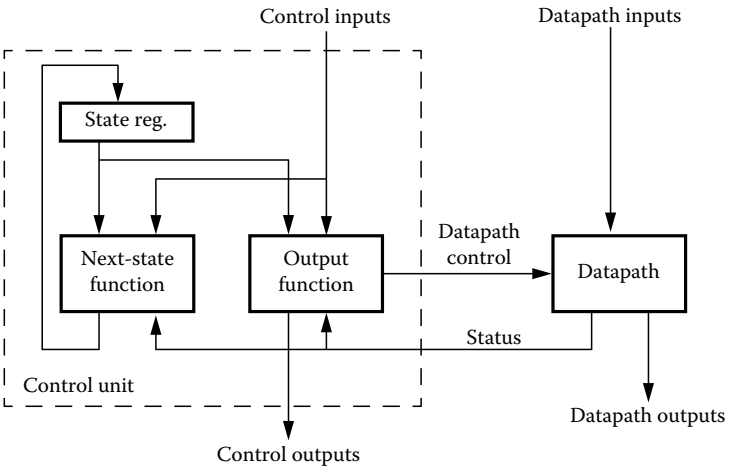


FIGURE 12.21 Generic FSMD block diagram.

values to be placed on the inputs to the FSMD interpreted model are either contained on the incoming token's information, or "color" fields, are supplied by the modeler via some outside source, or are derived using the techniques described in Section 12.4.2.1. The clock signal is either generated locally if the system-level model is globally asynchronous, or converted from the global clock into the proper format if the system-level model is globally synchronous. The functions performed by the I/U operator include releasing the tokens back into the performance model at the appropriate time and, if required, coloring them with new values according to the output signals of the interpreted element. The structure of these two parts of the mixed-level interface is shown in Figure 12.22. The U/I operator is composed of the following blocks: a Driver, an Activator, and a Clock_Generator. The I/U operator is composed of an Output_Condition_Detector, a Colorer, and a Sequential_Releaser.

In the U/I operator, the Activator is used to detect the arrival of a new token (packet of data) to the interpreted element, inform the Driver of a new token arrival, and to drive control input/s of the

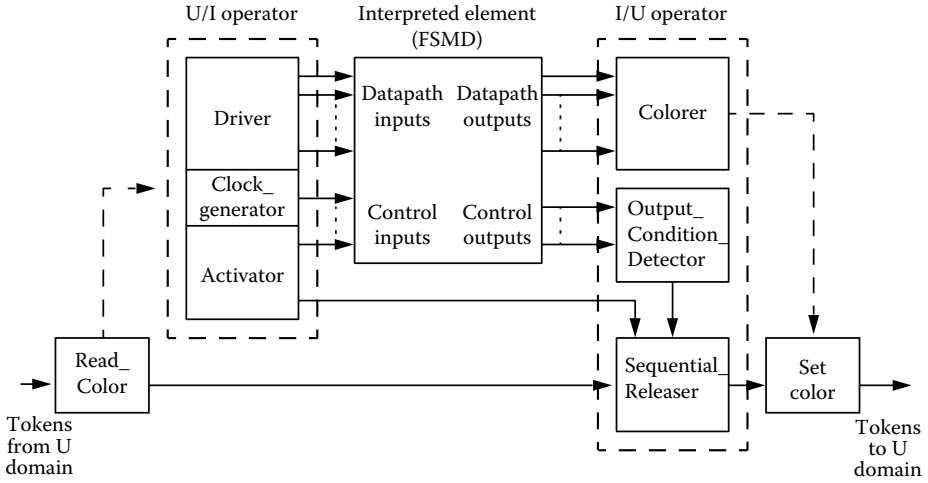


FIGURE 12.22 Mixed-level element structure.

interpreted element. The Activator's output is also connected to the I/U operator for use in gathering information on the delay through the interpreted element. The Driver reads information from the token's color fields and drives the proper input signals to the interpreted element according to predefined assignment properties. The Clock_Generator generates the clock signal according to the overall type of system-level model, either synchronous or asynchronous.

In the I/U operator, the Output_Condition_Detector detects the completion of the interpreted element data-processing operation, as discussed earlier, by comparing the element's outputs to predefined properties. The Colorer samples the datapath outputs and maps them to color fields according to predefined binding properties. The Sequential_Releaser, which "holds" the original token, releases it back to the uninterpreted model upon receiving the signal from the Output_Condition_Detector. The information carried by the token is then updated by the Colorer and the token flows back to the uninterpreted part of the model.

Given this structure, the operation of the mixed-level model can be described using the general example in Figure 12.22. Upon arrival of a new token to the mixed-level interface (U/I operator), the Read_Color module triggers the Activator component and passes the token to the Sequential_Releaser, where it is stored until the interpreted component is finished with its operation. Once triggered by the Read_Color module, the Activator notifies the Driver to start the data-conversion operation on a new packet of data. In the case of a globally asynchronous system-level model, the Activator will notify the Clock_Generator to start generating a clock signal. Since the interpreted element is a sequential machine, the Driver may need to drive sequences of values onto the inputs of the interpreted element. This sequence of values is supplied to the interpreted element, while the original token is held by the Sequential_Releaser. This token is released back to the uninterpreted model only when the Output_Condition_Detector indicates the completion of the interpreted element operation. The Output_Condition_Detector is parameterized to recognize the completion of data processing by the particular FSMD interpreted component. Once the Output_Condition_Detector recognizes the completion of data processing, it signals the Sequential_Releaser to release the token into the performance model. Once the token is released, it passes through the Colorer which maps the output data of the interpreted element onto color fields of the token. The new color information on the token may be used by the uninterpreted model for such things as delays through other parts of the model or for routing decisions.

12.4.2.1 Finding Values for the "Unknown Inputs" in an FSMD-Based Mixed-Level Model

As described previously, because of the abstract nature of a performance model, it may not be possible to derive values for all of the inputs to the interpreted component from the data present in the performance model. Typically, the more abstract the performance model (i.e., the earlier in the design cycle), the higher the percentage of input values that will be unknown. In some cases, particularly during the very early stages of the design process, it is possible that the abstract performance model will not provide any information to the interpreted element, other than the fact that new data has arrived. In this case, the data-abstraction gap will be large. This section describes an analytical technique developed to determine values for these "unknown inputs" such that a meaningful performance metric—best- or worst-case delay—can be derived from the mixed-level model.

If some (or all) inputs are not known from the performance model, some criteria for deriving the values on the unknown inputs must be made. Choosing a criterion is based on the objective of the mixed-level model. For the objective of timing verification, delays (number of clock cycles) through the interpreted element are of interest. The most common criterion in such cases is the worst-case processing delay. In some cases, best-case delay may be desired. If the number of unknown inputs is small, an exhaustive search for worst/best case may be practical. Therefore, it is desirable to minimize the number of unknown inputs which can affect the delay through the interpreted element. The methods for achieving this objective are described conceptually in Section 12.4.2.1.1. By utilizing these methods,

the number of unknown inputs is likely to be reduced but unknown inputs will not be eliminated completely. In this case, the performance metrics of best- and worst-case delay can be provided by a ''traversal'' of the state graph of the SDE component. Section 12.4.2.1.2 describes the traversal method developed for determining the delays through a sequential element.

Note that in the algorithms described below, the function of the SDE component is represented by the state transition graph (STG) or state table. These two representations are essentially equivalent and are easily generated from a behavioral VHDL description of the SDE, either by hand, or using some of the automated techniques that have been developed for formal verification.

12.4.2.1.1 Reducing the Number of Unknown Inputs

Although it is not essential, reducing the number of unknown inputs can simplify the simulation of a mixed-level model significantly. Since the FSMD elements being considered have output signals that can be monitored to determine the completion of data processing as discussed previously, other outputs may not be significant for performance analysis. Therefore, the ''nonsignificant'' (insignificant) outputs can be considered as ''don't cares.'' By determining which inputs do not affect the values on the significant outputs, it is possible to minimize the number of unknown delay affecting inputs (DAIs).

The major steps in the DAI detection algorithm are as follows:

Step 1. Select the ''insignificant'' outputs (in terms of temporal performance).
Step 2. In the state transition graph (STG) of the machine, replace all values for these outputs with a ''don't-care'' to generate the modified state machine.
Step 3. Minimize the modified state machine and generate the corresponding state table.
Step 4. Find the inputs which do not alter the flow in the modified state machine by detecting identical columns in the state table, and combining them by implicit input enumeration.

This method is best illustrated by an example. Consider the state machine which is represented by the STG shown in Figure 12.23. This simple example is a state machine with two inputs, X_1 and X_2, and two outputs, Y_1 and Y_2. This machine cannot be reduced, i.e., it is a minimal state machine. Assume that this machine is the control unit of an FSMD block and that the control output Y_1 is the output which indicates the completion of the ''data processing'' when its value is 1. Thus, output Y_2 is an ''insignificant output'' in terms of delay in accordance with Step 1. Therefore, a ''don't-care'' value is assigned to Y_2 as per Step 2. The modified STG is shown in Figure 12.24. As per Step 3, the modified STG is then reduced. In this example, states A, C, and E are equivalent (can be replaced by a single state, K) and the minimal

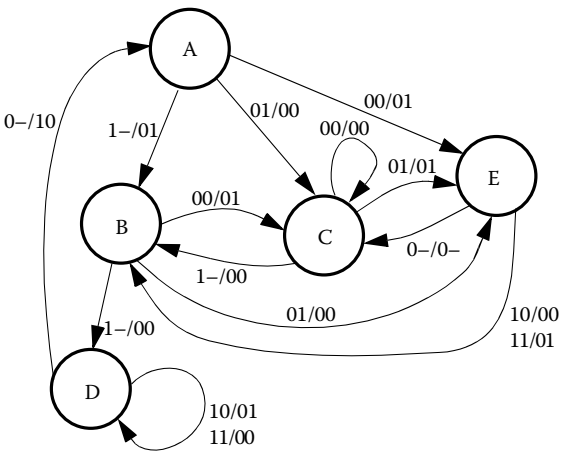


FIGURE 12.23 STG of a two-inputs two-outputs state machine.

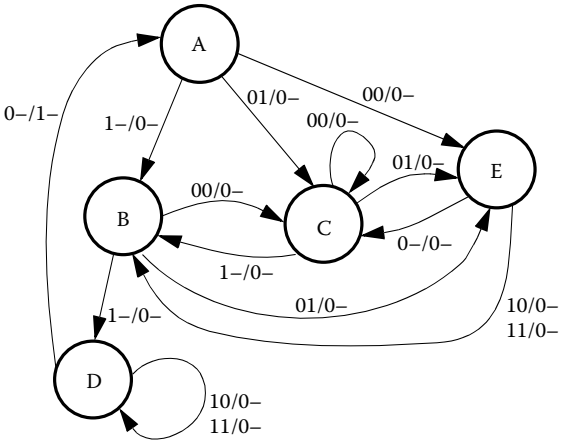


FIGURE 12.24 STG with "significant" output values only.

machine consists of three states, K, B, and D. This minimal machine is described by the state table shown in Table 12.2.

All possible input combinations appear explicitly in Table 12.2. However, it can be seen that the first two columns of the table are identical (i.e., the same next state and output value for all possible present states). Similarly, the last two columns of the table are identical. Therefore, in accordance with Step 4, these columns can be combined yielding the reduced state table shown in Table 12.3.

The reduced state table reveals that the minimal machine does not depend on the value of input X_2. Therefore, the conclusion is that input X_2 is not a DAI. This result implies that by knowing only the value of input X_1, the number of clock cycles (transitions in the original STG) required to reach the condition that output $Y_1 = 1$ can be determined regardless of the values of X_2. This is the case for any given initial state. It is important to emphasize that the paths in the *original* STG and their lengths are those which must be considered during the traversal process and that the modified state machine is used only for the purpose of detecting non-DAIs. The machine which is actually being traversed during the mixed-level simulation is the original state machine with all its functionality.

TABLE 12.2 Next State and Output Y_1 of the Minimal Machine

PS\\X_1X_2	00	01	10	11
K = (ACE)	K, 0	K, 0	B, 0	B, 0
B	K, 0	K, 0	D, 0	D, 0
D	K, 1	K, 1	D, 0	D, 0

TABLE 12.3 Minimal Machine with Implicit Input Enumeration

PS\\X_1X_2	0–	1–
K = (ACE)	K, 0	B, 0
B	K, 0	D, 0
D	K, 1	D, 0

To demonstrate the meaning of an input which is not a DAI, consider the original state machine represented by the graph in Figure 12.23 and assume that the initial state is A. Consider, for example, one possible sequence of values on input X_1 to be 0, 1, 0, 0, 1, 1, 0. By applying this input sequence, the sequence of values on output Y_1 is 0, 0, 0, 0, 0, 0, 1, regardless of the values applied to input X_2. Therefore, two input sequences which differ only in the values of the non-DAI input X_2 will produce the same sequence of values on the "significant" output Y_1. For example, the sequence $X_1X_2 = 00$, 10, 00, 01, 10, 10, 01 will drive the machine from state A to E, B, C, E, B, D, and back to A, and the sequence of values on output Y_1 will be as above. Another input sequence, $X_1X_2 = 01$, 11, 01, 00, 11, 11, 00, in which the values of X_1 are identical to the previous sequence, will drive the machine from state A to C, B, E, C, B, D, and back to A, while the sequence of values on output Y_1 is identical to the previous case. Therefore, the two input sequences will drive the machine via different states but will produce an identical sequence of values on the "significant" output (which also implies that the two paths in the STG have an equal length).

12.4.2.1.2 Traversing the STG for Best and Worst Delays

After extracting all possibilities for minimizing the number of unknown inputs, a method for determining values for those unknown inputs which are DAIs is required. The method developed for mixed-level modeling is based on the traversal of the original STG of the sequential interpreted element. As explained earlier, some combination of output values may signify the completion of processing the data. The search algorithm will look for a minimum or maximum number of state transitions (clock cycles) between the starting state of the machine and the state (Moore machine) or transition (state and input combination—Mealy machine), which generates the output values which signify the completion of data processing. Once this path is found, the input values for the unknown DAIs that will cause the SDE to follow this state transition path will be read from the STG and applied to the interpreted component in the simulation to generate the required best- or worst-case delay. Since the state machine is represented by the STG, this search is equivalent to finding the longest or shortest path between two nodes in a directed graph (digraph).

The search for the shortest path utilizes a well-known algorithm. Search algorithms exist for both single-source shortest path and all-pairs shortest path. One of the first and most commonly used algorithms is Dijkstra's algorithm [46], which finds the shortest path from a specified node to any other node in the graph. The search for all-pairs shortest path is also a well-investigated problem. One such algorithm by Floyd [47] is based on work by Warshall [48]. Its computation complexity is $O(n^3)$ when n is the number of nodes in the graph, which makes it quite practical for moderate-sized graphs. The implementation of this algorithm is based on Boolean matrix multiplication and the actual realization of all-pairs shortest paths can be stored in an $n \times n$ matrix. Utilizing this algorithm required some enhancements to make it applicable to mixed-level modeling. For example, if some of the inputs to the interpreted element are known (from the performance model), then the path should include transitions that include these known input values.

In contrast, the search for the longest path is a more complex task. It is an NP-complete problem that has not attracted significant attention. Since most digraphs contain cycles, the cycles need to be handled during the search to prevent a path from containing an infinite number of cycles. One possible restriction that makes sense for many state machines that might be used in mixed-level modeling is to construct a path that will not include a node more than once. Given a digraph $G(V, E)$ that consists of a set of vertices (or nodes) $V = \{v_1, v_2, \ldots\}$ and a set of edges (or arcs) $E = \{e_1, e_2, \ldots\}$, a *simple path* between two vertices v_{init} and v_{fin} is a sequence of alternating vertices and edges $P = v_{\text{init}}, e_n, v_m, e_{n+1}, v_{m+1}, e_{n+2}, \ldots, v_{\text{fin}}$ in which each vertex does not appear more than once. Although an arbitrary choice was made to implement the search allowing each vertex to appear in the path only once, the same algorithm could be easily modified to allow the appearance of each vertex a maximum of N times.

Given an initial node and a final node, the search algorithm developed for this application starts from the initial node and adds nodes to the path in a depth-first-search (DFS) fashion, until the final node is

reached. At this point, the algorithm backtracks and continues looking for a longer path. However, since the digraph may be cyclic, the algorithm must avoid the possibility of increasing the path due to a repeated cycle, which may produce an infinite path.

The underlying approach for avoiding repeated cycles in the algorithm *dynamically* eliminates the cycles while searching for the longest simple path. Let u be the node that the algorithm just added to the path. All the in-arcs to node u can be eliminated from the digraph at this stage of the path construction. The justification for this dynamic modification of the graph is that, while continuing in this path, the simple path cannot include u again. While searching forward, more nodes are being added to the path and more arcs can be removed temporarily from the graph. At this stage, two things may happen (1) either the last node being added to the path is the final node or (2) the last node has no out-arcs in the dynamically modified graph. These two cases are treated in the same way except that in the first case the new path is checked to see if it is longer than the longest one found so far. If it is, the longest path is updated. However, in both cases the algorithm needs to backtrack.

Backtracking is performed by removing the last node from the path, hence decreasing the path length by one. During the process of backtracking, the in-arcs to a node being removed from the path must be returned to the current set of arcs. This process will enable the algorithm to add this node when constructing a new path. At the same time, whenever a node is removed from the path, the arc that was used to reach that node is marked in the dynamic graph. This process will eliminate the possibility that the algorithm repeats a path that was already traversed. Therefore, by dynamically eliminating and returning arcs from/to the graph, a cyclic digraph can be treated as if it does not contain cycles. The process of reconnecting nodes, i.e., arcs being returned to the dynamic graph, requires that the original graph be maintained. A more detailed description of this search algorithm can be found in Ref. [49].

The restriction that a longest path does not include any *node* in the graph more than once may not be appropriate for all cases. In some mixed-level modeling cases, a more realistic restriction might be that the longest path does not include any *transition* (arc) more than once. A longest-path with no repeated arcs may include a node multiple times as long as it is reached via different arcs. In the case of more than one transition that meets the condition on the output combination, a search for the longest path should check all paths between the initial state and all of these transitions. However, such a path should include any of these transitions only once, and it should be the last one in the path.

Performing a search with the restriction that no arc is contained in the path more than once requires maintaining information on arcs being added or removed from the path. Maintaining this information during the search makes the algorithm and its implementation much more complicated relative to the search for the longest path with no repeated nodes. Therefore, a novel approach to this problem was developed. The approach used is to map the problem to the problem of searching for the longest path with no repeated nodes. This mapping can be accomplished by transforming the digraph to a new digraph, to be referred to as the transformed digraph (or Tdigraph). Given a digraph, $G(V, E)$, which consists of a set of nodes $V = \{v_1, v_2, \ldots, v_k\}$ and a set of arcs $E = \{e_1, e_2, \ldots, e_l\}$ the transformation τ maps $G(V, E)$ into a Tdigraph $TG(N, A)$, where N is its set of nodes and A its set of arcs. The transformation is defined as $\tau(G(V, E)) = TG(N, A)$, and contains the following steps:

Step 1. $\forall(e_i \in E)$ generate a node $n_i \in N$.
Step 2. $\forall(v \in V)$ and $\forall\left(e_p \in d^v_{\text{in}}, e_q \in d^v_{\text{out}}\right)$ generate an arc $a \in A$ such that a: $n_p \rightarrow n_q$.

The first step is used to create a node in the Tdigraph for each arc in the original digraph. This one-to-one mapping defines the set of nodes in the Tdigraph to be $N = \{n_1, n_2, \ldots, n_l\}$, which has the same number of elements found in the set E.

The second step creates the set of arcs, $A = \{a_1, a_2, \ldots, a_u\}$, in the Tdigraph. For each node in the original digraph and for each combination of in-arcs and out-arcs to/from this node, an arc in the Tdigraph is created. For example, given a node v with one in-arc e_i and one out-arc e_j, e_i is mapped to a node n_i, e_j is mapped to a node n_j, and an arc from n_i to n_j is created. In Step 2 of the transformation process, it is guaranteed that all possible connections in the original digraph are preserved as transitions

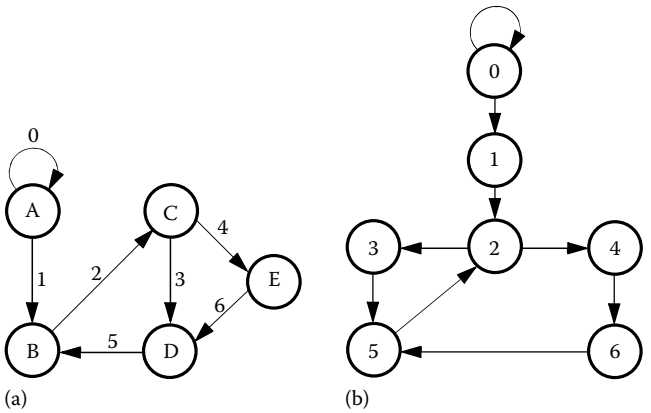


FIGURE 12.25 Transformation of a digraph.

between nodes in the Tdigraph. As a result of this transformation, the restriction on not visiting an arc more than once in the original digraph is equivalent to not visiting a node more than once in the Tdigraph. Therefore, by using this transformation, the problem of searching for the longest path with no repeated arcs in the original digraph is mapped to a search for the longest path with no repeated nodes in the Tdigraph. The algorithm described above can then be used to search the Tdigraph.

This transformation is best illustrated by a simple example. Consider the digraph shown in Figure 12.25a. The arcs in this digraph are labeled by numbers 0–6. The first step of the transformation is to create a node for each arc in the original digraph. Therefore, there will be seven nodes, labeled 0–6, in the Tdigraph as shown in Figure 12.25b. The next step is to create the arcs in the Tdigraph. As an illustration of this step, consider node C in the original digraph. Arc "2" is an in-arc to node C while arcs "3" and "4" are out-arcs from node C. Applying Step 2 results in an arc from node "2" to node "3" and a second arc from node "2" to node "4" in the Tdigraph. Considering node B in the original digraph, the Tdigraph will include an arc from node "1" to node "2" and an arc from node "5" to node "2." This process is repeated for all the nodes in the original digraph until the Tdigraph, as shown in Figure 12.25b, is formed. A search algorithm can now be executed to find the longest path with no repeated nodes.

A mixed-level modeling methodology, which is composed of all the methods described above, has been integrated into the ADEPT environment. The steps for minimizing the unknown inputs can be performed prior to simulating the mixed-level model. On the other hand, the search for longest/shortest possible delay must be performed during the simulation itself. This requirement arises because each token may carry different information which may alter the known input values and, therefore, alter the search of the STG. The STG traversal process has been integrated into the ADEPT modeling environment using the following steps: (1) when a token arrives to the mixed-level interface, the simulation is halted and the search for minimum/maximum number of transitions is performed and (2) upon completion of the search, the simulation continues while applying the sequence of inputs found in the search operation. The transfer of information between the VHDL simulator and the search program, which is implemented in C is done by using the simulator's VHDL/C interface. A component called the Stream_Generator has been created that implements the STG search process via this interface.

Since mixed-level models are part of the design process and are constructed by refining a performance model, it is likely that many tokens will carry identical relevant information. This information may be used for selective application of the STG search algorithm, hence increasing the efficiency of the mixed-level model simulation. For example, if several tokens carry exactly the same information (and assuming the same initial state of the FSM), the search is performed only once, and the results can be used for the following identical tokens.

12.4.2.2 Example of Mixed-Level Modeling with an FSMD Component

This section presents an example of the construction of a mixed-level model with an FSMD interpreted component. The example is based on the performance model of an execution unit of a particular processor. This execution unit is composed of an integer unit (IU), a floating-point unit (FPU), and a load-store unit (LSU). These units operate independently although they receive instructions from the same queue (buffer of instructions). If the FPU is busy processing one instruction and the following instruction requires the FPU, it is buffered, waiting for the FPU to be free again. Meanwhile, instructions which require the IU can be consumed and processed by IU at an independent rate. Both the FPU and the IU have the capability of buffering only one instruction. Therefore, if two or more consecutive instructions are waiting for the same unit, the other units cannot receive new instructions (since the second instruction is held in the main queue). One practical performance metric that can be obtained from this model is the time required for the execution unit to process a given sequence of instructions.

Because the FPU was identified as the most complex and time critical portion of the design, a behavioral description of a potential implementation of it was developed. At this point, a mixed-level model, in which the behavioral description of the FPU is introduced into the performance model, was constructed using the interface described above. The mixed-level model is shown in Figure 12.26. The mixed-level interface is constructed around the interpreted block which is the behavioral description of the FPU. This FPU is an FSMD type of element, and the interpreted model consists of a clock cycle-accurate VHDL behavioral description of this component. The inputs to the FPU include the operation to be performed (Add, Sub, Comp, Mul, MulAdd, and Div), the precision of the operation (single or double), and some additional control information. The number of clock cycles required to complete any instruction depends on these inputs.

Figure 12.27 shows the execution unit performance derived from the mixed-level model for three different instruction traces. In this case, only 40% of the inputs have values that are supplied by the abstract performance model, the remainder of the values for the inputs are derived using the techniques described in Section 12.4.2.1. The performance value is normalized by defining unity to be the amount of time required to process a trace according to the initial uninterpreted performance model. In this example, the benefit of the simulation results of the mixed-level model is clear. It provides performance bounds in terms of best- and worst-case delays, for the given implementation of the FPU.

12.4.3 Interface for Mixed-Level Modeling with Complex Sequential Components

A methodology and components for constructing a mixed-level interface involving general sequential interpreted components that can be described as FSMDs was detailed in the previous section. However, many useful mixed-level models can be constructed that include sequential interpreted components that are too complex to be represented as FSMDs, such as microprocessors, complex coprocessors, network interfaces, etc. In these cases, a more ''programmable'' mixed-level interface that is able to deal with the additional complexity in the timing abstraction problem was needed. This section describes the ''watch-and-react'' interface that was created to be a generalized, flexible interface between these complex sequential interpreted components and performance models.

The two main elements in the watch-and-react interface are the *Trigger* and the *Driver*. Figure 12.28 illustrates how the Trigger and Driver are used in a mixed-level interface. Both elements have ports that can connect to signals in the interpreted components of a model. Collectively, these ports are referred to as the *probe*. The primary job of the Trigger is to detect events on the signals attached to its probe, while the primary job of the Driver is to force values onto the signals attached to its probe. The Driver decodes information carried in tokens to determine what values to force onto signals in the interpreted model (the U/I interface) and the Trigger encodes information about events in the interpreted model onto tokens in the performance model (the I/U interface).

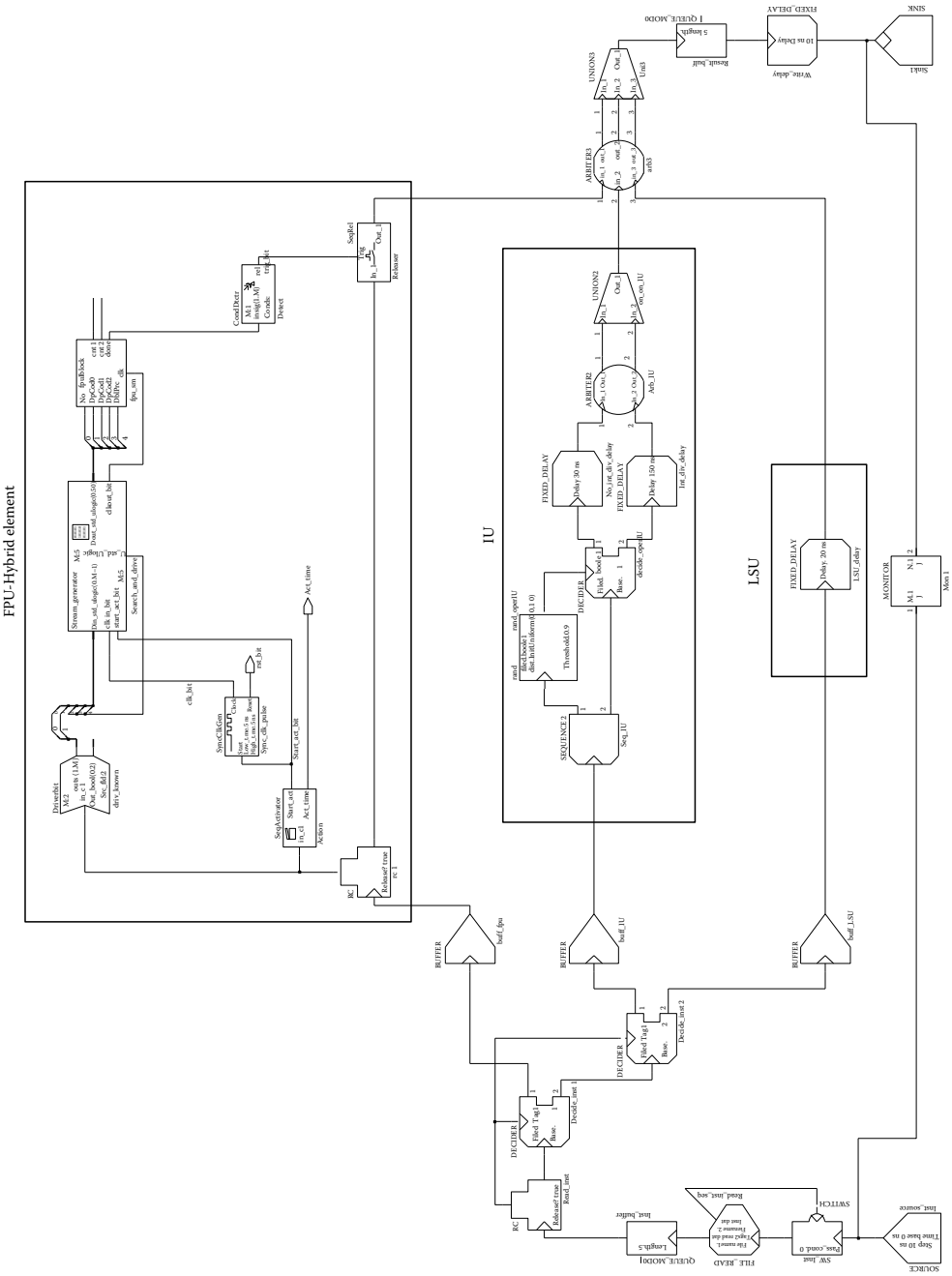


FIGURE 12.26 Mixed-level model with the FPU behavioral description.

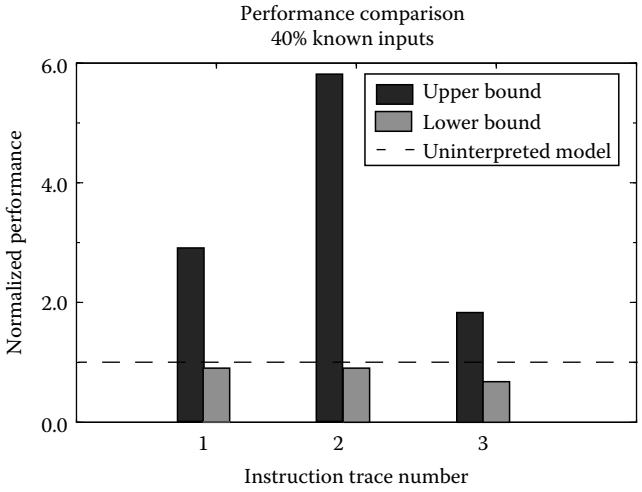


FIGURE 12.27 Performance comparison of the execution unit for three instruction traces.

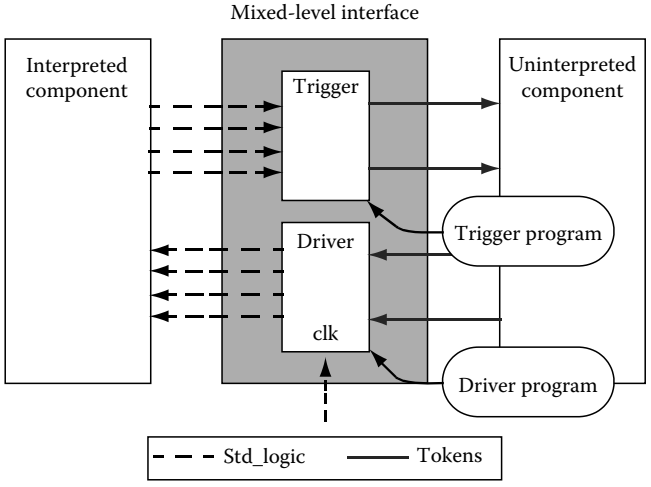


FIGURE 12.28 Watch-and-react mixed-level interface.

The Trigger and Driver were designed to be as generic as possible. A command language was designed that specifies how the Trigger and Driver should behave to allow users to easily customize the behavior of the Trigger and Driver elements without having to modify their VHDL implementation. This command language is interpreted by the Trigger and Driver during simulation. Because the language is interpreted, changes can be made to the Trigger and Driver programs without having to recompile the model, thus minimizing the time required to make changes to the mixed-level model.

The schematic symbol for the Trigger is shown in Figure 12.29. The primary job of the Trigger is to detect events in the interpreted model. The probe on the Trigger is a bus of std_logic signals `probe_size` bits wide, where `probe_size` is a generic on the symbol. There is one token output called `out_event_token` and one token output called `out_color_token`. Tokens generated by the Trigger when events are detected are placed on the `out_event_token` port. The condition number (an integer) of the event that caused the token to be generated is placed on the `condition`

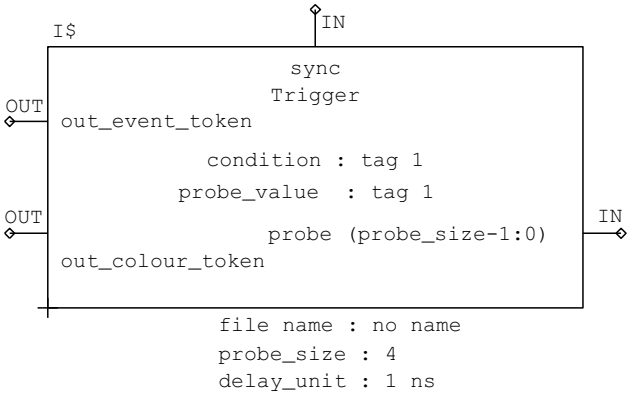


FIGURE 12.29 Schematic symbol for the Trigger.

tag field, which is specified as a generic on the symbol. Also, each time a signal changes on the probe, the color of the token on the `out_color_token` port changes appropriately regardless of whether an event was detected or not. The probe value is placed on the `probe_value` tag field of the `out_color_token` port. The sync port is used to synchronize the actions of the Trigger element with the Driver element as explained below.

The name of the file containing the Trigger's program is specified by the `filename` generic on the symbol. The `delay_unit` generic on the symbol is a multiple that is used to resolve the actual length of an arbitrary number of delay units specified by some of the interface language statements.

The schematic symbol for the Driver element is shown in Figure 12.30. The primary job of the Driver is to create events in the interpreted model by driving values on its probe. These values come from either the command program, or from the values of tag fields on the input tokens to the Driver. The probe on the Driver is a bus of std_logic signals `probe_size` bits wide, where the `probe_size` is a generic on the symbol. There is one token input called `in_event_token`, one token input called `in_color_token`, and a special input for a std_logic type clock signal called `clk`. The `clk` input allows Driver to synchronize its actions with an external interpreted clock source. The `sync` port is used to synchronize the actions of the Driver element with the Trigger element. The `filename` and `delay_unit` generic on the symbol function the same as for the Trigger.

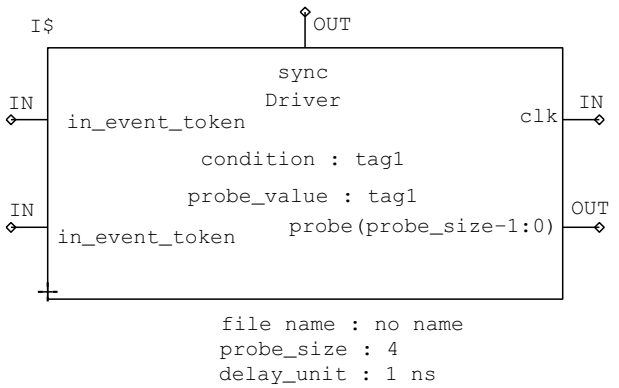


FIGURE 12.30 Schematic symbol for the Driver.

As discussed previously, a command language was developed to allow the user to program the actions of the Trigger and Driver. This command language is read by the Trigger and Driver at the beginning of the simulation. Constructs are available within the command language to allow waiting on events of various signals and driving values or series of values on various signals, either asynchronously, or synchronous with a specified clock signal. In addition, several looping and go-to constructs are available to implement complex behaviors more easily. A summary of the syntax of the command language constructs is shown in Table 12.4.

12.4.3.1 Example of Mixed-Level Modeling with a Complex Sequential Element

This section presents an example which demonstrates how the Trigger and Driver elements can be used to interface an interpreted model of a complex sequential component with an uninterpreted model. In this example, the interpreted model is a microprocessor-based controller and the uninterpreted model is that of a motor control system including a motor controller and a motor. The motor controller periodically asserts the microcontroller's interrupt line. The microcontroller reacts by reading the motor's current speed from a sensor register on the motor controller, calculating the new control information, and writing the control information to the motor controller.

The microcontroller system consists of interpreted models of eight-bit, RISC-like microprocessor, RAM, memory controller, I/O controller, and clock. The memory controller handles read and write requests issued by the processor to the RAM, while the I/O controller handles read and write request issued by the processor to an I/O device. In the system model, the I/O device is the uninterpreted model of a motor controller. A schematic of the model is shown in Figure 12.31.

Three Triggers and two Drivers are used to construct the mixed-level interface for the control system model. One of the Triggers is used to detect when the I/O controller is doing a read or write. The other two Triggers are used to collect auxiliary information about the operation, such as the address on the address bus and data on the data bus. One of the Drivers is used to create a microcontroller interrupt and the other Driver is used to force data onto the data bus when the processor reads from the speed sensor register on the motor controller.

The interrupt Driver's program is listed in Figure 12.32. The program begins by forcing the interrupt line to "Z" and then waiting for a token from the uninterpreted model of the motor controller to arrive. Once a token arrives, the program forces the interrupt line high for 10 clock cycles. This condition is accomplished by using a `for-next` statement with a `wait_on_rclk` as the loop body. After 10 clock cycles, the program jumps to line 10 where the cycle begins again.

The data Driver's program is listed in Figure 12.33. The program begins by also waiting for a token from the uninterpreted model of the motor controller to arrive. If the value on the condition tag field of the token is 1, then "ZZZZZZZZ" is forced onto the data bus. If the condition tag field value is 3, then the value on the `probe_value` tag field of the `in_color_token` input is forced on the data bus. This process is repeated for every token that arrives.

The I/O Trigger's program is listed in Figure 12.34. This Trigger waits until there is a change on the probe. Once there is a change, the program checks to see if the I/O device is being unselected, written to, or read from. If one of the `when` statements matches the probe value, then its corresponding `output` statement is executed. An output of 1 corresponds to the I/O device not being selected. An output of 2 corresponds to the processor writing control information to the motor controller. An output of 3 corresponds to the processor reading the motor's speed from the sensor register on the motor controller.

Figure 12.35 shows the results from the mixed-level model as a plot of the sensor output and the processor's control response. Some random error was introduced to the sensor's output to reflect variations in the motor's load as well as sensor noise. The target speed for the system was 63 ticks per sample time (a measure of the motor's RPM). The system oscillates slightly around the target value because of the randomness introduced into the system.

TABLE 12.4 Trigger and Driver Command Programming Language Constructs

Command	Element	Meaning
– <comment>	Both	Comment—no action
alert_user	Both	Print message in simulation window
delay_for <T>	Both	Delay for <T > time units where the time unit is specified as a generic on the module's symbol
end	Both	End the command program
for <N> <loop body> **next**	Both	Iterate *N* times over the sequence of statement in the loop body
goto <L>	Both	Go to line <L> in the command program
output_sync	Both	Generate a token on the sync output of the module
wait_on_sync	Both	Wait for an occurrence of a token on the sync port of the module
case_probe_is **when** <STD_LOGIC_VAL> <sequence of statements> **when** . . . **end_case**	Trigger	Conditionally execute some sequence of statements depending on the STD_LOGIC value of the probe signal
output <INTEGER_VAL> **after** <T>	Trigger	Generate a token on the Trigger's output with the value of <INTEGER_VAL> on the tag field specified by the generic on the symbol after <T > time units
trigger	Trigger	Must appear as the first statement in the Trigger's command program
wait_on <STD_LOGIC_VAL>	Trigger	Wait until the probe signal takes on the specified STD_LOGIC value
wait_on_probe	Trigger	Wait until there is ANY event on the probe signal
case_token_is **when** <**INTEGER_VAL**> <**sequence of statements**> **when . . .** **end_case**	Driver	Conditionally execute some sequence of statements depending on the integer value of the input token's tag field specified by the generic on the symbol
driver	Driver	Must appear as the first statement in the Driver's command program
dynamic_output_after <T>	Driver	Force the value from the specified tag field of the input token onto the probe after <T> time units
output <STD_LOGIC_VAL> **after** <T>	Driver	Force the specified STD_LOGIC value onto the probe signal after <T> time units
wait_on <INTEGER_VAL>	Driver	Wait until a token with the given integer value on the tag field specified by the generic on the symbol arrives on the in_event_token signal
wait_on_fclk	Driver	Wait until the falling edge of the clock occurs
wait_on_rclk	Driver	Wait until the rising edge of the clock occurs
wait_on_token	Driver	Wait until a token arrives on the in_event_token signal

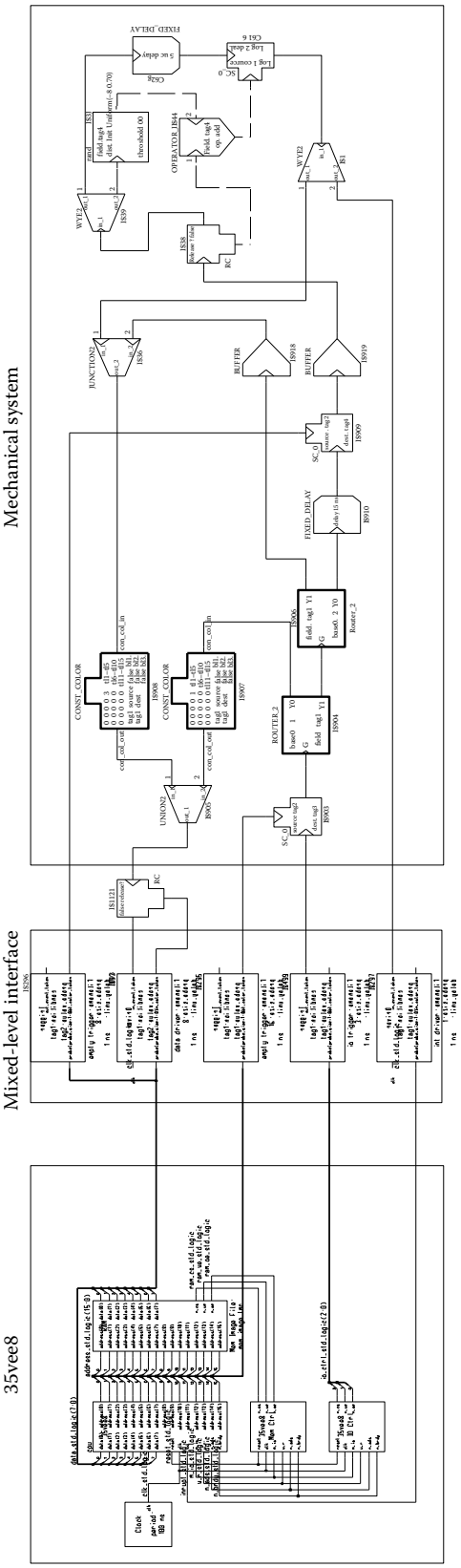


FIGURE 12.31 Schematic of control system model.

```
driver
10 output Z after 0
20 wait_on_token
30 output 1 after 0
40 for 10
50    wait_on_rclk
60 next
70 goto 10
80 end
```

FIGURE 12.32 Interrupt Driver's program for the control system.

```
driver
10 wait_on_token
20 case_token_is
30     when 1
40        -- sensor not selected
50        output ZZZZZZZZ after 0
60     when 3
70        -- sensor selected for reading
80        dynamic_output_after 0
90 end_case
100 goto 10
110 end
```

FIGURE 12.33 Data Driver's program for the control system.

```
trigger
10 wait_on_probe
20 case_probe_is
30     when 111
40        -- sensor not selected
50        output 1 after 0
60     when 001
70        -- sensor selected for writing
80        output 2 after 0
90     when 010
100       -- sensor selected for reading
110       output 3 after 0
120 end_case
130 goto 10
140 end
```

FIGURE 12.34 I/O Trigger's program for the control system.

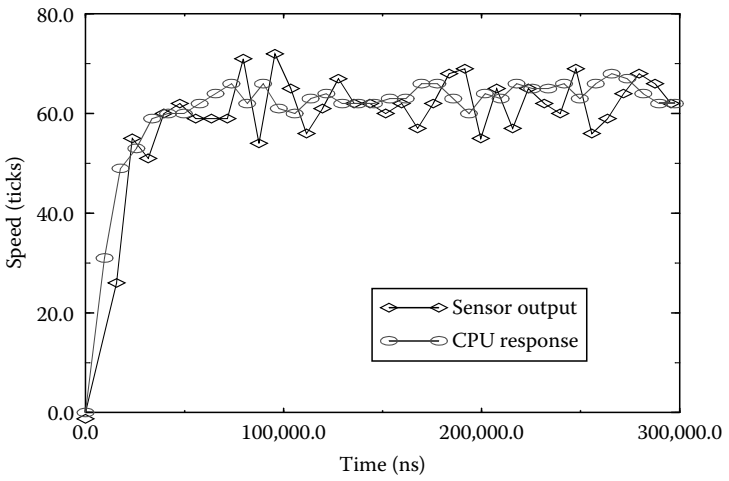


FIGURE 12.35 Sensor and processor outputs for the control system.

12.5 Performance and Mixed-Level Modeling Using SystemC

This section describes a performance-modeling environment capable of mixed-level modeling that is based on the SystemC language [50]. The environment is intended to model the system at the Processor Memory Switch level much like the Honeywell PML environment described earlier. The goal of this work was to show how SystemC could be used to construct a mixed-level modeling capability.

In the SystemC-based PBMT (Performance-Based Modeling Tool), the user begins by describing the functions executed by the system as a task graph. A task graph is a representation of the flow of execution through an application. The nodes in a task graph represent computational tasks, and the edges in a task graph represent the flow of control, or the actual transfer of data, between tasks. An example of a task graph for a simple application is shown in Figure 12.36. Note that the topology shown in the figure, the example application has the opportunity for some tasks, such as Task 2, Task 3, and Task 4, to be executed in parallel if the system architecture upon which the application is to be executed, allows for it.

Once the task graph model is constructed, the user then selects a system architecture on which the application will execute. The system architecture is specified by the number of processors in the architecture, and an interconnect topology used to provide communications between them. The available interconnect topologies include a bus (a single shared communications resource), a crossbar switch (a partially shared communications resource), or fully connected (a completely nonshared communication resource). Note that in this high-level architecture model, what actually constitutes a processor in the system is not specified. That is, a processor is simply modeled as a computational resource and may in implementation be a general-purpose processor (of any clock speed), a special purpose processor like a DSP, or custom hardware for a specific task.

Once the system architecture is specified, all that remains is for the user to specify upon which processor each of the tasks is to execute and what the total execution time for that task on the specific processor will be. This delay value may be either fixed, or dependent on the amount of data that is passed into the task by the previous task in the graph. Once this task-to-processor mapping and delay specification is done, the complete SystemC model is constructed and simulated using either the reference simulator, included as part of the SystemC distribution available in Ref. [10], or the commercial Mentor Graphics ModelSim simulator which includes the capability to cosimulate SystemC models along with Verilog or VHDL models.

Figure 12.37 shows the results of executing the task graph of Figure 12.36 on three different system architectures. All of the architectures utilize a single shared bus for communications. The first result is for an architecture with only a single processor. In this case, the obvious result is that all of the tasks execute in sequence on the single processor and the run time is simply the sum of the individual task execution times. The second result is for a three-processor architecture. In this case, after Task 1

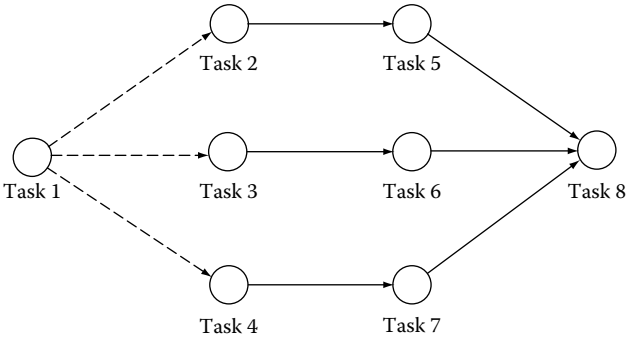


FIGURE 12.36 Example task graph.

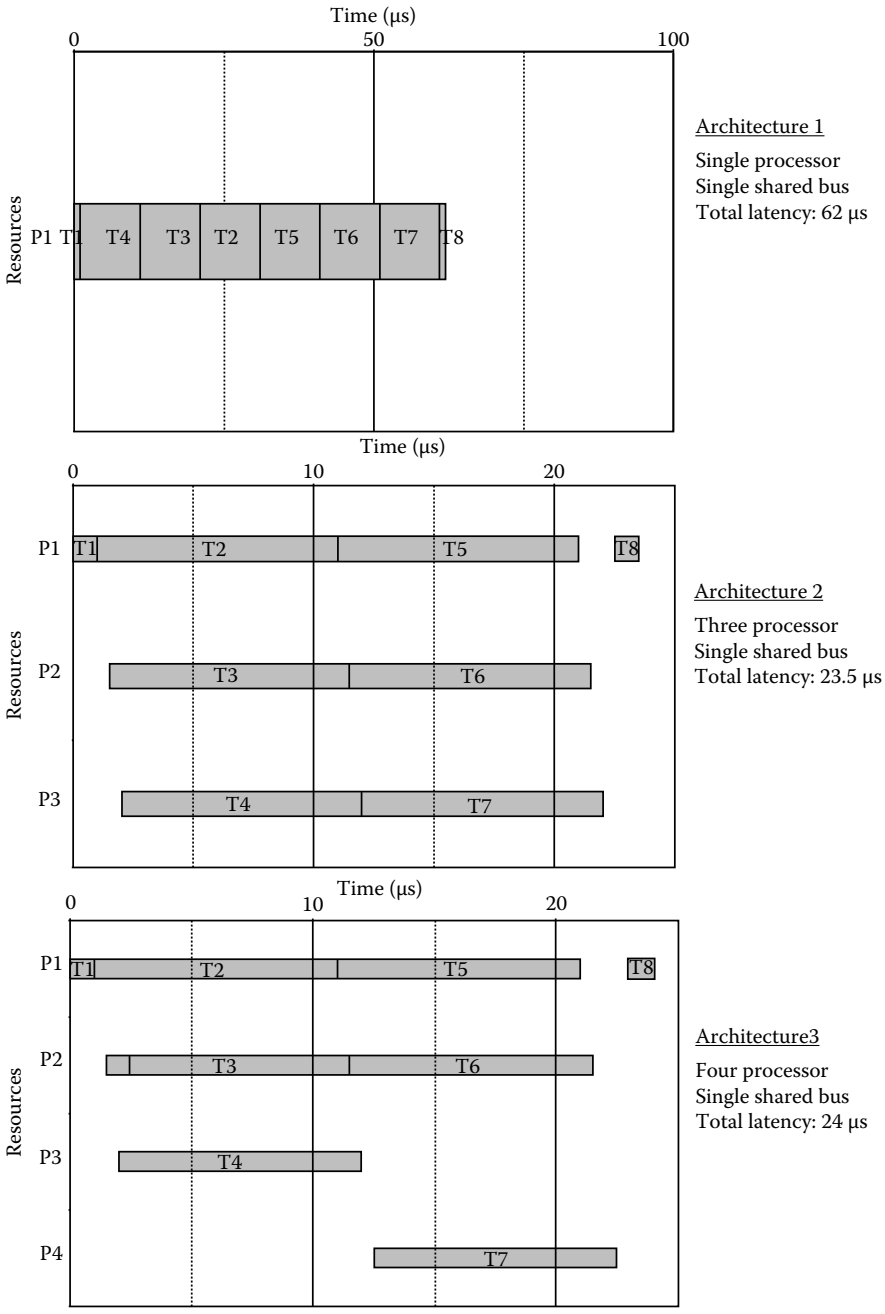


FIGURE 12.37 Parallel example shared bus timelines.

completes, some latency can be seen before Tasks 3 and 4 begin execution. This accounts for the communication time required to send data from the processor that executed Task 1 to the processors that are executing Tasks 3 and 4. In addition, the graph shows that Task 4 begins execution after Task 3 because of the contention for the single shared bus communications resource. Likewise, the latency between the end of execution for Tasks 5–7, and the start of execution for Task 8 accounts for the time required to communicate Tasks 6 and 7's results back to the single processor that is schedule to execute

Task 8. The overall run time for this configuration is much less than the first example because the inherent parallelism in the application is being exploited by the selected architecture. Finally, the third result is for a four-processor architecture. In this simulation, Tasks 1 and 8 are allocated to the fourth processor, separate from the other Tasks 2–7. This results in additional communications time being required using the single bus to transfer all of the data from, and back to, that extra processor. Thus, as can be seen from the graph this architecture actually takes longer than the three processor architecture to execute the application.

12.5.1 SystemC Background

This section describes some of the basic concepts of SystemC and how it is used to model systems at a high level. SystemC is a library extension to C++. Essentially, SystemC uses a set of classes and predefined functions to build a new ''language'' on top of C++. The basic unit of a SystemC model is the `SC_MODULE` class. The `SC_MODULE` is roughly equivalent to an entity in VHDL. It can have input, output, and bidirectional ports. However, it is a C++ class and as such it can, and does, have member functions. Any member function that is declared as public can be accessed just like the member function of any other class. Every `SC_MODULE` must have at least one public member function, its constructor. In the constructor the module is given a SystemC name, and declares if it has any processes, and does anything else necessary to get the model ready for simulation. There are two types of processes in SystemC, the `SC_THREAD` and the `SC_METHOD`. The behavior of a process must first be declared as a member function, then that member function can be declared to be either an `SC_THREAD` or an `SC_METHOD` in the constructor. The primary differences are how they behave when they reach the end of their definition, and when the scheduler activates them.

An `SC_THREAD` will terminate, and never be activated again, if it reaches the end of its description. Typically `SC_THREAD`s are infinite loops with one or more wait statements to break the execution, and wait on some signal change or other event. An `SC_METHOD` will be activated any time something it is sensitive to changes, and will run once through to the end of its description. If something an `SC_METHOD` is sensitive to changes again, it will run again. `SC_THREAD`s do have the ability to use a form of the wait command that is not available to `SC_METHOD`s. They may use a wait command with no parameters which will cause them to be reactivated and continue execution with the line after the wait statement when something in their sensitivity list has an event.

As indicated by the discussion of processes, SystemC uses an event-driven simulation methodology. The library provides basic signal classes that have a notify-update sequence much like VHDL. As mentioned in the Transaction Level Modeling discussion above, SystemC has a concept of channels. A channel is generally some means of moving information. The basic signal classes provided are base channels. A designer could potentially design some new base channel type that has the same interfaces as the existing base channels. However, building a new base channel is rather involved since its implementation must interact directly with the scheduler to implement the notify/update semantics of a signal. Additionally, such a user-designed base channel would still only have a basic interface, and could not have any internal processes, thus would not provide significant abstraction leveraging for the amount of time required to design it. Instead designers should use what is called a hierarchical channel for most modeling needs. A hierarchical channel is a channel that is made up of a number of elements of base classes. A hierarchical channel can have any number of ports, or methods. The methods that a module could use to access the channel, and convey information through it, are called interface methods. For a module to be able to access an interface method the channel needs to have a defined interface that it inherits. The `sc_interface` class is used as a base class to define such an interface. The file in Figure 12.40 shows one such interface class. Once an interface class has been defined and inherited by the channel, then any module with a port of that type can be bound to the channel's interface. In addition to the base channel classes, the SystemC library provides an event object that can be waited upon using the same syntax as a wait for a signal change event. For more details on the syntax, classes, and functions of SystemC the reader may refer to Ref. [10].

At the time of this writing, there are several options for simulating SystemC models. The two most robust SystemC simulators are the reference simulator, available with the SystemC distribution, and the ModelSim simulator as mentioned above. Because of some internal implementation details which differ between the two simulators, there are minor code differences required in the performance models between the two simulators. The differences and the techniques for enabling the models to be compiled and run in either simulators are described in the sections below.

12.5.2 SystemC Performance Models

The performance models described herein take advantage of the C++ foundations of SystemC to provide a highly parameterizable simulation environment that loads most of the parameters at run time. Figure 12.38 shows the class definition for the top level of the simulation model, and the relevant comments. The top-level entity makes use of a pointer to an object of type `SIM`, declared in the header file *generic.h*, to allow the constructor arguments to be read in from the *top_config.txt* file rather than statically specified in the source code. When the simulation is loaded the SystemC constructor (`SC_CTOR` in the code) will be called. The constructor will create a `streambuf` object, a number of processors variable, a connection type variable, as well as a pointer variable to point to an object of type `SIM`. It will then open the *top_config.txt* file and associate it with the `streambuf` variable. Then it will read an unsigned value into the number of processors variable, then read an unsigned value into the connection type variable. Once those values are read in then the constructor uses the ''new'' command to instantiate a new object of type `SIM`. The ''new'' command allows passing variable constructor arguments to the objects constructor, so the new `SIM` object's constructor builds the object with the value of `num_processors` processors, and the value of `connect_type` interconnect number. If the number of processors or interconnect type was specified via a template argument, or a fixed constructor

```
#ifndef MY_MAIN
#define MY_MAIN

#include <systemc.h>
#include "generic.h"

SC_MODULE(main_sim)
{
   SIM *simulation;

   SC_CTOR(main_sim)
   {
      ifstream topFile;
      unsigned num_processors;
      //1 to 9, could be to 10^13 or memory constraints except for modelsim
namebinding
      unsigned connect_type;   //1,2,3 are valid
      topFile.open("top_config.txt");//this doesn't do error check!
      topFile>>num_processors;
      //get info from file & set num_processors,connect_type
      topFile>>connect_type;// read in the interconnect type to use
      simulation = new SIM("simulation",num_processors,connect_type);
      // systemC name,number of processor ,connection class
   };

};
#endif
```

FIGURE 12.38 Top-level class definition.

```
#include <systemc.h>
#include "generic_main.h"
#ifdef MTI_SYSTEMC
SC_MODULE_EXPORT(main_sim);
#else
//for OSCI reference simulator
int sc_main(int ac, char *av[] )
{
    main_sim my_sim("my_sim");
    sc_start(500, SC_NS);
    sc_close_vcd_trace_file(main_trace);
    return 0;
};
#endif
```

FIGURE 12.39 Top-level C++ file.

value (e.g., "`simulation = new SIM("it,9,1);`") in the source code then the simulation would have to be recompiled every time something was changed.

Notice that all the implementation details of the SystemC modules are declared in header files. This is the recommended way of describing models to compile the SystemC code for the ModelSim simulator. The included guards are also a must for any `SC_MODULE` definitions to ensure they are not included multiple times by the ModelSim SystemC compiler, SCCOM. The actual C++ file that is compiled to generate the simulation models is shown in Figure 12.39. The `SC_MODULE_EXPORT(module_class_name)` is the function used to create the ModelSim model for the specified entity. The `MTI_SYSTEMC` compiler definition is defined specifically for compilation using SCCOM, and allows having a single set of code for both the ModelSim and reference simulator. Any ModelSim specific code can be placed inside a `#ifdef MTI_SYSTEMC` compiler directive statement so that the compiler only uses it when compiling for ModelSim. The `#else` statement prevents the ModelSim compiler from trying to compile the reference simulator-specific portions, and the `#endif` closes out the if-else statement.

The *generic.h* header file defines the class `SIM`. It makes use of the shared base class `my_basic_rw_port` to declare a pointer that will allow assignment of an instantiation of any of the three channel types developed for the environment. The `my_basic_rw_port` base class definition is shown in Figure 12.40. Notice the use of the keyword `virtual`. Since the method is declared as virtual, all

```
#ifndef INTERFACE_TYPE
#define INTERFACE_TYPE
include <systemc.h>
/* this header describes the interface type...*/

class my_basic_rw_port
  : public virtual sc_interface
{
public:
  // basic read/write interface
  //virtual bool read(int address, int data) = 0; //not used anymore!
  virtual bool read(int address, int source_address, int data,
                    int dest_task) = 0;
  virtual bool write(const int source, int address, int data,
                     int dest_task) = 0;
  virtual bool non_blk_write(const int dest_address, int source,
                             int data, int dest_task)=0;
}; // end class
#endif
```

FIGURE 12.40 "interface_type.h."

classes that inherit from this class must either provide, or inherit, an implementation for the `read`, `write`, and `non_blk_write` methods. This common interface is also what allows the type compatibility for pointer assignment used in the *generic.h* file. This is described in more detail in Section 12.5.4.

12.5.3 Processor Model

The processor model has a fairly simple structure. There are three methods in this model, the constructor, and two member functions. The first member function describes the performance only behavior of the processor, the second describes the mixed-level behavior. The mixed-level functionality will be described in detail in Section 12.5.9.

In addition to those methods, the processor model has a number of objects that are members of the class. It has three integer variables for passing command arguments to the interface methods, a pointer for a `command_in` object that opens the command file and parses the model execution commands for the processor model, a `command_type` object that is used to return the commands from the `command_in` object, a signal of enumerated type `action_type` to display the current action, an unsigned signal to display the current task number, and a pointer for a refined computation model. Figure 12.41 graphically shows the objects in the processor model. On the far right is the IO port. It is mapped to the interconnect interface. Next on the right are the three integer variables used to pass information to the interconnect calls. Below the variables are boxes representing the two signals that allow the state of the processor to be viewed in the ModelSim waveform window. Then to the left there are the performance and mixed-level descriptions. One of these member functions will be turned into an `SC_THREAD` and will control the model's behavior during simulation. On the top labeled as refined model, is an outlined box representing the pointer to a refined model object. Below is a storage location (labeled command) that the `command_in` object will return values in. In the bottom left is a dashed box representing a pointer to a `command_in` object. In the top left is a box representing the constructor. The IO port and constructor are both on the edge of the processor model because they are the only ones that interact with other objects in the simulation.

The constructor receives a processor number, and creates a `command_in` object with the appropriate processor number for the processor the current instance represents. The `command_in` object creates a string with the proper processor number in the middle, and uses it to open the processor's command file. It is called by the active behavior to read in the next command once the previous one has executed. Its primary function is to remove command parsing from the processor model. Having it as a separate object makes changing how the commands for the processor are read in or generated a simple matter of including a different implementation of the object. Once the `command_in` object is created and initialized, the processor model constructor then instantiates either the performance only or mixed-level implementation, and opens its log file. It does this by registering the proper member function with

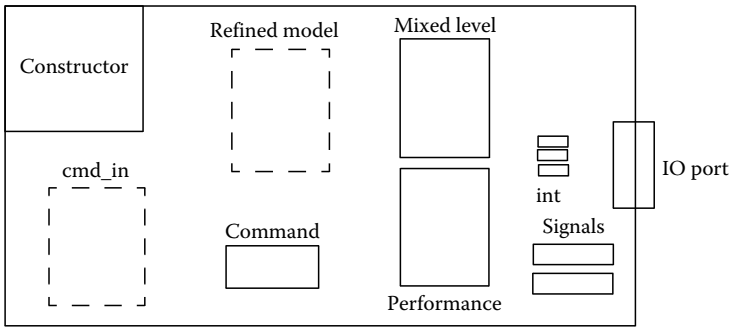


FIGURE 12.41 Graphical representation of processor model.

```
void tlm_behav()
{
   int temp2;
   double temp_time;
   while(1)
   {
      cmd_input->get_command(&this_command);
      switch (this_command.command)
      {
      case 1://ie send
         {...
            break;}
      case 2: //ie receive
         {...
            break;};
      case 0: //ie compute
         {...
            break;};
      case 3: //non_blocking_send
         {...
            break;};
      case 4: //io done
         {...
            break;};
      case 5: //loop/branch
         {...
            break;};
      default: //idle the processor
         {...
            break;};
      };//end of switch
   wait(SC_ZERO_TIME); //to break things up in the text output...
   };//end of infinite while loop
};
```

FIGURE 12.42 Main processor loop.

the simulation kernel as an `SC_THREAD`. During simulation the processor model essentially reads in a command from a file, then uses a case statement to perform whatever command was read in and repeats until it reaches a done command, an end of file, or some command it does not recognize. Figure 12.42 shows the framework of the main processor loop.

Notice in the source code that the processor has a port of the same type as the base type for the interconnect channels `my_basic_rw_ port`. This port is bound to the interface of the channel object. If the command read in from the command file is a send or receive command, then the processor uses the port as a pointer to the interface to the channel object and accesses the appropriate interface method to perform the send or receive operation. In the models here the thread in the processor model actually executes all the code in the blocking send and receive methods, so the processor model is incapable of doing anything else until the blocking io function returns.

12.5.4 Channels

The channels used in these models are considered hierarchical channels. They are not any of the predefined SystemC primitive channel types, they are composed of multiple objects, and they contain a number of threads. To allow for a variable number of processors to connect via the channels they have only an interface and no ports. In SystemC, all ports must be bound to something, be it another port, an interface, or a signal. Interfaces however may exist even if nothing is bound to them. So for maximum

flexibility, channels should provide an interface, and any connected modules should have a port of the type of the interface and have that port bound to the interface on the channel.

Since multiple ports can be bound to a single interface, the channel object can have any number of processors bound to it. However, the crossbar and fully connected channels' behavior is determined in part by the number of processors present, so all channels are passed a constructor argument that tells them how many processors are present in the simulation. The channel models are the most extensive models since their behavior is an abstract representation of all of the characteristics of an interconnect topology. They model the arbitration, data transfer, and blocking/nonblocking characteristics of the interconnect without restricting the designer to a particular implementation. Since nonblocking sends are allowed they also implicitly model a sending queue.

All the channel models are based on the `comm_medium` class. The `comm_medium` class provides a logical connection between processors, with signals for the source processor number, the destination processor number, and transaction type, as well as blocking and nonblocking read and write methods, and an arbitration thread. In addition, the `comm_medium` class provides two threads to allow for nonblocking reads and writes. Figure 12.43 shows the members of the `comm_medium` class. The four signals shown in the top left are signals to show the current state of the logical connection in the waveform window. To the right are the two wait queues, one for write request, and one for read request. New transaction requests received via the interface methods are placed into these queues. The Boolean `no_match` variable maintains whether there is currently a match between read operations and received data, the integers below it are used to store the values of the current processor for the sender and receiver, and to store the location in the queue of the current send request being executed, and the current receive request being executed. Below the integers is a dummy variable whose sole purpose is to fix an existing bug in the implementation of the SystemC simulator. In the bottom left of the figure are the member functions of the object. The functions all the way to the left are the functions intended to be accessed by other objects, the remaining four are intended for internal use only, though they are declared as public and thus visible to other objects. The functions with a star after them are registered with the simulation scheduler as `SC_THREAD`s. The top two events in the bottom right of the figure are used by the read and write methods to notify the arbitrator process that there may be new pairs of requests that could be activated to communicate. The event in the very bottom is used to coordinate the execution of two threads when they communicate. In the top right of the figure are two transaction pointers. These

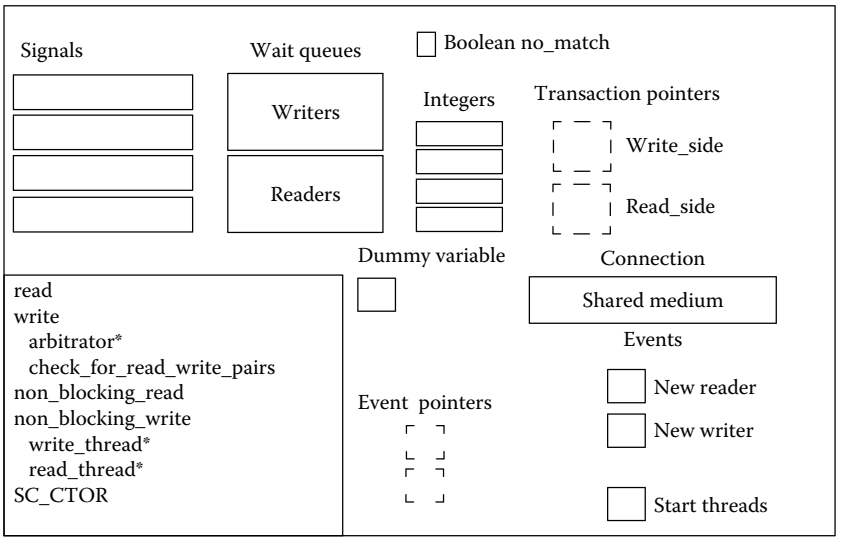


FIGURE 12.43 Graphical representation of ''comm_medium'' class.

pointers are used by the arbitrator to keep track of which two transactions it is dealing with. The event pointers in the bottom center of the figure are also used by the arbitrator. When the arbitrator has selected two transaction requests to communicate it does not actually handle the communication. There is a write thread, and read thread that execute the `transact()` code in each of the two transaction objects. For any blocking transaction objects the calling processor's thread is suspended in the interface call waiting on the transaction to notify it that the transaction is complete. Any nonblocking transaction objects have already had their calling thread return to the processor model code. The arbitrator has no need to check for any potential request pairs until after the two threads have completed their transaction. These two pointers are set to allow the arbitrator to suspend until both threads have completed, rather than wasting simulation resources polling to see if they are done.

While support exists in the `comm_medium` object for nonblocking reads, the channels do not have methods to give access to that functionality to the processor models. This was done on purpose to avoid having to check data dependencies before beginning a computation. This also keeps the simulation simple, and more efficient in terms of simulation time. The functionality was built into the `comm_medium` object because it was easy to do and makes adding nonblocking reads at some future point much easier. The arbitration scheme provided is a longest waiting first scheme. As soon as at least one transaction pair, a matching send and receive, is present the pair with the largest sum of positions in the wait queues is selected to transact next. The crossbar channel uses a variant of the `comm_medium` class. In the crossbar variant there are pointers to the transaction wait queues which allow a single set of queues to be used by all of the logical channels.

The `comm_medium` class also makes use of the class `transaction_wait_queue`, which is a specialized linked list to allow for a large number of waiting transactions without allocating a fixed large amount of memory. The elements of the linked list are of the class transaction_element, which contains all the essential information about a transaction request. The only item of importance from the linked list is the class that actually holds the transaction information. This `transaction_element` class contains all of the information about the transaction request. Figure 12.44 graphically depicts the key elements of the `transaction_element` class.

The integers in the bottom center contain the source processor number, the destination processor number, the destination task's id number, and the size, in nanoseconds, associated with the transaction. The Boolean variables in the top right tell whether the transaction element is a write or read, and whether it is blocking or not. The handshakes object in the upper middle is a set of four events that are used by the transact method to logically ''perform'' the transaction. The complement pointer below the handshakes is a pointer to a `transaction_element` object. To communicate two `transaction_element` objects must be paired up the channel's arbitrator process. It does this pairing by setting a read and a write's compliment pointer to each other. The `activate_me` event in the top left is used by the arbitrator to activate the thread executing the element's side of a transaction.

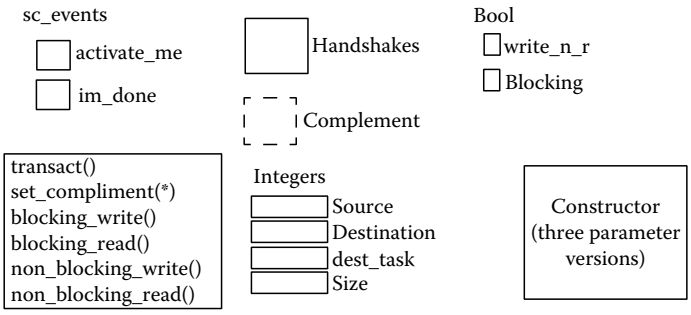


FIGURE 12.44 Graphical representation of ''transaction_element'' class.

While the `im_done` event is used to notify the arbitrator, and in the case of a blocking transaction the requesting processor's thread, that the transaction is complete. The constructor for this object depicted in the bottom left has three different implementations with different parameter lists, the first constructor implementation that sets all of the transaction values is the one used in the current version of the models. The others were left to maintain backward compatibility with previous versions, and may be useful for future versions. The purpose of the methods displayed in the bottom left of the figure are self-explanatory. The `transact` method is what controls the actual behavior of a transaction pair once it has been scheduled. The nondebugging parts are repeated below in Figure 12.45. The blocking and nonblocking versions of the read and write routines are the same in this version of the models.

The nondebugging versions of the blocking read and write methods are shown in Figure 12.46. These methods show how the four-way handshaking is implemented. The use of a full four-way handshaking in the channel model is somewhat arbitrary, but it makes incremental refinement of the channel easier.

```
   void transact() //method for creating thread to call (blocking read
or write)
   {
      wait(activate_me); //wait for arbitration process to activate me
      if (blocking)
      {
         if (write_n_r) //then it's a blocking write
            blocking_write(); //call the blocking write routine
         else //it's a blocking read
            blocking_read();  //call the blocking read routine
      }
      else
      {
         if (write_n_r) //then it's a non-blocking write
            non_blocking_write(); //call the non-blocking write routine
         else //it's a non-blocking read
            non_blocking_read();  //call the non-blocking read routine
      };
      im_done.notify(SC_ZERO_TIME);
      return;
   };
```

FIGURE 12.45 Transaction_element transact() method.

```
   void blocking_write()
   {
      compliment->handshakes.start_write.notify(SC_ZERO_TIME);
      wait(handshakes.start_ack);
      compliment->handshakes.write_done.notify(SC_ZERO_TIME);
      wait(handshakes.done_ack);
   };

   void blocking_read()
   {
      wait(handshakes.start_write);

compliment->handshakes.start_ack.notify(size,SC_NS);//SC_ZERO_TIME);
      wait(handshakes.write_done);
      compliment->handshakes.done_ack.notify(SC_ZERO_TIME);
   };
```

FIGURE 12.46 Transaction_element write and read methods.

```
    void blocking_write()
    {
       wait(handshakes.start_ack);
    };

    void blocking_read()
    {

  compliment->handshakes.start_ack.notify(size,SC_NS);//SC_ZERO_TIME);
       wait (size,SC_NS)
    };
```

FIGURE 12.47 Alternative write and read methods.

However, with the abstract behavior described here a single line for the write method and two for the read method would be sufficient. Figure 12.47 shows how the methods could be implemented in this way.

All of the channel models also read in parameter information from the *channel_param.txt* file located in the directory that the simulation is running in. This file contains two lines. The first line is the bus speed in megabytes per second. The second line is the fixed communication overhead per communication transaction. In the top-level channel models the data size parameter passed to read interface method is run through a `data_to_delay` function that return the delay in nanoseconds that the communication should take based on the specified bandwidth and communication overhead.

12.5.5 Shared Bus Model

The shared bus architecture consists of a single logical communication medium, which is an object of class `comm_medium`. The behavior is encapsulated in the `shared_bus_io` class. This class inherits the virtual interface functions from the `my_basic_rw_port` class, and must provide an implementation for them. The implementation for these functions is shown in Figure 12.48. The

```
   inline bool read(int dest_address, int source_address, int data_size, int
dest_task)
   //blocking read function with writer's address
      //read function takes data to be size of data transmission
   {
      int temp_size;
      temp_size = data_to_delay(data_size);
      shared_bus.read(dest_address, source_address, temp_size, dest_task);
      return 1;
   };
   inline bool write(const int dest_address, int source, int data,
                     int dest_task)
      //blocking write function here
   {
      shared_bus.write(dest_address, source, data, dest_task);
      return 1;
   };
   inline bool non_blk_write(const int dest_address, int source, int data,
                             int dest_task)
   //non-blocking write function here
   {
      shared_bus.non_blocking_write(dest_address,source,data,dest_task);
      return 1;
   };
```

FIGURE 12.48 Implementation of inherited virtual interface functions.

functions essentially map the interface functions to the methods of the `comm_medium` object. The `data_to_delay` function takes the `data_size` passed to the read method, and calculates the required transaction time in nanoseconds based on the bandwidth, and then adds the fixed communication overhead that the channel's constructor reads in from the channel parameter file.

12.5.6 Fully Connected Model

The fully connected architecture is built around the same `comm_medium` object as the Shared Bus model. The fully connected architecture creates a `comm_array` object which contains all the logical connections, and copies the addresses of the `comm_medium` objects into a two-dimensional array that it uses to map a processor's request to the overall communication architecture to the appropriate logical connection. The fully connected architecture passes the number of processors to the `comm_array` object which then instantiates the number of `comm_medium` objects needed to provide a dedicated shared bus between each pair of processors in the simulation.

12.5.7 Crossbar Model

The crossbar architecture is similar to the fully connected architecture except that it only requires four `cross_comm_medium` objects since with nine processors only four concurrent connections are allowed. As mentioned earlier, it uses its own version of the basic `comm_medium` object. This is necessary because the logical connections in the crossbar are not associated with any particular processor, and the communication requests are not associated with any of the logical connections. Just to clarify in the fully connected architecture there is a logical connection between every processor modeled by a `comm_medium` object. The fully connected architecture model simple directed the requests it received to the correct logical connection. In the fully connected architecture there is only one connection that all requests are intended for, but in the crossbar the number of logical connections is equal to the number of processors divided by two, and every request could potentially communicate over any of them.

12.5.8 SystemC Performance Modeling Examples

This section contains a number of examples of performance models constructed using the SystemC modeling modules described above. The examples are presented as a demonstration that the models execute correctly and also that they demonstrate performance of the system they are intended to model. The first example is a trivial example with a set of four tasks all executing on one processor. Since data communication inside a processor is assumed to take no time, the description should take simulation time equal to the sum of the computation time of all the tasks. The second example is the same four tasks allocated to two processors such that each task must send the data over the communication channel to the next task. This second example should take longer, with three 100 byte sends being sent over the communication channel. The third example is the same task graph description with varying bus parameters. The fourth and fifth examples have bus contention, to show that contention is handled properly. Each example lists the simulated latency, and a timeline showing the simulation results.

12.5.8.1 Single Processor

Figure 12.49 shows the simple sequential task graph for the first example. Here all the tasks are allocated to Processor 0. Each task has a compute value of 10 μs, and each edge has a data value of 100 bytes. Since communication within a processor is assumed to take no time, the latency for this description should be the sum of the compute times, which is 40 μs.

In addition to the processing timeline shown previously, the models generate a text output stream that describes the actions each module is taking at a given simulation time. The text output for this simulation

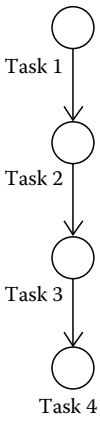


FIGURE 12.49 Single processor example task graph.

is shown below. Note that the final task is completed at 40 μs of simulation time which is exactly as expected.

```
shared bus architecture
0 s proc#0 Task number 1 Computing for 10000 ns
0 μs proc#0 Task number 2 Computing for 10000 ns
20 μs proc#0 Task number 3 Computing for 10000 ns
30 μs proc#0 Task number 4 Computing for 10000 ns
40 μs proc#0 done!
sim done!?
```

12.5.8.2 Dual Processor

The task graph for the second example is shown in Figure 12.50. The graph also shows the allocation of the tasks to two processors. Notice that the sequential tasks are on different processors so the data must be transferred across the communication channels before the computations can begin. Here the communication channel's bandwidth determines how long a communication transaction should take to complete. The length of time is the data size in bytes divided by the bandwidth in megabytes per second. The communication channel can also take into account communication overhead, in nanoseconds, if it is specified. The channel bandwidth and communication overhead are read in from a file. If this file is not present or an item is missing it will take on its default value. The value specified for this example is 100 Mbyte/s for bandwidth and 5 ns for channel overhead.

Since all the communication is of the same size, the expected latency for this example can be determined using the following equation:

$$\text{Computation time} + \text{numtrans} * \left(\frac{\text{Data size (byte)}}{\text{Bandwidth (Mbyte/s)}} + \text{Communication overhead (ns)} \right)$$

which, for this example, evaluates as follows:

$$40\ \mu\text{s} + 3 * \left(\frac{100\ \text{byte}}{10\ \text{Mbyte/s}} + 5\ \text{ns} \right) = 40\ \mu\text{s} + \frac{300}{100 * 10^6}\ \text{s} + 15\ \text{ns} = 40\ \mu\text{s} + \frac{300}{100}\ \mu\text{s} + 15\ \text{ns}$$

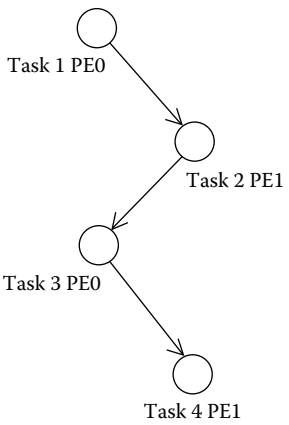


FIGURE 12.50 Dual processor example task graph.

Thus the expected latency is 40 μs for computation plus 3 μs for the actual data transmission, plus 15 ns for communication overhead. That gives a total latency of 43,015 ns. The timeline output for this simulation is shown in Figure 12.51. Note that the final task, Task 4, completes at 43 μs on the graph.

12.5.8.3 Parallel Communications Example

The next example shows the effect of various communications topologies on an application with requirements for simultaneous communications. The task graph for this example is shown in Figure 12.52. Each task (called nodes in this graph) computes for a fixed period of time and then sends data to a second task causing it to begin execution. The tasks are allocated to processors such that after completion of the first set of tasks, all processors attempt to send data to another processor. Because the first tasks all have the same execution time, all the communications become ready to begin at the same time. Thus, if an architecture has parallel

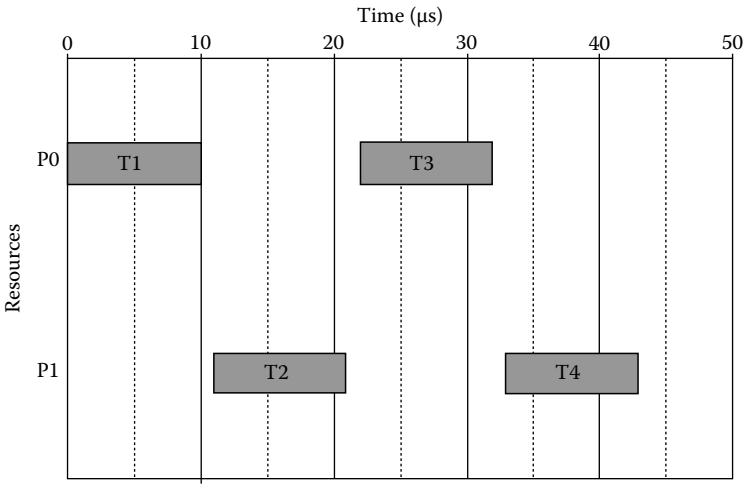


FIGURE 12.51 Dual processor example timeline.

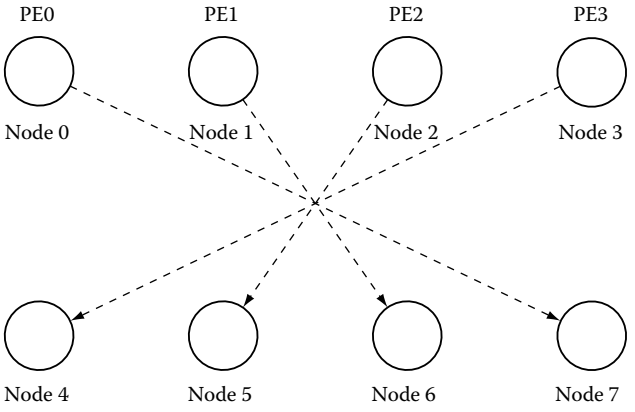


FIGURE 12.52 Bus contention example task graph.

communication paths, this will result in a decrease in the total application run time. The four start tasks (nodes) in this example all compute for 10 μs, then attempt to do a nonblocking send of size 100 byte to a task allocated to another processor. They then move on to start the read required to begin their next task.

For all of the results discussed below, the channel parameters are set to a bus bandwidth of 1 Mbyte/s and a communication overhead of 0 ns.

12.5.8.3.1 Shared Bus Simulation Results

The first set of results is for a system with a single shared bus. On this system Tasks 0–3 all execute in parallel on the four processors. At this point there will be four-way contention for the single system bus. The communications operations will be assigned priority on a first-come-first-serve basis. In the current implementation of SystemC, the task that will get first priority to communicate its data cannot be determined ahead of time, however the tasks will all run in the same order every time the simulation is run.

With a shared bus architecture, the latency is 100 μs for all processors to compute in parallel plus 4*(data size/bandwidth)*1000 (ns), or 400 μs, for each of the four sends to occur in series, plus 100 μs for the last receiver to compute after completing their receive. Thus the overall latency should be 600 μs.

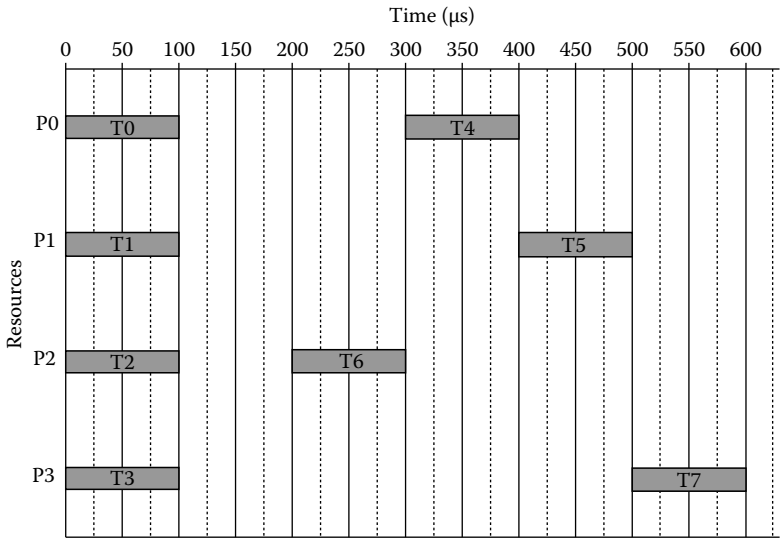


FIGURE 12.53 Shared bus contention example timeline.

The timeline for this example is shown in Figure 12.53. The timeline shows all the tasks beginning their blocking read then having to wait for the arbitrator to select them to communicate across the bus. Once the individual communications have taken place, the destination task, Tasks 4–7, execute. The timeline correctly shows the last task completing execution at 600 μs.

12.5.8.3.2 Fully Connected Simulation Results

In the fully connected architecture there is a dedicated communications channel between each pair of processors. However, in this architecture, it was decided to model a system where a processor cannot both send and receive a message from the same processor at the same time. Because of the connectivity of the task graph for this application, after the first set of tasks execute in parallel, each processor needs to send and receive a message before it can execute the next task. For example, Processor 0 cannot send to Processor 3 and receive from Processor 3 at the same time. Rather it must do one, then the other. Thus, for this example, each channel in the fully connected architecture is effectively a half duplex connection.

During execution the run time is 100 μs for all processors to compute the first four tasks in parallel plus 100 μs for the first set of sends, plus 100 μs for the second set of sends—during which the tasks started by the first set of sends also execute, then finally 100 μs for the last two tasks to compute in parallel. Thus the overall latency for the fully connected architecture should be 400 μs. The timeline for this example is shown in Figure 12.54.

12.5.8.3.3 Crossbar Simulation Results

As mentioned above, the crossbar architecture behaves like a fully connected architecture where the maximum number of connections is limited to the number of processors divided by two. Thus for this four processor example, the crossbar architecture will only allow two communications at a time. This characteristic means that for this example, the crossbar architecture will have the same latency as the fully connected architecture for this example. This result is shown in Figure 12.55.

12.5.8.4 Second Contention Example

This second example expands on the previous example by showing a slightly different set of contention conditions. In the first example, the communication requirements specified by the task graph required

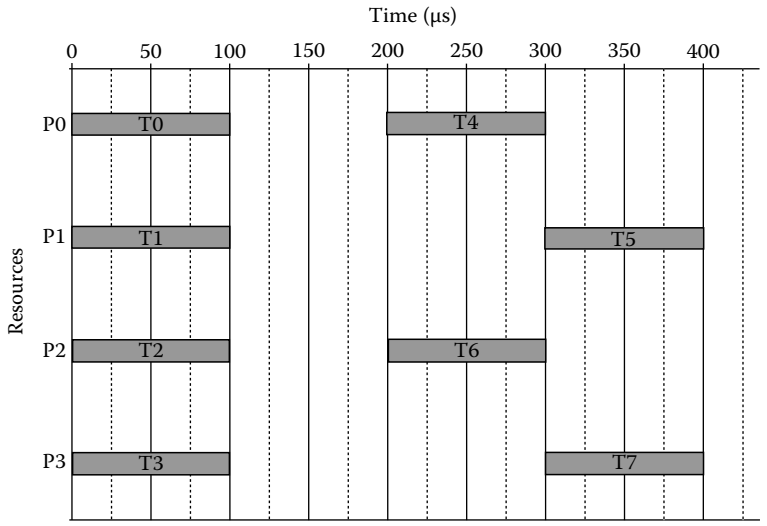


FIGURE 12.54 Fully connected contention example timeline.

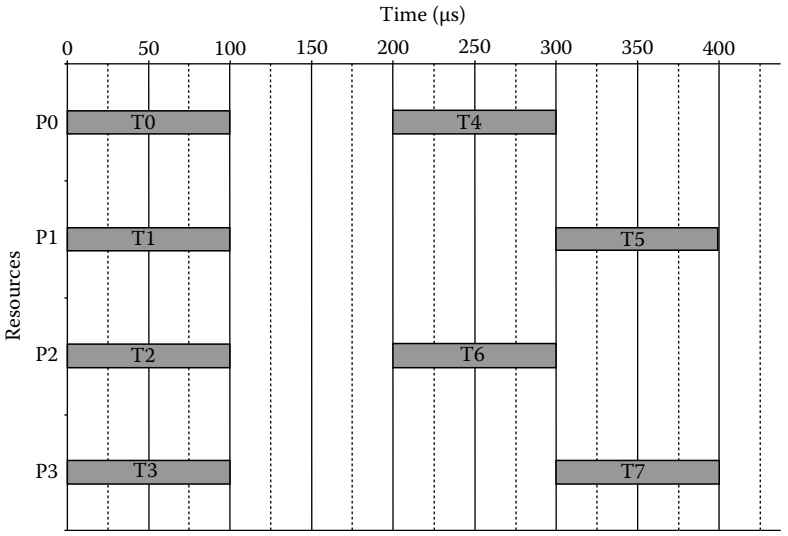


FIGURE 12.55 Crossbar contention example timeline.

the processors to send and receive data from the same processor. This effectively allowed for only two active communication transactions on the fully connected architecture. In this example, as shown in Figure 12.56, the processors will be sending and receiving data from different processors during the communications portion of the application. This set of communication requirements will allow all of the available communication channels to be used concurrently with the fully connected architecture.

Figure 12.57 shows the results for this example for the shared bus, fully connected, and crossbar architectures. Note that in this example, in the fully connected architecture, all of the communication operations occur in parallel which allows the entire application to execute in 300 μs.

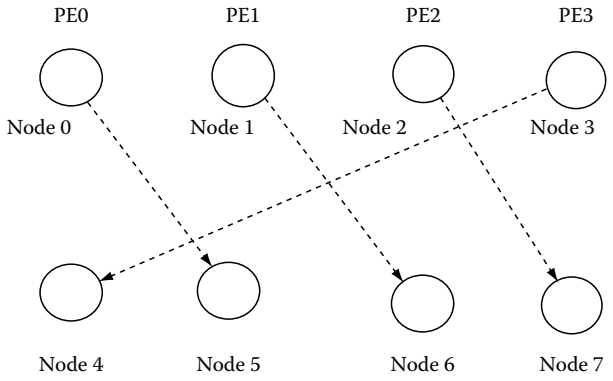


FIGURE 12.56 Second contention example task graph.

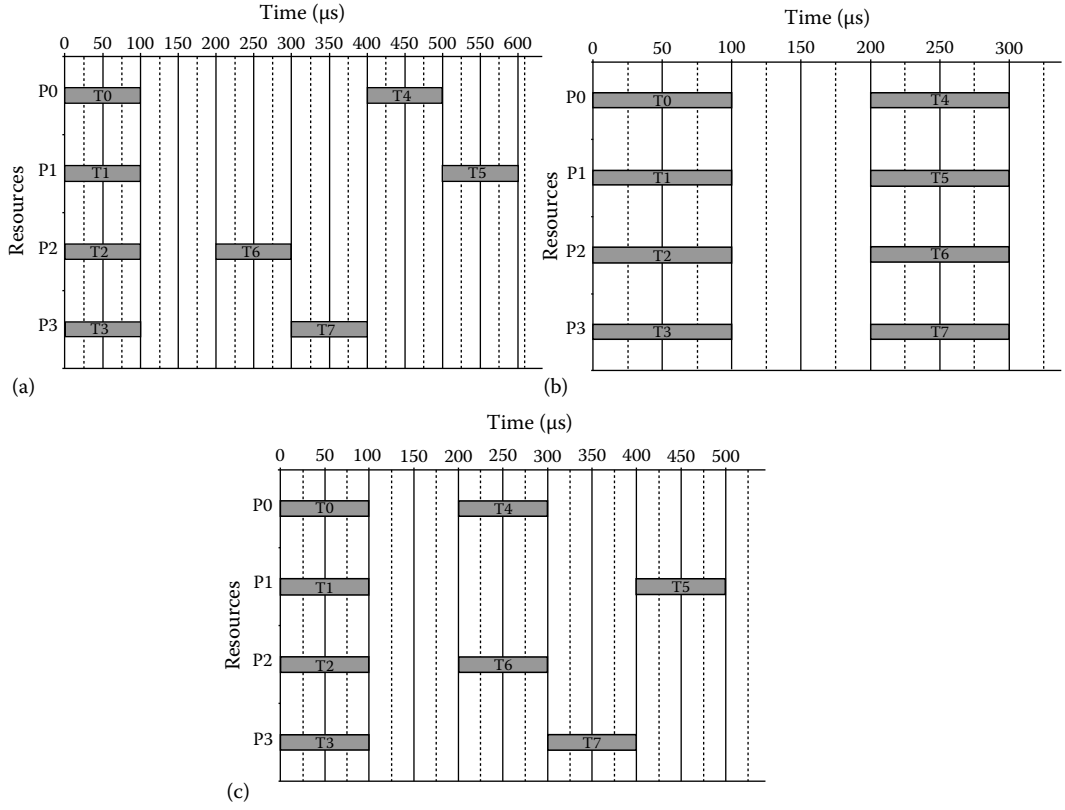


FIGURE 12.57 Second contention example timelines: (a) shared bus, (b) fully connected, and (c) crossbar.

12.5.9 Mixed-Level Processor Model

Although simplistic in nature, the above examples show that the SystemC-based performance modeling methodology can be used to model the execution of different applications on various system architectures. In addition to this capability, as mentioned earlier, the SystemC module for the processor has the

ability to replace the performance only computation delay with a refined computation model that is described in VHDL or Verilog. This mixed-level modeling capability allows the model to be refined to a lower level in a step-wise fashion.

The replacement of an abstract processor model with a refined (RTL or gate-level model) is accomplished by using the ModelSim `SC_FORIEGN_MODULE` syntax. The `SC_FORIEGN_MODULE` provided by ModelSim allows a non-SystemC model that has been compiled for ModelSim to be instantiated by a SystemC model. Incidentally, it also allows an already compiled SystemC module to be loaded in the same manner.

The processor model opens its processor command file and looks at the first line during the execution of its constructor. If the first line is "mixed" then the processor model knows that it should run as a mixed-level model with a refined computation model. If the model is to be a mixed-level one, the constructor of the processor model will then look for a *mixed_processor.txt* text file that specifies the ModelSim path for the refined computation model to use. The relevant part of the constructor that instantiates the refined model is shown in Figure 12.58.

The `refined_computation` object is the one that opens the *mixed_processorX.txt* file, where *X* is the processor number. Once this file is opened, the constructor uses the path contained within it to open the refined model object. In this example, it instantiates an object of class `rng_comp_tb` that is shown in Figure 12.59.

This class is essentially a wrapper for the actual precompiled model VHDL or Verilog model. This class/module definition would map to a VHDL entity definition like the one shown in Figure 12.60. Note that this entity corresponds to what is effectively a test bench for the refined model.

```
        cmd_input = new command_in(proc_num);
        string_temp.assign(cmd_input->get_proc_type());
        if(/*1st_line*/string_temp=="mixed")
        {
           SC_THREAD(mixed_behav);
           //only make the relevant behavior a thread!
           hardware_compute=new
  refined_computation("name",proc_num);
           //may need to add an argument this class...
        }
```

FIGURE 12.58 Refined computation part of processor model constructor.

```
class rng_comp_tb : public sc_foreign_module
{
public:
    sc_in<sc_logic> start;
    sc_out<sc_logic> done;

    rng_comp_tb(sc_module_name nm, const char* hdl_name)
     : sc_foreign_module(nm, hdl_name),
       start("start"),
       done("done")
    {
      //cout<<"Building hardware model\n";
   };

     ~rng_comp_tb()
     {}

};
```

FIGURE 12.59 Code for "rng_comp_tb" class.

```
ENTITY rng_comp_tb IS
   port(
     start : in std_logic;
     done : out std_logic
     );

END rng_comp_tb;
```

FIGURE 12.60 Sample VHDL refined computation entity declaration.

The intent for this interface is for it to be easy to integrate into an existing test bench for the refined model of computation. The start and done signals are active high. So when the start goes high to a logical ''1,'' the refined model should start a ''computation.'' This computation is effectively the test bench applying a set of predefined stimulus waveforms to the refined model. As described below, these stimulus waveforms can be derived for the specific refined model in a number of different ways depending on the objectives for the mixed-level model.

When the refined model has finished its ''computation'' it should raise the done signal. Both signals should be low at the start of the simulation. When the processor model puts the start signal high, the refined model begins its computation. The presumption is that the refined model is something like a test bench, with a model of the actual hardware, or some other more detailed model, instantiated inside of it. The refined model presents any required data to the detailed model, and watches for whatever condition indicates that it has completed the computation. Once the computation is completed, it raises the done signal telling the abstract processor model it is done. The processor model will then lower the start signal, and the refined model will respond by lowering the done signal. Figure 12.61 shows the basic timing diagram for the interface.

Figure 12.62 shows the portion of the processor model that raises and lowers the start signal and waits for the done signal. This code is located in the `refined_computation` object. The `refined_computation` object is instantiated by the processor model's constructor when it reads in from the command file and determines that it should be a mixed-level model.

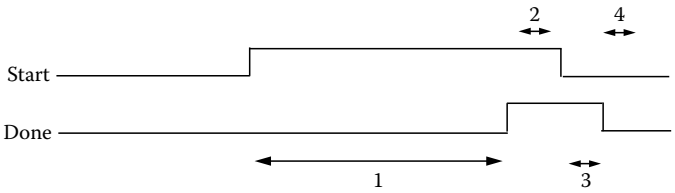


FIGURE 12.61 Refined computation model interface-timing diagram.

```
   bool start_compute()
   {/* basic algorithm is:
      send start signal
      wait for done signal
      return
      */
#ifdef MTI_SYSTEMC
      temp= true; //'1';
      start = temp;
      while (done != temp)  //ie while done != 1
         wait(done.value_changed_event());//wait until event and
loop..
      temp = false;
      start = temp;
#endif
      return 0;
   };
```

FIGURE 12.62 SystemC start/done interface code.

```
-------------------------------------------------------------------
-- Process to respond to start and create go and done. JA
-------------------------------------------------------------------
   CompBench: PROCESS (start, tb_done)
   Begin
      if (start='1') then
         if ((tb_done'event) AND (tb_done='1')) then
            go   <= '0';
            done <= '1';
         ELSE
            go   <= '1';
            done <= '0';
         end if;
      end if;
   End process CompBench;
```

FIGURE 12.63 Sample VHDL start/done interface code.

Figure 12.63 shows a portion of sample VHDL test bench that implements the refined computation model's side of the start/done interface. The start signal from the start/done interface activates the process. The process uses the go signal to cause the test bench to perform a computation, and the test bench raises the `tb_done` signal when it is finished with a single computation.

12.5.10 Mixed-Level Examples

The following examples are mixed-level examples where the computation part of the processor is modeled in more detail. Since the rest of the simulation is at a more abstract level and does not have all of the stimuli needed for the refined model, the stimuli need to be created in some way. The following examples focus mainly on different ways of generating the data that the various refined models need to function.

12.5.10.1 Fixed Cycle Length

The first example is one where an abstract processor is replaced with a random number generator. The model for the random number generator is an RTL discrete digital model of a random number generator described in Ref. [51]. The model attempts to describe a number of elements that are extremely sensitive to initial conditions, and thus in reality exhibit more random behavior than can be modeled with a solely digital model. As it is, the model always generates the same nonrepeating sequence of values. For this example, the existing test bench was modified to put the generator through its reset cycle, then through a single random number generation. The only inputs to the refined model are a set of control signals and a clock. When the processor, that it is the refined computation model for, gets a compute command; it will send the start signal to the test bench, which will then go through the reset and generate phases, and signal back with the done signal once a number has been generated. In this particular example the internal signals continue to oscillate between compute commands. Since this model takes a fixed number of cycles using the refined model, and the values generated are not passed elsewhere, this example is less efficient and no more accurate than putting in the actual compute time for the abstract compute command.

12.5.10.2 Variable Cycle Length

The rest of the mixed-level examples presented use an RTL booth multiplier model. The booth multiplier takes a variable number of cycles to complete the binary multiplication. The number of cycles required depends on the numbers being multiplied. The inputs to the model are the two numbers to be multiplied, and a clock, the outputs are the result and a control signal indicating that the multiplication is complete.

There is a slight propagation delay for the result to appear on the output after the done signal appears. If the input clock continues to cycle after the multiplication is complete, then the result will become invalid after a clock cycle. Thus the test bench allows a half cycle to elapse before considering the computation done. Note that the effect of this refined model is that the computational delay is dependent on the data being applied to it for the computation. This delay mechanism is a more accurate representation of how a refined model would be used in, and add additional accuracy to, a system-level performance model. However, as discussed in the section above on VHDL-based mixed-level modeling, the important question is how to generate the data that is input to the refined model in such a way as to accurately represent the performance of the refined component in the real system. Typically, this can be done by either presenting the refined model with random data to develop a statistical representation of average system performance, or presenting the refined model with predefined data. This data can be generated by the designer to represent typical system performance, or to exercise the best-, or worst-case delay scenarios.

12.5.11 Random Data Generation

In this example, the booth multiplier is presented with two random numbers generated by two other entities instantiated in the test bench. Since random number generation is not present in standard VHDL library, ModelSim's mixed language ability is utilized to allow a SystemC random number generator module to be used. The SystemC module uses the C++ `rand()` function to generate a pseudo random number, which is passed back to the VHDL test bench. Since the VHDL test bench is instantiated by the SystemC performance simulation, this is in fact a SystemC–VHDL–SystemC hierarchy. ModelSim's mixed language interface allows the designer to use whatever language is best suited for the task at hand. Here the random number must be passed back to the VHDL test bench as an `sc_logic` vector, which is automatically translated into a VHDL `std_logic_vector` by the simulator. Since both numbers are pseudorandom, the computer will take a variable amount of time to complete. Although it is possible that the use of the `rand()` function will result in repeating sequence of numbers, most simulations will not run long enough for this to be noticeable. If a more random distribution, or a particular type of distribution is desired, a specialized random number generator package can be used in the model. Since most of the computations at the system level, will consist of multiple operations, to make this example more typical, a means of generating a set multiple numbers to be multiplied was needed. While it is possible to generate a fixed size data set this example goes just a little farther and generates a pseudorandom size data set of pseudorandom numbers. The generation of the random size of the data set is done using the same SystemC module that generates the random data itself. A few changes to the test bench needed to be made so that it did not send the done signal back until all of the multiplications were completed. Figure 12.64 shows two waveforms from two different mixed-level simulations, each with two different size sets of random numbers multiplied together.

12.5.11.1 Data Set from File

As described above, when a sample set of data that exercises a specific scenario for the refined model is available, it can be advantageous to use it rather than generating a new set that may or may not be close to the actual data. Using this predefined sample data set ensures accurate performance for that data set, and removes any guesswork as to what a realistic data set might be. Since VHDL has standard file access capabilities that are easy to use, the test bench for this example reads the values directly from the input file, and does not send the done signal until it reaches the end of file. Unfortunately, the current implementation ModelSim used for this example did not allow passing generic information across the language boundary, thus specification of the file to use had to be done in the VHDL code. An extension to this approach would be to read the filename to use from a configuration file similar to what is done to specify the refined model to use in the SystemC performance code. Figure 12.65 shows the waveform of the mixed-level model simulation where the data for the refined component were read in from a file.

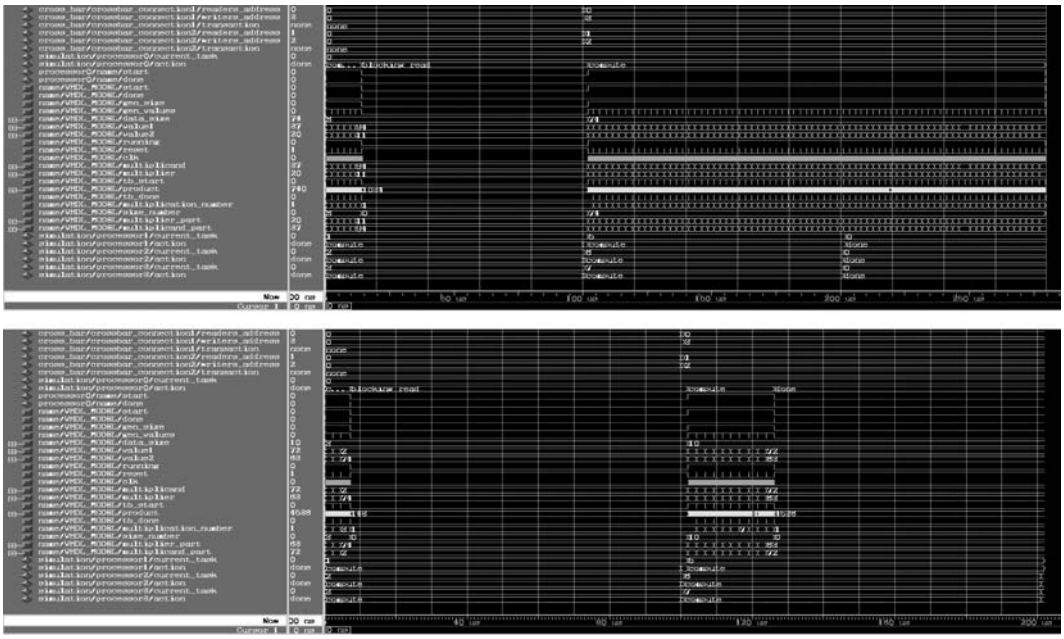


FIGURE 12.64 First- and second-random data set waveforms.

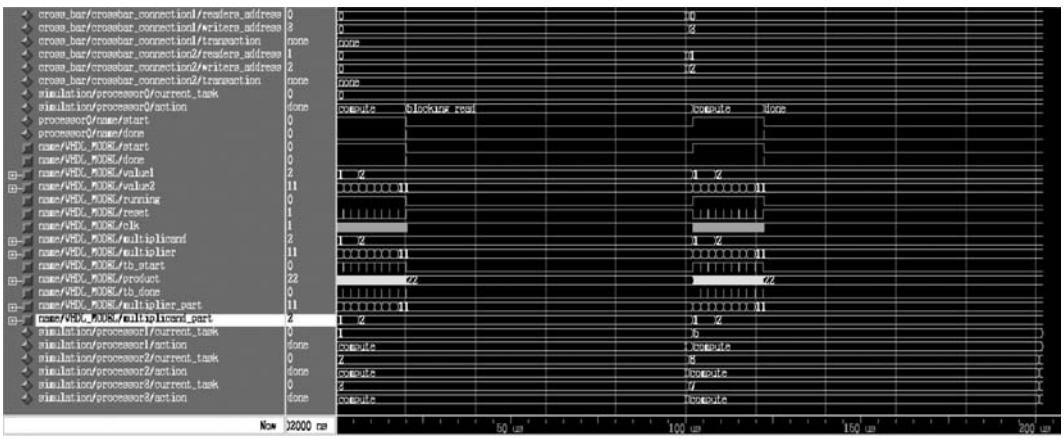


FIGURE 12.65 Data from file example waveform.

12.5.12 Mixed-Level Example Summary

Mixed-level modeling techniques in SystemC-based performance models have been demonstrated. These examples utilized simple pseudorandom data generation, variable size pseudorandom data set generation, reading a data set from single file, and reading multiple data sets from multiple files. Multiple variations and combinations of the above approaches can be used to generate stimuli for a wide variety of refined computation models. Anything from randomly selecting a file from which to read a data set, to using a file to parameterize random data generation is possible depending on whether realistic data are available and on the simulation objective. It is clear to see, in any case, that including refined behavioral

RTL or gate-level components in a performance model in SystemC is fairly easily done and can increase the overall accuracy of the performance model, just as was possible in the VHDL-based performance model.

12.6 Conclusions

Integration of performance modeling into the design process such that it can actually drive the refinement of the design into an implementation has clear advantages in terms of design time and quality. The capability to cosimulate detailed behavioral models and abstract system-level models is vital to the development of a design environment that fully integrates performance modeling into the design process. One methodology and implementation for cosimulating behavioral models of individual components with an abstract performance model of the entire system was presented. This environment results in models that can provide estimates of the performance bounds of a system that converge as the refinement of the overall model increases.

This chapter has only scratched the surface on the possible improvements that performance- or system-level modeling can have on the rapid design of complex VLSI systems. As more and more functionality can be incorporated into the embedded VLSI systems and these systems find their way into safety-critical applications, measures of dependability such as reliability and safety at the system-level are becoming of great interest. Tools and techniques are being developed that can use the performance model from which to derive the desired dependability measures. In addition, behavioral fault simulation and testability analysis are finding their way into the early phases of the design process. In summary, the more attributes of the final implementation that can be determined from the early and often incomplete model, the better the resulting design and shorter the design cycle.

References

1. ASIC and EDA Magazine, January 1993, Reed Business Information, a division of Reed Elsevier Inc., New York, 1993.
2. D. Hill and W. vanCleemput, SABLE: A tool for generating structured, multi-level simulations, *Proceedings of the 16th ACM/IEEE Design Automation Conference*, San Diego, CA, June 25–27, 1979, pp. 272–279.
3. D.W. Franke and M.K. Purvis, Hardware/software codesign: A perspective, *Proceedings of the 13th International Conference on Software Engineering*, Austin, TX, May 13–17, 1991, pp. 344–352.
4. D.W. Franke and M.K. Purvis, Design automation technology for codesign: Status and directions, *International Symposium on Circuits* and *Systems*, San Diego, CA, 1992, pp. 2669–2671.
5. G.A. Frank, D.L. Franke, and W.F. Ingogly, An architecture design and assessment system, *VLSI Design*, 6(8): 30–50, August 1985.
6. C. Terry, Concurrent hardware and software design benefits embedded systems, *EDN*, 148–154, July 1990.
7. S. Napper, Embedded-system design plays catch-up, *IEEE Computer*, 31(8): 118–120, August 1998.
8. F.W. Zurcher and B. Randell, Iterative multi-level modeling—A methodology for computer system design, *Proceedings of IFIP Congress '68*, Edinburgh, UK, August 5–10, 1968, pp. 867–871.
9. Martin Marietta Laboratories, RASSP First Annual Interim Technical Report (CDRL A002), Moorestown, NJ, October 31, 1994.
10. Open SystemC Initiative Website, www.systemC.org.
11. P.A. Wilsey and S. Dasgupta, A formal model of computer architectures for digital system design environments, *IEEE Transaction on Computer-Aided Design*, 9(5): 473–486, May 1990.
12. C.A. Giumale and H.J. Kahn, Information models of VHDL, *Proceedings of the 32nd Design Automation Conference*, San Francisco, CA, 1995, pp. 678–683.

13. P. Kollaritsch, S. Lusky, D. Matzke, D. Smith, and P. Stanford, A unified design representation can work, *Proceedings of the 26th Design Automation Conference*, ACM Press, Las Vegas, NV, June 25–29, 1989, pp. 811–813.
14. J. Peterson, Petri nets, *Computing Surveys*, 9(3): 223–252, September 1997.
15. M.K. Molloy, Performance analysis using stochastic petri nets, *IEEE Transactions on Computers*, C-31(9): 913–917, September 1982.
16. M.A. Holliday and M.K. Vernon, A generalized timed petri net for performance analysis, *IEEE Transactions on Software Engineering*, SE-13(12): 1297–1310, December 1987.
17. L. Kleinrock, *Queuing Systems*, Vol. 1: *Theory*, Wiley, New York, 1975.
18. G.S. Graham, Queuing network models of computer system performance, *Computing Surveys*, 10(3): 219–224, September 1978.
19. G. Balbo, S.C. Bruell, and S. Ghanta, Combining queuing networks and generalized stochastic petri nets for solutions of complex models of system behavior, *IEEE Transactions on Computers*, 37(10): 1251–1268, October 1988.
20. G. Frank, Software/hardware codesign of real-time systems with ADAS, *Electronic Engineering*, 95–102, March 1990.
21. *SES/Workbench User's Manual*, Release 2.0, Scientific and Engineering Software Inc., Austin, TX, January 1991.
22. L. Maliniak, ESDA boosts productivity for high-level design, *Electronic Design*, 41: 125–128, May 27, 1993.
23. Integrated Design Automation System (IDAS), JRS Research Laboratories Inc., Orange, CA, 1988.
24. Application Note for TRANSCEND/VANTAGE Optium Cosimulation, TD Technologies, pp. 1–33, 1993.
25. Application Note for TRANSCEND Structural Ethernet Simulation, TD Technologies, August 1993, pp. 1–13.
26. R. Bargodia and C. Shen, MIDAS: Integrated design and simulation of distributed systems, *IEEE Transactions on Software Engineering*, 17(10): 1042–1058, October 1991.
27. E. Lee et al., Mini Almagest, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, April 1994.
28. E.A. Lee and D.G. Messerschmitt, Synchronous data flow, *Proceedings of the IEEE*, 75(9): 1235–1245, September 1987.
29. VHDL Hybrid Models Requirements, Honeywell Technology Center, Version 1.0, December 27, 1994.
30. Honeywell Technology Center, *VHDL Performance Modeling Interoperability Guideline*, Version Draft, August 1994.
31. F. Rose, T. Steeves, and T. Carpenter, VHDL performance models, *Proceedings of the 1st Annual RASSP Conference*, Arlington, VA, August 1994, pp. 60–70.
32. J.A. Rowsen and A. Sangiovanni-Vincentelli, Interface-based design, *Proceedings of the 34th ACM/IEEE Design Automation Conference (DAC-97)*, ACM Press, Anaheim Convention Center, Anaheim, CA, June 9–13, 1997, pp. 178–183.
33. L. Cai and D. Gajski, Transaction level modeling: An overview, *Proceedings of CODES + ISSS'03*, 2003, pp. 19–24.
34. IEEE, *IEEE Standard VHDL Language Reference Manual*, IEEE Std. 1076–1993, New York, NY, June 6, 1994.
35. A.P. Voss, R.H. Klenke, and J.H. Aylor, The analysis of modeling styles for system level VHDL simulations, *Proceedings of the VHDL International Users Forum*, IEEE Computer Society, Washington, DC, Fall 1995, pp. 1.7–1.13.
36. J.H. Aylor, R. Waxman, B.W. Johnson, and R.D. Williams, The integration of performance and functional modeling in VHDL, in *Performance and Fault Modeling with VHDL*, J.M. Schoen (Ed.), Prentice-Hall, Englewood Cliffs, NJ, 1992, pp. 22–145.

37. K. Jensen, Colored petri nets: A high level language for system design and analysis, in *High-Level Petri Nets: Theory and Application*, K. Jensen and G. Rozenberg (Eds.), Springer, Berlin, Germany, 1991, pp. 44–119.
38. F.T. Hady, A methodology for the uninterpreted modeling of digital systems in VHDL, MSc thesis, Department of Electrical Engineering, University of Virginia, Charlottesville, VA, January 1989.
39. A.P. Voss, Analysis and enhancements of the ADEPT environment, MSc thesis, Department of Electrical Engineering, University of Virginia, May 1996.
40. J.B. Dennis, Modular, asynchronous control structures for a high performance processor, *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation, SECTION: Asynchronous Modular Systems*, Association for Computing Machinery, ACM Press, New York, 1970, pp. 55–80.
41. G. Swaminathan, R. Rao, J. Aylor, and B. Johnson, Colored petri net descriptions for the UVa primitive modules, CSIS Technical Report No. 920922.0, University of Virginia, Charlottesville, VA, September 1992.
42. ADEPT Library Reference Manual, CSIS Technical Report No. 960625.0, University of Virginia, Charlottesville, VA, June 6, 1996.
43. M. Meyassed, R. McGraw, J. Aylor, R. Klenke, R. Williams, F. Rose, and J. Shackleton, A framework for the development of hybrid models, *Proceedings of the 2nd Annual RASSP Conference*, Arlington, VA, July 1995, pp. 147–154.
44. R.A. MacDonald, R. Williams, and J. Aylor, An approach to unified performance and functional modeling of complex systems, *IASTED Conference on Modeling and Simulation*, Pittsburg, PA, April 1995.
45. D. Gajski, N. Dutt, A. Wu, and S. Lin, *High-Level Synthesis: Introduction to Chip and System Design*, Kluwer Academic, Dordrecht, the Netherlands, 1992.
46. E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik*, 1: 269–271, 1959.
47. R. Floyd, Algorithm 97 (shortest path), *Communications of the ACM*, 5(6): 345, 1962.
48. S. Warshall, A theorem on Boolean matrices, *Journal of the ACM*, 9(1): 11–12, 1962.
49. M. Meyassed, System-level design: Hybrid modeling with sequential interpreted elements, PhD dissertation, Department of Electrical Engineering, University of Virginia, Charlottesville, VA, January 1997.
50. J.J. Hein, J.H. Aylor, and R.H. Klenke, A performance-based design tool for hardware/software systems, *Proceedings of the IEEE Workshop on Rapid System Prototyping*, San Diego, CA, June 2003.
51. S.T. Mitchum and R.H. Klenke, Design and fabrication of a digitally synthesized, digitally controlled, ring oscillator, *Proceedings of the IASTED Circuits Signals and Systems Conference*, Marina Del Rey, CA, October 24–26, 2005.

13

Embedded Computing Systems and Hardware/ Software Codesign

13.1 Introduction **13**-1
13.2 Uses of Microprocessors **13**-1
13.3 Embedded System Architectures **13**-3
13.4 Hardware/Software Codesign **13**-6
Models • Cosimulation • Performance Analysis • Hardware/Software Cosynthesis • Design Methodologies

Wayne Wolf
Princeton University

13.1 Introduction

This chapter describes embedded computing systems that make use of microprocessors to implement part of the system's function. It also describes hardware/software codesign, which is the process of designing embedded systems while simultaneously considering the design of its hardware and software elements.

13.2 Uses of Microprocessors

An embedded computing system (or more simply an embedded system) is any system which uses a programmable processor but itself is not a general-purpose computer. Thus, a personal computer is not an embedded computing system (though PCs are often used as platforms for building embedded systems), but a telephone or automobile which includes a CPU is an embedded system. Embedded systems may offer some amount of user programmability—3Com's PalmPilot, for example, allows users to write and download programs even though it is not a general-purpose computer—but embedded systems generally run limited sets of programs. The fact that we know the software that we will run on the hardware allows us to optimize both the software and hardware in ways that are not possible in general-purpose computing systems.

Microprocessors are generally categorized by their word size, since word size is associated both with maximum program size and data resolution. Commercial microprocessors come in many sizes; the term microcontroller is used to denote a microprocessor which comes with some basic on-chip peripheral devices, such as serial input/output (I/O) ports. Four-bit microcontrollers are extremely simple but capable of some basic functions. Eight-bit microcontrollers are workhorse low-end microprocessors.

Sixteen- and 32-bit microprocessors provide significantly more functionality. A 16/32-bit microprocessor may be in the same architectural family as the CPUs used in computer workstations, but microprocessors destined for embedded computing often do not provide memory management hardware. A digital signal processor (DSP) is a microprocessor tuned for signal processing applications. DSPs are often Harvard architectures, meaning that they provide separate data and program memories; Harvard architectures provide higher performance for DSP applications. DSPs may provide integer or floating-point arithmetic.

Microprocessors are used in an incredible variety of products. Furthermore, many products contain multiple microprocessors. Four- and eight-bit microprocessors are often used in appliances: for example, a thermostat may use a microcontroller to provide timed control of room temperature. Automatic cameras often use several eight-bit microprocessors, each responsible for a different aspect of the camera's functionality: exposure, shutter control, etc. High-end microprocessors are used in laser and ink-jet printers to control the rendering of the page. Many printers use two or three microprocessors to handle generation of pixels, control of the print engine, and so forth. Modern automobiles may use close to 100 microprocessors, and even inexpensive automobiles generally contain several. High-end microprocessors are used to control the engine's ignition system—automobiles use sophisticated control algorithms to simultaneously achieve low emissions, high fuel economy, and good performance. Low-end microcontrollers are used in a number of places in the automobile to increase functionality: for example, four-bit microcontrollers are often used to sense whether seat belts are fastened and turn on the seat belt light when necessary.

Microprocessors may replace analog components to provide similar functions, or they may add totally new functionality to a system. They are used in several different ways in embedded systems. One broad application category is signal conditioning, in which the microprocessor or DSP performs some filtering or control function on a digitized input. The conditioned signal may be sent to some other microprocessor for final use. Signal conditioning allows systems to use less-expensive sensors with the application of a relatively inexpensive microprocessor. Beyond signal conditioning, microprocessors may be used for more sophisticated control applications. For example, microprocessors are often used in telephone systems to control signaling functions, such as determining what action to take based on the reception of dial tones, etc. Microprocessors may implement user interfaces; this requires sensing when buttons, knobs, etc. are used, taking appropriate actions, and updating displays. Finally, microprocessors may perform data processing, such as managing the calendar in a personal digital assistant.

There are several reasons why microprocessors make good design components in such a wide variety of application areas. First, digital systems often provide more complex functionality than can be created using analog components. A good example is the user interface of a home audio/video system, which provides more information and is easier use than older, non-microprocessor-controlled systems. Microprocessors also allow related products much more cost-effectively. An entire product family, including models at various price and feature points, can be built around a single microprocessor-based platform. The platform includes both hardware components common to all the family members and software running on the microprocessor to provide functionality. Software elements can easily be turned on or off in various family members. Economies of scale often mean that it is cheaper to put the same hardware in both expensive and cheap models and to turn off features in the inexpensive models rather than to try to optimize the hardware and software configurations of each model separately. Microprocessors also allow design changes to be made much more quickly. Many changes may be possible simply by reprogramming; other features may be made possible by adding memory or other simple hardware changes along with some additional programming. Finally, microprocessors aid in concurrent engineering. After some initial design decisions have been made, hardware and software can be designed in parallel, reducing total design time.

While embedded computing systems traditionally have been fabricated at the board level out of multiple chips, embedded computing systems will play an increasing role in integrated circuit design as well. As VLSI technology moves toward the ability to fabricate chips with billions of transistors, integrated circuits will increasingly incorporate one or several microprocessors executing embedded

software. Using microprocessors as components in integrated circuits increases design productivity, since CPUs can be used as large components which implement a significant part of the system functionality. Single-chip embedded systems can provide much higher performance than board-level equivalents, since chip-to-chip delays are eliminated.

13.3 Embedded System Architectures

Although embedded computing spans a wide range of application areas, from automotive to medical, there are some common principles of design for embedded systems. The application-specific embedded software runs on a hardware platform. An example hardware platform is shown in Figure 13.1. It contains a microprocessor, memory, and I/O devices. When designing on a general-purpose system such as a PC, the hardware platform would be predetermined, but in hardware/software codesign the software and hardware can be designed together to better meet cost and performance requirements.

Depending on the application, various combinations of criteria may be important goals for the system design. Two typical criteria are speed and manufacturing cost. The speed at which computations are made often contributes to the general usability of the system, just as in general-purpose computing. However, performance is also often associated with the satisfaction of deadlines—times at which computations must be completed to ensure the proper operation of the system. If failure to meet a deadline causes a major error, it is termed a hard deadline. And missed deadlines, which result in tolerable but unsatisfactory degradations, are called soft deadlines. Hard deadlines are often (though not always) associated with safety-critical systems. Designing for deadlines is one of the most challenging tasks in embedded system design. Manufacturing cost is often an important criteria for embedded systems. Although the hardware components ultimately determine manufacturing cost, software plays an important role as well. First, the size of the program determines the amount of memory required, and memory is often a significant component of the total component cost. Furthermore, the improper design of software can cause one to require higher-performance, more-expensive hardware components than are really necessary. Efficient utilization of hardware resources requires careful software design. Power consumption is becoming an increasingly important design metric. Power is certainly important in battery-operated devices, but it can be important in wall socket-powered systems as well—lower power consumption means smaller, less-expensive power supplies and cooling and may result in environmental ratings that are advantageous in the marketplace. Once again, power consumption is ultimately determined by the hardware, but software plays a significant role in power characteristics. For example, more efficient use of on-chip caches can reduce the need for off-chip memory access, which consumes much more power than on-chip cache references.

Figure 13.1 shows the hardware architecture of a basic microprocessor system. The system includes the CPU, memory, and some I/O devices, all connected by a bus. This system may consist of multiple chips for high-end microprocessors or a single-chip microcontroller. Typical I/O devices include analog/digital (ADC) and digital/analog (DAC) converters, serial and parallel communication devices, network and bus interfaces, buttons and switches, and various types of display devices. This configuration is a complete, basic, embedded computing hardware platform on which application software can execute.

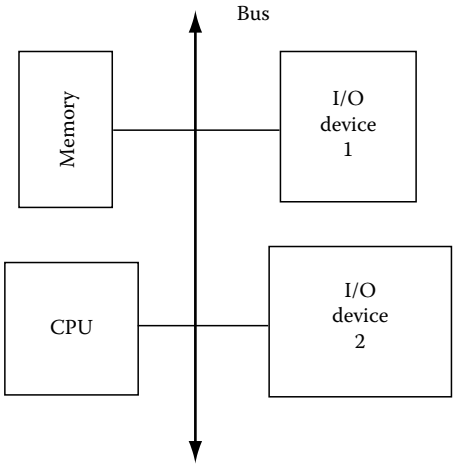


FIGURE 13.1 Hardware structure of a microprocessor system.

The embedded application software includes components for managing I/O devices and for performing the core computational tasks. The basic software techniques for communicating with I/O devices are polling and interrupt-driven. In a polled system, the program checks each device's status register to determine if it is ready to perform I/O. Polling allows the CPU to determine the order in which I/O operations are completed, which may be important for ensuring that certain device requests are satisfied at the proper rate. However, polling also means that a device may not be serviced in time if the CPU's program does not check it frequently enough. Interrupt-driven I/O allows a device to change the flow of control on the CPU and call a device driver to handle the pending I/O operation. An interrupt system may provide both prioritized interrupts to allow some devices to take precedence over others and vectored interrupts to allow devices to specify which driver should handle their request.

Device drivers, whether polled or interrupt-driven, will typically perform basic device-specific functions and hand-off data to the core routines for processing. Those routines may perform relatively simple tasks, such as transducing data from one device to another, or may perform more sophisticated algorithms such as control. Those core routines often will initiate output operations based on their computations on the input operations.

Input and output may occur either periodically or aperiodically. Sampled data is a common example of periodic I/O, while user interfaces provide a common source of aperiodic I/O events. The nature of the I/O transactions affects both the device drivers and the core computational code. Code which operates on periodic data is generally driven by a timer which initiates the code at the start of the period. Periodic operations are often characterized by their periods and the deadline for each period. Aperiodic I/O may be detected either by an interrupt or by polling the devices. Aperiodic operations may have deadlines, which are generally measured from the initiating I/O event. Periodic operations can often be thought of as being executed within an infinite loop. Aperiodic operations tend to use more event-driven code, in which various sections of the program are exercised by different aperiodic events, since there is often more than one aperiodic event which can occur.

Embedded computing systems exhibit a great deal of parallelism which can be used to speed up computation. As a result, they often use multiple microprocessors which communicate with each other to perform the required function. In addition to microprocessors, application-specific ICs (ASICs) may be added to accelerate certain critical functions. CPUs and ASICs in general are called processing elements (PEs). An example multiprocessor system built from several PEs along with I/O devices and memory is shown in Figure 13.2.

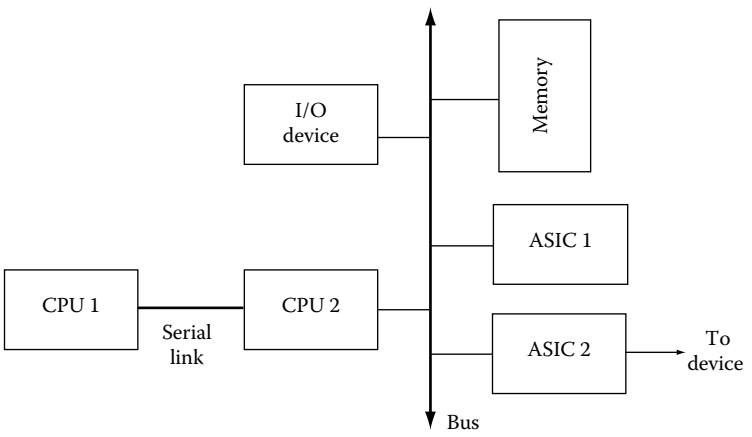


FIGURE 13.2 Heterogeneous embedded multiprocessor.

The choice of several small microprocessors or ASICs rather than one large CPU is primarily determined by cost. Microprocessor cost is a nonlinear function of performance, even within a microprocessor family. Vendors generally supply several versions of a microprocessor which run at different clock rates; chips which run at varying speeds are a natural consequence of the variations in the VLSI manufacturing process. The slowest microprocessors are significantly less expensive than the fastest ones, and the cost increment is larger at the high end of the speed range than at the low end. As a result, it is often cheaper to use several smaller microprocessors to implement a function.

When several microprocessors work together in a system, they may communicate with each other in several different ways. If slow data rates are sufficient, serial data links are commonly used for their low hardware cost. The I^2C bus is a well-known example of a serial bus used to build multi-microprocessor embedded systems; the CAN bus is widely used in automobiles. High-speed serial links can achieve moderately high performance and are often used to link multiple DSPs in high-speed signal processing systems. Parallel data links provide the highest performance thanks to their sheer data width. High-speed busses such as PCI can be used to link several processors.

The software for an embedded multiprocessing system is often built around processes. A process, as in a general-purpose computing system, is an instantiation of a program with its own state. Since problems complex enough to require multiprocessors often run sophisticated algorithms and I/O systems, dividing the system into processes helps manage design complexity. A real-time operating system (RTOS) is an operating system specifically designed for embedded, and specifically real-time applications. The RTOS manages the processes and device drivers in the system, determining when each executes on the CPU. This function is termed scheduling. The partitioning of the software between application code which executes core algorithms and an RTOS which schedules the times to which those core algorithms are executed is a fundamental design principle in computing systems in general and is especially important for real-time operation.

There are a number of techniques which can be used to schedule processes in an embedded system—that is, to determine which process runs next on a particular CPU. Most RTOSs use process priorities in some form to determine the schedule. A process may be in any one of three states: currently executing (there can obviously be only one executing process on each CPU), ready to execute, or waiting. A process may not be able to execute until, for example, its data has arrived. Once its data arrives, it moves from waiting to ready. The scheduler chooses among the ready processes to determine which process runs next. In general, the RTOS's scheduler chooses the highest-priority ready process to run next; variations between scheduling methods depend in large part on the ways in which priorities are determined. Unlike general-purpose operating systems, RTOSs generally allow a process to run until it is preempted by a higher-priority process. General-purpose operating systems often perform time-slicing operations to maintain fair access of all the users on the system, but time-slicing does not allow the control required for meeting deadlines.

A fundamental result in real-time scheduling is known as rate-monotonic scheduling. This technique schedules a set of processes which run independently on a single CPU. Each process has its own period, with the deadline happening at the end of each period. There can be arbitrary relationships between the periods of the processes. It is assumed that data does not in general arrive at the beginning of the period, so there are no assumptions about when a process goes from waiting to ready within a period. This scheduling policy uses static priorities—the priorities for the processes are assigned before execution begins and do not change. It can be shown that the optimal priority assignment is based on period—the shorter the period, the higher the priority. This priority assignment ensures that all processes will meet their deadlines on every period. It can also be shown that at most, 69% of the CPU is used by this scheduling policy. The remaining cycles are spent waiting for activities to happen—since data arrival times are not known, it is not possible to utilize 100% of the CPU cycles.

Another well-known, real-time scheduling technique is earliest deadline first (EDF). This is a dynamic priority scheme—process priorities change during execution. EDF sets priorities based on the impending deadlines, with the process whose deadline is closest in the future having the highest priority.

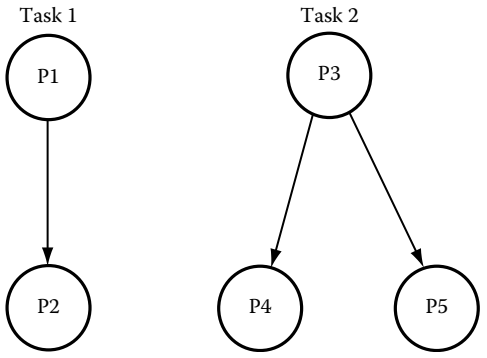


FIGURE 13.3 Task graph with two tasks and data dependencies between processes.

Clearly, the rate of change of process priorities depends on the periods and deadlines. EDF can be shown to be able to utilize 100% of the CPU, but it does not guarantee that all deadlines will be met. Since priorities are dynamic, it is not possible in general to analyze whether the system will be overloaded at some point.

Processes may be specified with data dependencies, as shown in Figure 13.3, to create a task graph. An arc in the data dependency graph specifies that one process feeds data to another. The sink process cannot become ready until all the source processes have delivered their data. Processes which have no data dependency path between them are in separate tasks. Each task can run at its own rate. Data dependencies allow schedulers to make more efficient use of CPU resources. Since the source and sink processes of a data dependency cannot execute simultaneously, we can use that information to eliminate some combinations of processes which may want to run at the same time. Narrowing the scope of process conflicts allows us to more accurately predict how the CPU will be used.

A real-time operating system is often designed to have a small memory footprint, since embedded systems are more cost-sensitive than general-purpose computers. RTOSs are also designed to be more responsive in two different ways. First, they allow greater control over the order of execution of processes, which is critical for ensuring that deadlines are met. Second, they are designed to have lower context-switching overhead, since that overhead eats into the time available for meeting deadlines. The kernel of an RTOS is the basic set of functions that is always resident in memory. A basic RTOS may have an extremely small kernel of only a few hundred instructions. Such microkernels often provide only basic context-switching and scheduling facilities. More complex RTOSs may provide high-end operating system functions such as file systems and network support; many high-end RTOSs are POSIX (a Unix standard) compliant. While running such a high-end operating system requires more hardware resources, the extra features are useful in a number of situations. For example, a controller for a machine on a manufacturing line may use a network interface to talk to other machines on the factory floor or the factory coordination unit; it may also use the file system to access a database for the manufacturing process.

13.4 Hardware/Software Codesign

Hardware/software codesign refers to any methodology which takes into account both hardware and software during the design of an embedded computing system. When the hardware and software are designed together, the designer has more opportunities to optimize the system by making trade-offs between the hardware and software components. Good system designers intuitively perform codesign, but codesign methods are increasingly being embodied in computer-aided design (CAD) tools. We will discuss several aspects of codesign and codesign tools, including models of the design, cosimulation, performance analysis, and various methods for architectural cosynthesis. We will conclude with a look at design methodologies that make use of these phases of codesign.

13.4.1 Models

In designing embedded computing systems, we make use of several different types of models at different points in the design process. We need to model basic functionality. We must also capture nonfunctional requirements: speed, weight, power consumption, manufacturing cost, etc.

In the earliest stages of design, the task graph is an important modeling tool. The task graph does not capture all aspects of functionality, but it does describe the various rates at which computations must be performed and the expected degrees of parallelism available. This level of detail is often enough to make some important architectural decisions. A useful adjunct to the task graph are the technology description tables, which describe how processes can be implemented on the available components. One of the technology description tables describes basic properties of the processing elements, such as cost and basic power dissipation. A separate table describes how the processes may be implemented on the components, giving execution time (and perhaps other function-specific parameters like precise power consumption) on a processing element of that type. The technology description is more complex when ASICs can be used as processing elements, since many different ASICs at differing price/performance points can be designed for a given functionality, but the basic data still applies.

A more detailed description is given by either high-level language code (C, etc.) for software or hardware description language code (VHDL, Verilog, etc.) for software components. These should not be viewed as specifications—they are, in fact, quite detailed implementations. However, they do provide a level of abstraction above assembly language and gates and so can be valuable for analyzing performance, size, etc. The control-data flow graph (CDFG) is a typical representation of a high-level language: a flowchart-like structure describes the program's control, while data flow graphs describe the behavior within expressions and basic blocks.

13.4.2 Cosimulation

Simulation is an important tool for design verification. The simulation of a complete embedded system entails modeling both the underlying hardware platform and the software executing on the CPUs. Some of the hardware must be simulated at a very fine level of detail—for example, busses and I/O devices may require gate-level simulation. On the other hand, the software can and should be executed at a higher level of abstraction. While it would be possible to simulate software execution by running a gate-level simulation of the CPU and modeling the program as residing in the memory of the simulated CPU, this would be unacceptably slow.

We can gain significant performance advantages by running different parts of the simulation at different levels of detail: elements of the hardware can be simulated in great detail, while software execution can be modeled much more directly. Basic functionality aspects of a high-level language program can be simulated by compiling the software on the computer on which the simulation executes, allowing those parts of the program to run at the native computer speed. Aspects of the program which deal with the hardware platform must interface to the section of the simulator which deals with the hardware. Those sections of the program are replaced by stubs which interface to the simulator. This style of simulation is a multirate simulation system, since the hardware and software simulation sections run at different rates: a single instruction in the software simulation will correspond to several clock cycles in the hardware simulation. The main jobs of the simulator are to keep the various sections of the simulation synchronized and to manage communication between the hardware and software components of the simulation.

13.4.3 Performance Analysis

Since performance is an important design goal in most embedded systems, both for overall throughput and for meeting deadlines, the analysis of the system to determine its speed of operation is an important element of any codesign methodology. System performance—the time it takes to execute a particular aspect of the system's functionality—clearly depends both on the software being executed and the underlying hardware platform. While simulation is an important tool for performance analysis, it is not sufficient, since simulation does not determine the worst-case delays. Since the execution times of most programs are data-dependent, it is necessary to give the simulation of the program the proper set

of inputs to observe worst-case delay. The number of possible input combinations makes it unlikely that one will find those worst-case inputs without the sort of analysis that is at the heart of performance analysis.

In general, performance analysis must be done at several different levels of abstraction. Given a single program, one can place an upper bound on the worst-case execution time of the program. However, since many embedded systems consist of multiple processes and device drivers, it is necessary to analyze how these programs interact with each other, a phase which makes use of the results of single-program performance analysis.

Determining the worst-case execution time of a single program can be broken into two subproblems: determining the longest execution path through the program and determining the execution time of that program. Since there is at least a rough correlation between the number of operations and the actual execution time, we can determine the longest execution path without detailed knowledge of the instructions being executed—the longest path depends primarily on the structure of conditionals and loops. One way to find the longest path through the program is to model the program as a control-flow graph and use network flow algorithms to solve the resulting system.

Once the longest path has been found, we need to look at the instructions executed along that path to determine the actual execution time. A simple model of the processor would assume that each instruction has a fixed execution time, independent of other factors such as the data values being operated on, surrounding instructions, or the path of execution. In fact, such simple models do not give adequate results for modern high-speed microprocessors. One problem is that in pipelined processors, the execution time of an instruction may depend on the sequence of instructions executed before it. An even greater cause of performance variations is caching, since the same instruction sequence can have variable execution times, depending on whether the code is in the cache. Since cache miss penalties are often 5X or 10X, the cost of mischaracterizing cache performance is significant. Assuming that the cache is never present gives a conservative estimate of worst-case execution time, but one that is so over-conservative that it distorts the entire design. Since the performance penalty for ignoring the cache is so large, it results in using a much faster, more expensive processor than is really necessary. The effects of caching can be taken into account during the path analysis of the program—path analysis can determine bound how often an instruction present in the cache.

There are two major effects which must be taken into account when analyzing multiple-process systems. The first is the effect of scheduling multiple processes and device drivers on a single CPU. This analysis is performed by a scheduling algorithm, which determines bounds on when programs can execute. Ratemonotonic analysis is the simplest form of scheduling analysis—the utilization factor given by ratemonotonic analysis tells one an upper limit on the amount of active CPU time. However, if data dependencies between processes are known, or some knowledge of the arrival times of data is known, then a more accurate performance estimate can be computed. If the system includes multiple processing elements, more sophisticated scheduling algorithms must be used, since the data arrival time for a process on one processing element may be determined by the time at which that datum is computed on another processing element.

The second effect which must be taken into account is interactions between processes in the cache. When several programs on a CPU share a cache, or when several processing elements share a second-level cache, the cache state depends on the behavior of all the programs. For example, when one process is suspended by the operating system and another process starts running, that process may knock the first program out of the cache. When the first process resumes execution, it will initially run more slowly, an effect which cannot be taken into account by analyzing the programs independently. This analysis clearly depends in part on the system schedule, since the interactions between processes depends on the order in which the processes execute. But the system scheduling analysis must also keep track of the cache state—which parts of which programs are in the cache at the start of execution of each process. Good accuracy can be obtained with a simple model which assumes that a program is either in the cache or out of it,

without considering individual instructions; higher accuracy comes from breaking a process into several subprocesses for analysis, each of which can have its own cache state.

13.4.4 Hardware/Software Cosynthesis

Hardware/software cosynthesis tries to simultaneously design the hardware and software for an embedded computing system, given design requirements such as performance as well as a description of the functionality. Cosynthesis generally concentrates on architectural design rather than detailed component design—it concentrates on determining such major factors as the number and types of processing elements required and the ways in which software processes interact.

The most basic style of cosynthesis is known as hardware/software partitioning. As shown in Figure 13.4, this algorithm maps the given functionality onto a template architecture consisting of a CPU and one or more ASICs communicating via the microprocessor bus. The functionality is usually specified as a single program. The partitioning algorithm breaks that program into pieces and allocates pieces either to the CPU or ASICs for execution. Hardware/software partitioning assumes that total system performance is dominated by a relatively small part of the application, so that implementing a small fraction of the application in the ASIC leads to large performance gains. Less performance-critical sections of the application are relegated to the CPU.

The first problem to be solved is how to break the application program into pieces; common techniques include determining where I/O operations occur and concentrating on the basic blocks of inner loops. Once the application code is partitioned, various allocations of those components must be evaluated. Given an allocation of program components to the CPU or ASICs, performance analysis techniques can be used to determine the total system performance; performance analysis should take into account the time required to transfer necessary data into the ASIC and to extract the results of the computation from the ASIC. Since the total number of allocations is large, heuristics must be used to search the design space. In addition, the cost of the implementation must be determined. Since the CPU's cost is known in advance, that cost is determined by the ASIC cost, which varies as to the amount of hardware required to implement the desired function. High-level synthesis can be used to estimate both the performance and hardware cost of an ASIC which will be synthesized from a portion of the application program.

Basic, cosynthesis heuristics start from extreme initial solutions: We can either put all program components into the CPU, creating an implementation which is minimal cost but probably does not meet performance requirements, or put all program elements in the ASIC, which gives a maximal-performance, maximal-expense implementation. Given this initial solution, heuristics select which program component to move to the other side of the partition to either reduce hardware cost or increase performance, as desired. More sophisticated heuristics try to construct a solution by estimating how critical a component will be to overall system performance and choosing a CPU or ASIC implementation accordingly. Iterative improvement strategies may move components across the partition boundary to improve the design.

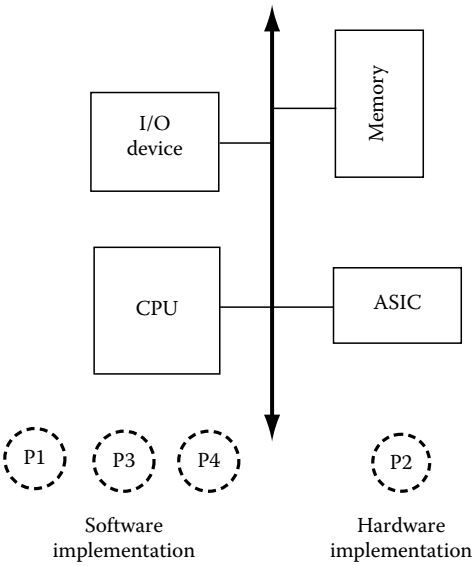


FIGURE 13.4 Hardware/software partitioning.

However, many embedded systems do not strictly follow the one CPU, one bus, *n* ASIC architectural template. These more general architectures are known as distributed embedded systems. Techniques for designing distributed embedded systems rest on the foundations of hardware/software partitioning, but they are generally more complicated, since there are more free variables. For example, since the number and types of CPUs is not known in advance, the cosynthesis algorithm must select them. If the number of busses or other communication links is not known in advance, those must be selected as well. Unfortunately, these decisions are all closely related. For example, the number of CPUs and ASICs required depends on the system schedule. The system schedule, in turn, depends on the execution time of each of the components on the available hardware elements. But those execution times depend on the processing elements available, which is what we are trying to determine in the first place. Cosynthesis algorithms generally try to fix several designs and vary only one or a few, then check the results of a design decision on the other parameters. For example, the algorithm may fix the hardware architecture and try to move processes to other processing elements to make more efficient use of the available hardware. Given that new configuration of processes, it may then try to reduce the cost of the hardware by eliminating unused processing elements or replacing a faster, more expensive processing element with a slower, cheaper one.

Since the memory hierarchy is a significant contributor to overall system performance, the design of the caching system is an important aspect of distributed system cosynthesis. In a board-level system with existing microprocessors, the sizes of second-level caches is under designer control, even if the first-level cache is incorporated on the microprocessor and therefore fixed in size. In a single-chip embedded system, the designer has control over the sizes of all the caches. Cosynthesis can determine hardware elements such as the placement of caches in the hardware architecture and the size of each cache. It can also determine software attributes such as the placement of each program in the cache. The placement of a program in the cache is determined by the addresses used by the program—by relocating the program, the cache behavior of the program can be changed. Memory system design requires calculating the cache state when constructing the system schedule and using the cache state as one of the factors to determine how to modify the design.

13.4.5 Design Methodologies

A codesign methodology tries to take into account aspects of hardware and software during all phases of design. At some point in the design process, the hardware and software components are well-specified and can be designed relatively independently. But it is important to consider the characteristics of both the hardware and software components early in design. It is also important to properly test the system once the hardware and software components are assembled into a complete system.

Cosynthesis can be used as a design planning tool, even if it is not used to generate a complete system architectural design. Because cosynthesis can evaluate a large number of designs very quickly, it can determine the feasibility of a proposed system much faster than a human designer. This allows the designer to experiment with what-if scenarios, such as adding new features or speculating on the effects of lower component costs in the future. Many cosynthesis algorithms can be applied without having a complete program to use as a specification. If the system can be specified to the level of processes with some estimate of the computation time required for each process, then useful information about architectural feasibility can be generated by cosynthesis.

Cosimulation plays a major role once subsystem designs are available. It does not have to wait until all components are complete, since stubs may be created to provide minimal functionality for incomplete components. The ability to simulate the software before completing the hardware is a major boon to software development and can substantially reduce development time.

14

Design Automation Technology Roadmap

14.1 Introduction **14**-1
14.2 Design Automation: Historical Perspective **14**-3
The 1960s: The Beginnings of Design Automation • The 1970s: The Awaking of Verification • The 1980s: Birth of the Industry • The 1990s: The Age of Integration
14.3 The Future **14**-27
International Technology Roadmap for Semiconductors • EDA Impact
14.4 Summary **14**-38
References **14**-38

Donald R. Cottrell
Silicon Integration Initiative, Inc.

14.1 Introduction

No invention in the modern age has been as pervasive as the semiconductor and nothing has been more important to its technological advancement than has electronic design automation (EDA). EDA began in the 1960s both for the design of electronic computers and because of them. It was the advent of the computer that made possible the development of specialized programs that perform the complex management, design, and analysis operations associated with electronics and electronic systems. At the same time, it was the design, management, and manufacture of the thousands (now tens of millions) of devices that make up a single electronic assembly that made EDA an absolute requirement to fuel the semiconductor progression. Today, EDA programs are used in electronic packages for all business markets from computers to games, telephones to aerospace guidance systems, and toasters to automobiles. Across these markets, EDA supports many different package types such as integrated circuit (IC) chips, multichip modules (MCMs), printed circuit boards (PCBs), and entire electronic system assemblies.

No electronic circuit package is as challenging to EDA as the IC. The growth in complexity of ICs has placed tremendous demands on EDA. Mainstream EDA applications such as simulation, layout, and test generation have had to improve their speed and capacity characteristics with this ever-increasing growth in the number of circuits to be processed. New types of design and analysis applications, new methodologies, and new design rules have been necessary to keep pace. Yet, even with the technological breakthroughs that have been made in EDA across the past four decades, it is still having difficulty keeping up with the breakthroughs being made in the semiconductor technology progression that it fuels. Decrease in size and spacing of features on the chip is causing the number of design elements per chip to increase at a tremendous rate. The decrease in feature size and spacing coupled with the increase in operating frequency is causing additional levels of complexity to be approximated in the models used by design and analysis programs.

In the period from 1970 to the present semiconductor advances such as the following have had a great impact on EDA technology (Figure 14.1):

- IC integration has grown from tens of transistors on a chip, to beyond tens of millions.
- Feature size on production ICs has shrunk from 10 μm to 90 nm and smaller.
- On-chip clock frequency has increased from a few megahertz to many gigahertz.

Playing an essential part in the advancement of EDA have been advances in computer architectures that run the EDA applications. These advances have included the following:

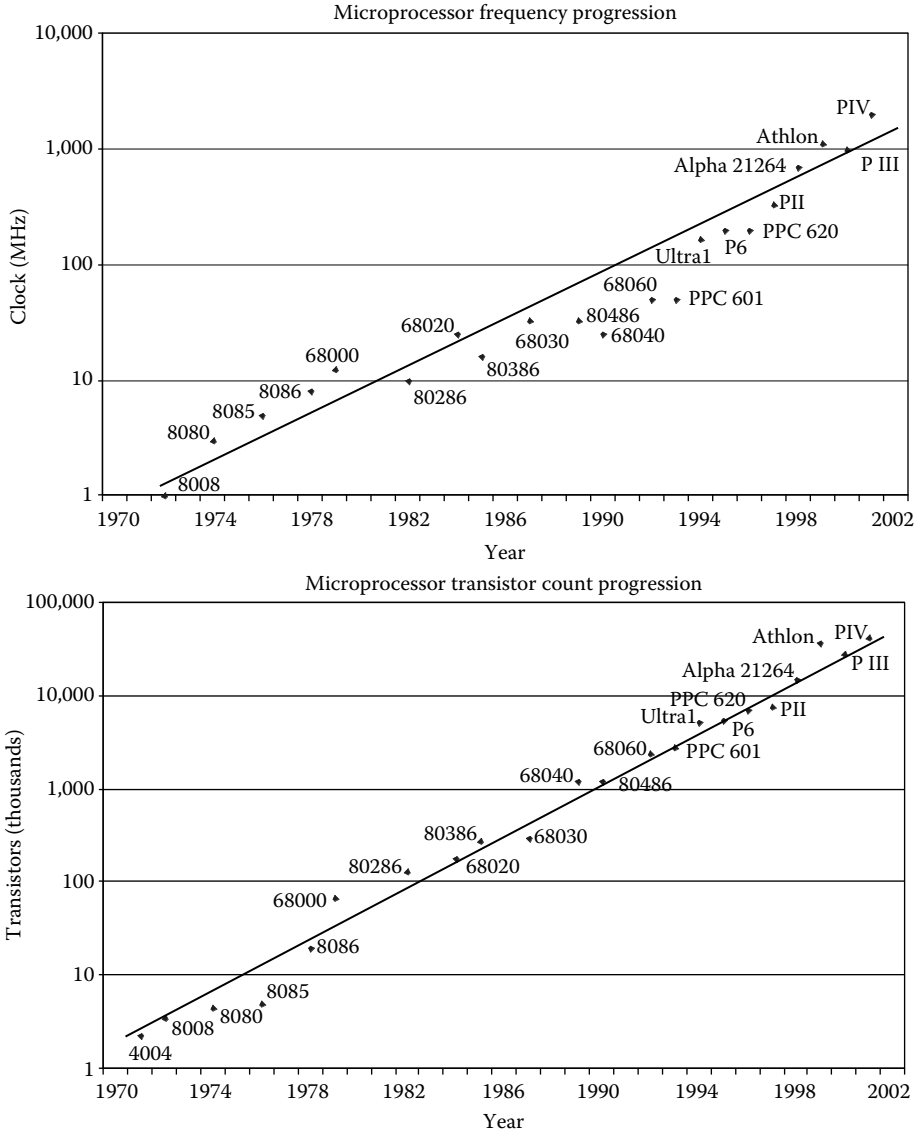


FIGURE 14.1 Microprocessor development.

- Computer CPU speed: from less than a million instructions per second (MIPS) of shared mainframe to hundreds of MIPS on a dedicated workstation
- Computer memory: from <32 KB to >500 GB
- Data archive: from voluminous reels of (rather) slow-speed tape to virtually limitless amounts of high-speed electronic storage

However, these major improvements in computing power alone would not have been sufficient to meet the EDA needs of semiconductor advancement. Major advances have also been made to fundamental EDA algorithms, and entirely new design techniques and design paradigms have been invented and developed to support semiconductor advancement. This chapter will trace the more notable advancements made in EDA across its history for electronics design and discuss important semiconductor technology trends predicted across the next decade along and the impact they will have on EDA for the future. It is important to understand these trends and projections, because if the EDA systems cannot keep pace with the semiconductor projections, then these projections cannot be realized. Although it may be possible to build foundries that can manufacture ultradeep submicron wafers and even to acquire the billions of dollars of capital required for each, without the necessary EDA support these factories will never be fully utilized. SEMATECH reports that chip design productivity has increased at a compounded rate somewhere between 21% and 30%, while Moore's Law predicts the number of transistors on a chip to increase at a compound rate of 56%. This means that at some time, bringing online new foundries that can produce smaller, denser chips may reach a point of diminishing returns, because the ability to design and yield chips with that many transistors may not be possible.

14.2 Design Automation: Historical Perspective

14.2.1 The 1960s: The Beginnings of Design Automation

Early entries into design automation were made in the areas of design (description) records, PCB wiring, and manufacturing test generation. A commercial EDA industry did not exist and developments were made within companies with the need, such as IBM [1] and Bell Labs, on mainframe computers such as the IBM 7090. The 7090 had addressable 36-bit words and a limit of 32,000 words of main storage (magnetic cores). This was far less than the typical 256+ MB of RAM on the average notebook PC, and certainly no match for a high-function workstation with gigabytes of main store. Though computer limitations continue to be an on-going challenge for EDA development, the limitation of the computers of the 1960s was particularly acute.

In retrospect, the limitation of computers in the 1960s was a blessing for the development of design automation. Because of these limitations, design automation developers were forced to invent highly creative algorithms that operated on very compact data structures. Many of the fundamental concepts developed during this period are still in use within commercial EDA systems today. Some notable advances during this period were

- Fundamental "stuck-at" model for manufacturing test and a formal algebra for the generation of tests and diagnosis of faults
- Parallel fault simulation, which provided simulation of many fault conditions in parallel with the good-machine (nonfaulty circuit) to reduce fault-simulation run times
- Three-valued algebra for simulation which yields accurate results using simple delay models, even in the presence of race conditions within the design.
- Development of fundamental algorithms for the placement and wiring of components
- Checking of designs against prescribed electrical design requirements (rules)

Also, there was development of fundamental heuristics for placement and wire routing, and for divide-and-conquer concepts supporting both. One such concept was the hierarchical division of a wiring image

into cells, globally routing between cells, and then performing detailed routing within cells, possibly subdividing them further. Many of these fundamental concepts are still applied today, although the complexities of physical design (PD) of today's large-scale integration (LSI) are vastly more complex.

The 1960s represented the awakening of design automation and provided the proof of its value and need for electronics design. It would not be until the end of this decade when the explosion of the number of circuits designed on a chip would occur and the term LSI would be coined. EDA development in the 1960s was primarily focused on printed circuit assemblies, but the fundamental concepts developed for design entry, test generation, and PD provided the basics for EDA in the LSI era.

14.2.1.1 Design Entry

Before the use of computers in electronics design, the design schematic was a hard-copy drawing. This drawing was a draftsman's rendering of the notes and sketches provided by the circuit designer. The drawings provided the basis for manufacturing and repair operations in the field. As automation developed, it became desirable to store these drawings on storage media usable by computers so that the creation of input to the automated processes could, itself, be automated. So, the need to record the design of electronic products and assemblies in computers was recognized in the late 1950s and early 1960s. In the early days of design automation, the electronics designer would develop the design using paper and pencil and then transcribe it to a form suitable for keyed entry to a computer. Once keyed into the computer, the design could be rendered in a number of different formats to support the manufacturing and field operations. It was soon recognized that these computerized representations of the circuit design drawing could also drive design processes such as the routing of printed circuit traces or the generation of manufacturing test patterns. Finally, from there, it was but a short step to the use of computers to generate data in the form required to drive automated manufacturing and test equipment.

Early design entry methods involved the keying of the design description onto punch cards that were read into the computer and saved on a persistent storage device. This became known as the design's database, and is the start of the design automation system. From the database, schematic diagrams and logic diagrams were rendered for use in engineering, manufacturing, and field support. This was typically a two-step process, whereby the designer drew the schematic by hand and then submitted it to another for conversion to the transcription records, keypunch, and entry to the computer. Once in the computer, the formal automated drawings were generated, printed, and returned to the designer. Although this process seems archaic by today's standards, it did result in a permanent record of the design in computer readable format. This could be used for many forms of records management, engineering change history, and as input to design, analysis, and manufacturing automation that would soon follow.

With the introduction of the alphanumeric terminal, the keypunch was replaced as the window into the computer. With this, new design description languages were developed, and the role of the transcription operator began to move back to the designer. Although these description languages still represented the design at the device or gate level, they were free format and keyword oriented and design engineers were willing to use them. The design engineer now had the tools to enter design descriptions directly into the computer thus eliminating the inherent inefficiencies of the ''middleman.'' Thus, a paradigm shift began to evolve in the method by which design was entered to the EDA system. Introduction of the direct access storage devices (disks) in the late 1960s also improved the entry process as well as the entire design system by providing online, high-speed direct access to the entire design or any portion of it. This was also important to the acceptance of design entry by the designer as the task was still viewed as a necessary overhead rather than a natural part of the design task. It was necessary to get access to the other evolving design automation tools, but typically, the real design thought process took place with pencil and paper techniques. Therefore, any change that made the entry process faster and easier was eagerly accepted.

The next shift occurred in the later part of the 1970s with the introduction of graphics terminals. With these, the designer could enter design into the database in schematic form. This form of design entry was a novelty at first, but not a clear performance improvement. In fact, until the introduction of the

workstation with dedicated graphics support, graphic entry was detrimental to design productivity in many cases. Negative effects such as less than effective transaction speed, time lost in making the schematic esthetically pleasing and the low level of detail all added to less than obvious advances. In contrast, use of the graphics display to view the design and make design changes proved extremely effective and was a great improvement over the red-lined hard-copy prints. For this reason, use of computer graphics represented a major advance and this style of design entry took off with the introduction of the workstation in the 1980s. In fact, the graphic editor's glitz and capability was often a major selling point in the decision to use one EDA system over another. Further, to be considered a commercially viable system, graphics entry was required. Nevertheless, as the density of ICs grew, graphics entry of schematics would begin to yield to the productivity advantages of (alphanumeric) description languages. As EDA design and analysis tool technology advanced, entry of design at the register-transfer level (RTL) would become commonplace and today the design engineer is able to represent his design ideas at many levels of abstraction and throughout the design process. System-level and RTL design languages have been introduced and the designer is able to verify design intent much earlier in the design cycle.

By the 1990s and after the introduction of synthesis automation, design entry using RTL descriptions was the generally accepted approach for entry of design, although schematic entry remained the accepted method for PCB design and many elements of custom ICs. There is no doubt that the introduction of graphics into the design automation system represents a major advance and a major paradigm shift. The use of graphics to visualize design details, wiring congestion, and timing diagrams is of major importance. The use of graphics to perform edit functions is standard operating procedure.

Large system design, often, entails control circuitry, dataflow, and functional modules. Classically, these systems span across several chips and boards and employ several styles of entry for the different physical packages. These may include

- Schematics—graphic
- RTL and behavioral level languages—alphanumeric
- Timing diagrams—graphic
- State diagrams—alphanumeric
- Flowcharts—graphic

Today, these entry techniques can be found in different EDA tools and each is particularly effective for different types of design problems. Schematics are effective for the design of ''glue'' logic that interconnects functional design elements such as modules on a PCB and for custom IC circuitry. Behavioral languages are useful for system-level design, and particularly effective for dataflow behavior. Timing diagrams lend themselves well to describe the functional operations of ''black-box'' components at their I/Os without needing to describe their internal circuitry. State diagrams are a convenient way to express the logical operation of combinational circuits. Flowcharts are effective for describing the operations of control logic, much like use of flowcharts for specification of software program flow. With technology advances, the IC is engulfing more and more of the entire system and all of these forms of design description may be prevalent on a single chip. It is even expected that the design of ''black-box'' functions will be available from multiple sources to be embedded onto the chip similar to the use of modules on a PCB. Thus, it is likely that future EDA systems will support a mixture of design description forms to allow the designer to represent sections of the design in a manner most effective to each. After all, design is described in many forms by the designer outside the design system.

14.2.1.2 Test Generation

Testing of manufactured electronic subassemblies entails the use of special test hardware that can provide stimulus (test signals) to selected (input) pins of the part under test and measure for specified responses on selected (output) pins. If the measured response matches the specified response, then the part under test has passed that test successfully. If some other response is measured, then the part has failed that test

and the presence of a defect is indicated. Manufacturing test is the successive application test patterns that cause some measurable point on the part under test to be sensitized to the presence of a manufacturing defect. That is, some measurable point on the part under test will result in a certain value if the fault is present and a different one if no fault were present. The collection of test patterns causes all (or almost all) possible manufacturing failures to render a different output response than would the nondefective part. For static DC testers, each stimulus is applied and after the part under test settles to a steady state, the specified outputs are measured and compared with the expected results for a nondefective part.

To bound the test generation problem, a model was developed to represent possible defects at the abstract gate level. This model characterizes the effects of defects as stuck-at values. This model is fundamental to most of the development in test generation and is still in use today. It characterizes defects as causing either a stuck-at-one or a stuck-at-zero condition at pins on a logic gate. It assumes hard faults (i.e., if present, a fault remains throughout the test) and that only one fault occurs at a time. Thus, this model became known as the single stuck-at fault model. Stuck-at fault testing assumes that the symptom of any manufacturing defect can be characterized by the presence of a stuck-at fault some place within the circuit and that it can be observed at some point on the unit under test. By testing for the presence of all possible stuck-at faults that can occur, all possible manufacturing hard-defects in the logic devices can thus be tested.

The stuck-at fault models for NAND and NOR gates are shown in Figure 14.2. For the NAND gate, the presence of an input stuck-at-one defect can be sensitized (made detectable) at the gate's output pin by setting the good-machine state for that input to 0, and setting the other input values to 1. If a zero is observed at the gate's output node, then a fault (stuck-at-one) on the input pin set to 0 is detected. A stuck-at-zero condition on any specific input to the NAND gate is not distinguishable from a stuck-at-zero on any other input to the gate, thus is not part of the model. However, a stuck-at-one is modeled for the gate's output to account for such a fault or a stuck-at-one fault on the gate's output circuitry. Similarly, a stuck-at-zero is modeled on the gate's output.

The fault model for the NOR gate is similar except that here input faults are modeled as stuck-at-zero, as the stuck-at-one defect cannot be isolated to a particular input.

Later in time, additional development would attack defects not detectable with this stuck-at fault model; for example, bridging faults where nodes are shorted together, and delay faults where the output response does not occur within the required time. The stuck-at fault model cannot detect these fault types and they became important as the development of CMOS progressed. Significant work was performed in both these areas beginning in the 1970s, but it did not have the impact on test generation development that the stuck-at fault model did.

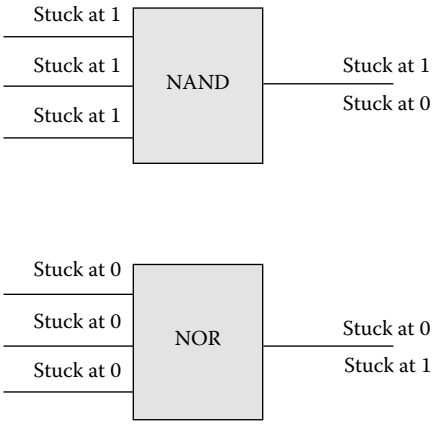


FIGURE 14.2 Stuck-at fault models.

The creation of the fault model was very important to test generation as it established a realistic set of objectives to be met by automated test set generation that could be achieved in a realistic amount of computational time. A formal algebra was developed by Roth [2] called the D-ALG that formalized an approach to test generation and fault diagnosis.

The test generation program could choose a fault based on the instances of gates within the design and the fault models for the gates. It could then trace back from that fault to the input pins of the design and, using the D-ALG calculus, it could find a set of input states that would sensitize the fault. Then, it could trace forward from that fault to the design's output pins, sensitizing the path (causing the good-machine value to be the opposite of the stuck-at fault value along that path) to at least one observable pin (Figure 14.3).

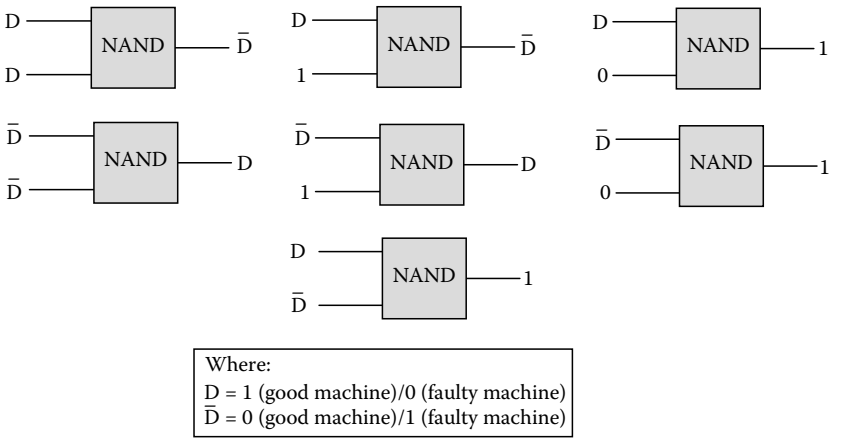


FIGURE 14.3 D-ALG calculus.

The use of functional patterns as the test patterns in lieu of heuristic stuck-at test generation was another approach for manufacturing test. However, this required an extreme number of tests to be applied and provided no measurable objective to be met (i.e., 100% of all possible stuck-at faults) thus, depended on the experience and skill of the designer to create quality tests. The use of fault models and automatic test generation produced a minimum set of tests and greatly reduced intensive manual labor.

The exhaustive method to assure coverage of all the possible detectable defects would be to apply all possible input states to the design under test and compare the measured output states with the simulated good-machine states. For a design with n input pins, however, this may theoretically require the simulation of 2^n input patterns for combinatorial logic and at least 2^{n+m} for sequential logic (where m is the number of independent storage elements). For even a relatively small number of input pins, this amount of simulation would not be possible even on today's computers, and the time to apply this number of patterns at the tester would be grossly prohibitive.

The amount of time that a part resides on the tester is critical in the semiconductor business, as it impacts a number of parts that can be produced in a given amount of time and the capital cost for testers. Thus, the number of test patterns that need to be applied at the tester should be kept to a minimum. Early work in test generation attacked this problem in two ways. First, when a test pattern was generated for a specific fault, it was simulated against all possible faults one at a time. In many cases, the application of a test pattern that is targeted for one specific fault will also detect several other faults at the same time. The presence of a specific fault may be detectable on one output pin, for example, but at the same time additional faults may be observable at the other output pins. The use of fault simulation (discussed in Section 14.2.1.3) detected these cases and provided a mechanism to mark as tested those faults that were "accidentally" covered. This meant that the test generation algorithm did not have to generate a specific test for those faults. Second, schemes were developed to merge test patterns together into a smaller test pattern set to minimize the number of patterns required for detection of all faults. This is possible when two adjacent test patterns require the application of specific values on different input pins, each allowing all the other input pins to be at a do not-care state. In these cases, the multiple test patterns can be merged into one. With successive analysis in this way, all pairs of test patterns (pattern n with $n+1$, or the merger of m and $m+1$ with pattern $m+2$) are analyzed and merged into a reduced set of patterns.

Sequential design elements severely complicate test generation, as they require the analysis of previous states and the application of sequences of patterns. Early work in test generation broke feedback nets, inserted a lumped delay on them, and analyzed the design as a combinatorial problem using a Huffman model. Later work attempted to identify the sequential elements within the design using sophisticated topological analysis and then used a Huffman model to analyze each unique element. State tables for each

unique sequential element were generated and saved for later use as lookup tables in the test generation process. The use of three-value simulation within the analysis reduced the analysis time as well as guaranteed that the results were always accurate. Huffman analysis required 2^x simulations (where x is the number of feedback nets) to determine if critical hazards existed in the sequential elements. Using three-valued simulation [3] (all value transitions go through an X state), this was reduced to a maximum of 2^x simulations.

This lumped delay model did not account for the distribution of delays in the actual design, thus it often caused pessimistic results. Often simulated results yielded do not-know (X-state) conditions when a more accurate model could yield a known state. This made it difficult to generate patterns that would detect all faults in the model. As the level of integration increased, the problems associated with automatic test generation for arbitrary sequential circuits became unwieldy. This necessitated that the designer be called back into the problem of test generation most often to develop tests that would detect those that were missed by the test generation program. New approaches that ranged from random pattern generation to advanced algorithms and heuristics, which use a combination of different approaches, were developed. However, by the mid-1970s the need to design-for-test was becoming evident to many companies.

During the mid-1970s the concept of scan design such as IBM's Level Sensitive Scan Design (LSSD), was developed [4]. Scan design provides the ability to externally control and observe internal state variables of the design. This is accomplished by the use of special scan latches into the design at the points to be controlled and observed. These latches are controllable and can accept one of the two different data inputs depending on an external control setting. One of these data inputs is the node within the IC to be controlled/observed and the other is used as the test input. These latches are connected into a shift register chain that has its stage-0 test input and stage-n output connected to externally accessible pins on the IC. Under normal conditions, signals from within the design are passed through individual latches via the data input/output. Under test conditions, test vectors can be scanned, under clock control, onto the scan register's externally accessible input pin and into the scan latches via their test data input. Once scanned into the register, the test vector is applied to the internal nodes. Similarly, internal state values within the design are captured in individual scan latches and scanned through the register out to its observable output pin.

The development of scan design was important for two reasons. First, LSSD allowed for external control and observability of all sequential elements within the design. With specific design-for-test rules, this reduced the problem of test generation to one of a set of combinatorial circuits. Rigid design rules such as the following assure this, and checking programs and scan chain generation algorithms were implemented to assure that they were adhered to before entering test generation:

- All internal storage elements implemented in hazard-free polarity-hold latches
- Absence of any global feedback loops
- Latches may not be controlled by the same clock that controls the latches feeding them
- Externally accessible clocks control all shift register latches

Second, scan design allowed for external control and observability at otherwise nonprobable points between modules on an MCM or PCB. During the 1980s an industry standard was developed, called Boundary Scan (IEEE 1149.1 Joint Test Action Group) [5], which uses scan design techniques to provide control and observability for all chip or module I/Os from pins of their next level package (MCM or PCD, respectively).

Scan design requires additional real estate in the IC design and has a level of performance overhead. However, with the achievable transistor density levels on today's ICs and the test and diagnosis benefits accrued, these penalties are easily justified in all but the most performance critical designs (Figure 14.4).

During the 1980s, IC density allowed for the design of built-in self-test (BIST) circuitry on the chip itself. Operating at hardware speed, it became feasible to generate complete exhaustive tests and large numbers of random tests never possible with software and stored program testers. BISTs are generated

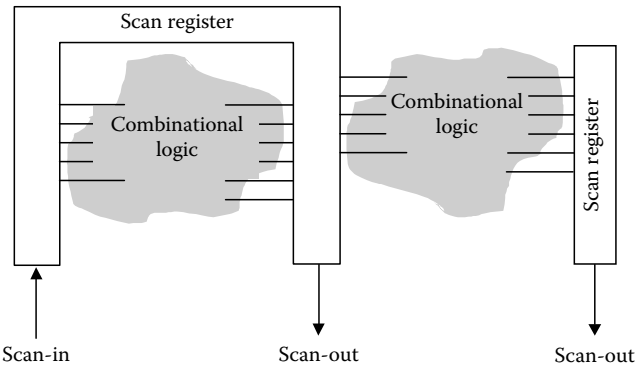


FIGURE 14.4 Scan design.

on the tester by the device under test, reducing the management of data transfer from design to manufacturing. A binary counter is used or linear feedback shift registers (LFSR) are used to generate the patterns for exhaustive or random tests, respectively. In the latter case, a pseudorandom bit sequence is formed by the exclusive-OR of the bits on the LFSR and this result is then fed back into the LFSR input. Thus, a pseudorandom bit sequence whose sequence length is based on the number of LFSR stages and the initial LFSR state can be generated. The design can be simulated to determine the good-machine state conditions for the generated test patterns, and these simulated results are compared with the actual device-under-test results observed at the tester.

BIST techniques have become common for the test of on-chip RAM and ROS. BIST is also used for logic sections of chips either with fault-simulated weighted random test patterns or good machine-simulated exhaustive patterns. In the latter case, logic is partitioned into electrically isolated regions with a smaller number of inputs to reduce the number of test patterns. Partitioning of the design reduces the number of exhaustive tests from 2^n (where n is the total number of inputs) to

$$\sum_{i=1}^{m} 2^{n^i}$$

where

m is the number of partitions

n^i ($n^i < n$) is the number of inputs on each logic partition

Since the use of BIST implies an extremely large number of tests, the simulated data transfer and test measurement time is reduced greatly by the use of a compressed signature to represent the expected and actual test results. Thus, only a comparison of the simulated signatures for each BIST region needs to be made with the signatures derived by the on-chip hardware, rather than the results of each individual test pattern. This is accomplished by feeding the output bit sequence to a single input serial input LFSR after exclusive-OR with the pseudorandom pattern generated by that LFSR. In this way, a unique pattern can be observed for a sequence of test results, which is a function of the good-machine response and a pseudorandom number (Figure 14.5).

Today, testing of ICs typically consists of combinations of different test strategies. These may include stored program stuck-at fault tests, BIST, delay (or at-speed) test (testing for signal arrival times in addition to state), and I_{DDQ} tests (direct drain quiescent current testing checks for CMOS defects that cause excessive current leakage). The latter two test techniques are used to detect defect conditions not identified by stuck-at fault tests such as shorts (bridging faults) between signal nets or gate oxide defects

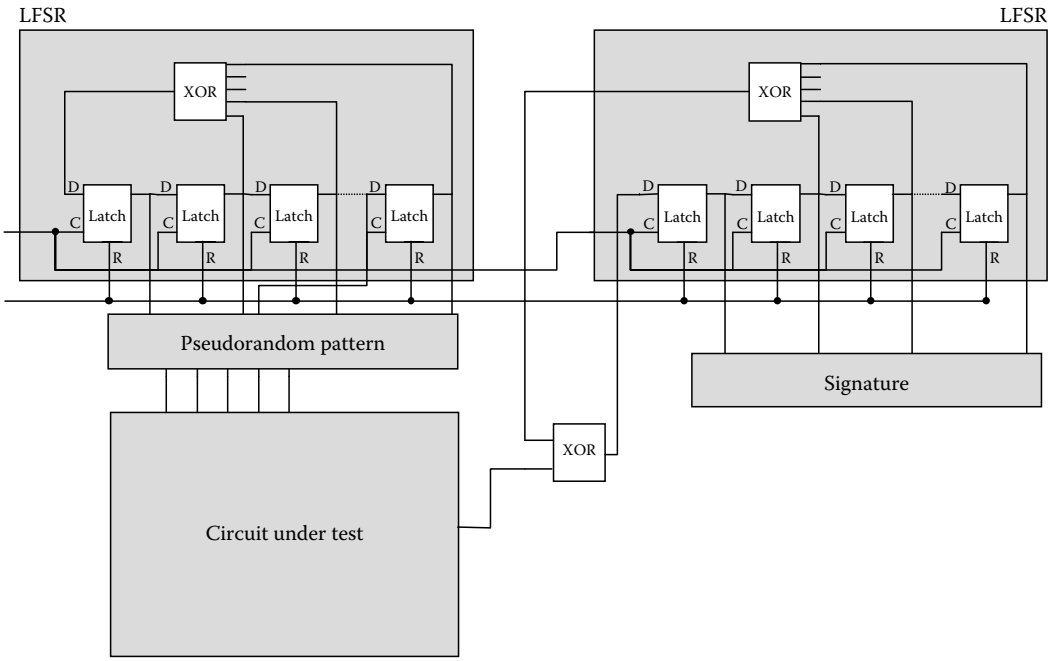


FIGURE 14.5 BIST logic.

that cause incorrect device operation and parametric variability resulting from process variations in the wafer foundry. However, the stuck-at fault model and scan techniques have been and will continue to be fundamental ingredients for the manufacturing test recipe.

14.2.1.3 Fault Simulation

Fault simulation is used to predict the state of a design at observable points when in the presence of a defect. This is used for manufacturing test and for field diagnostic pattern generation. Early work in fault simulation relied on the stuck-at fault model and performed successive simulations of the design with each single fault independent of any other fault; thus, a single stuck-at fault model was assumed. Because even these early designs consisted of thousands of faults, it was too time consuming to simulate each fault serially and it was necessary to create high-speed fault simulation algorithms.

For manufacturing test, fault simulation took advantage of three-valued zero-delay simulation. The simulation model of the design was levelized and compiled into an executable program. Levelization assured that driver gates were simulated before the gates receiving the signals; thus, allowing a state resolution in a single simulation pass. Feedback loops were cut and the *X*-transition of three-valued simulation resolved race conditions. The inherent instruction set of the host computer (e.g., AND, OR, and XOR.) allowed the use of a minimum set of instructions to simulate a gate's function.

The parallel-fault simulation algorithm that allowed many faults to be simulated in parallel was developed during the 1960s. Using the 32-bit word length of the IBM 7090 computer architecture, for example, simulation of 31 faults in a single pass (using the last bit for the good-machine) was possible. For each gate in the design, two host machine words were assigned to represent its good-machine state and the state for 31 single stuck-at faults. The first bit position of the first word was set to 1 or 0 representing the good-machine state for the node, and each of the successive bit position was set to the 1/0 state that would occur if the simulated faults were present (each bit-position representing a single fault). The corresponding bit position in the second word was set to 0 if it was a known state

and one if it was an *X*-state. The entire fault list was divided into *n* partitions of 31 faults and each partition was then simulated against all input patterns. In this way, the run time for simulation is a function of

$$\sum_{i=1}^{n=F/31} Patterns$$

where *F* is the total number of single stuck-at faults in the design.

Specific faults are injected within the word representing their location within the design by the insertion of a mask that is AND'd or OR'd at the appropriate word. For stuck-at-one conditions the mask contains a 1-bit in the position representing the fault (and 0-bit at the others) and it is OR'd to the gate's memory location. For stuck-at-zero faults the mask contains a 0-bit at the positions representing the fault (and 1-bit at the others) and it is AND'd with the gate's memory location.

As the level of integration increased, so did the number of faults. The simulation speed improvement realized from parallel-fault simulation was limited to the number of faults simulated in parallel and because of the algorithm overhead, it did not scale with the increase in faults.

Deductive-fault simulation was developed early in the 1970s and required only one simulation pass per test pattern. This is accomplished by simulating only the good-machine behavior and using deductive techniques to determine each fault that is detectable along the simulated paths. Note here that fault detection became the principal goal and fault isolation (for repair purposes) was ignored, as by now the challenge was to isolate from a wafer bad chips that were not repairable. Because lists of faults detectable at every point along the simulated path need to be kept, this algorithm requires extensive memory, far more than the parallel-fault simulation one. However, with increasing memory on host computers and the inherent increase in fault simulation speed, this technique won the favor of many fault simulators.

Concurrent-fault simulation refined the deductive algorithm by recognizing the fact that paths in the design quickly become insensitive to the presence of most faults, particularly after some initial set of test patterns is simulated (an observation made in the 1970s was that a high percentage of faults is detected in a low percentage of the initial test patterns, even if these patterns are randomly generated). The concurrent-fault simulation algorithm simulates the good machine and concurrently simulates a number of faulty machines. Once it is determined that a particular faulty machine state is the same as the good-machine one, simulation for that fault ceases. Since on logic paths most faults will become insensitive rather close to the point where the fault is located, the amount of simulation for these faulty machines was kept small. This algorithm required even more memory, particularly for the early test patterns; however, host machine architectures of the late 1970s were supporting, what then appeared as, massive amounts of addressable memory.

With the introduction of scan design in which all sequential elements are controllable from the tester, the simulated problem is reduced to that of a combinatorial circuit whose state is deterministic based on any single test pattern, and is not dependent on previous patterns or states. Parallel-pattern fault simulation was developed in the late 1970s to take advantage of this, by simulating multiple test patterns in parallel against a single fault. A performance advantage is achieved because compiled simulation could again be utilized as opposed to the more costly event-based approach. In addition, because faults not detected by the initial test patterns are typically only detectable by a few patterns and for these a sensitized path often disappears within a close proximity to the fault's location, simulation of many patterns does not require a complete pass across the design.

Because of the increasing number of devices on chips, the test generation and fault simulation problem continued to face severe challenges. With the evolution of BIST however, the use of fault simulation was relaxed. With BIST, only the good-machine behavior needs to be simulated and compared with the actual results at the tester.

14.2.1.4 Physical Design

As with test generation, PD has evolved from early work on PCBs. PD automation programs place devices and generate the physical interconnect routing for the nets that connect them into logic paths assuring that electrical and physical constraints are met. The challenge for PD has become ever greater since its early development for PCBs where the goal was simply to place components and route nets typically looking for the shortest path. Any nets that could not be auto-routed were routed (embedded) manually, or as a last resort with nonprinted (yellow) wires. As the problem moved on to the ICs, the ability to use nonprinted wires to finish routing was no more. Now, all interconnects had to be printed circuits and anything less than a 100% solution is unacceptable. Further, as the IC densities increased, so did the number of nets which necessitated the invention of smarter wiring programs and heuristics. Even a small number of incomplete (overflow) routes became too complex of a task for manual solutions.

As IC device sizes shrank and the gate delays decreased, the delay caused by interconnect wiring also became an important factor for a valid solution. No longer was any wiring solution a correct solution. Complexity increased by the need to find wiring solutions that fall within acceptable timing limits. Thus, the wiring lengths and thickness needed to be considered. As IC features become packed closer together cross-coupled capacitance (cross talk) effects between them is also an important consideration and for the future, wiring considerations will expand into a three-dimensional space. PD solutions must consider these complex factors and still achieve a 100% solution that meets the designer-specified timing for IC designs that contain hundreds of millions of nets.

Because of these increasing demands on the PD, major paradigm changes have taken place in the design methodology. In the early days, there was a clear separation of logic design and PD. The logic designer was responsible for creating a netlist that correctly represented the logic behavior desired. Timing was a function of the drive capability of the driving circuit and the number of receivers. Different power levels for drivers could be chosen by the logic designer to match the timing requirements based on the driven circuits. The delay imposed by the time-of-flight along interconnects and owing to the parasitics on the interconnect was insignificant. Therefore, the logic designer could hand off the PD to another, more adept at using the PD programs and manually embedding overflow wires. As the semiconductor technology progressed, however, there needed to be more interactions between the logic designer and the physical designer, as the interconnect delays became a more dominant factor across signal paths. The logic designer had to give certain timing constraints to the physical designer and if these could not be met, the design was often passed back to the logic designer. The logic designer, in turn, then had to choose different driver gates or a different logical architecture to meet his design specification. In many cases, the pair had to become a team or there was a merger of the two previously distinct operations into one ''IC designer.''

This same progression of merging logic design and PD into one operational responsibility has also begun at the EDA system architecture level. In the 1960s and 1970s, front-end (design) programs were separate from back-end (physical) programs. Most often they were developed by different EDA development teams and designs were transferred between them by means of data files. Beginning in the 1980s, the data transferred between the front-end programs and the back-end ones included specific design constraints that must be met by the PD programs—the most common being a specific amount of allowed delay across an interconnect or signal path. Moreover, as the number of constraints that must be met by the PD programs increases so does the challenge to achieve a 100% solution. Nonetheless, many of the fundamental wiring heuristics and algorithms used by PD today spawned from work done in the 1960s for PCBs.

Early placement algorithms were developed to minimize the total length of the interconnect wiring using Steiner trees and Manhattan wiring graphs. In addition, during these early years, algorithms were developed to analyze wiring congestion that would occur as a result of placement choices and minimize it to give routing a chance to succeed. Later work in the 1960s led to algorithms that performed a hierarchical division of the wiring image and performed global wiring between these subdivisions

(then called cells) before routing within the cells [6]. This divide-and-conquer approach simplified the problem and led to quicker and more complete results. Min-cut placement algorithms often used today are a derivative of this divide-and-conquer approach. The image is divided into partitions and the placements of these partitions are swapped to minimize interconnect length between them and possible wiring congestion. Once a global solution for the cells is found, placement is performed within them using the same objectives. Many current placement algorithms are based on these early techniques, although they now need to consider more physical and electrical constraints.

Development of new and more efficient routing algorithms progressed. The Lee algorithm [7] finds a solution by emitting a ''wave'' from both the source and target points to be wired. This wave is actually an ordered identification of available channel positions—where the available positions adjacent to the source or destination are numbered 1, and the available positions adjacent to them are numbered 2, etc. Successive moves and sequential identification is made (out in all directions as would a wave) until the source and destination moves meet (the waves collide). Then a backtrace is performed from the intersecting position in reverse sequential order along the numbered track positions back to the source and destination. At points where a choice is available (i.e., there are two adjacent points with the same order number), the one which does not require a change in direction is chosen.

The Hightower line-probe technique [8], also developed during this period and speeded up routing, by use of emanating lines rather than waves. This algorithm emanated a line from both the source and destination points, toward each other. When either line encounters an obstacle, then another line is emanated from a point just missing the edge of the obstacle on the original line at a right angle to the original line, and toward the target or source. Thus, the process is much like walking blindly in an orthogonal line toward the target and changing direction only after bumping into a wall. This process continues until the lines intersect at which time the path is complete.

In today's ICs, the challenge of completed wiring that meets all constraints is of crucial importance. Unlike test generation, which can be considered successful when a very high percentage of the faults are detected by the test patterns, 100% complete is the only acceptable answer for PD. Further, all of the interconnects must fall within the required electrical and physical constraints. Nothing $<$100% is acceptable! Today these constraints include timing, power consumption, noise, and yield and this list will become more complex as IC feature sizes and spacing are reduced further.

14.2.2 The 1970s: The Awaking of Verification

Before the introduction of large-scale integrated (LSI) circuits in the 1970s, it was common practice to build prototype hardware to verify the design correctness. PCB packages containing discrete components, single gates, and small-scale integrated modules facilitated engineering rework of real hardware within the verification cycle. Prototype PCBs were built and engineering used test stimulus drivers and oscilloscopes to determine whether the correct output conditions resulted from input stimuli. As design errors were detected, they were repaired on the PCB prototype, validated, and recorded for later engineering into the production version of the design. Use of the wrong logic function within the design could easily be repaired by replacing components in error with the correct ones. Incorrect connections could easily be repaired by cutting a printed circuit and replacing it with a discrete wire. Thus, design verification (DV) was a sort of trial-and-error process using real hardware.

The introduction of LSI circuits drastically changed the DV paradigm. Although the use of software simulation to verify system design correctness began during the 1960s, it was not until the advent of the LSI circuits that this concept became widely accepted. With large-scale integration, it became impossible to use prototype hardware or to repair a faulty design after it was manufactured. The 1970s are best represented by a quantum leap into verification before manufacture through the use of software modeling (simulation). This represented a major paradigm shift in electronics design and was a difficult change for some to accept. DV on hardware prototypes resulted in a tangible result that could be touched and held. It was a convenient tangible, which could be shown by management to represent real progress.

Completed DV against a software model did not produce the same level of touch and feel. Further, since the use of computer models was a relatively new concept, it met with the distrust of many. However, the introduction of LSI circuits demanded this change and today, software DV is a commonplace practice used in all levels of electronic components, subassemblies, and systems.

Early simulators simulated gate-level models of the design with two-valued (1 and 0) simulation. Since gates had nearly equal delays and the interconnect delay was insignificant by comparison, the use of a unit of delay for each gate with no delay assigned to interconnects was common. Later simulators exploited the use of three-valued simulation (1, 0, and unknown) to resolve race conditions and identify oscillations within the design more quickly. With the emergence of LSI circuits, however, these simple models had to become more complex and, additionally, simulators had to become more flexible and faster—much faster. In the first half of the 1970s, important advances were made to DV simulators that included

- Use of abstract (with respect to the gate-level) models to improve simulation performance and enable verification throughout the design cycle (not just at the end)
- More accurate representations of gate and interconnect delays to enhance the simulated accuracy

In the latter half of the decade, significant contributions were made to facilitate separation of function verification from timing verification and formal approaches for verification. However, simulation remains a fundamental DV tool and the challenge to make simulators faster and more flexible continues even today.

14.2.2.1 Simulation

Though widely used for DV, simulation has a couple of inherent problems. First, unlike test generation or PD, there is no precise measure of completeness. Test generation has the stuck-at fault model and PD has a finite list of nets that must be routed. However, there is no equivalent metric to determine when verification of the design is complete, or when enough simulation is done. During the 1970s research began to develop a metric for verification completeness, but to this day none has been generally accepted. Use is made of minimum criteria such as all nets must switch in both directions, and statistical models using random patterns. Recent work in formal verification is applying algorithmic approaches to validate coverage of paths and branches within the model. However, the generally accepted goal is to ''do more.''

Second, it is a hard and time-consuming task to create effective simulation vectors to verify a complex design. DV simulators typically support a rich high-level language for the stimulus generation, but still require the thought, experience, and ingenuity of the DV engineer to develop and debug these programs. Therefore, it is desirable to simulate the portion of the design being verified in as much of the total system environment as possible and have the simulation create functional stimulus for the portion to be verified.

Finally, it is tedious to validate the simulated results for correctness, making it desirable to simulate the full system environment where it is easier to validate results. Ideally, the owner of an application-specific integrated circuit (ASIC) chip being designed for a computer could simulate that chip within a model of all of the computer's hardware, the microcode, the operating system, and use example software programs as the ultimate simulation experiment. To even approach this goal, however, simulator technology must continuously strive for rapid and more rapid techniques.

Early DV simulators often used compiled models (where the design is represented directly as a computer program), but this technique gave way to interpretative event-driven simulation. Compiled simulators have the advantage of higher speeds because host machine instructions are compiled in-line to represent the design to be verified and are directly executed with minimum simulator overhead. Event-based simulators require additional overhead to manage the simulation operations, but provide a level of flexibility and generality not possible with the compiled model. This flexibility was necessary to provide for simulation of timing characteristics as well as function, and to handle general sequential designs. Therefore, this approach was generally adopted for DV simulators.

14.2.2.1.1 Event-Based Simulation

There are four main concepts to an event-based simulator:

- Netlist, which provides the list of blocks (gates at first, but any complex function later), connections between blocks, and delay characteristics of the blocks.
- Event time queues, which are lists of events that need to be executed (blocks that need to be simulated) at specific points in (simulation) time. Event queues contain two types of events—update and calculate. Update-events change the specified node to the specified value, then schedule calculate-events for the blocks driven from that node. Calculate-events call the simulation behavior for the specified block and, on return from the behavior routine, schedule update-events to change the states on the output nodes to the new values at the appropriate (simulation) time
- Block Simulation Behavior (the instructions that will compute that block's output(s) states when there is a change to its input(s) state—possibly also scheduling some portion of the block's behavior to be simulated at a later time).
- Value list—the current state of each node in the design.

Simulation begins by scheduling update-events in appropriate time queues for the pattern sequence to be simulated. After the update-events are stored, the first-time event queue is traversed and, one-by-one, each update-event in the queue is executed. Update-events update the node in the value list and, if the value new value is different from the current value (originally set to unknown), schedule calculate-events for blocks driven from the updated node. These calculate-events are saved back in time queues based on delay specifications for later execution—which will be discussed later. After all update-events are executed and removed from the queue, the simulator selects calculate-events in the queue sequentially, interprets its function, and passes control to the appropriate block simulation behavior for calculation.

The execution of a calculate-event causes simulation of a block to take place. This is accomplished by passing control to the simulation behavior of the block with pointers to the current state-values on its inputs (in the value list). When complete, the simulation routine of the block will pass control back to the simulator with the new state condition for its output(s). The simulator then schedules the block output(s) value update by placing an update-event for it in the appropriate time queue. The decision of which time queue the update-events are scheduled within is based on what the delay value is for the block.

Once all calculate-events are executed and removed from the queue, the cycle begins again at the next time queue and the process of executing update-events followed by calculate-events for the current time queue repeats. This cycle repeats until there are no more events or until some specified maximum simulation time. To keep update-events separated from calculate-events within the linked-list queues, it is common to add update-events at the top of the linked-list and calculate-events at the bottom, updating the chain-links accordingly.

Because it is not possible to predetermine how many events will reside in a queue at any time, it is common to create these as linked lists of dynamically allocated memory. Additionally, time queues are linked since the required number of time queues cannot be determined in advance and similar to events, new time queues can be inserted into that chained list as required (Figure 14.6).

A number of techniques have been developed to make queue management fast and efficient as they are the heart of the simulator. Because of its generality, the event-based algorithm was the clear choice for DV. One advantage of event-based simulation over the compiled approach is that it easily supports the simulation of delay. Delays can be simulated for blocks and nets and even early simulators typically supported a very complete delay model, even before sophisticated delay calculators were available to take advantage of it.

Treatment of delay has had to evolve and expand since the beginnings of software DV. Unit delay was the first model used—where each gate in the design is assigned one unit of delay and interconnect delays are zero. This was a crude approximation, but allowed for high-speed simulations because of the

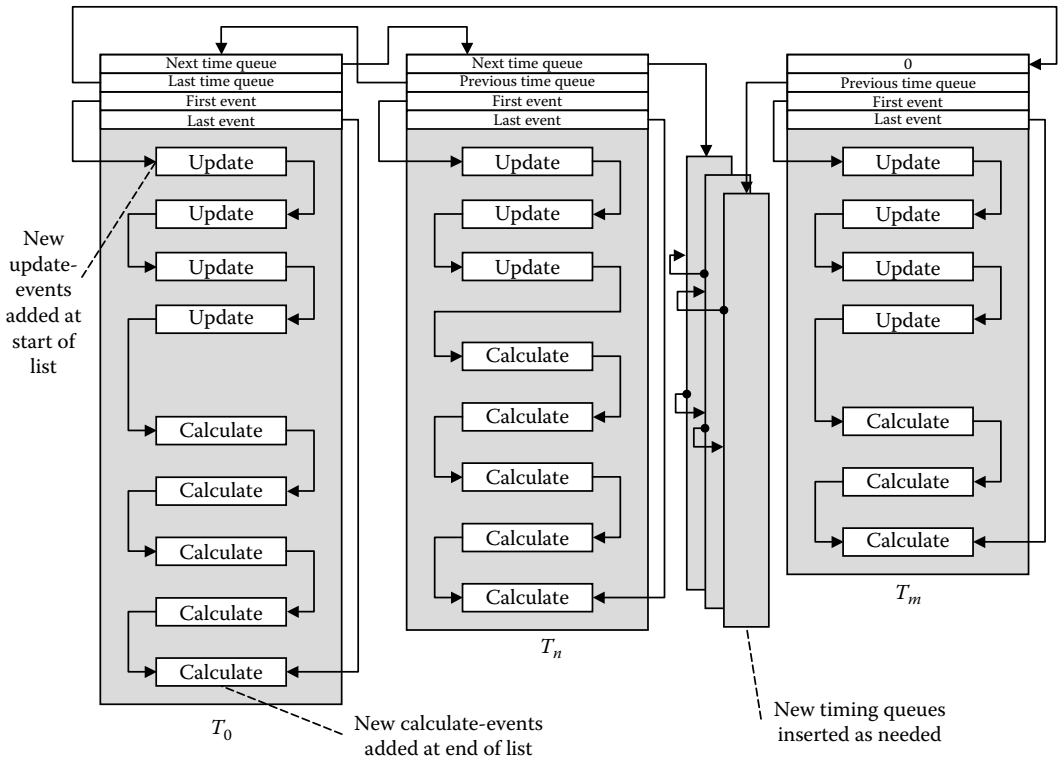


FIGURE 14.6 Event queues.

simplifying assumptions and it worked reasonably well. As the IC evolved, however, the unit delay model was replaced by a lumped delay model in which, each gate could be assigned a unique value for delay—actually, a rise-delay and a fall-delay. This was assigned by the technologist based on some average load assumption. At this time, also, the beginnings of development of delay calculators began. These early calculators used simple equations, adding the number of gates driven by the gate being calculated, and then adding the additional delay to the intrinsic delay values of that gate. As interconnect wiring became a factor in the timing of the circuit the pin-to-pin delay came into use. Though delay values used for simulation in the 1970s was crude by today's norm, the delay model Figure 14.7 was rich and supported specification and simulation for

- Intrinsic block delay (T_{block})—the time required for the block output to change state relative to the time that a controlling input to that block changed state
- Interconnect delay (T_{int})—the time required for a specific receiver pin in a net to change state relative to the time that the driver pin changed state
- Input–output delay (T_{io})—the time required for a specific block output to change state relative to the time the state changed on a specific input to that block

Today's ICs, however, require delay calculation based on very accurate distributed RC models for interconnects, as these delays have become more significant with respect to gate delays. Future ICs will require even more precise modeling for delays, as will be discussed later in this chapter, and consider inductance (L) in addition to the RC parasitics. Additionally, transmission line models will be necessary for the analysis of certain critical global interconnects. However, the delay model defined in the early years of DV and the capabilities of the event-based simulation algorithm stand ready to meet this challenge.

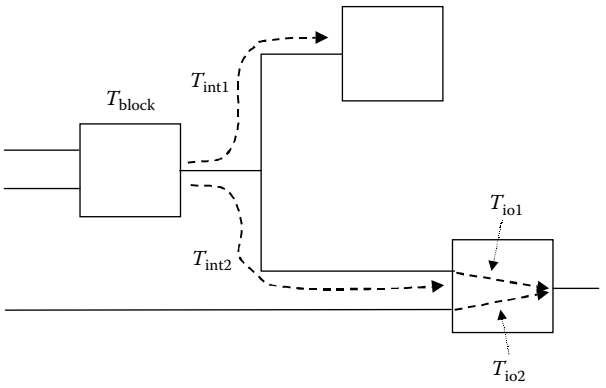


FIGURE 14.7 Simulation delay types.

Another significant advantage of the event-based simulation is that it easily supports simulation of blocks in the netlist at different levels of abstraction. Recall that one of the fundamental components of event simulation is the block simulation behavior. A calculation-event for a block passes control to the subroutine that simulates the behavior using the block's input states. For gate-level simulation, these behavior subroutines are quite simple—AND, OR, NAND, and NOR. However, since the behavior is actually a software program, it can be arbitrarily complex as well. Realizing this, early work took place to develop description languages that could be coded by designers and compiled into block simulation behaviors to represent arbitrary sections of a design as a single (netlist) block. For example, a block simulation behavior might look like the following:

```
  FUNCTION (CNOR);
/* Complimentary output NOR function with 3 inputs. Input delay=2,
   intrinsic delay for true output=10, intrinsic delay for complement
   output=12*/
  INPUT (a,b,c); /*Declare for inputs a, b, c*/
  OUTPUT (x,y);
  DECLARE input1, input2, input3; /*Declare storage for inputs*/
  DECLARE out, cout;              /*Declare storage for outputs*/

  GET a,input1, b,input2, c,input3;

  DELAY 2, Entry1;
    Entry1: out=input1|input2|input3;
    cout=¬out;

    DELAY 10, (x=out); /*Schedule this update-event for 10 time units
                         later */
    DELAY 12 (y=cout); /*Schedule this update-event for 12 time units
                         later */
    END Entry1;

  ENDCNOR;
```

The sophisticated use of these behavioral commands supported the direct representation not only for simple gates, but also for complex functions such as registers, MUXs, RAM, and ROS. In doing so, simulation performance was improved because what was treated before as a complex interconnection of gates could now be simulated as a single block.

By generalizing the simulator system, block simulation routines could be loaded from a library at simulation run time only when required (by the use of a block with that function in the netlist), and dynamically linked with the simulator control program. This meant that the block simulation behaviors could be developed by anyone, compiled independently from the simulator, stored in a library of simulation behavioral models, and used as needed by a simulation program. This is common practice in DV simulators today, but this concept in the early 1970s represented a significant breakthrough and supported a major paradigm shift in design—namely, the concept of top-down design and verification.

With top-down design, the designer no longer had to design the gate-level netlist before verification could take place. High-level (abstract) models could be written to represent the behavior of sections of the design not yet complete in detail, and they could be used in conjunction with gate-level descriptions of the completed parts to perform full system verification. Now, simulation performance could be improved by use of abstracted high-speed behavior models in place of detailed gate-level descriptions for portions of the overall design. Now, concurrent design and verification could take place across design teams. Now, there was a formal method to support design reuse of system elements without the need to expose internal design details for reusable elements. In the extreme case, the system designer could write a single functional model for the entire system and verify it with simulation. The design could then be partitioned into subsystem elements and each could be described with an abstract behavioral model before being handed off for detailed design. During the detailed design phase, individual designers could verify their subsystem in the full system context even before the other subsystems were completed. Also, the behavioral model concept was particularly valuable for generation of the functional patterns to be simulated as they could now be generated by the simulation of the other system components. Thus, verification of the design no longer had to wait until the end, it could now be a continuous process throughout the design.

Throughout the period, improvements were made to DV simulators to improve performance and in the formulation and capability of behavioral description languages. In addition, designers found increasingly novel ways to use behavioral models to describe full systems containing nondigital system elements and peripherals such as I/O devices. Late in the 1970s and during the 1980s two important formal languages for describing system behavior were developed:

- VHDL [9] (very high-speed integrated circuit [VHSIC] high-level description language), sponsored by DARPA
- Verilog [10], a commercially developed RTL description language

These two languages are now accepted industry-standard design description languages.

14.2.2.1.2 Compiled Simulation

Synchronous design such as that used for scan-based design which emerged in the 1970s provides for a clean separation of timing verification and functional verification of combinatorial circuits. Because of this, the use of compiled simulation returned for high-speed verification for designs meeting constraints. A simulation technique called cycle simulation that was developed yielded a major performance advantage over event-based simulation. Cycle-simulation treats the combinational sections of a scan-based design as compiled zero-delay models moving data between them at clock-cycle boundaries. The simulator executes each combinatorial section at each simulated cycle by passing control to the compiled routine for it along with its input states. The resulting state values at the outputs of these sections are assumed to be correctly captured in their respective latch positions. That is, the clock circuitry and path delays are assumed correct, and are not simulated during this phase of verification. The (latched) output values are used as the input states to the combinatorial sections they drive at the next cycle, and the process repeats for each simulated cycle. Each simulation pass across the compiled models represents one cycle of the design's system clock, starting with an input state and resulting in an output state. To assure that only one pass is required, the gates or RTL statements for the combinational sections are levelized

before compilation into the host-machine instructions. This assures that value updates occur before calculations and only one pass across a section of the model is required to achieve the correct state response at the outputs.

Simulation performance was greatly improved with cycle simulation because of the compiled model and because the clock circuitry did not have to be simulated repeatably with each simulated-machine cycle. Cycle simulation did not replace event simulation even for constrained synchronous designs as the clock circuitry needs to be verified; however, it proved to be effective for many large systems and these early developments provided the foundations for modern compiled simulators used today.

14.2.2.1.3 Hardware Simulators

During the 1980s, research and development took place on custom hardware simulators and accelerators. Special-purpose hardware-simulators use massively parallel instruction processors with much customized instruction sets to simulate gates. These provide simulation speeds that are orders of magnitude faster than the software simulation on general-purpose computers. However, they are expensive to build, lack the flexibility of software simulators, and the hardware technology they are built in soon becomes outdated (although general parallel architectures may allow the incremental addition of additional processors). Hardware accelerators use custom hardware to simulate portions of design in conjunction with the software simulator. These are more flexible, but still have the inherent problems that their big brothers have. Nonetheless, use of custom hardware to tackle the simulation performance demands has gained acceptance in many companies and they are commercially available today using both gate-level and hardware description language (HDL) design descriptions.

14.2.2.2 Timing Analysis

The practice of divide and conquer in DV started in the 1970s with the introduction of the behavioral model. Another divide-and-conquer style born in the 1970s, which achieved wide popularity in the 1990s, is to separate verification of the design's function from its timing. With the invention of scan design, it became possible to verify logic as a combinational circuit using high-speed compiled cycle simulators. Development of path tracing algorithms to verify timing resulted in a technique to verify timing without simulation; thus providing a complete solution which is not dependent on completeness of any input stimulus as required by simulation. For this reason alone, this technique coined static timing analysis (STA) was a major contribution to EDA—one that became key to the notion of ''signoff'' to the wafer foundry.

STA is used to analyze projected versus required timing along signal paths from primary inputs to latches, latches to latches, latches to primary outputs, and primary inputs to primary outputs. This is done without the use of simulation, but by summing the min–max delays along each path. At each gate, the STA program computes the min–max time in which that gate will change in state based on the min–max arrival times of its input signals. STA tools do not simulate the gate function, they only add its contribution to the path delay, although the choice of using rise or fall times for the gate is based on whether it has a complementary output, or not. Because the circuitry between the latches is combinational, only one pass needs to be made across the design. The addition can be based on the minimum rise or fall delay for gates or both, providing a min–max analysis. The designer specifies the required arrival times for paths at the latches or primary outputs, and the STA program compares these with the actual arrival times. The difference between the required arrival and the actual arrival is defined as slack. The STA tool computes the slack at the termination of each path, sorts them numerically, and provides a report. The designer then verifies the design correctness by analysis of all negative slacks.

Engineering judgment is applied during the analysis, including the elimination of false-path conditions. A false-path is a signal transition that will never occur in the real operation of the design. Since the STA tool does not simulate the behavior of the circuit, it cannot automatically eliminate all false-paths. Through knowledge of the signal polarity, STA can eliminate false-paths caused by the fan-out and reconvergence of certain signals. However, other forms of false-paths are only identifiable by the designer.

14.2.2.3 Formal Verification

Development of formal methods to verify the equivalence of two different representations of a design began during the 1970s. Boolean verification analyzes a circuit against a known good reference and provides a mathematical proof of equivalence. An RTL model, for example, of the reference circuit is verified using standard simulation techniques. The Boolean verification program then compiles the known-good reference design into a canonical NAND–NOR equivalent circuit. This equivalent circuit is compared with the gate-level hardware design using sophisticated theorem provers to determine equivalence. To reduce processing times, formal verification tools may preprocess the two circuits to create an overlapping set of smaller logic *cones* to be analyzed. These *cones* are simply the set of logic traversed by backtracing across the circuit from an output node (latch positions or primary outputs) to the controlling input nodes (latch positions or primary inputs). User controls specify the nodes that are supposed to be logically equivalent between the two circuits.

Early work in this field explored the use of test generation algorithms to prove equivalence. Two *cones* to be compared can be represented as $F_{\text{cone1}} = f(a, b, c, X, Y, Z, \dots)$ and $F_{\text{cone2}} = f(d, e, f, \dots, X', Y', Z', \dots)$. F_{cone1} and F_{cone2} are user defined output nodes to be compared for equivalence. The terms a, b, c and d, e, f represent the set of input nodes for the function and X, Y, Z are the computational subfunctions. User inputs define the equivalence between a, b, c and d, e, f. If F_{cone1} and F_{cone2} are functionally equivalent, then the value of G must be 0 for all possible input states. If the two cones are equivalent then, use of D-ALG test generation techniques for $G = F_{\text{cone1}}$ XOR F_{cone2} will be unable to derive a test for the stuck-at-zero fault on the output of G. Similarly, the use of random pattern generation and simulation can be applied to prove equivalence between *cones* by observing the value of G for all input states.

Research during the 1980s provided improvements in Boolean equivalence checking techniques and models (such as binary decision diagrams), and modern Boolean equivalence checking programs may employ a number of mathematical and simulation algorithms to optimize the overall processing. However, Boolean equivalence checking methods require the existence of a reference design against which equivalence is proved. This implies there must be a complete validation of the reference design against the design specification. Validating the reference design has typically been a job for simulation and, thus, is vulnerable to the problems of assuring coverage and completeness of the simulation experiment. Consequently, formal methods to validate the correctness of functional-level models have become an important topic in EDA research. Modern design validation tools use a combination of techniques to validate the correctness of a design model. These typically include techniques used in software development to measure completeness of simulation test cases such as

- Checking for coverage of all instructions (in the model)
- Checking to assure that all possible branch conditions (in the model) were exercised

They may also provide more formal approaches to validation such as

- Checking (the model) against designer-asserted conditions (or constraints) that must be met
- Techniques that construct a proof that the intended functions are realized by the design

These concepts and techniques continue to be the subject of research and will gain more importance as the size of IC designs stretches the limits of simulation-based techniques.

14.2.2.4 Verification Methodologies

With the use of structured design techniques, the design to be verified (Figure 14.8) can be treated as a set of combinational designs. With the arsenal of verification concepts that began to emerge during this period, the user had many verification advantages not previously available. Without structured design, delay simulation of the entire design was required. Use of functional models intermixed with gate-level descriptions of subsections of the design provided major improvements, but was still very costly and time consuming. Further, to be safe, the practical designer would always attempt to delay simulate the entire design at the gate level.

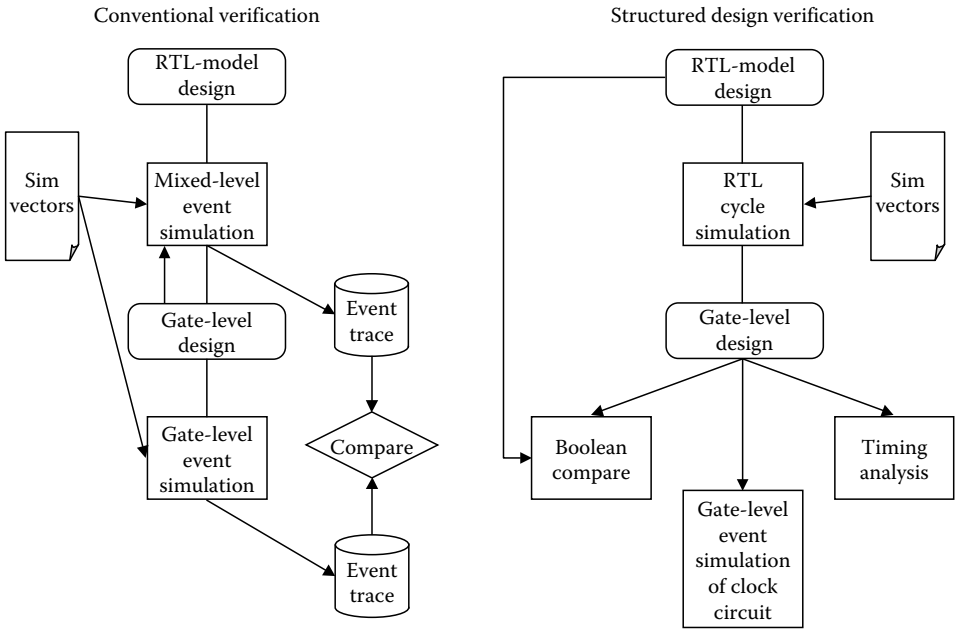


FIGURE 14.8 Verification methodology.

With structured design techniques, the designer could do the massive simulations at the RTL focusing of the logic function only, regardless of the timing. Cycle simulation improved the simulation performance by 1 to 2 orders of magnitude by eliminating repetitive simulations of the clock circuitry and use of compiled (or table lookup) simulation. For some, the use of massively parallel hardware simulators offered even greater simulation speeds—the equivalent of hundreds of millions events per second. Verification of the logic function of the gate-level design could be accomplished by formally proving its equivalence with the simulated RTL model. STA provided the basis to verify the timing of all data paths in a rigorous and methodical manner. Functional and timing verification of the clock circuitry could be accomplished with the standard delay-simulation techniques using event simulation. Collectively, these tools and techniques provided major design productivity improvements. However, the IC area and performance overhead required for these structured design approaches limited the number of designs taking advantage of them. During the 1980s as the commercial EDA business developed, these new verification tools and techniques remained as in-house tools in a few companies. Commercial availability did not emerge until the 1990s when the densities and complexities of ICs began to demand the change.

14.2.3 The 1980s: Birth of the Industry

Up to the 1980s design automation was, for the most part, developed in-house by a small number of large companies for their own proprietary use. Almost all EDA tools operated against large mainframe computers using company-specific interfaces. High-level description languages were unique and most often proprietary, technology rule formats were proprietary, and user interfaces were unique. As semiconductor foundries made their manufacturing lines available for customer-specific chip designs; however, the need for access to EDA tools grew. With the expansion of the ASIC, the need for commercially available EDA tools exploded. Suddenly, a number of commercial EDA companies began to emerge and electronics design companies had the choice of developing tools in-house or

purchasing them from a variety of vendors. The EDA challenge now often became that of integrating tools from multiple suppliers into a homogeneous system. Therefore, one major paradigm shift of the 1980s was the beginnings of EDA standards to provide the means to transfer designs from one EDA design system to another or from a design system to manufacturing. VHDL and Verilog matured to become IEEE industry-standard HDLs. Electronic data interchange format (EDIF) [11] was developed as an industry standard for the exchange of netlist and GDSII (developed in the 1970s at Calma) became a standard interface for transferring mask pattern data. (In 2003, SEMI introduced a new mask pattern exchange standard, OASIS, that compacts mask pattern data to one-tenth or less of the bytes used by GDSII.)

A second paradigm shift of the 1980s was the introduction of the interactive workstation as the platform for EDA tools and systems. Although some may view it as a step backward, the ''arcade'' graphics capabilities of this new hardware and its scalability caught the attention of design managers and made it a clear choice over the mainframe wherever possible. For a time, it appeared that many of the advances made in alphanumeric HDLs were about to yield to the pizzazz of graphical schematic editors. Nevertheless, although the graphics pizzazz may have dictated the purchase of one solution over another, the dedicated processing, and the ability to incrementally add compute power made the move from the mainframe to the workstation inevitable. Early commercial EDA entries such as from Daisy (Logician) and Valid (SCALD) were developed on custom hardware using commercial microprocessors from Intel and Motorola. This soon gave way, however, to commercially available workstations using RISC-based microprocessors, and UNIX became the de facto operating system standard. During the period, there was a rush of redevelopment of many of the fundamental algorithms for EDA onto the workstation. However, with the development of commercial products and with the new power of high-function graphics, these applications were vastly improved along the way. Experience and new learning streamlined many of the fundamental algorithms. The high-function graphic display provided the basis for enhanced user interfaces to the applications and high-performance high-memory workstations allowed application speeds to compete with the mainframe. The commercial ASIC business provided focus on the need for technology libraries from multiple manufacturers. Finally, there was significant exploration into custom EDA hardware such as hardware simulators and accelerators and parallel processing techniques.

From an IC design perspective however, the major paradigm shift of the 1980s was synthesis. With the introduction of synthesis, automation could be used to reduce an HDL description of the design to the final hardware representation. This provided major productivity improvements for ASIC design, as chip designers could work at the HDL level and use automation to create the details. Also, there was a much higher probability that the synthesis-generated design would be correct than for manually created schematics. The transgression from the early days of IC design to the 1980s is similar to what occurred earlier in software. Early computer programming was done at the machine language level. This could provide optimum program performance and efficiency, but at the maximum labor cost. Programming in machine instructions proved too inefficient for the vast majority of software programs, thus the advent of assembly languages. Assembly language programming offers a productivity advantage over machine instructions because the assembler abstracts up several of the complexities of machine language. Thus, the software designer works with less complexity, using the assembler to add the necessary details and build the final machine instructions. The introduction of functional-level program languages (such as FORTRAN then, and C++ now) provided even more productivity improvements by providing a set of programming statements for functions that would otherwise require many machine or assembler instructions to implement. Thus, the level at which the programmer could now work was even higher, allowing him/her to construct programs with far fewer statements. The analog in IC design is the progression of transistor-level design (machine level), to gate-level design (assembler level), to HDL-based design. Synthesis provided the basis for HDL-based design and its inherent productivity improvements, and major changes to IC design methodologies.

14.2.3.1 Synthesis

Fundamentally, synthesis is a three-step process:

- Compile an HDL description of a design into an equivalent NAND–NOR description
- Optimize the NAND–NOR description based on design targets
- Map the resulting NAND–NOR description to the technology building blocks (cells) supported for the wafer foundry (process) to be used

Although work on synthesis techniques has occurred on and off since the beginnings of design automation and back to Transistor–Transistor Logic (TTL)-based designs, it was not until gate array style ICs reached a significant density threshold that it found production use. In the late 1970s and 1980s, considerable research and development in industry and universities took place on the high-speed algorithms and heuristics for synthesis. In the early 1980s, IBM exploited the use of synthesis on the ICs used in the 3090 and AS400 computers. These computers used chips that were designed from qualified sets of predesigned functions interconnected by personalized wiring. These functions (now called cells) represented the basic building blocks for each specific IC family. The computers used a high number of uniquely personalized chip designs so it was advantageous to design at the HDL level and use automation to compile to the equivalent cell-level detail. The result was significant improvements to the overall design productivity.

Today, synthesis is a fundamental cornerstone of the design methodology (Figure 14.9) for ASICs supporting both VHDL and Verilog as the standard input. As the complexities of IC design have increased, so have the challenges for synthesis. Early synthesis had relatively few constraints to be observed in its optimization phase. Based on user controls, the design was optimized for minimum area, minimum fan-out (minimum power), or maximum fan-out (maximum performance). This was accomplished by applying a series of logic reduction algorithms (transforms) that provide different types of reduction and refinement [12]. Cell behavior could also be represented in primitive-logic equivalent form and a topological analysis could find matches between the design gate patterns and equivalent cells.

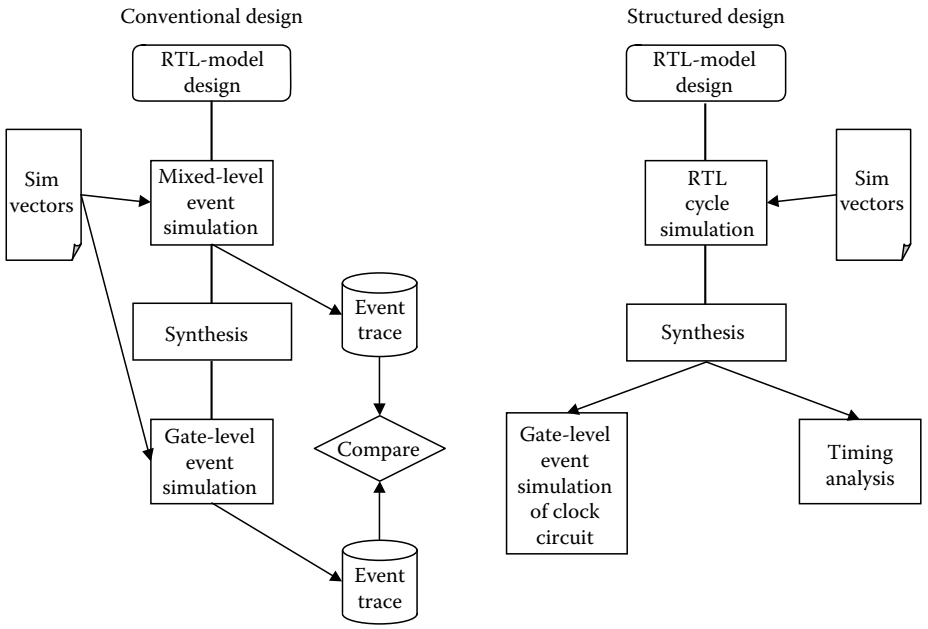


FIGURE 14.9 Synthesis methodology.

The mapping phase would then select the appropriate cells based on function and simple electrical and physical parameters for each cell in the library such as drive strength and area. Previously, synthesis was not overly concerned with signal delays in the generated design nor for other electrical constraints that the design may be required to meet. As IC feature sizes decreased, however, these constraints became critical in synthesis and automation became required to consider them in its solution.

The design input to modern synthesis tools is not the functional HDL design alone, but now includes constraints such as the maximum allowed delay along a path between two points in the design. This complicates the synthesis decision process as it must now generate a solution that meets the required function with a set of cells and interconnections that will fall within the required timing constraints. Therefore, additional trade-offs must be made between the optimization and mapping phases, and the effects of interconnection penalty need to be considered. Additionally, synthesis must now have technology characteristics available to it such as delay for cells and wiring. To determine this delay it is necessary to understand the total load seen by a driver cell as a result of the input capacitance of the cells it drives and the RC parasitics on the interconnect wiring. However, the actual wiring of the completed IC is not yet available (since routing has not taken place) and estimates for interconnect parasitics must be made. These are often based on wire load models that are created from empirical analysis of other chip designs in the technology. These wire load models are tables that provide an estimate of interconnect length and parasitic values based on the fan-out and net density.

As interconnect delay became a more dominant portion of path delay it became necessary to refine the wire load estimations based on more than total net count for the chip. More and more passes through the synthesis-layout design loop became necessary to find a final solution that achieved the required path delays. The need for a new design tool in the flow became apparent as this estimation of interconnect delay was too coarse. The reason is that different regions on a typical chip have different levels of real estate (block and net) congestion. This means that wire load estimates differ across different regions on the chip because the average amount of wiring length for a net with a given pin count differs with the region. Conversely, the actual placement of cells on the chip effect the congestion and therefore, the resulting wire load models. Consequently, a new tool, called floor planning, was inserted in the flow between synthesis and layout.

14.2.3.2 Floor Planning

The purpose of floor planning is to provide a high-level plan for the placement of functions on the chip, which is used to both refine the synthesis delay estimations and direct the final layout. In effect, it inserts the designer into the synthesis-layout loop by providing a number of automation and analysis functions used to effectively partition and place functional elements on the chip. As time progressed, the number of analysis and checking functions integrated into the floor planner grew to a point where today, it is more generally thought of as design planning [13]. The initial purpose of the floor planner was to provide the ability to partition the chip design functions and develop a placement of these partitions that optimized the wiring between them. Optimization of the partitioning and placement may be based on factors such as minimum wiring or minimum timing across a signal path. A typical scenario for the use of a design planner is as follows:

- Run synthesis on the RTL description and map to the cell level.
- Run the design planner against the cell-level design with constraints such as path delays and net priorities.
  - Partition and floor plan the design, either automatically or manually, with the design planner.
  - Create wire load models based on the congestion within the physical partitions and empirical data.
- Rerun synthesis to optimize the design at the cell level using the partitioning, global route, and wire load model data.

User-directed graphics and autoplacement-wiring software algorithms are used to place the partitions and route the interpartition (global) nets. One key to the planner is the tight coupling of partitioning and routing capability with electrical analysis tools such as power analysis, delay calculation, and timing analysis. Checking of the validity of a design change (such as placement) can be made immediately on that portion of the design that changed. Another key is that it needs to be reasonably tightly connected to synthesis, as it is typical to pass through the synthesis-planner-synthesis loop a number of times before reaching a successful plan for the final layout tools.

After a pass in the floor planner, more knowledge is available to synthesis. With this new knowledge, the optimization and mapping phases can be rerun against the previous results to produce a refined solution. The important piece of knowledge now available is the gate density within different functions in the design (or regions on the chip). With this, the synthesis tool can be selective about which wire load table to use and develop a more accurate estimate of the delay resulting from a specific solution choice.

Floor planning has been a significant enhancement to the design process. In addition to floor planning, modern design planners include support for clock tree, design power, bus design, I/O assignment, and a wealth of electrical analysis tools. As will be discussed later, semiconductor technology trends will place even more importance on the design planner and dictate an ever-tightening integration between it with both synthesis and PD. Today, synthesis, design planning, and layout are three discrete steps in the typical design flow. Communication between these steps is accomplished by file-based interchange of the design data and constraints. As designs become more dense and with the growing complexities of electrical analysis required, it will become necessary to integrate these three steps into one tight process as are the functions within the modern design planners.

14.2.4 The 1990s: The Age of Integration

Before EDA tools were commercially available, software architecture standards could be established to assure smooth integration of the tools into complex EDA systems supporting the entire design flow. Choice of languages, database and file exchange formats, host hardware and operating systems, and user interface conventions could be made solely by the developing company. Moreover, the need for comprehensive EDA systems supporting the entire flow through design and manufacturing release in a managed process is of paramount importance. Design of ICs with hundreds of thousands of gates (growing to hundreds of millions) across a design team requires gigabytes of data contained within thousands of files. All these data must be managed and controlled to assure its correctness and consistency with other design components. Design steps must be carried out and completed in a methodical manner to assure correctness of the design before release to manufacturing. The EDA system that encompasses all of this must be efficient, effective, and manageable. This was always a challenge for EDA system integrators, but the use of EDA tools from multiple external vendors compounded the problem. EDA vendors may develop their tools to different user interfaces, database and file formats, computer hardware, and operating systems. In turn, it is the EDA system integrator who must find a way to connect them in a homogeneous flow that is manageable, yet provides the necessary design efficiency across the tools. By the late 1980s it was said that the cost to integrate an EDA tool into the in-house environment was often over twice that of the tool itself. Consequently, the CAD Framework Initiative (CFI) was formed whose charter was to establish software standards for critical interfaces necessary for EDA tool integration. Thus began the era of the EDA Framework.

The EDA Framework [14] is a layer of software function between the EDA tool and the operating system, database, or user. For this software layer CFI defined a set of functional elements, each of which had a standard interface and provided a unique function in support of communication between different tools. The goal was to provide EDA system integrators with the ability to choose EDA tools from different vendors and be able to plug them into a homogeneous system to provide an effective system operation across the entire design flow. The framework components included

- Design information access—a formal description of design elements and a standard set of software functions to access each, called an application program interface (API)
- Design data management—a model and API to provide applications the means to manage concurrent access and configuration of design data
- Intertool communication (ITC)—an API to communicate control information between applications and a standard set of controls (called messages)
- Tool encapsulation specification (TES)—a standard for specifying the necessary information elements required to integrate a tool into a system such as input files and output files
- Extension language (EL)—specification of a high-level language that can be used external to the EDA tools to access information about the design and perform operations on it
- Session management—a standard interface used by EDA tools to initiate and terminate their operation and report errata
- Methodology management—APIs used by tools so that they could be integrated into software systems to aid the design methodology management
- User interface—a set of APIs to provide consistency in the graphical user interface (GUI) across tools

Each of these functional areas needs to be considered for integration of EDA tools and to provide interoperability between them. In its first release, CFI published specifications for design representation (DR) and access, ITC, TES, and EL. Additionally, the use of MOTIF and X-Windows as specified as the GUI toolkit. However, for many practical, business, and human nature reasons, these standards did not achieve widespread adoption across the commercial EDA industry and the ''plug-and-play'' motto of CFI was not realized. However, the early work at CFI did serve to formalize the science of integration and communication, and a number of the information models developed for the necessary components are used today, but often, slightly modified and within a proprietary interface.

As learned, it was not only expensive to change EDA tools to use a different internal interface or data structures, but it was also not considered good business to allow tools to be integrated into commercial system offerings from competitors. For a period, several framework offerings became available, but these quickly became the product end in themselves rather than a means to an end. These products were marketed as a means to bring outside EDA tools into the environment, but few vendors modified their tools to support this by adopting the CFI specifications. Instead, to meet the need for tool integration, the commercial industry focused on data flow only and chose an approach that provides communication between tools by standard data file formats and sequential data translation. Consequently, EDA flows of the 1990s were typically a disparate set of heterogeneous tools stitched together by the industry-standard data files. These files are created from one tool and translated by another tool (within the design flow) to its internal data structures. Many of these formats evolved from proprietary formats developed within commercial companies that were later transferred or licensed for use across the industry. Common EDA file formats that evolved were

- Electronic data interchange format (EDIF)—netlist exchange (EIA → IEC)
- Physical data exchange format (PDEF)—floor plan exchange (Synopsys → IEEE)
- Standard delay file (SDF)—delay data exchange (Cadence Design Systems → IEEE)
- Standard parasitic exchange file (SPEF)—parasitic data exchange (Cadence Design Systems → IEEE)
- Library exchange file (LEF)—physical characteristics information for a circuit family (Cadence → OpenEDA)
- Data exchange format (DEF)—physical data exchange (Cadence → OpenEDA)
- Library data format (.lib)—ASIC cell characterization (Synopsys)
- ASIC library format (ALF)—ASIC cell characterization (Open Verilog International)
- VISIC high-level design language (VHDL)—behavioral design description language (IEEE)
- Verilog—RTL design description language (Cadence Design Systems → IEEE)

These "standard" formats have provided the desired engineering and business autonomy necessary to get adoption and their use across the industry has improved the ability to integrate tools into systems drastically compared with the 1980s. It has proved to be an effective method for the design of today's commodity ICs above 0.25 μm. However, the inherent ambiguities, translation requirements, rapidly increasing file sizes (owing to increasing IC density), and the fundamental nonincremental sequential nature of design flows based on them will soon give way to IC technology advances.

14.3 The Future

It may be said that the semiconductor industry is the most predictable industry on earth. Whether it was an astute prediction or the cause, the prediction of Gordon Moore (now known as Moore's Law) has held steadily over the past two decades. Every 18 months, the number of transistors or bits on an IC doubles. To accomplish this, feature sizes are shrinking, spacing between features is shrinking, and chip die sizes are increasing. In addition, the fundamental physics that govern the electrical properties of these devices and between them has been documented in electrical engineering textbooks for some time. However, there are points within this progression where paradigm shifts occur in the elemental assumptions and models required to characterize these circuits, and where fundamental EDA design and analysis must change. Earlier, we discussed paradigm shifts in DV because of the inability to repair. We also discussed shifts resulting from the number of elements in a design versus the necessary tool performance and designer productivity. Through the past three decades, we have seen shifts in test, verification, design abstraction, and design methodologies. When the 0.25 μm node was passed, another paradigm shift was required. Feature packing, decreased rise times, increased clock frequencies, increased die sizes, and the explosion of the number of switching transistors are all interacting to place new demands of models, tools, and EDA systems. After that and including the present, new challenges came about in design tools, rules, systems, and schools for problematic areas of delay, signal integrity, power, test, and manufacturability.

14.3.1 International Technology Roadmap for Semiconductors

SEMATECH periodically publishes a report called the *International Technology Roadmap for Semiconductors* [15] (ITRS). This report is jointly sponsored by the European Semiconductor Industry Association, Japan Electronics and Information Technology Industries Association, Korea Semiconductor Industry Association, Taiwan Semiconductor Industry Association, and the Semiconductor Industry Association. The objective of the ITRS "is to ensure advancements in the performance of integrated circuits." Thus ITRS characterizes planned semiconductor directions and advances across the next decade and the necessary technology advancements in areas including

- Design and test
- Process integration devices and structures
- Front-end (manufacturing) processes
- Lithography
- Interconnect
- Factory integration
- Assembly and packaging
- Environment, safety, and health
- Defect reduction
- Metrology
- Modeling and simulation

ITRS provides a wealth of information on the semiconductor future as well as the problematic areas that will arise and areas where inventive new technology will be required. Some relevant ITRS-projected

TABLE 14.1 2004 ITRS Microprocessor Roadmap

Characteristic	2001	2004	2007	2010	2013	2016
Transistor gate length (nm)	90	53	35	25	18	13
Chip size (mm^2)	310	310	310	310	310	310
Million transistors/mm^2	0.89	1.78	3.57	7.14	14.27	28.54
Million transistors/chip	276	553	1106	2212	4424	8848
Wire pitch (nm), metal	350	210	152	104	76	54
Clock frequency (GHz) (on-chip)	1.684	4.171	9.285	15.079	22.980	39.683
Supply voltage (V) (low power)	1.1	0.9	0.8	0.7	0.6	0.5
Maximum power (W) (high-performance)	130	158	189	198	198	198

semiconductor advancements for MPU/ASIC ICs are shown in Table 14.1 and compared with previous ITRS numbers for year 2001. The reader is encouraged to study the technology characteristics in the ITRS, as the following is just a summary of certain points for the purpose of illustration.

To understand some of the implications of the ITRS data, it is convenient to review the effects of scaling on a number of electrical parameters [16]. *Gate delay* is a function of the gate capacitance and transistor resistance. The gate capacitance ($C = k\varepsilon_o A/T_{gox}$) is a function of the gate area (A), and the gate-oxide thickness (T_{gox}). Assuming ideal scaling (all features scale down in direct proportion to the change in transistor size), gate capacitance scales in direct proportion to transistor-size scaling (ΔSize) since

$$\Delta C_{gate} \propto \left(\Delta W_{gate} \Delta L_{gate}\right)/\Delta T_{gox}$$

therefore

$$\Delta C_{gate} \propto \Delta\text{Size}$$

The transistor resistance (R_{tr}) is proportional to supply voltage and inversely proportional to current both of which scale down proportionally, so the R_{tr} is constant. Thus, the gate delay (D_{gate}) decreases in direct proportion to ΔSize since

$$D_{gate} = R_{tr} C_{gate},$$

therefore

$$\Delta D_{gate} \propto 1^{*}\Delta\text{Size} = \Delta\text{Size}$$

Thus, gate delays scale down in proportion to feature size. However, the effects of interconnect delay must be considered to understand the impact of scaling on timing. For $R_{int} < R_{tr}$ interconnect effects have little affect on signal path timing. As R_{int} increases, however, delay analysis must also take into account the interconnect parasitics.

Interconnect delay is a function of the *RC*-time constant, which is a result of resistance and distributed capacitance of the interconnect. Assuming ideal scaling, the interconnects' cross-sectional area scales in direct proportion to ΔSize.

Interconnect resistance is proportional to the length and inversely proportional to the cross-sectional area of the interconnect. Since

$$R_{int} \propto L_{int}/(W_{int} H_{int}),$$

therefore

$$\Delta R_{\text{int}} \propto L_{\text{int}}/\Delta\text{Size}^2$$

where, L_{int}, H_{int}, and W_{int} are the length, height, and width of the interconnect, respectively.

The self-capacitance ($C_{\text{self}} = k\varepsilon_o A/t$) of the interconnect is proportional to its plate area ($A = W_{\text{int}}\, L_{\text{int}}$) and inversely proportional to dielectric layer thickness (t). Assuming ideal scaling for t,

$$\Delta C_{\text{self}} \propto \Delta L_{\text{int}}\Delta\text{Size}/\Delta\text{Size}$$

therefore

$$\Delta C_{\text{self}} \propto \Delta L_{\text{int}}$$

The resulting interconnect *RC* constant, therefore, scales as

$$\Delta R_{\text{int}}\Delta C_{\text{self}} \propto \left(\Delta L_{\text{int}}/\Delta\text{Size}^2\right)(\Delta L_{\text{int}}) = \Delta L_{\text{int}}^2/\Delta\text{Size}^2$$

For *local nets* the interconnect length (L_{int}) scales with ΔSize thus, the *RC* constant remains relatively constant

$$\Delta(R_{\text{int}}C_{\text{self}})_{\text{local}} \propto \Delta L_{\text{int}}^2/\Delta\text{Size}^2 = \Delta\text{Size}^2/\Delta\text{Size}^2 = 1$$

Local interconnect delay does not scale down in proportion to gate delay on successive technology generations. Therefore, its relative impact on the overall timing across a signal path is increasing in importance.

For *global nets*, the interconnect length does not scale down with the feature size. Here, the die size (S_{die}) is the more predominant determinant. Therefore, the *RC* constant for global nets scales as

$$\Delta(R_{\text{int}}C_{\text{self}})_{\text{gobal}} \propto \Delta L_{\text{int}}^2/\Delta\text{Size}^2 = 1/\Delta\text{Size}^2$$

Global interconnect delay increases in indirect proportion to gate delay for successive technology generations. Thus, global interconnect routing and delay analysis has become a critically important area for EDA.

For complete analysis on interconnect delay, the cross-coupling effect of closely spaced features on the chip must also be considered. Capacitance results whenever two conducting bodies are charged to different electric potentials and may be viewed as the reluctance of voltage to build or decay quickly in response to injected power. The self-capacitance between the interconnect surfaces and ground is only a part of the overall capacitance to be considered. Another important consideration is the mutual capacitance between interconnects in close proximity. To get an accurate measure of the total capacitance along any interconnect it is necessary to consider all conducting bodies within a sufficiently close proximity. This includes not only the ground plane, but also adjacent interconnects as there is a mutual capacitance between these. This mutual capacitance is a function of interconnect sidewall area, the distance between it and adjacent interconnects, and the dielectric constant of the dielectric separating them.

Figure 14.10 depicts three equally spaced adjacent interconnect wires on a signal plane. An approximate formula for the capacitance [17] of the center wire is

$$C = \text{self-capacitance} + \text{mutual capacitance} = C_{\text{int-ground}} + 2C_{\text{m}}$$

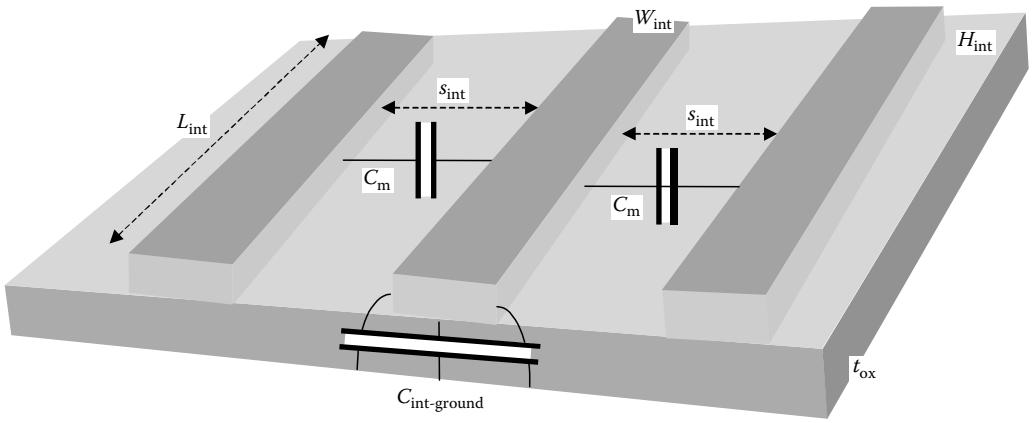


FIGURE 14.10 Mutual capacitance.

The mutual capacitance is a function of the plate area along the sides of the interconnects and the spacing between them and may be simply approximated as

$$C_m = k\varepsilon_o L_{int} H_{int} / S_{int}$$

where S_{int} is the distance between the interconnects.

Therefore (for a constant k-value), the change in mutual capacitance as a function of ideal scaling is given by

$$\Delta C_{m(Local)} \propto \Delta Size^2 / \Delta Size = \Delta Size \quad \text{and} \quad \Delta C_{m(Global)} \propto \Delta Size / \Delta Size = 1$$

Thus, mutual capacitance for local interconnects scales with feature size, while for global interconnects, it does not. Because of increased frequencies, electrical cross talk effects between interconnects must be considered. Further, mutual capacitance between only adjacent interconnects may not be complete. Full analysis may require consideration of all interconnects within a sphere around the interconnect being analyzed or even on wiring layers above or below.

Noise is the introduction of unwanted voltage onto a signal line, and is proportional to mutual capacitance between them and inversely proportional to the rate at which the voltage potential between them changes. Therefore, because the full effect of mutual capacitance is also a function of the voltage potential between interconnects (Miller effect) analysis of the simultaneous signal transitions on each becomes an important consideration for delay. Mutual capacitance injects current into adjacent (victim) signal lines proportional to the rate of change in voltage on the exciting (aggressor) signal line. The amount of induced current can be approximated by the following equation [18]:

$$I_{mC} = C_m \left(dV_{aggressor} / dt \right)$$

The amount of noise induced onto the victim net is proportional to the resistance on the victim net and the mutual capacitance, and inversely proportional to the rise time (τ_r) of the aggressor waveform:

$$V_{mC} = I_m Z_{victim} = C_m \left(\Delta V_{aggressor} / \tau_r \right) R_{victim}$$

Noise can increase or even decrease [19] the signal delay on the victim net. Voltage glitches induced can also give rise to faulty operation in the design (e.g., if the glitch is of sufficient amplitude and duration to cause a latch to go to an unwanted logic state.

Inductance can also cause noise when large current changes occur in very short times according to the following relationship:

$$V_L = L(dI/dt)$$

Sudden changes in large amounts of current can be seen on the power grid resulting from a large number of circuits switching at the same time. The inductance in the power grid and this large current draw over short periods of time (di/dt) results in a self-induced electromagnetic force whose voltage is equal to Ldi/dt. This induced voltage causes the power supply level to drop, resulting in a voltage glitch that is proportional to the switching speed, the number of switching circuits, and the effective inductance in the power grid.

14.3.2 EDA Impact

Scaling effects have impact on EDA beyond delay and timing, and the list of DF*x* (design-for-*x*) topics is increasing beyond design-for-test and design-for-timing (closure). Scaling down V_{dd} decreases power for a transistor, but the massive scaling up of total transistors on the IC dictates extensive analysis of high currents and leakage (design for power). Lithographic limitations (discussed later) now require consideration of diffraction physics (design for manufacturability [DFM]). Further, while the design complexities owing to electrical effects such as these are increasing, the number of circuits on the IC continues to increase. The fundamental challenge for EDA is more analysis on many more circuit elements with no negative effect on design cycle times.

14.3.2.1 EDA System Integration

It is now necessary to use EDA applications with increased accuracy and scope for electrical analysis to assure that designs will meet intended specifications and to integrate EDA design flows in ways that contain design cycle times. EDA systems must now take a data-centric view of the design process as opposed to the tool-centric view of the past. Data must be organized and presented in ways that allow EDA to exploit: abstraction and hierarchy, shared access by design tools, incremental processes (whereby only portions of a design that have changes need be reanalyzed), and concurrent execution of design and analysis tools within the design flow.

EDA systems have evolved or are evolving toward integration and database technology that meets the above requirements to one degree or another. EDA systems have evolved from vendor-specific sets of design and analysis applications supporting data exchange between custom, often proprietary, file formats to systems that support interchange between vendors through industry-standard files. Over the past decade they have moved toward systems of applications communicating through vendor-specific integrated database technology supporting intersystem exchange via the same industry-standard files. Because of the increasing need for new and additional EDA applications companies performing IC design continue to require the ability to use EDA applications from many vendors and develop flows that provide efficient and complete intervendor integration. With increasing feature counts and complexities for successive technology generations, this dictates the need to reduce the overhead and possible data loss caused by file-based interchange (Figure 14.11). In 2000, a multicompany effort under Si2 was initiated to develop an industry-open data model that could be used by commercial EDA companies and for university research and (EDA customer) proprietary EDA development. This effort resulted in the publication of an EDA data model specification, which includes an application program software interface, and development of a production-quality *reference* database that is compliant with that specification. Collectively, this specification and database is called OpenAccess (see www.si2.org) and was made available to the industry in 2003 on a royalty-free basis. OpenAccess has an excellent chance of being the foundation on which the next point of EDA systems' evolution will be founded. At that

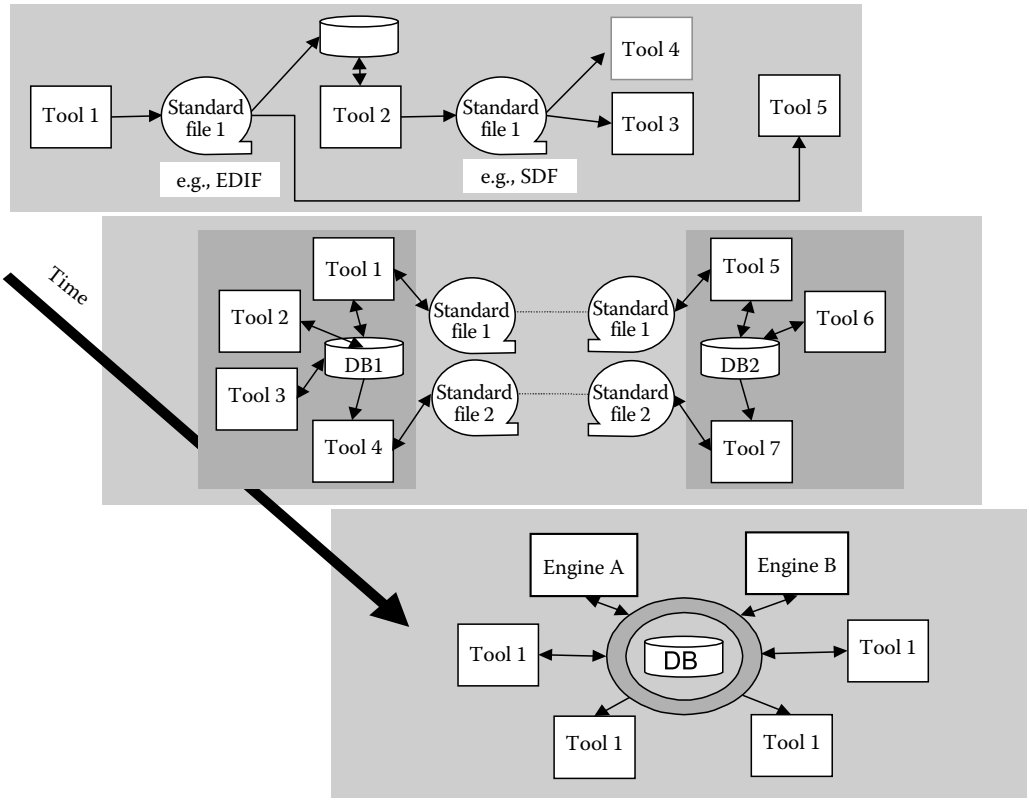


FIGURE 14.11 EDA system architecture progression.

point, highly efficient EDA flows will be built using EDA tools from multiple sources and integrated around a single database meeting the fundamental requirements mentioned above.

14.3.2.2 Delay

Simplifying assumptions and models for delay have provided the foundation for high-speed event-driven delay simulation since its initial use in the 1970s. More accurate waveform simulation such as that provided by SPICE implementations has played a crucial role in the characterization of IC devices during over three decades and continues to do so. SPICE-level simulations are important for characterization and qualification of ASIC cells and custom macros of all levels of complexity. However, the run times required for this level of device modeling are too large to support its use across the entire ICs. SPICE is often used to analyze the most critical signal paths, but SPICE-level simulation on circuits with hundreds of millions of transistors is not practical today. Consequently, simulation and STA at the abstracted gate-level, or higher, is essential for IC timing analysis. However, simplifying models used to characterize delay for discrete-event simulation and STA have become more complex as feature sizes on ICs have shrunk, and the importance of interconnect resistance and cross talk increased. In the future, these models will need to improve even more (Figure 14.12).

When ICs were considerably >1 μm, gate delay was the predominant factor in determination of timing. Very early simulation of TTL used a simple model which assigned a fixed unit of delay to each gate and assumed no delay across the interconnects. Timing was based only on the number of gates through which a signal traversed along its path. As LSI advanced delay was based on a foundry-specified gate delay value (actually a rise delay and a fall delay) that was adjusted based on the capacitive load of

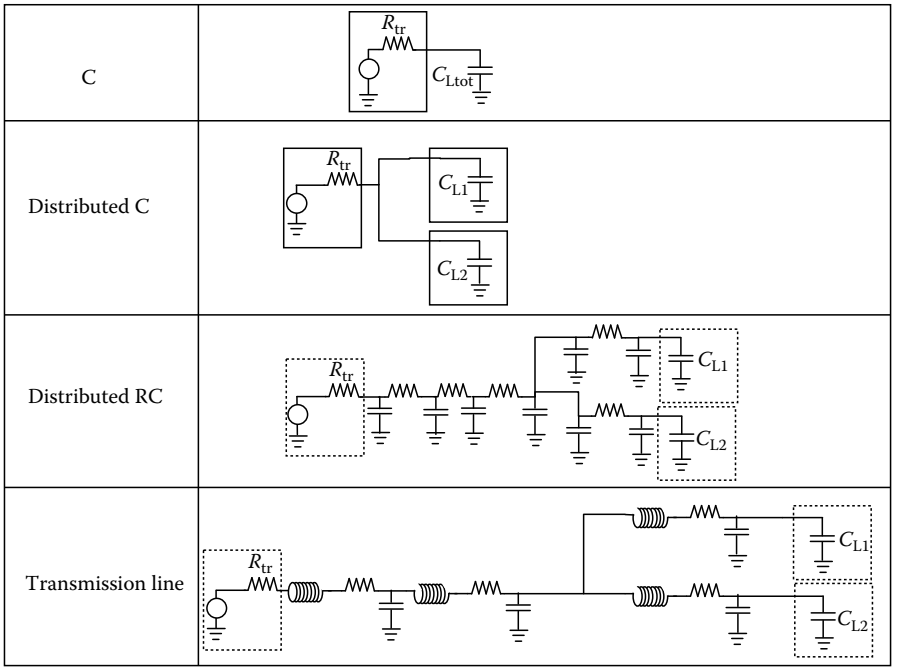


FIGURE 14.12 Delay models.

the gates it fanned out to (receiver gates). At integration levels above 1 μm, the load seen by a gate was dominated by the capacitance of its receivers, so this delay model was sufficiently accurate. As feature sizes crossed below 1 μm, however, the parasitic resistance and capacitance along the interconnects became a significant factor. More precise modeling of the total load effect on gate delay and interconnect delay had to be taken into account. By 0.5 μm the delay attributed to global nets almost equaled that of gates, and by 0.35 μm the delay attributed to short nets equaled the gate delay.

Today, a number of models are used to represent the distributed parasitics along interconnects using lumped-form equations. The well-known π-model, for example, is a popular method to model these distributed RCs. Different degrees of accuracy can be obtained by adding more sections to the equivalent-lumped RC model for interconnect.

As the importance of timing-driven design tools grows, integration of multiply sourced design tools into an EDA flow becomes problematic. As more EDA vendors use custom built-in models for computation of gate and interconnect delays, the calculated values may differ across different design tools. This results in difficult and time-consuming analysis on the designers' part to correlate the different results. In mid-1994, an industry effort began to develop an open architecture supporting common delay-calculation engines [20]. The goal of this effort was to provide a standard language usable by ASIC suppliers to specify the delay calculation models and equations (or table lookups) for their technology families. In this way, the complexities of deciding appropriate models to be used to characterize circuit delay and the resulting calculation expressions could be placed in the hands of the semiconductor supplier rather than across the EDA vendors. These models could then be compiled into a form that could be directly used by any EDA application requiring delay calculation and in a way that would protect the intellectual property contained within them. By allowing the semiconductor supplier to provide a single software engine for the calculation of delay, all applications in the design flow could provide consistency in the computed results. The Delay Calculation Language (DCL) technology, originally developed by IBM Corporation, was contributed to industry as the basis for this. Today,

DCL has been extended to cover power calculation in addition to delay and it has been ratified by the IEEE as an open industry standard (Delay and Power Calculation System [DPCS], IEEE 1481–1999). For a number of technical and business reasons, adoption of this standard has failed to take hold. Although release and adoption of the Synopsys.lib format, used to provide technology parameters used by delay calculation in a standard form, greatly improved this situation, problems with accuracy and consistency of delay calculation remain. To that end, a renewed interest in developing an industry-wide solution came about in 2005 and the Open Modelling Coalition (OMC) has been formed under Si2 to readdress the challenge with industry partners.

At 0.25 μm, cross talk noise resulting from mutual capacitance between signal lines begins to have a significant effect on delay. At this technology generation, it became necessary to consider effects of mutual capacitance between interconnects. Today, sophisticated extraction tools must analyze the proximity of features (including wires, vias, and pads) and properties of dielectrics between them. In addition, EDA design and analysis applications must account for mutual parasitics and cross talk. This is an important element of layout, parasitic extraction, delay calculation, and timing analysis. Further, the development of wafer fabrication techniques such as dielectric air gaps to reduce effective *k*-values will bring further challenges to EDA.

With future technology generations, more sophisticated models that consider inductance may also become necessary. Distributed *RC* models used to characterize delay are very accurate approximations so long as the rise time (t_r) of the signals is much larger than the time-of-flight (t_{of}) across the interconnect wire. As the transistor size scales down, so do the signal rise times. As t_r approaches t_{of}, transmission line analysis may be necessary to accurately model timing. Between these points is a gray area where transmission line analysis may be necessary depending on the criticality of the timing across a particular path. Published papers [21,22] address the question of where this level of analysis is important and the design rules to be considered for time-critical global lines. For future EDA systems, this means more complexities in the design and analysis of circuits. Extraction tools will then need to derive inductance of interconnects so that the effects of the magnetic fields surrounding these lines can be factored into the delay calculations. Effects of mutual inductance also becomes important, particularly along the power lines which will drive large amounts of current in very short times ($e = L\ \mathrm{d}i/\mathrm{d}t$). Design planners and routers will need to be aware of these effects when designing global nets and power bus structures.

14.3.2.3 Test

Manufacturing test of ICs today is becoming a heuristic process involving a number of different strategies in combination to assure coverage and maximize generation costs and throughput at the tester. Today, IC test employs the use of one or more of the following approaches:

- Static stuck-at fault test using stored program testers: The test patterns may be derived algorithmically or from random patterns and the test is based on final steady-state conditions independent of arrival times. The goal of stuck-at fault test is to achieve 100% detection of all stuck-at faults (except for redundancies and certain untestable design situations that can result from the design). Algorithmic test generation for all faults in general sequential circuits is often not possible, but test generation for combinational circuits is. Therefore the use of design-for-test strategies (such as scan design) is becoming more accepted.
- Delay (at-speed) test: Patterns for delay testing may be algorithmically generated or functional patterns derived from DV. The tester measures output pins for the logic state at the specified time after the test pattern is applied.
- BIST: BISTs are generated at the tester by the device under test and output values are captured in scan latches for comparison with simulated good-machine responses. BISTs tests may be exhaustive or random patterns and the expected responses are determined by simulation. BIST requires special circuitry on the device-under-test (LFSR and scan latches) for pattern generation and result

capture. To improve tester times, output responses are typically compressed into a signature and comparison of captured results and simulated results is made only at the end of sequences of tests.

- Quiescent current test (I_{DDQ}): I_{DDQ} measures quiescent current draw (current required to keep transistors at their present state) on the power bus. This form of testing can detect faults not observable by stuck-at fault tests (such as bridging faults) that may cause incorrect system operation.

Complicated semiconductor trends that will affect manufacturing test in the future are the speed at which these future chips operate and the quiescent current required for the large density of transistors on the IC. To perform delay test, for example, the tester must operate at speeds of the device-under-test. That is, if a signal is to be measured at a time n after the application of the test pattern, then the tester must be able to cycle in n units of time. It is possible that the hardware costs required to build testers that operate at frequencies above future IC trends will be prohibitive. I_{DDQ} tests are a very effective means to quickly identify faulty chips as only a small number of patterns are required. Further, these tests find faults not otherwise detected by stuck-at fault test. Electric current measurement techniques are far more precise than voltage measurements. However, the amount of quiescent current draw required for the millions of transistors in future ICs may make it impossible to detect the small amount of excess current resulting from small numbers of faulty transistors. New inventions and techniques may be introduced into the test menu. However, at present it appears that in the future manufacturing test must rely on static stuck-at fault tests and BIST. Therefore, it is expected that more and more application of scan design will be prevalent in future ICs.

14.3.2.4 Design Productivity

The ITRS points out that future ICs will contain hundreds of millions of transistors. Even with the application of the aforementioned architectural features and new and enhanced EDA tools, it is unclear that design productivity (number of good circuits designed per person year) will be able to keep pace with semiconductor technology capability. Designing and managing anything containing over 100,000,000 subelements is almost unthinkable. Doing it right and within the typical 18-month product cycles of today seems impossible. Yet, if this is not accomplished, semiconductor foundries may run at less than full production, and the possibility of obtaining returns against the exorbitant capital expenditures required for new foundries will be low. Ultimately, this could negate the predictions in the ITRS.

Over the history of IC design, a number of paradigm shifts enhanced designer productivity, the most notable being high-level design languages and synthesis. Use of design rules such as LSSD to constrain the problem has also represented productivity advances. Algorithm advances and faster processing computers will necessarily play a crucial role, as the design cycle time is a function of computer resources required. Many of the EDA application algorithms, however, are not linear with respect to design size and processing times can increase as an exponential function of transistors. Therefore exploitation of hierarchy, abstraction, shared, incremental, and concurrent EDA system architectures will play an important role to overall productivity. Even with all of this, there is a major concern that industry will not be able to design a sufficient number of good circuits fast enough. Consequently, there is a major push in the semiconductor industry toward design reuse. There is a major activity in the EDA industry (Virtual Sockets Interface Alliance [VSIA] www.vsia.org) to define the necessary standards, formats, and test rules to make design reuse a reality with ICs.

Design reuse is not new to electronic design or EDA. Early TTL modules were an example of reuse, as are standard cell ASICs and PCB-level MSI modules. With each, the design of the component is done once, then qualified, then reused repeatedly in application-specific designs. Reuse is also common where portions of a previous design are carried forward to a new system or technology node, and where a common logical function used multiple times across the system. However, the ability to embed

semiconductor designs qualified for one process into chips manufactured on another, or for which the logical design was developed by one company to be used by another one results in new and unique challenges. This is the challenge that the VSIA is addressing.

The VSIA has defined the following three different types of reusable property for ICs:

1. Hard macros: These functions have been designed and verified, and have a completed layout. They are characterized by being a technology-fixed design and a mapping of manufacturing processes is required to retarget them to another fabrication line. The most likely business model for hard macros is that they will be available from the semiconductor vendor for use in application-specific designs being committed to that supplier's fabrication line. In other words, hard macros will most likely not be generally portable across foundries except in cases where special business partnerships are established. The complexities of plugging a mask-level design for one process into another process line are a gating factor for further exploitation at present.
2. Firm macros: These can be characterized as reusable parts that have designed down to the cell level through partitioning and floor planning. These are more flexible than hard macros since they are not process dependent and can be retargeted to other technology families for manufacture.
3. Soft macros: These are truly portable design descriptions, but are only completed down through the logical design level. No technology mapping or PD is available.

To achieve reuse at the IC level, it will be necessary to define or otherwise establish a number of standard interfaces for design data. The reason for this is that it will be necessary to integrate design data from other sources into the design and EDA system being used to develop the IC. VSIA was established to determine where standard interfaces or data formats are necessary and choose the right standard. For soft macros these interfaces will need to include behavioral descriptions, simulation models, and timing models. For firm macros the scope will additionally include cell libraries, floor plan information, and global wiring information. For hard macros GDSII may be supplied.

To reuse designs that were developed elsewhere (hereafter called intellectual property [IP] blocks), the first essential requirement is that the functional and electrical characteristics of the available IP blocks be available. Since internal construction of firm and hard IP may be highly sensitive and proprietary, it will become more important to describe the functional and timing characteristics at the I/Os. This will drive the need for high-level description methods such as use of VHDL behavioral models, DCL, I/O Buffer Information Specification (IBIS) models, and dynamic timing diagrams. Further, the need to test these embedded IP blocks will mandate more use of scan-based test techniques such as Joint Test Action Group (JTAG) boundary scan. Standard methods such as these for encapsulating IP blocks will be of paramount importance for broad use across many designs and design systems.

There are risks, however, and reuse of IP will not cover all design needs. Grand schemes for reusable software yielded less than desired results. Extra effort to generalize the IP block design points for broad use, and describe its characteristics in standard formats is compounded by the ability to identify an IP block that fits a particular design need. The design and characterization of reusable IP will need to be robust in timing, power, reliability, and noise in addition to function and cost. Further, the broad distribution of IP information may conflict with business objectives. Moreover, even if broadly successful, use of reusable IP will not negate the fundamental need for major EDA advances in the areas described. First, tools are needed to design the IP. Second, tools are needed to design the millions of circuits that will interconnect IP. Finally, tools and systems will require major advances to accurately design and analyze the electrical interactions between, over, and under IP and application-specific design elements on the IC. Standards are essential, but they are not by themselves sufficient for success. Tools, rules, and systems will be the foundation for future IC design as they have been over the past. Nevertheless, the potential rewards are substantial and there is an absence of any other clear EDA paradigm shift. Consequently, new standards will emerge for design and design systems, and new methods for characterization, distribution, and lookup of IP will become necessary.

14.3.2.5 DFM

14.3.2.5.1 Lithography

Because of the physical properties of diffraction, as printed features shrink, the ability to print them with the required degree of fidelity becomes a challenge for mask making. The scaling effects for lithographic resolution can be generally viewed using the Rayleigh equation for a point light source:

$$\text{Smallest resolvable feature} = K(\lambda/\text{NA})$$

Here

Value of K is a function of the photoresist process used
λ is the wavelength of the coherent light source used for the exposure
NA is the numerical aperture of the lens system

As feature shrink continues, the ITRS projects that design wafer foundry lithographic systems will keep pace by moving to light sources of smaller wavelength, photoresist systems with lower values for K, and immersion lithography. (Immersion lithography increases the maximum achievable numerical aperture ($\text{NA} = I \sin a$) beyond that in air (in air, the refractive index (I) = 1, thus $\text{NA}_{\max} = 1$), because of higher refractive indices for water and other fluids). However, the physics of light diffraction is now affecting the design of photomasks, and this will dramatically increase in importance with future technology generations.

Without attempting to discuss the physics behind them, two complications need to be accounted for in the design of a mask. First, the intensity of the light source passing through the mask and onto the photoresist is nonlinear. That is, the intensity at the edges of small features is less than that in the center. Thus, edges of small features do not print with fidelity. To compensate for this, for example on the corners of lines (which will expose as rounded edges) and line ends (which will be rounded and short), a series of reticle (mask) enhancement techniques are used. These techniques add subresolution features to the pattern data. Some examples are the addition of serifs on corners and hammerheads on line ends to extend as well as square them.

The second complication is the interference of diffracted light between adjacent features. That is, the light used to expose densely packed features may interfere with the exposure of other features in close proximity. This may be corrected for by phase-shift mask techniques or be accounted for by adjusting feature placement (spreading densely packed areas and using scatter bars to fill sparsely packed areas) and biasing feature widths to adjust for the interference.

Though these corrections are necessary for two distinct physical properties, they are generally collectively performed by optical proximity correction (OPC) programs. OPC in today's design systems is generally performed a step after the design is complete and ready for tape-out. The desired design pattern data is topographically analyzed by OPC against a set of rules or models that determine if and what corrections need to be made in the mask pattern to assure wafer fidelity. These corrections result in the incorporation of additional features to be cut into the mask such as serifs, hammerheads, and scatter bars.

With successive technology generations, the number of OPCs for a mask set will increase and cause mask pattern data to increase beyond that of Moore's Law for features. This, in turn, causes increased time (and resulting costs) in capital-intensive mask manufacturing. Therefore, future design systems will need to consider lithography limitations early in the design process (synthesis and PD) rather than at the end, and new OPC methods to optimize the mask pattern volume will be required.

14.3.2.5.2 Yield

Traditionally, wafer yield has been viewed as the responsibility of the wafer foundry alone. Hard defects caused by random particles or incorrect process settings are under control of the foundry manager and defects in feature parametrics were sufficiently bounded by design rules to be insignificant. As features shrink, small variations in print fidelity or process parameters have an increasingly important impact on

feature parasitics and this can be a significant factor in the yield of properly operating chips. Traditionally, parasitic extraction performed on the PD-generated geometry was sufficient to analyze a parasitic effect on, for example, timing. In future, because of lithography limitations, the analysis of parasitics may require extraction based on the simulated geometries that are projected to print on the wafer. Traditionally, EDA design applications can operate within the specific bounds of the design rules to yield good design. Future design applications may require that statistical analysis of their decisions be performed to account for intrachip parametric variations and to provide a design known to be good within a certain degree of probability. The topic of yield will become an important aspect of the next EDA paradigm shift.

Future design system flows will require a closer linkage between the classical EDA flow and Technology Computer Aided Design (TCAD) systems and mask making. In addition, the need for rich models that can be used to accurately predict the impacts of design choices on the results of foundry and mask manufacturing processes will become important. New design tools such as Statistical Static Timing Analysis (SSTA) have become available and are a topic of much discussion. How to incorporate the effects of variability on yield in, for example PD and synthesis will be a topic for future research. High-speed lithographic simulation, capable of full chip analysis, will be an important field for EDA R&D. Finally, database and integration technology to support this increased level of collaboration between IC designers, mask manufacturers, and the fabricating engineers will be critical.

14.4 Summary

Over the past four decades, EDA has become a critical science for electronics design. Where it may have begun as a productivity enhancement, it is now a fundamental requirement for the design and manufacture of electronic components and products. The criticality of EDA for semiconductors is the focus of much attention because of the expanding semiconductor advancements. Fundamental approaches in test, simulation, and PD are being constantly emphasized by these semiconductor advancements. New EDA disciplines are opening up. Fundamental electrical models are becoming more important and at the same time, more complex. New problems in design and IC failure modes will surely surface.

The next decade of EDA will prove to be as challenging and exciting as its past!

References

1. Case, P.W., Correia, M., Gianopulos, W., Heller, W.R., Ofek, H., Raymond, T.C., Simel, R.L., and Stieglitz, C.B., Design automation in IBM, *IBM Journal of Research and Development*, 25(5): 631, 1981.
2. Roth, J.P., Diagnosis of automata failures: A calculus and a method, *IBM Journal of Research and Development*, 10: 278, 1966.
3. Eichelberger, E.B., Hazard detection in combinational and sequential switching circuits, *IBM Journal of Research and Development*, 9: 90, 1965.
4. Eichelberger, E.B. and Williams, T.W., A logic design structure for LSI testability, *Proceedings of the 14th Design Automation Conference*, New Orleans, LA, pp. 462–468, June 1977.
5. IEEE Std 1149.1-1990, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE, Washington, 1990.
6. Hitchcock, R.B., Cellular wiring and the cellular modeling technique, *Proceedings of the 6th Annual Design Automation Conference*, ACM, New York, pp. 25–41, 1969.
7. Lee, C.Y., An algorithm for path connection and its applications, *IRE Transactions on Electronic Computers*, 10(1): 364–365, 1961.
8. Hightower, D.W., The interconnection problem: A tutorial, *Proceedings of the 10th Annual Design Automation Workshop*, IEEE Press Piscataway, NJ, pp. 1–21, 1973.
9. *1076-1993 IEEE Standard VHDL Language Reference Manual*, IEEE, Washington, 1993.

10. *1364-2005 IEEE Standard Hardware Description Language Based on the Verilog® Hardware Description Language*, IEEE, Washington, 2006.
11. Electronic Industry Associates, *Electronic Design Interchange Format (EDIF)*, IHS, www.globalihs.com.
12. Stok, L., Kung, D.S., Brand, D., Drumm, A.D., Sullivan, A.J., Reddy, L.N., Heiter, N., et al., BooleDozer: Logic synthesis for ASICs, *IBM Journal of Research and Development*, 40(4): 407, 1996.
13. Sayah, J.Y., Gupta, R., Sherlekar, D.D., Honsinger, P.S., Apte, J.M., Bollinger, S.W., Chen, H.H., et al., Design planning for high-performance ASICs, *IBM Journal of Research and Development*, 40(4): 431, 1996.
14. Barnes, T.J., Harrison, D., Newton, R.A., and Spickelmier, R.L., *Electronic CAD Frameworks*, Kluwer Academic, Dordrecht, the Netherlands, Chapter 10, 1992.
15. Semiconductor Industry Association, *International Technology Roadmap for Semiconductors, Technology Needs, 2001 Edition*, www.sematech.org.
16. Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, Reading, MA, Chapters 1–7, 1990.
17. Goel, A.K., *High-Speed VLSI Interconnections, Modeling, Analysis & Simulation*, Wiley, New York, Chapter 2, 1994.
18. Johnson, H.W. and Graham, M., *High-Speed Digital Design, A Handbook of Black Magic*, Prentice-Hall PTR, New York, Chapter 1, 1993.
19. Pandini, D., Scandolaro, P., and Guardiani, C., Network reduction for crosstalk analysis in deep submicron technologies, *ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Austin, TX, pp. 280, December 1997.
20. Cottrell, D.R., Delay calculation standard and deep submicron design, *Integrated Systems Design*, July 1996.
21. Fisher, P.D., Clock cycle estimates for future microprocessor generations, *IEEE 1997 ISIS: International Conference on Innovative Systems in Silicon*, Austin, TX, pp. 61–71, October 1997.
22. Deutsh, A., Kopcsay, G.V., Restle, P., Katopis, G., Becker, W.D., Smith, H., Coteus, P.W., et al., When are transmission-line effects important for on-chip interconnections, *Electronic Components and Technology Conference*, San Jose, CA, pp. 704, 1997.

Index

A

Advanced design environment prototype tool (ADEPT)
  design flow, 12-17
  EDIF netlist, 12-17–12-18
  handshaking and token passing mechanism
    event sequence, 12-13
    status field, 12-12–12-13
    three-module, 12-14–12-15
    token propagation, 12-15–12-16
  interpreted and uninterpreted models, 12-9
  module categories, 12-16–12-17
  revolutionary approach, 12-9
  simulation results, 12-21–12-23
  three-computer system model
    Data-Dependent Delay A (DDA) element, 12-20
    Fixed Delay modules, 12-19
    interconnection, 12-18
    unit_step delay, 12-20
  token implementation
    CPN representation, 12-11–12-12
    definition, 12-10
    Petri net, 12-11
    STATUS field, 12-11
  VHDL behavioral description, 12-9–12-10
Analog circuit (AC) simulation
  active device models, 9-5–9-6
  analysis
    frequency response, 9-9–9-10
    gain and input impedances, 9-10
    noise, 9-10–9-11
bipolar
  DC models, 8-40
  high-frequency model, 8-42–8-43
BSIM extraction strategy
  channel length modulation (CLM), 8-36
  drain-induced barrier lowering (DIBL), 8-37–8-38
  MOS models, 8-38–8-40
  steps, 8-35–8-36
  wide-short device, 8-36–8-37
  zero-order parameters, 8-35
circuit equations, 9-3–9-4
commercial simulators, 9-19
DC (steady-state) analysis
  Newton's method, 9-7–9-8
  simulation, 9-8
device modeling and characterization, 8-21–8-22
extraction
  drain current, 8-27–8-28
  linear region equation, 8-27
  vs. optimization, 8-28–8-29
fast simulation methods, 9-18–9-19
least-squares curve fitting
  analytical solutions, 8-23–8-25
  Gauss–Newton method, 8-26–8-27
  MOS transistor, 8-24
  numerical techniques, 8-25–8-27
  root mean square (RMS) error, 8-23
  steepest descents method, 8-25–8-26
macro modeling, 8-44
modified nodal analysis, 9-4–9-5
MOS and bipolar capacitance models
  BJT model, 8-41–8-42
  junction diode capacitor, 8-41
  overlap, 8-42
MOS DC models
  BSIM model, 8-30–8-31
  extraction strategy, 8-31–8-35
  SPICE, 8-29–8-30
netlists, 9-2–9-3
purpose, 9-1–9-2
skew parameter files, 8-43
strategies, 8-29
TCAD environment, 8-44
transient analysis
  differential amplifier, 9-12–9-13
  numerical method, 9-11–9-12
Verilog-A, 9-13–9-17
Analog design centering problem
  convexity approaches
    convex programming, 4-23
    feasible region, 4-22
    largest Hessian ellipsoid, 4-22–4-23
    log-barrier function, 4-22
    steps, 4-21–4-22

ellipsoidal method, 4-21
problem description, 4-19
simplicial approximation method
design, 4-20
elongated feasible regions, 4-21
hypersphere, in polytope, 4-20
steps, 4-19–4-20
Application-specific heterogeneous multiprocessor (ASHM), 11-12
Application-specific integrated circuit (ASIC) chip, 14-14

B

Berkeley short-channel IGFET model (BSIM)
channel length modulation (CLM), 8-36
drain-induced barrier lowering (DIBL), 8-37–8-38
MOS DC model, 8-30–8-31
MOS models, 8-38–8-40
proprietary models, 8-31
steps, 8-35–8-36
wide-short device, 8-36–8-37
zero-order parameters, 8-35
Bipolar junction transistor (BJT) model, 8-40

C

Circuit performance modeling
applications
clock distribution circuit, 1-15
predicted *vs.* simulated values, 1-16
central composite design, 1-8–1-9
circuit parameters
model-building procedures, 1-4–1-5
nominal component, 1-3
statistically varying component, 1-4
digital circuits *vs.* analog circuits, 1-2
factorial experimental design
advantages, 1-8
pictorial representation, 1-6
resolution designs, 1-7–1-8
two-factor interaction effect, 1-6–1-7
group method of data handling (GMDH), 1-14–1-15
input–output relationships, 1-3
Latin hypercube sampling (LHS) method, 1-9–1-11
least-square fitting method
basis functions, 1-12
coefficient of determination, 1-12–1-13
definition, 1-11
maximally flat quadratic interpolation (MFQI) method, 1-14–1-15
stochastic performance modeling, 1-14
Taguchi's orthogonal arrays method, 1-9
variable screening method, 1-13–1-14
CMOS digital circuits
area model, 4-14
convex programming approach
gradient calculations, 4-18
posynomial functions, 4-17–4-18
suboptimal solution, 4-17
delay modeling
circuit technique, 4-12–4-14
gate level, 4-11–4-12
transistor, 4-11
feasible direction method
delay estimation algorithm, 4-15–4-16
generalized gradient, 4-16
method of feasible directions (MFD) algorithm, 4-15–4-16
scaling, 4-17
Lagrangian multipliers method, 4-17
problem description, 4-10–4-11
sensitivity-based TILOS algorithm
algorithm steps, 4-14–4-15
sensitivity calculation, 4-15
statistical variations, 4-18–4-19
two-step optimization, 4-17
Cockpit program
Applet, 10-24
core functionality, 10-27
execute message, 10-19
graphical user interface, 10-17
Java file system (JFS), 10-27
Minbits encoder, 10-19
multithreads, 10-17
physical synthesis, 10-19
productions and ratings, 10-17
query handling, 10-20
reports subtask success, 10-21
root flow, 10-24
system windows, 10-23
task-specific knowledge, 10-16
tool invocation, 10-17
whiteboard, 10-10
Commercial off-the-shelf (COTS) modules, 11-2
Compacted modified nodal analysis matrix (CMNAM), 2-14–2-15
Compiled simulation, EDA, 14-18–14-19
Complementary metal-oxide semiconductor (CMOS) inverter, 7-3
Computer aided design (CAD)
conceptual design
applications, 7-13–7-14
coarse filter, 7-6
decision assistance, 7-10–7-12
design planning, 7-8–7-10
least-commitment strategy, 7-12–7-13
memory technology, 7-6
performance trade-offs, 7-7
design entry
computational algorithms, 7-4–7-5
graphic and electrical devices, 7-3
language and mechanism, 7-3
Moore finite state machine, 7-5
physical and logical information, 7-3
schematic capture, 7-4
state diagram, 7-5
VHDL model, 7-5–7-6
dynamic timing analysis, 7-17–7-18
electrical monitoring techniques, 7-26–7-27
event-driven stimulation, 7-19–7-20

fault modeling test
ad hoc techniques, 7-26
advantages, 7-25
boundary scan, 7-26
coding techniques, 7-26–7-27
integrated circuit, 7-23
sequential automatic program, 7-24–7-25
single stuck line model, 7-23–7-24
strategies, 7-24
stuck-at techniques, 7-24
filter performance prediction, 7-10–7-11
frameworks
data management, 7-28
definition, 7-27
object-oriented paradigms, 7-28–7-29
process management, 7-28
user interface services, 7-27–7-28
integrated circuit, 7-22–7-23
message-driven stimulation, 7-20
signal transition requirements, 7-17
signature techniques, 7-26
simulation engines, 7-21
static timing analysis, 7-18–7-19
synthesis techniques
design process, 7-14–7-15
pipelining schedules, 7-16
silicon compliers, 7-14
taxonomy, 7-15
test design process, 7-21–7-22
verification
emulation, 7-21
simulation, 7-19–7-21
synthesis task, 7-16
timing analysis, 7-17–7-19
Computer-aided engineering (CAE), *see* Computer aided design (CAD)
Control/data-flow graphs (CDFG), 11-4
Circuit simulation
AC analysis
cascode amplifier, 8-6–8-9
macro model elements, 8-6–8-7
numerical method, 8-6
phasor analysis, 8-5
PSPICE, 8-7
bipolar junction transistors (BJTs), 8-1
DC (steady-state) analysis
Newton's method, 8-2–8-3
SPICE, 8-4–8-5
steps, 8-2–8-3
voltage regulator, 8-3–8-5
device simulation
conventional lumped models, 8-18
lattice heat equation, 8-18–8-19
MOSFET, 8-20
SUPREM-IV curves, 8-19
distributed systems, 8-2
IC design process, 8-13–8-14
logic integrated circuits (ICs), 8-2
process simulation
diffusion, 8-14–8-15
ion implantation, 8-16–8-17
NMOS transistor, 8-17
oxidation, 8-15–8-16
software, 8-20
transient analysis
analog multiplier, 8-12
MOSFET models, 8-13
numerical method, 8-9–8-10
phase-locked loop (PLL) circuit, 8-10–8-13
Crout's algorithm, 3-12–3-13

D

Delay calculation language (DCL) technology, 14-33
Delay modeling
circuit diagram, 4-12
gate delays, 4-14
posynomial function, 4-13
program evaluation and review technique (PERT), 4-12
steps, 4-13
Design for manufacturability (DFM)
lithography, 14-37
wafer yield, 14-37–14-38
Design rule checking (DRC), 6-4
Design verification (DV) simulators, 14-13–14-19
DFM, *see* Design for manufacturability (DFM)
Digital signal processor (DSP), 13-2
Discrete circuits
derivative-based method
Hessian matrix, 5-17–5-18
orthogonal array (OA) sampling, 5-18
parametric sampling technique, 5-17
sampling yield estimation, 5-16
yield gradient, 5-16–5-17
heuristic method
centers of gravity method, 5-14–5-15
Gaussian adaptation, 5-15
stochastic approximation (SA) approach, 5-20–5-23
Doolittle's algorithm
computation cost, 3-15–3-16
matrix factorization, 3-12

E

Ebers–Moll model, 9-5–9-6
EDA, *see* Electronic design automation (EDA)
Electronic design automation (EDA)
EDA impact
delay models, 14-32–14-34
design for manufacturability (DFM), 14-37–14-38
design productivity, 14-35–14-36
system integration, architecture progression, 14-31–14-32
IC test, 14-34–14-35
historical perspective
design entry, 14-4–14-5
fault simulation, 14-10–14-11
formal verification, 14-20
fundamental concept, 14-3–14-4
hard-copy drawing, 14-4
industry birth developed
floor planning, 14-24–14-25
integration age, 14-25–14-27
synthesis, 14-21–14-23
integration age, 14-25–14-27
microprocessor development, 14-2
physical design, 14-12–14-13
register-transfer level (RTL), 14-5
test generation
built-in self-test (BIST), 14-8–14-10

D-ALG calculus, 14-6–14-7
pseudorandom number, 14-9
scan design, 14-8–14-9
stuck-at fault models, 14-6
timing analysis, 14-19
verification
methodologies, 14-20–14-21
printed circuit boards (PCBs), 14-13
simulation, 14-14–14-19
Electronic design interchange format (EDIF) netlist, 12-17
Elmore time constant, 4-11
Embedded computing systems
core computational tasks, 13-4
CPU resources, 13-6
data dependencies, 13-6
hardware architecture, 13-3
hardware/software codesign
cosimulation, 13-7
cosynthesis, 13-9–13-10
design methodology, 13-10
models, 13-6–13-7
performance analysis, 13-7–13-9
heterogeneous, 13-4
interrupt-driven, 13-4
microprocessors
analog components, 13-2
digital signal processor (DSP), 13-2
peripheral devices, 13-1
VLSI technology, 13-2–13-3
nonlinear function, 13-5
processing elements (PEs), 13-4
real-time operating system (RTOS), 13-5–13-6
scheduling technique, 13-6
soft deadlines, 13-3
Equation formulation method
Kirchhoff's law
branch currents and characteristics, 3-1–3-2
reduced incidence matrix, 3-2
reference convention, 3-2
voltage around loop, 3-3
modified nodal analysis, 3-7–3-8
nodal analysis, 3-4–3-7
sparse tableau method, 3-4
stamping methods
modified nodal formulation, 3-9–3-11
nodal formulation, 3-8–3-9
voltage-controlled current source (VCCS), 3-4
Event-based simulation, EDA
system behavior languages, 14-18
simulation delay model, 14-16–14-17
simulator concept, 14-15
time queues, 14-15–14-16

F

Fast simulation methods
linear solution time reduction, 9-18
matrix assembly time reduction
simplify method, 9-18
waveform relaxation method, 9-18–9-19
Fault modeling test, CAD
ad hoc techniques, 7-26
advantages, 7-25
boundary scan, 7-26
coding techniques, 7-26–7-27
integrated circuit, 7-23
sequential automatic program, 7-24–7-25
single stuck line model, 7-23–7-24
strategies, 7-24
stuck-at techniques, 7-24

G

Gaussian elimination, 2-10
Geometrical methods, disadvantages, 4-23–4-24
Group method of data handling (GMDH), 1-14–1-15

H

Hardware simulators, EDA, 14-19
Hardware/software codesign
cosimulation, 13-7
cosynthesis, 13-9–13-10
design methodology, 13-10
models, 13-6–13-7
performance analysis
cache state, 13-8–13-9
embedded systems, 13-7
multiple-process systems, 13-8

I

IMEDA implementation and execution environment
chip synthesis node, 10-18
Cockpit program
graphical user interface, 10-17
task-specific knowledge, 10-16
communications model, 10-26
external tools
actual design execution, 10-24
association and execution, 10-25
Cockpit Applet, 10-24
Java Development Kit (JDK), 10-26
manager programs
backtracking, 10-17
circuit area or latency, 10-18
production execution, 10-17
query handling, 10-18
one-hot encoder, 10-19
scheduling
automatic task execution, 10-21
filter design process, 10-20
linear scheduler, 10-19
linear scheduling algorithm, 10-22
rollback signal, 10-23
system Cockpit
components, 10-24
Java-enabled Web browser, 10-23
Integrated circuits (IC)
chip reusable property, 14-36
derivative-based method, 5-19–5-20
stochastic approximation (SA) method
convolution function smoothing, 5-23–5-25
gradient estimation, 5-23
International Technology Roadmap for Semiconductors (ITRS), 14-27–14-31
interconnect delay, resistance, 14-28
mutual capacitance, 14-30
semiconductor advancements, 14-27–14-28

Internet-based microelectronic design automation (IMEDA) system
agent-based approach, 10-9
components, 10-9–10-10
electronic design automation (EDA), 10-3
execution environment
Cockpit program, 10-16–10-17
execution example, 10-18–10-19
manager program, 10-17–10-18
scheduling, 10-19–10-23
features, 10-11–10-12
flow graph
bipartite acyclic directed graph, 10-12
field-programmable gate array configuration file, 10-13
tool invocation, 10-12
framework functional requirements
building blocks, 10-4
design methodology, 10-3
execution environment, 10-7
literature surveys, 10-7–10-9
process automation, 10-4
process specification, 10-6–10-7
workflow management, 10-4–10-5
grammars
child specification type, 10-13
sample graph derivation, 10-15
simulation and partition alternatives, 10-14
implementation
communications model, 10-26
external tools, 10-24–10-26
system Cockpit, 10-23–10-24
user interface, 10-27–10-30
microelectronic industry, 10-1
process specifications, 10-11
rapid design exploration and optimization (RaDEO), 10-2
real-world processes, 10-11
unified process specification language, 10-3

J

Java Development Kit (JDK), 10-26–10-27

K

Kernighan–Lin (K–L)
algorithm, 6-12
graph bisections, 6-10
locked/unlocked vertices, 6-11–6-12
vertex exchanges, 6-11

L

Latin hypercube sampling (LHS) method, 1-9–1-11
Least-square fitting method
basis functions, 1-12
coefficient of determination, 1-12–1-13
definition, 1-11
Lee's algorithm, 6-16–6-18
Levenberg–Marquardt algorithm, 8-27
Linear equations, decompositon methods
backward substitution, 3-14
forward substitution, 3-13
matrix factorization
computation cost, 3-15–3-16
Crout's algorithm, 3-12–3-13
Doolittle's algorithm, 3-12
pivoting, 3-14–3-15

M

Maximally flat quadratic interpolation (MFQI) method, 1-15
Microprocessors
analog components, 13-2
digital signal processor (DSP), 13-2
peripheral devices, 13-1
VLSI technology, 13-2–13-3
Mixed integer-linear programming (MILP), 11-6
Model-building procedures, 1-4–1-5
MOSFET model, 9-6
MOS models
bipolar capacitance models
BJT model, 8-41–8-42
junction diode capacitor, 8-41
overlap, 8-42
DC analysis
Berkeley short-channel IGFET model (BSIM), 8-30–8-31
extraction strategy, 8-31–8-35
SPICE, 8-29–8-30
Multichip modules (MCMs), 11-5
Multi-level modeling approach
advantages, 12-8
architecture design and assessment system (ADAS), 12-7
behavioral and performance models, 12-6
functional models, 12-5
heterogeneous, 12-8–12-9
interpretation, 12-6
Petri nets, 12-6–12-7
Ptolemy project, 12-8
SystemC language
background, 12-44–12-45
C++ foundations, 12-45–12-47
channels, 12-48–12-52
crossbar contention, 12-56–12-58
crossbar model, 12-53
dual processor, 12-54
fixed cycle length, 12-61
fully connected architecture, 12-53
mixed-level processor model, 12-58–12-61
parallel communications, 12-54–12-56
performance-based modeling tool (PBMT), 12-42
performance model, 12-63–12-64
processor model, 12-47–12-48
random data generation, 12-62–12-63
shared bus architecture, 12-52–12-53
single processor, 12-53–12-54

system architecture, 12-42–12-43
task graph model, 12-42–12-44
variable cycle length, 12-61–12-62
transaction-level model (TLM), 12-8
VHDL models, 12-7–12-8

O

OPC, *see* Optical proximity correction
Operational amplifiers (op-amps)
characteristics, 2-14
compacted MNA matrix (CMNAM), 2-14–2-15
components
norator, 2-13
nullator, 2-12–2-13
nullor, 2-13
definition, 2-12
gyrator circuit, 2-15
Optical proximity correction (OPC) programs, 14-37
Optimization algorithms
constrained methods
direction method, 4-6
Lagrange multiplier, 4-5
linear programming, 4-5
others, 4-8
penalty function method, 4-5–4-6
Vaidya's convex programming, 4-6–4-8
definitions
concave function, 4-4
convex function, 4-3–4-4
convex hull, 4-3
convex programming problem, 4-4
convex set, 4-2–4-3
posynomial function, 4-4
multicriterion
design, 4-9
formulation, 4-8
minmax function, 4-10
Pareto critical point, 4-8–4-9
weighted sum function, 4-9–4-10
nonlinear problem, 4-2

P

PCBs, *see* Printed circuit boards (PCBs)
PD, *see* Physical design
Performance modeling and analysis
ADEPT design environment
design flow, 12-17
EDIF netlist, 12-17–12-18
handshaking and token passing mechanism, 12-12–12-16
interpreted and uninterpreted models, 12-9
module categories, 12-16–12-17
revolutionary approach, 12-9
token implementation, 12-10–12-12
VHDL behavioral description, 12-9–12-10
design methodology, 12-4
disadvantages, 12-2
embedded systems, 12-4
engineering/system-level design environments, 12-3
hardware/software system, 12-3–12-4
interfaces, 12-5
mixed-level modeling
behavioral models, 12-25
characteristics, 12-25
complex sequential components, 12-34–12-41
FSMD components, 12-26–12-34
interpreted and uninterpreted models, 12-23–12-24
structure, 12-24
taxonomy, 12-25–12-26
multi-level modeling approach
advantages, 12-8
architecture design and assessment system (ADAS), 12-7
behavioral and performance models, 12-6
functional models, 12-5
heterogeneous, 12-8–12-9
interpretation, 12-6
Petri nets, 12-6–12-7
Ptolemy project, 12-8
transaction-level model (TLM), 12-8
VHDL models, 12-7–12-8
product's development cycle, 12-1–12-2
queue lengths *vs.* simulation time, 12-23
risk driven expanding information models (RDEIMs), 12-4–12-5
SystemC language
background, 12-44–12-45
C++ foundations, 12-45–12-47
channels, 12-48–12-52
crossbar contention, 12-56–12-58
crossbar model, 12-53
dual processor, 12-54
fixed cycle length, 12-61
fully connected architecture, 12-53
mixed-level processor model, 12-58–12-61
parallel communications, 12-54–12-56
performance-based modeling tool (PBMT), 12-42
performance model, 12-63–12-64
processor model, 12-47–12-48
random data generation, 12-62–12-63
shared bus architecture, 12-52–12-53
single processor, 12-53–12-54
system architecture, 12-42–12-43
task graph model, 12-42–12-44
variable cycle length, 12-61–12-62
three-computer system
ADEPT model, 12-19–12-20
Data-Dependent Delay A (DDA) element, 12-20
Fixed Delay (FDA and FDB) modules, 12-19
interconnection, 12-18

simulation results, 12-21–12-23
unit step delay, 12-20
Phase-locked loop circuit (PLL)
analog and mixed signal simulator model (AMS), 9-15–9-17
detector, 9-14–9-15
Ebers–Moll model, 9-16–9-17
loop filter, 9-15
VCO, 9-14
Physical design (PD), 14-12–14-13
Physical design automation
circuit blocks and pins, 6-15
compaction
classification, 6-21–6-22
fabrication process, 6-21
others, 6-23
shadow-propagation algorithm, 6-22–6-23
cycles
compaction, 6-6
partitioning process, 6-4
placement, 6-4–6-6
routing algorithms, 6-6
design styles
cell library, 6-8
full-custom structure, 6-6–6-7
gate array, 6-8
I–O pads, 6-7
nonhierarchical structure, 6-8–6-9
standard cells, 6-8
detailed routing algorithms
greedy channel router, 6-18–6-19
Lee's algorithm, 6-16–6-18
other algorithms, 6-18–6-19
global routing algorithms, 6-16
partitioning process
classification, 6-9–6-10
disadvantages, 6-9
Kernighan–Lin (K–L) algorithm, 6-10–6-12
placement phase
annealing algorithm, 6-13–6-14
classification, 6-13
flexible blocks, 6-12–6-13
techniques, 6-14
routing regions, 6-15–6-16
very large-scale integration (VLSI) design cycle
circuit design, 6-3
design verification and fabrication, 6-4
functional and logic design, 6-2
packaging, testing, and debugging, 6-4
physical design, 6-3–6-4
processing steps, 6-3
system specification, 6-2
Printed circuit boards (PCBs), 14-5, 14-12
Probability density function (p.d.f)
circuit simulator variables, 5-2–5-3
discrete resistor, 5-3–5-4
manufacturing yield, 5-10–5-12
yield gradient formula, 5-17
Program evaluation and review technique (PERT), 4-12–4-14

R

Rapid prototyping of application specific signal processor (RASSP), 10-2
Real-time operating system (RTOS), 13-5–13-6
Register-transfer level (RTL), 14-5
Response surface model (RSM), 1-4–1-5
Routing algorithms
Acker's scheme, 6-18
circuit blocks, 6-15
greedy channel router, 6-18–6-19
grid-based models, 6-16
LEA and YACR2, 6-19–6-20
Lee's algorithm, 6-16–6-18
sequential and concurrent approach, 6-16
split nets, 6-19
VLSI chip, 6-15
RTL, *see* Register-transfer level

S

Short-circuit admittance parameters, 2-9
Simulation program with integrated circuit emphasis (SPICE), *see* Computer aided design (CAD)
Sparse tableau method
equation formulation method, 3-4
symbolic analysis techniques, 2-3
Statistical design optimization
acceptability region approximation
active RC filter, 5-6–5-7
c-dimensional hyperplanes, 5-9
cutting-plane methods, 5-10
definition, 5-6
e-space, 5-6–5-7
indicator function, 5-8
integrated circuits, 5-6, 5-8
point-based methods, 5-8–5-9
circuit performance, 5-3
circuit (simulator) variables
CMOS transistor parameters, 5-5–5-6
discrete active RLC circuits, 5-3
discrete resistor, 5-3–5-4
level set, 5-4
passive IC elements and device model parameters, 5-4–5-5
θ parameters, 5-6
three-dimensional *e*-space, 5-5
designable parameters, 5-2
manufacturing (parametric) yield
definition, 5-10
Monte Carlo algorithm, 5-11–5-12
performances, functional *vs.* parametric circuits, 5-10
variance reduction methods, 5-12
problems and methodologies, 5-1–5-2
random and circuit variables, 5-2–5-3
yield optimization
case study, 5-25–5-26
circuit-parameter space *e*, 5-13
derivative-based methods, 5-16–5-18
discrete circuit, 5-12–5-13
enhancement, process optimization, 5-25–5-26

heuristic method, 5-14–5-15
integrated circuits, 5-19–5-20
large-sample *vs.* small sample, 5-14
objective, 5-12
performance-space oriented design centering methods, 5-12–5-13
standard deterministic algorithms, 5-14
stochastic approximation (SA) approach, 5-20–5-25
variability and Taguchi circuit optimization problems, 5-26–5-29
Stochastic performance modeling, 1-14
SUPREM, *see* Computer aided design (CAD)
Symbolic analysis techniques
applications
circuit operation, 2-18
circuit response optimization, 2-16
circuit sizing, 2-17
in education, 2-18
fault diagnosis of analog circuits, 2-17–2-18
frequency response evaluation, 2-16
parameter extraction in device modeling, 2-17
sensitivity analysis, 2-17
statistical analysis method, 2-17
common analysis methods
hybrid analysis, 2-3
modified nodal analysis (MNA), 2-3–2-4
sparse tableau analysis, 2-3
determinant-based solution method
Cramer's rule, 2-7
two-port circuit representation, 2-8
operational amplifiers (op-amps)
characteristics, 2-14
compacted MNA matrix (CMNAM), 2-14–2-15
components, 2-12–2-13
definition, 2-12
gyrator circuit, 2-15
parameter reduction solution method
two-port terminated circuit, 2-10–2-11
variable suppression, 2-9–2-10
software packages
add-on packages (toolboxes), 2-19
symbolic simulation programs, 2-18
symbolic transfer function, 2-2
Symbolic transfer function, 2-2
SystemC language
background, 12-44–12-45
C++ foundations
class definition, 12-45–12-46
interface_type.hfile, 12-46–12-47
ModelSim compiler, 12-46
top config.txt file, 12-45
channels
channel_param.txt file, 12-51–12-52
comm_medium class, 12-49
data_to_delay function, 12-52
graphical representation, 12-49–12-50
hierarchical models, 12-48–12-49
im_done event, 12-51
processor models, 12-50
transaction_element object, 12-50–12-51
transact() method, 12-51
crossbar contention, 12-56–12-58
crossbar model, 12-53
dual processor, 12-54
fixed cycle length, 12-61
fully connected architecture, 12-53
mixed-level processor model
computation model, 12-60
gate-level model (RTL), 12-58–12-59
interface code, 12-60–12-61
refined_computation object, 12-59
rng_comp_tb class, 12-59–12-60
parallel communications
fully connected and crossbar simulation, 12-56
shared bus architecture, 12-55–12-56
task graph, 12-54–12-55
performance-based modeling tool (PBMT), 12-42
performance model, 12-63–12-64
processor model, 12-47–12-48
random data generation, 12-62–12-63
shared bus architecture, 12-52–12-53
single processor, 12-53–12-54
system architecture, 12-42–12-43
task graph model, 12-42–12-44
variable cycle length, 12-61–12-62
System-level design
characteristics, 11-10
design space, 11-3
Gantt chart, 11-7–11-8
integrated circuit (IC), 11-1
interconnection strategy, 11-9
packaging styles, 11-9
partitioning methods
constructive technique, 11-5–11-6
iterative technique, 11-6–11-7
multichip modules (MCMs), 11-5
philosophies, 11-2–11-3
processing modules, 11-7–11-8
research survey
ASHM image processing, 11-12
critical path method (CPM), 11-14
embedded systems, 11-15
genetic algorithms, 11-13
imprecise computation, 11-13–11-14
macropipelined design, 11-12
network chip, 11-15
nonpipelined design, 11-11–11-12
partitioning, 11-11
performance bounds theory, 11-14–11-15
proabilistic models, 11-14
rate-monotonic scheduling (RMS), 11-13
specification, 11-11

stochastic simulation, 11-14
system data structure (SDS), 11-11
word-length selection, 11-15
specification, 11-3–11-4
storage modules, 11-8–11-9
word-length determination, 11-10

T

Taguchi's orthogonal arrays method, 1-9
TCAD, *see* Technology Computer Aided Design
Technology Computer Aided Design (TCAD) systems, 14-38
Two-factor interaction effect, 1-6–1-7
Two-port admittance matrix, 2-9

U

User interface
design flow graph properties, 10-27
Java file system (JFS), 10-27
key framework properties, 10-29–10-30
macro substitution
children flow objects, 10-29
property inheritance, 10-28
property macros, 10-28
property window and property inheritance, 10-29

V

Vaidya's convex programming algorithm
algorithmic process, 4-7
objective, 4-6–4-7
pictorial representation, 4-8
Variable screening method, 1-13–1-14
Variable suppression, 2-9; *see also* Gaussian elimination
Verilog-A
equations and elements, 9-13
phase-locked loop circuit (PLL)
analog and mixed signal simulator model (AMS), 9-15–9-17
detector, 9-14–9-15
Ebers–Moll model, 9-16–9-17
loop filter, 9-15
VCO, 9-14
Verilog hardware descriptive language (VHDL), 11-3
ADEPT design environment, 12-9–12-10
description and simulation, 12-17
multi-level modeling, 12-7–12-8

Y

Yield optimization statistical methods
case study, 5-25–5-26
circuit-parameter space *e,* 5-13
discrete circuits
derivative-based method, 5-16–5-18
heuristic method, 5-14–5-15
stochastic approximation (SA) approach, 5-20–5-23
enhancement, process optimization, 5-25–5-26
heuristic method, 5-14–5-15
integrated circuits
derivative-based method, 5-19–5-20
stochastic approximation (SA) method, 5-23–5-25
large-sample *vs.* small sample, 5-14
objective, 5-12
performance-space oriented design centering methods, 5-12–5-13
standard deterministic algorithms, 5-14
stochastic approximation (SA) approach, 5-20–5-25
variability and Taguchi circuit optimization
generalized loss function (GLF), 5-28
variability minimization, 5-26–5-27
weight function (wα), 5-28–5-29

Electrical Engineering

Upon its initial publication, *The Circuits and Filters Handbook* broke new ground. It quickly became **the** resource for comprehensive coverage of issues and practical information that can be put to immediate use. Not content to rest on his laurels, in addition to updating the second edition, editor Wai-Kai Chen divided it into tightly focused texts that made the information easily accessible and digestible. These texts have been revised, updated, and expanded so that they continue to provide solid coverage of standard practices and enlightened perspectives on new and emerging techniques.

Computer Aided Design and Design Automation draws together international contributors who address topics, such as the modeling of circuit performances, symbolic analysis methods, numerical analysis methods, design by optimization, statistical design optimization, and physical design automation. Building on this, they then turn their attention to high performance simulation, formal verification, system-level design, Internet-based microelectronic design automation framework, performance modeling, and embedded computing systems design.

Features

- Addresses the usefulness of symbolic analysis as a computer-aided design tool and its application to integrated circuit design
- Presents an overview of the fundamental concepts of VLSI physical design automation
- Examines the three primary simulation modes: DC, AC, and transient analysis with an overview of numerical techniques used and a discussion of common pitfalls
- Discusses analog circuit simulators and the theory behind them and provides examples that illustrate their use
- Presents a collaborative engineering framework that coordinates distributed design activities through the Internet
- Describes embedded computing systems that make use of microprocessors to implement part of the system's function

This volume will undoubtedly take its place as the engineer's first choice in looking for solutions to problems encountered in computer aided design and design automation.

59181

ISBN: 978-1-4200-5918-2

90000

9 781420 059182

CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
270 Madison Avenue
New York, NY 10016
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK